MEMORANDUM REPORT ARBRL-MR-03142

A GENERAL CURVILINEAR GRID GENERATION PROGRAM FOR PROJECTILE CONFIGURATIONS

J. L. Steger
C. J. Nietubicz
K. R. Heavey

October 1981

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

Approved for public release; distribution unlimited.
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.

The use of trade names or manufacturers' names in this report does not constitute endorsement of any commercial product.
A general curvilinear grid generation program for projectile configurations

J.L. Steger, C.J. Nietubicz, & K.R. Heavey

U.S. Army Ballistic Research Laboratory
(ATTN: DRDAR-BLL)
Aberdeen Proving Ground, Maryland 21005

US Army Armament Research & Development Command
US Army Ballistic Research Laboratory (DRDAR-BL)
Aberdeen Proving Ground, MD 21005

Approved for public release, distribution unlimited.

A planar grid generation routine has been developed for use with standard and non-conventional projectile shapes. Three-dimensional grid generation has been obtained by generating a sequence of planar grids about axis normal cross-sections. The method is basically automatic and generates smoothly varying grids for arbitrary body shapes and allows for grid point clustering. The program is modular and can be used to generate planar Cartesian-like grids, C-grids, O-grids, or any portion thereof. The routine can be used for axisymmetric projectiles with or without stings, symmetric tubular projectiles.
20. ABSTRACT (Continued)

and for any configuration with an axisymmetric nose. Sample grids are presented along with the program listing and sample output.
### LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Standard Projectile Grid with Sting</td>
<td>19</td>
</tr>
<tr>
<td>2a</td>
<td>Cartesian-like Projectile Grid with Sting</td>
<td>19</td>
</tr>
<tr>
<td>2b</td>
<td>Special Case Isolated Boattail</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>Standard Projectile Grid with Base, Special Case of O-Grid with Symmetry</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>Tubular Projectile Grid of C-Grid</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>Projectile Blunt Body Grid (Fraction of O-Grid)</td>
<td>21</td>
</tr>
<tr>
<td>6</td>
<td>Projectile with Symmetric Nose and Nonsymmetric Afterbody</td>
<td>21</td>
</tr>
<tr>
<td>7</td>
<td>Projectile Cross Section with Periodic B.C. (O-Grid)</td>
<td>22</td>
</tr>
<tr>
<td>8</td>
<td>Projectile Cross Section with Symmetry Plane (O-Grid)</td>
<td>22</td>
</tr>
<tr>
<td>9</td>
<td>Mapping from Physical Space to Computational Space</td>
<td>23</td>
</tr>
<tr>
<td>10a</td>
<td>Standard Projectile Grid in the Physical Plane</td>
<td>24</td>
</tr>
<tr>
<td>10b</td>
<td>Grid Detail near Projectile</td>
<td>25</td>
</tr>
<tr>
<td>11</td>
<td>Stretching Function, Points $j$ are Specified along with $x_0$, $x_f$, $\Delta x_0$, and $\nabla x_f$.</td>
<td>26</td>
</tr>
<tr>
<td>12</td>
<td>Outer Boundary Structure and Terminology for Two Classes of Grid</td>
<td>27</td>
</tr>
<tr>
<td>13a</td>
<td>Overview of Straight Ray Grid for Tubular Projectile</td>
<td>28</td>
</tr>
<tr>
<td>13b</td>
<td>Grid Detail near Lower Trailing Edge</td>
<td>29</td>
</tr>
<tr>
<td>14a</td>
<td>Standard Projectile Grid with Controlled Reclustering along Lines of Constant $\xi$.</td>
<td>30</td>
</tr>
<tr>
<td>14b</td>
<td>Boattail Base Detail Showing Specified Clustering in $n$ near the Body</td>
<td>31</td>
</tr>
<tr>
<td>15</td>
<td>Hybrid Straight Ray and Elliptic Solver Grid Detail for a Tubular Projectile</td>
<td>32</td>
</tr>
</tbody>
</table>
I. INTRODUCTION

Modern finite difference procedures for solving the partial differential equations which describe fluid flow frequently utilize curvilinear mapping procedures. Because boundary surfaces in the physical plane can be mapped onto rectangular surfaces in the transformed plane, a finite difference algorithm for the transformed equations can be readily applied to a variety of different body shapes. Even unsteady body motion is easily incorporated into the governing equations. To take advantage of the generality of the transformed equations, however, one needs a fairly automatic method of generating smoothly varying grids that fit arbitrary bodies and allow grid point clustering. The problem of grid generation, restricted to arbitrary projectile shapes, is the subject of this report.

A modular general purpose grid generation routine has been written for use with standard and nonconventional projectiles shapes. Three-dimensional grid generation capability is satisfied by generating a sequence of planar grids about axis normal cross sections. The grid generation routine can be used to generate planar Cartesian-like grids, C-grids, O-grids, or any portion thereof. The routine can be used for axisymmetric projectiles with or without stings, symmetric tubular projectiles, and for any projectile with an axisymmetric nose.

The modular grid generation program developed here contains its own body surface representation and grid point distribution routines. Another set of routines allows the user to build up an arbitrary outer boundary curve and grid point distribution. Finally, the mesh itself is formed using either algebraic straight-line rays to connect inner and outer boundary points, or by using an elliptic solver. Clustering along grid lines is accomplished with an exponential clustering routine that allows the user to specify a given grid point spacing along the inner boundary curve. The modular structure of the grid generation program allows the user to substitute alternate clustering and grid generation routines of his own design.

Figures 1-8 show the various classification of grids which the code is designed to handle. These include axisymmetric projectiles with or without a sharp leading edge and with or without an afterbody sting, Figures 1-3. Isolated boattails or flares can also be meshed (Figure 2b). The code can also treat tubular projectiles (ring airfoils) that are axisymmetric, Figure 4. The isolated axisymmetric blunt body problem, Figure 5, is actually a special case of Figure 1.

In Section II the mechanics of the grid generation routine are described. Various grids are displayed to illustrate the ideas. Additional discussion of the computer grids is given in Section III, while complete documentation of the codes is given in the Appendices. The option of selecting a 2 or 3 dimensional grid is included and a plotting routine is presented in the appendix.
II. GRID GENERATION

The purpose of the grid generation routine is to generate a network of constant lines of \( \xi \) and \( \eta \) in the physical x-y plane as indicated in Figure 9a. Corresponding uniform values of \( \xi \) and \( \eta \) in the computational space define a one to one mapping between points \( j,k \) in the physical plane to points \( j,k \) in the computational plane, see Figure 9b. The mapping functions are described, at least numerically, once \( \xi_{j,k} \) and \( \eta_{j,k} \) are known in the physical plane as a function of \( x_{j,k} \) and \( y_{j,k} \); or conversely, once \( x_{j,k} \) and \( y_{j,k} \) are determined in the transform plane. The metric quantities \( \xi_x, \xi_y, \eta_x, \) and \( \eta_y \) needed in the transformed flow equations can then be determined numerically (see, for example, References 1-3).

In Figures 10a and 10b we show a typical grid generated for an axisymmetric projectile shape. In this case a spherical cap is placed at the end of the boattail (see Figure 10b) in order to avoid a highly discontinuous corner. The grid as it stands is suitable for axisymmetric or \( n \)-invariant flow calculations. A three-dimensional grid can be formed by rotating the grid about the axis and defining constant lines in \( \theta \). When viewed in this manner the grid is equivalent to warping a spherical coordinate about a nonspherical projectile shape. Using the notation defined in Figure 9a, the axis corresponds to the \( \xi = 0 \) and \( \xi = \xi_{\text{max}} \) lines. The inner and outer boundaries correspond to \( n = 0 \) and \( n = n_{\text{max}} \).

Depending on the projectile shape a warped hemispherical coordinate may be used (Figure 1) or a warped Cartesian coordinate (Figure 3). A ring airfoil (tubular projectile) can be meshed with a C-grid (Figure 4). One could also spin an airfoil O-grid (Figure 7) about the tubular projectiles axis of symmetry. Whichever class of grid is used, however, it must map onto the uniform computational plane shown in Figure 9b. The \( r = 0 \) plane is not necessarily restricted to only the body surface. It may for example, include the forward cut of Figure 2a or the lower and upper cuts of Figure 4.


In generating a projectile grid such as those indicated in Figures 1-8, one first decides what class of grid fits the given problem. The grid generation problem can then be broken into three main tasks as follows: (1) define the body shape, possible sting or cut, and distribute grid points along the \( n = 0 \) boundary (i.e., specify \( \xi \) as a function of \( x \) and \( z \) along \( n = 0 \)). Points along this boundary should be clustered to flow field gradients, e.g., the forward stagnation point, expansions, shocks; (2) define the outer boundary curve and distribute grid points along the \( n = n'_{\text{max}} \) boundary. Here we restrict \( \xi = 0 \) and \( \xi = \xi_{\text{max}} \) to be vertical or horizontal straightline rays in order to simplify programming logic, thus the endpoints of the \( n'_{\text{max}} \) curve must properly align with those of the \( n = 0 \) boundary; (3) once the outer boundaries are defined, they are "connected" by generating the interior grid with appropriate clustering functions in \( n \).

In the remainder of this Section the procedures used to generate the \( n = 0 \) boundary, the \( n = n'_{\text{max}} \) boundary, and the interior clustered grid, will be described.

A. Surface Representation and Grid Point Distribution

The first step in generating the grid is to represent and distribute points along the body surface. A sting or cut may also have to be included.

The body shape is expected to have either an analytic description or be described as a table of \( x, y \) ordinates. In either case the data is assumed to be nondimensional with respect to a reference length which can, of course, be taken as 1 so the data remains unaltered.

The present code allows for either a parabolic arc or standard class projectile, such as a projectile with a sharp tangent-ogive or blunt secant ogive nose, cylindrical body, boattail, and spherical cap. If the analytic body shape differs from the above mentioned shapes, then the user must supply his own description. In this case values of \( x \) along the body axis (or chord) will be distributed by contiguously combining segments of the clustering function

\[
x_j = x_0 + a\psi_j + b\psi_j^2 + c\psi_j^3 \quad \text{for} \quad x_0 < x_j < x_f
\]

\[
j_0 < j < j_f
\]

where \( \psi_j = (j-j_0)/(j_f-j_0) \) and \( j \) is an index value such that points \( j_0 \) to \( j_f \) lie in the interval \( x_0 \) to \( x_f \) and \( x_{j_0} = x_0 \) while \( x_{j_f} = x_f \). Equation (1) is used to cluster \( x_j \) as a function of \( j \) as indicated in Figure 11. The user determines the shape of the clustering function by specifying the initial and final increments of \( x \), that is
\[\Delta x_0 = x_{j_0+1} - x_{j_0}\]  

\[\nabla x_f = x_{j_f} - x_{j_f-1}\]

(2a)  

(2b)

Since \(x_0\) and \(x_f\) are also specified, \(a\), \(b\), and \(c\) are determined

\[
c = \frac{\nabla x_f + \Delta x_0 - 2h(x_f-x_0)}{(h - 3h^2 + 2h^3)}
\]

\[
b = \frac{\Delta x_0 - h(x_f-x_0) - c(h^3 - h)}{(h^2 - h)}
\]

\[
a = x_f - x_0 - b - c
\]

where \(h = (j_f - j_0)^{-1}\).

The amount of clustering at each point is determined by the specified values of \(\Delta x_0\) and \(\nabla x_f\). Moreover, because \(\Delta x_0\) and \(\nabla x_f\) are specified, the user can smoothly patch functions together to form a general clustering function. Examples of this are indicated in the computer output presented in Appendix C. One drawback to the clustering function, Eq. (1), is that the function is not guaranteed to be monotone in the interval. This can happen, for example, if \(\Delta x_0\) is too small and \(\nabla x_f\) too large. Again, the output in Appendix C indicates practical values to choose.

In the case of a nonanalytic body shape \(x\) and \(y\) are read in as a table of values. Here either an axis length or surface arc length is used as a clustering function. If the axis length is chosen, \(x\) is given and corresponding \(y\) values are found from the table of \(x,y\) coordinates using cubic spline interpolation. Alternately, the surface arc length can be used. In this approach the arc length \(s\) is computed from the table of \(x,y\) values, and the length is normalized. A new normalized clustered arc length is then defined, using Eq. (1) with \(s\) (the arc length) in place of \(x\). Both \(x\) and \(y\) are then interpolated from the tables \(x\) versus \(s\) and \(y\) versus \(s\). Again cubic spline interpolation is used.

Finally, a sting or cut may be added to the configuration. Again, the clustering relation, Eq. (1), is used to distribute points along the sting or cut.

---

B. Outer Boundary Formation and Grid Point Distribution

Usually the shape of the outer boundary curve is not as well defined as the inner boundary. However, the shape of the outer curve may be partially prescribed. For example, in Figure 4, the axis of symmetry of the ring airfoil has a specified location. The side and upper boundaries need only be smooth curves far-removed from the body. If, however, the ring airfoil or standard projectile is tested in a wind tunnel, a numerical simulation of the experiment requires a fixed wall outer boundary. In this case the top portion of the outer boundary curve must be specified.

A part of the grid generation problem then is the formation of the arbitrary outer boundary. Here this boundary is built up by connecting contiguous cubic segments, which in the degenerate case can be straight lines. Figures 12a and 12b illustrate two typical outer boundary curves. In Figure 12a three cubic segments make up the boundary $n = n_{\text{max}}$. Each segment, from $a$ to $b$ for example, is formed by specifying the endpoints $x, y$, and angle $\theta$, where $\theta$ is the angle between the curve and the $x$ axis. In the example, Figure 12a, $\theta_a = 90^\circ$, $\theta_b = \theta_c = 0^\circ$ or $180^\circ$ and $\theta_d = 90^\circ$.

The data $x, y, \theta$ at each endpoint determines the shape of the parametric curves

$$\begin{align*}
x &= x_o + \alpha_1 t + \alpha_2 t^2 \\
y &= y_o + \beta_1 t + \beta_2 t^2
\end{align*}$$

which are equivalent to a cubic

$$y = y_o + \gamma_1 (x-x_o) + \gamma_2 (x-x_o)^2 + \gamma_3 (x-x_o)^3$$

The parametric cubic is used because the condition $\frac{dy}{dx} = \infty$ can be specified, e.g., segment bc of Figure 12b has this constraint at both endpoints.

The solution of the parameters $\alpha_1, \alpha_2, \beta_1,$ and $\beta_2$ are given by

$$\begin{align*}
\alpha_1 &= m_o \\
\alpha_2 &= (x_f - x_o) - \alpha_1 \\
\beta_1 &= n_o \\
\beta_2 &= (y_f - y_o) - \beta_1
\end{align*}$$
where

\[
\begin{align*}
    m_0 &= \frac{dx}{dt}|_0 \\
    n_0 &= \frac{dy}{dt}|_0 \\
    m_1 &= \frac{dx}{dt}|_1 \\
    n_1 &= \frac{dy}{dt}|_1
\end{align*}
\]

and

\[
\begin{align*}
    n_0 &= m_0 \frac{dy}{dx}|_0 \\
    m_0 &= n_0 \frac{dx}{dy}|_0 \\
    n_1 &= m_1 \frac{dy}{dx}|_1 \\
    m_1 &= n_1 \frac{dx}{dy}|_1
\end{align*}
\]

The solutions for \(n_0, m_0, n_1,\) and \(m_1\) are conditional insofar that infinite-slopes are avoided. Then the regular solutions are

\[\text{(7a)}\]

i) If \(y_x|_0 < y_y|_0, y_x|_1 < y_y|_1,\) and \(y_x|_1 \neq y_y|_0\)

\[
\begin{align*}
    m_1 &= 2 \left[ (x_f - y_0) - (x_f - x_0) y_x|_0 \right] / (y_x|_1 - y_x|_0) \\
    m_0 &= 2 (x_f - x_0) - m_1 \\
    n_0 &= m_0 y_x|_0 \\
    n_1 &= m_1 y_x|_1
\end{align*}
\]

\[\text{(7b)}\]

ii) If \(y_x|_0 < y_y|_0, y_x|_1 < y_y|_1,\) and \(y_x|_1 y_y|_0 \neq 1\)

\[
\begin{align*}
    m_0 &= 2 \left[ (x_f - x_0) - (y_f - y_0) y_x|_1 \right] / (1 - y_y|_1 y_x|_0) \\
    n_1 &= 2 (y_f - y_0) - m_0 y_x|_0 \\
    n_0 &= m_0 y_x|_0 \\
    m_1 &= n_1 y_y|_1
\end{align*}
\]

\[\text{(7c)}\]

iii) If \(y_x|_0 < y_y|_0, y_x|_1 < y_y|_1,\) and \(y_x|_1 y_y|_0 \neq 1\)

\[
\begin{align*}
    n_0 &= 2 \left[ (y_f - y_0) - (x_f - x_0) y_x|_1 \right] / (1 - y_x|_1 y_y|_0) \\
    m_1 &= 2 (x_f - x_0) - n_0 y_y|_0 \\
    m_0 &= n_0 y_y|_0 \\
    n_1 &= m_1 y_x|_1
\end{align*}
\]
iii) If \( x_0 | y_0 < y_x | x_1, x_1 | y_1 < y_x | 1 \), and \( x_0 | y_1 \neq y_0 | y_0 \)

\[
\begin{align*}
    n_1 &= 2 \left[ (x_f - x_0) - (y_f - y_0) \right] (x_1 | y_1 - x_0 | y_0) \\
    n_o &= 2 (y_f - y_0) - n_1 \\
    m_o &= n_0 x_0 | y_0 \\
    m_1 &= n_1 x_1 | y_1
\end{align*}
\]

Whenever the third constraint is violated (for example in case (i), if \( y_x | y_0 = y_x | 0 \)) a linear curve is used. In this case \( \alpha_2 = \beta_2 = 0 \) and \( \alpha_1 = x_f - x_0, \beta_1 = y_f - y_0 \). The segments ab, bc, and ea of Figure 12b are examples of the straight line segments.

The outer boundary curve is made up of contiguous cubic segments starting from the \( \xi = 0 \) boundary. Points are distributed along this curve either as a uniform distribution of arc length, or as a specified arc length distribution using the previously defined clustering scheme, Eq. (1). Since the true arc length is not specified a priori, precise alignment of points along the outer boundary can be specified only after the cubic segments are specified and the arc length is computed. Without such knowledge a normalized clustering function should be used.

C. Grid Generation and Clustering

The task of generating a grid is undertaken once the boundary curves are specified and points are distributed on the \( \eta = 0 \) and \( \eta_{\text{max}} \) boundaries. Two types of grid generation procedures were used and are discussed below.

In the first case, lines of constant \( \xi \) (i.e., the rays emerging from the body) are formed by simply connecting straight lines from points along \( \eta = 0 \) to points along \( \eta = \eta_{\text{max}} \). The spacing in \( \eta \) along each such line is either uniform or is determined by the relation

\[
\Delta s_k = \Delta s_0 (1 + \epsilon)^{k-1}, \quad k = 1, k_{\text{max}} - 1
\]

Here \( \Delta s_0 \) is the specified constant grid spacing at the inner boundary. The parameter \( \epsilon \) is determined by a Newton-Raphson iteration process so that the sum of the above increments matches the known arc length between the \( \eta = 0 \) and \( \eta = \eta_{\text{max}} \) for points which have the same values of \( \xi \). Figures 13a and 13b illustrate a straight ray grid with clustering in \( \eta \) for a tubular projectile.
In the second case, the grid is generated with elliptic partial differential equations following References 6, 7, 8. The grid generating equations are solved on the specified computational space for unknowns $x_{j,k}$ and $y_{j,k}$:

$$\alpha x_{\xi,\xi} - 2\beta x_{\xi,\eta} + \gamma x_{\eta,\eta} = -\nu^2 (\bar{p} x_{\xi} + \bar{Q} x_{\eta})$$

(9a)

$$\alpha y_{\xi,\xi} - 2\beta y_{\xi,\eta} + \gamma y_{\eta,\eta} = -\nu^2 (\bar{p} y_{\xi} + \bar{Q} y_{\eta})$$

(9b)

where

$$\alpha = x_n^2 + y_n^2, \quad \beta = x_n x_{\xi} + y_n y_{\xi}, \quad \gamma = x_{\xi}^2 + y_{\xi}^2, \quad J = x_{\xi} y_{\eta} - x_{\eta} y_{\xi}$$

and

$$\bar{p} = p_0 e^{-a(n-n_0)} + p_m e^{-a(n-n_{\text{max}})}$$

$$\bar{Q} = Q_0 e^{-b(n-n_0)} + Q_m e^{-b(n-n_{\text{max}})}$$

Here $p_0$, $Q_0$, $p_m$, $Q_m$, $a$ and $b$ are prescribed clustering parameters. Along the $n = 0$ and $n = n_{\text{max}}$ boundaries, $x_{j,k}$ and $y_{j,k}$ have been previously prescribed. Along the $\xi = 0$ and $\xi = \xi_{\text{max}}$, which are either vertical or horizontal lines in the physical space, the following boundary conditions are enforced: either

$$x \text{ is given and } y_{\xi} = 0$$

(10a)

on a vertical boundary, or

$$x_{\xi} = 0 \text{ and } y \text{ is given}$$

(10b)

on a horizontal boundary.

---


For periodic grids as indicated in Figure 7, these boundary conditions in $\xi$ are replaced by the usual periodic relations.

The derivative expressions on the left hand side of Eq. (9) are all differenced with conventional second order central difference operators, that is

$$
\begin{align*}
x_\xi &= (x_{j+1,k} - x_{j-1,k})/(2\Delta \xi) \\
x_\eta &= (x_{j,k+1} - x_{j,k-1})/(2\Delta \eta) \\
x_{\xi\xi} &= ((x_{j+1,k} - 2x_{j,k} + x_{j-1,k})/(\Delta \xi)^2) \\
x_{\xi\eta} &= (x_{j+1,k+1} - x_{j+1,k-1} - x_{j-1,k+1} + x_{j-1,k-1})/(4\Delta \xi \Delta \eta) \\
x_{\eta\eta} &= (x_{j,k+1} - 2x_{j,k} + x_{j,k-1})/(\Delta \eta)^2
\end{align*}
$$

while derivatives of $y$ are treated identically. The Jacobian $J$ is formed with central differencing. The right hand side companion terms to $P$ and $Q$, however, are backward or forward differenced depending on the sign of $P$ and $Q$. If $P$ is positive, $x_\xi$ and $y_\xi$ are forward differenced. The terms $x_\eta$, $y_\eta$ are differenced in the same way.

The one sided differencing for the right side term was chosen assuming $J$ is a constant. Preliminary analysis with local linearization of terms like $J^2 x_\xi$ suggests one sided differencing should also be used in $J$ to keep balanced coefficients. This however has not been evaluated.

The difference equations to Eq. (9) are solved with a successive line overrelaxation (SLOR) procedure. As an initial guess for the relaxation procedure we use the straight line ray procedure previously described. For the most part, if coefficients $P$ and $Q$ are large, the SLOR procedure is very difficult to converge. Consequently, we recommend using the algebraic clustering function, Eq. (8).

In the algebraic clustering approach the elliptic solver is used to generate a grid with $P = Q = 0$. The $x,y$ points along a $\xi =$ constant line are then redistributed along this line as a function of arc length. The clustering function Eq. (8) is used for this purpose. This procedure works quite well and provides excellent control of the grid spacing near the body surface. Further details are given in Reference 8. The grid shown in Figure 14 was generated in this manner.

The elliptic solver need not be used over the entire range in $\xi$. Because of the boundary condition, Eq. (10), the elliptic equations can be joined to a straight ray along any vertical or horizontal boundary line in $\xi$. Figure 15 shows details of such a procedure used in the previous tubular projectile case. Here the $\xi$-region over the tubular projectile is meshed using the elliptic equations while the remainder is meshed with straight rays. After the basic grid is formed, the entire grid is clustered in $\eta$ using Eq. (8).
D. Grid Plotting

An integral part of the grid development program is the ability to plot the computed grid in a timely manner. The plot program which was developed and utilized allowed almost instantaneous viewing of the computed grid. This capability significantly reduced the grid generator development time.

The plot program was written using Tektronix Plot 10 software on the BRL Cyber 176 computer. A program listing is presented in Appendix D.

The only input required for the plotting program is the converged grid file and the minimum and maximum $x,y$ values of the grid. The interactive program uses prompts for the remaining input.

III. DISCUSSION OF RESULTS AND CONCLUDING REMARKS

Figures 10, 13, 14, and 15 give the reader a reasonably clear picture of the capability of the grid generation routine. The other grids classified in Figures 1-8 simply use the same program elements in different arrangements.

An analytic shape that was meshed using the elliptic equation approach is illustrated in Figure 10. Along the body, points are clustered to the nose, boattail junctures, and base. No cuts or stings are used. Only two cubic segments are used to define the outer boundary and along this curve points are uniformly distributed. Solution of Eq. (9) with no additional reclustering completes the grid generation problem. Note that Eq. (10b) is well satisfied along the $\xi = 0$ and $\xi = \xi_{\text{max}}$ axis.

In Figure 13 a grid for a tubular projectile is shown. The body is defined by $x,y$ ordinates and here upper and lower cuts are used. The outer boundary is defined using four cubic segments, two of which degenerate to straight lines. From the trailing edge on back, the point distribution along the outer boundary matches that of the cuts. In this way vertical rays are used over the cut, although this is not required. Straight line rays make up the interior grid, and along these rays points in $n$ are exponentially clustered using Eq. (8). The controlled grid spacing along the body is illustrated in Figure 13b.

The case shown in Figure 14 is similar to Figure 10 only here the grid generated using Eq. (9) was reclustered along lines of constant $\xi$. The grid spacing near the body is now controlled as before.

Finally, the case shown in Figure 15 is similar to that of Figure 13 only now an elliptic solver is used over the airfoil. The cut region is again treated with vertical rays.

Grids for nonaxisymmetric bodies with axisymmetric noses, Figure 6, can be generated as follows. For the axisymmetric nose, Figure 5, the grid is generated in a plane and then spun around the axis forming a three-dimensional grid. The remaining grid can then be generated by taking planar cuts normal to the axis at various increments $\Delta x$ (x aligned with the axis, see Figure
6. At each cut a planar grid is generated and the combinations of these grids form the three-dimensional mesh. At each cross section one generates the 0-type grids shown in Figures 7 and 8 being careful to maintain continuity in x.

Completed computer code documentation is provided in Appendices A and B. Input and output to obtain the grid shown in Figure 15 is included in Appendix C and the plotting code is given in Appendix D.

The modular program developed here has proven to be quite flexible, and should find application in determining grids for various conventional and nonconventional projectile shapes.
REFERENCES


Figure 1. Standard Projectile Grid with Sting

Figure 2a. Cartesian-like Projectile Grid with Sting

Figure 2b. Special Case Isolated Boattail
Figure 3. Standard Projectile Grid with Base, Special Case of 0-Grid with Symmetry

Figure 4. Tubular Projectile Grid or C-Grid
Figure 5. Projectile Blunt Body Grid (Fraction of O-Grid)

Figure 6. Projectile with Symmetric Nose and Nonsymmetric Afterbody
Figure 7. Projectile Cross Section with Periodic B.C. (O-Grid)

Figure 8. Projectile Cross Section with Symmetry Plane (O-Grid)
Figure 9. Mapping from Physical Space to Computational Space
Figure 11. Stretching Function, Points $j$ are Specified along with $x_0$, $x_f$, $\Delta x_0$, and $\nabla x_f$
Figure 12. Outer Boundary Structure and Terminology for Two Classes of Grid
Figure 13a. Overview of Straight Ray Grid for Tubular Projectile
Figure 15. Hybrid Straight Ray and Elliptic Solver Grid Detail for a Tubular Projectile
LIST OF SYMBOLS

\( j \)  \( j \) index value in \( \xi \) direction
\( k \)  \( k \) index value in \( \mu \) direction
\( s \)  arc length
\( \mathbf{x,y} \) physical cartesian coordinates
\( J \) Jacobian of the transformation between the physical and the computational coordinates
\( P,Q \) clustering parameters for the elliptical solver
\( \Delta \) forward finite difference
\( \Delta s \) grid spacing at the inner boundary
\( \nabla \) backward finite difference
\( \varepsilon \) clustering parameter in \( \eta \) direction
\( \theta \) angle between segments of the outer boundary and the \( x \)-axis
\( \xi,\eta \) computational coordinates in the axial and radial directions
\( L \) model length

Subscripts

\( f \) final value
\( o \) initial value
The computer program is a highly modular code. The main program is divided into three parts: inner boundary, outer boundary, and grid generation.

A. Inner Boundary

In forming the inner boundary, subroutine BODY is called to define the body shape and to distribute points along the body. Subroutine BODAN is called for an analytic shape. The user can modify this routine to supply his own body function. Subroutine BODAN calls subroutine BODIS which is the routine that clusters according to Eq. (1). If the body is not analytic, a table of x,y ordinates are read from BODY. These ordinates are normalized and then distributed as a function of axis length (chord) or arc length with calls to BODIS. The newly distributed body points are interpolated from the table of ordinates using cubic splines (subroutine CSPLIN). For example, ordinates of y versus arc length s are interpolated to form y, as a function of the distribution arc length s. Subroutine BODY then returns to main. At this point a sting and/or cut can be added by calls to STING (i.e., a sting as in Figure 1 or upper cut as in Figure 4) and CARTB (i.e., forward cut in Figure 2, lower cut in Figure 4).

Points are distributed along the sting and/or cut using BODIS. In subroutine STING, points are read in from 1 to NGRD. NGRD-1 points are added to the total count in ξ. Likewise in CARTB a set of points along the cut are added to the previous number. The final number of x,y inner boundary points is printed in main.

B. Outer Boundary

MAIN calls subroutine OUTER which forms the outer boundary. Here the cubic segments, as defined by Eq. (4), are read in and joined together. Allowance is made for 8 possible segments. Finally, points are distributed along this boundary as a function of arc length. Either a uniform distribution is used, or again subroutine BODIS is employed. Interpolation of the distributed points is again obtained by cubic splines, but the cubic spline function is restricted within an originally defined segment. Thus in Figure 12b the cubic spline interpolation is not carried from a to c, but is carried a to b, b to c, etc. In this way the discontinuous corner is not spline fit.

C. Grid Generation

Finally MAIN calls subroutine ALGRD. In subroutine ALGRD the straight line ray grid is formed using uniform clustering in n. Any segment of the grid between vertical or horizontal boundaries can then be regenerated using subroutine RELAX to obtain an SLOR solution to Eq. (9). Finally, the grid lines can be exponentially reclustered in n using Eq. (8). The grid is then stored for display or computational purposes.
D. Subsidiary Subroutines

With storage of the grid the program ends. Besides those subroutines described above, several other routines are called. Subroutine CLUST is called by BODIS and this is the routine that literally corresponds to the distribution function, Eq. (1). Subroutines TRIB and TRIP are routines for the solution of the tridiagonal matrix which must be inverted in the successive line overrelaxation procedure used by RELAX.

The TRIB routine is for conventional tridiagonal matrices, the TRIP routine is for periodic tridiagonal matrices. Finally, subroutine INIPQ is used to input $P$ and $Q$ of Eq. (9). Use of these terms is not currently recommended.

Two subroutines are called from BODAN to describe the blunt, secant-ogive, nose projectile. SCALC computes x-values associated with the nose cap. The fuse height is used to vary the degree of bluntness. SECANT is then called to provide the analytic functions used to compute the points along the remaining body configuration.

Subroutine GSPIN is called only when a three-dimensional grid is required. Both three- and two-dimensional grids are written, however, the former is used for flowfield computations and the latter is used for plotting.
PROGRAM MAIN (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE9=TAPE10)

C CHECK FOR 30 GRID
COMMON JMAX, KMAX, JG, K, NBOO, JBOO
COMMON (300000, XX1(100), YY1(100), XS(100), YS(100), SS(100))
1, T(100), TS(100)
COMMON /CNP0/ (XX1(100), YY1(100))

C READ (6,100) I30,NO,LMAX
WRITE (6,100) I30,NO,LMAX

C DISTRIBUTE POINTS ALONG INNER BOUNDARY
WRITE (6,70)
CALL BODY
READ (5,60) NFLAG
IF (NFLAG.LT.0) GO TO 10
READ (9,60) NCGRO,NCART
WRITE (6,80) NCGRO,NCART
IF (NCGRO.GT.0) CALL STING (NCGRO)
IF (NCART.GT.0) CALL CARTB (NCART)
JMAX-JBOO
MWRITE 16,90 JMAX
IF (NCGRO.GT.0.OR.NCART.GT.0) GO TO 30
GO TO 10

10 WRITE (6,100)
2 CONTINUE
30 CONTINUE

C FORM OUTER BOUNDARY
WRITE (6,120)
READ (5,60) NSEGS, IOUTO
WRITE (6,130) JMAX, NSEGS
CALL OUTER (NSEGS, IOUTO)

C GRID GENERATION
WRITE (6,140)
CALL ALGRO (ISTOR)

C FORM 30 GRID. LMAX IS CIRCUMFERENTIAL DIRECTION
IF (I3D.EQ.1) CALL GPIN (I3D, ND, ISTOR, LMAX)
40 STOP

50 FORMAT (1H0,11H30,NO=LMAX+15)
60 FORMAT (10X)
70 FORMAT (1H1,13HINNER BOUNDARY INNER BOUNDARY INNER BOUNDARY)
80 FORMAT (1H1,13HNCGRO,NCART +15)
90 FORMAT (1H1,13HNCART +15)
100 FORMAT (1H1,13HNCART +15)
110 FORMAT (1H1,13HSS +15)
120 FORMAT (1H1,13HOUTER BOUNDARY OUTER BOUNDARY OUTER BOUNDARY)
130 FORMAT (1H1,13HJMAX+15)
140 FORMAT (1H1,13HGRID GENERATION GRID GENERATION )
END

37
SUBROUTINE ALCR (ISTOR)

COMMON JMAX, KMAX, JM, KM, NBDD, JBDD
COMMON JBDDY, XX(10DD), YY(10DD), XS(10DD), YS(10DD), SS(10DD), SS(10DD)
1, T(10DD), TS(10DD)
COMMON GRID/ X(8DD,6D), Y(8DD,6D)
COMMON ARRAY/ A(100), B(100), C(100), D(100), F(100), H(100)

FORM ALGEBRIC GRID OR ELLIPTIC EQ. GENERATE GRID
READ (5,153) KM, KMAX, ITERM, IPER, NCLUS, ISTOR, JELLI
WRITE (6,150) KM, ITERM, IPER, NCLUS, ISTOR, JELLI
K=KMAX-1
READ (5,170) DS, OMEGA
WRITE (6,170) DS, OMEGA
STRAIGHT RAY GRID USED IF ITERM .LE. 0
OTHERWISE USED AS INITIAL GUESS
GO TO 10
40 D0 = R/(KM-1)
GO TO 20
10 EPS = EPSIL(2.0,0.0,4.0,0.0,2.0,0.0,2.0,0.0,2.0,0.0,2.0)
DD = DS
20 DD = DS
X(J+1) = X(J)
Y(J+1) = Y(J)
TR = 0.
DD 30 K = 2, KMAX
TR = TR + DS*(1.0 + EPS)**(K-2)
TT = TR/R
X(J,K) = X(J) + (X1 - X(J)) * TT
Y(J,K) = Y(J) + (Y1 - Y(J)) * TT
30 CONTINUE
40 CONTINUE
C IF (ITERM .LT. 0) GO TO 70
C ELLIPTIC P.D.E. GRID GENERATION SCHEME
C IF (JELLI .NE. 1) GO TO 50
CALL RELAX (ITERM, IPER, JMAX, OMEGA)
GO TO 70
50 DD 60 LL = 1, JELLI
READ (5,153) JI, JF
WRITE (6,150) JI, JF
60 CALL RELAX (ITERM, IPER, JI, JF, OMEGA)
70 CONTINUE
C CLUSTERING OPTION
C IF (NCLUS .LT. 0) GO TO 130
DD 120 J = 1, JMAX
T(1) = 0.
DD 80 K = 2, KMAX
DO 110 K = 1, KMAX
  X(J,K) = X(J,K-1) + DS*(1 + EPS)*N
  DO 100 K = 1, KMAX
    X(J,K) = X(J,K-1)
    CALL CSPLIN (X(J,K), Y(J,K-1), X(J,K), Y(J,K), M, 2, EPS, Y(J,K))
  END
  DO 110 K = 2, KMAX
  X(J,K) = X(J,K-1)
  CALL CSPLIN (X(J,K), Y(J,K-1), X(J,K), Y(J,K), M, 2, EPS, Y(J,K))
  110 CONTINUE
  120 CONTINUE

SURPRESS THE PRINTOUT
K = KMAX/2
DO 20 J = 1, JMAX
  WRITE (6, 203) X(J,JMAX), Y(J,JMAX), X(J,JMAX), Y(J,JMAX), X(J,JMAX), Y(J,JMAX)
  WRITE (6, 203) X(J,JMAX), Y(J,JMAX), X(J,JMAX), Y(J,JMAX), X(J,JMAX), Y(J,JMAX)
  20 CONTINUE

OPTIONAL STORE OF DATA
IF (ISTOR.LT.0) GO TO 140
  WRITE (9, 9185) (X(J,K), J = 1, JMAX, K = 1, KMAX), (Y(J,K), J = 1, JMAX, K = 1, KMAX)
  140 CONTINUE
  RETURN

150 FORMAT (8I5)
160 FORMAT (1HO, 6I8, 6I5)
170 FORMAT (1HO, 6I8, 6I5)
180 FORMAT (1HO, 6I8, 6I5)
190 FORMAT (1HO, 6I8, 6I5)
END
SUBROUTINE BOOAN (ISECS)
COMMON /CALC/ X0, XATF, AS, ICI, FLAG, POYOX, FXF, REN
COMMON /MP5/ XI, X2, X3, X4, RAD, DYDX, CHORD, FUSE, AS, RADS
COMMON /MAX/ KMAX, JM, KN, MRDE, JBOO
COMMON /BOUDT/ YI, Y2, Y3, Y4, RAO, THETA, CHORD, FUSE, AS, RAOS
COMMON /BOUDT/ X1000, Y1000, X51000, Y51000, SSS1000, S11000
1 = T1000, TS1000
READ 15, 103 TAU, FLAG
WRITE (6, 110) TAU, FLAG
JWRITE = JBOO
ICT = 0
IF (FLAG .LE. 1.0) CALL SCALC
CALL BOOIS (ISECS, JM, JBOO)
DO 10 J = 1, JBOO
10 XX(J) = S(J)

C ANALYTIC BODY SHAPE
IF (FLAG .LE. 0.1) GO TO 30
DO 20 J = 1, JBOO
20 XI(J) = S(J)/S(J)
GO TO 90
30 CONTINUE
IF (FLAG .LE. 0.1) GO TO 40
CALL SECANT
GO TO 90

PROJECTILE WITH TANGENT OGIVE, CYLINDER, BOATTAIL, CIRCULAR CAP
40 READ 15, 103 XI, X2, X3, X4, RAD, THETA, CHORD
THETA = THETA * 0.17433
DYDX = TAN(THETA)
BASE = RAD + (X4 - X3) * DYDX
XE = BASE * DYDX * SORT1(1 + DYDX**2)
S = XE / XE

C NONDIMENSIONAL OPTION
IF (CHORD .GE. 0.1 CHORD = 3 - XI
CHORD = ABS(CHORD)
WRITE (6, 120) XI, X2, X3, X4, RAD, THETA, CHORD
RCH = 1 / CHORD
XI = XI / RCH
X2 = X2 / RCH
X3 = X3 / RCH
X4 = X4 / RCH
X5 = X5 / RCH
BASE = BASE / RCH
XE = XE / RCH
RAD = RAD / RCH
DO 80 J = 1, JBOO
80 IF (XX(J) .GE. X2) GO TO 50
XDG = X2 - XI
XBAR = (XX(J) - XI) / XDG
VLAM = XDG / RAD
V52 = VLAM**2
RBAR = V52 + 0.25
RADI = 1 - V52*(1 - XBAR)**2 / (RBAR**2)
YY(J) = 1.2 - 2 * RBAR**1.5

50 IF (XX(J) .GE. X3) GO TO 60
YY(J) = RAD
GO TO 80
60 IF (XX(J) .GE. X4) GO TO 70
YY(J) = RAD - (XX(J) - X3) / DYDX
GO TO 80
70 R5 = BASE**211 * (1 + DYDX**2)
XBAR = (XX(J) - X4)
RADI = R5 * XBAR / BASE * DYDX**2
YY(J) = SORT(RADI)
80 CONTINUE
END OF PROJECTILE
C END OF PROJECTILE
90 CONTINUE
RETURN
C
102 FORMAT (8F10, 0)
110 FORMAT (10H0, 8HTAU, ?LAC, 2F1*.5)
111 FORMAT (10H0, 8HTAU, ?LAC, 2F1*.5)
120 FORMAT (10H0, 8HTAU, ?LAC, 2F1*.5)
END
SUBROUTINE BOOIS (ISEGS, JWRITE)
COMMON /CALC/ X0, XATF, BS, ICT, FLAG, PDYDX, FXF, RPM
COMMON JMAX, KMAX, JN, KM, NBOO, JBOO
COMMON /BOUND/ XX100, YY100, XS100, YS100, SS100, TS100
COMMON /BOY/ JOOII, YOOII, JOOII, YOOII, JOOII, YOOII
>'I OOOI. T(IOO), T(SIOO)
COMMON /BOY/ XX100, Y(Y100), Z(S100)
S DISTRIBUTION ON BOY

10 WRITE (6,30) JI, JF, XI, XF, OXI, OXF
CALL CLUST (JI, JF, XI, XF, OXI, OXF, S)
10 CONTINUE
RETURN

20 FORMAT (2I5,6F10.0)
30 FORMAT (1H0,2I5 JI, JF, XI, XF, OXI, OXF, 2I5, 4F12.5)
40 FORMAT (1H1, 10F11.5)
END
SUBROUTINE BODY
COMMON JMAX, KMAX, JM, KM, NBOD, JBOO BODY 2
COMMON /BODY/ X(100), Y(100), X5(100), Y5(100), SS(100), S(100) BODY 3
L, T(100), TS(100) BODY 4
COMMON /ARRAY/ A(100), B(100), C(100), D(100), F(100), H(100) BODY 5
COMMON /COMP/ X1(100), Y(100) BODY 6
COMMON /IXORD/ XX(100), TY(100), X5(100), YS(100), SS(100), SHOO BODY 7
C
READ (5,110) NBOD, JBOO, IXORS, ISEG
WRITE (6,120) NBOD, JBOO, IXORS, ISEG
C
IF NBOD IS NEGATIVE, ANALYTIC SHAPE IS USED
IF (NBOD.GE.0) GO TO 10
NBOD=-NBOD
GO TO 10
CALL BOOAN (ISEGS)
GO TO 110
C
10 CONTINUE
WRITE (6,130)
READ (5,150) CHORD
WRITE (6,140)
READ (5,150) CHORD
READ CARDS IN CLOCKWISE NOSE TO TAIL
00 20 1-J-1, NBOD
WRITE (5,150) X(J), Y(J)
WRITE (6,160) J, X(J), Y(J)
XX(J)-X(1)*SS(J)/NBOD
Y(J)-Y(J)/NBOD
20 CONTINUE
C
COMPUTE NORMALIZED ARC LENGTH TO USE AS A MONOTONE PARAMETER
SS(J)=0.
00 30 J=2, NBOD
30 SS(J)=SS(J)+SQRT((X(J)-X(J-1))**2+(Y(J)-Y(J-1))**2)
00 40 J=2, NBOD
CALL BOOAN (ISEGS)
30 SS(J)=SS(J)/SS(NBOD)
C
C
COMPUTE A NORMALIZED CLUSTERED PARAMETRIC FUNCTION FOR DISTRIBUTION
OF BODY POINTS
JWRITE=JBOO
CALL BOOAN (ISEGS, JWRITE)
SAY=SEL
SS(J)=0.
00 50 J=2, NBOD
50 SS(J)=SS(J)+ABS((S(J)-SAY))
SAY=S(J)
50 SS(J)=SS(J)/SS(NBOD)
C
OPTION WHETHER TO SET X EQUAL TO S DISTRIBUTION
IF (IXORS.LT.0) GO TO 70
CALL CSPLIN (S, SS, X, Y, A, B, C, D, F, H, 1, NBOD, 1, NBOD)
CALL CSPLIN (S, SS, X, Y, A, B, C, D, F, H, 1, NBOD, 1, NBOD)
GO TO 90
70 CONTINUE
DO 80 J=1, 1+BOD
80 XX(J)=X(J)*(SS(NBOD)-X(J))
C
CALL CSPLIN (XX, YY, XX, YY, A, B, C, D, F, H, 1, NBOD, 1, NBOD)
C
90 CONTINUE
WRITE (6,170)
DO 100 J=1, 1+BOD
WRITE (6,180) J, XX(J), YY(J)
100 CONTINUE
RETURN
C
110 FORMAT (4I5)
L=15, 2HNBOD, JBOO, IXORS, ISEG, +4I5
120 FORMAT (1H0, 3H X, Y INPUT DEFINING BODY... SUB. BODY )
130 FORMAT (1H0, 3H X, Y NORMALIZING CHORD LENGTH INPUT ,F13.5)
140 FORMAT (1H0, 3H X, Y DISTRIBUTED X AND Y )
150 FORMAT (2F10.0, 1H0, J, X, Y, ON BODY ,15, 2F14.5)
160 FORMAT (1H0, 3H J X, Y , ON BODY 15, 2F14.5)
170 FORMAT (1H1, 32H J AND BODY DISTRIBUTED X AND Y )
42
SUBROUTINE CSPIN (I3D, NO, ISTOR, LMAX)
COMMON JMAX, KMAX, JM, KL, Y80D, J80D
COMMON /GRID/ X(80,60), Y(80,60)
COMMON /GRID30/ X3(240,60), Y3(240,60), Z3(240,60)
LEVEL 2:X3,Y3,Z3
PI=4.*ARCTAN(1.)

DS=PI/(LMAX-3)
N1=KMAX*LMAX
DO 10 K=1,NMAX
OT=-2.*OS
DO 20 L=1,NMAX
OT=OT+DS
KL=(K-1)*N3+L
DO 30 J=1,KMAX
Y3(KL,J)=Y(J,K)*SIN(OT)
Z3(KL,J)=Y(J,K)*COS(OT)
X3(KL,J)=X(J,K)
10 CONTINUE
IF (ISTOR.E.0) GO TO 40

REMIND 9
WRITE (9) ((X3(KL,J),XL*lt41,J>l,jn*X),L(r3(KL,J)<KL+1,N1,J>1,JMAX),I(23IKL.J)<KL+1,N1.J>1,JMAX)

C
C REWRITE 3D DATA FOR 2D PLOTTING
C
DO 30 J=1,JMAX
DO 20 KL=2,N1*LMAX
N1=1*(1-23/KL)+N1
X13,N)=X3(KL,J)
Y(J,N)=23(KL,J)
20 CONTINUE
30 CONTINUE
WRITE (10) ((X(J,N),J=1,JMAX),N=1,<MAX),((Y(J,N),J=1,JMAX),N=1,
1*KMAX)

40 CONTINUE
WRITE (6,90)
DO 60 L=2,LMAX+1
DO 50 K=1,LMAX+1
KL=(K-1)*ND*L
WRITE (6,70) L,K,L,K+1
KD=KD+1
50 WRITE (6,80) J,X3(KL,J),Y3(KL,J),Z3(KL,J),J,X3(KL2,J),Y3(KL2,J),
L Z3(KL2,J)
60 CONTINUE
RETURN

70 FORMAT (14,H9,J +2MK=12,3H L=12,9X,LMX=0X+12X+12X+12X+13X)
3H J=3MK=12,3H L=12,9X,LMX=10X+10X+11X+12X)
80 FORMAT (1H +1X+1X+1X+1X+1X+1X+1X+1X+1X+1X+1X+1X+1X+1X+1X)
15*2X*F10.5)
90 FORMAT (1HL)
END
SUBROUTINE OUTER (NSEGS, IOUTD)
COMMON JMAX, KMAX, JX, JY, NBOI, JBO0
COMMON /BOUOY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)
COMMON /ARRAY/ A(100), B(100), C(100), D(100), F(100), H(100)
COMMON JMAX, KMAX, JX, JY, NBOI, JBO0
COMMON /BOUOY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)
COMMON /ARRAY/ A(100), B(100), C(100), D(100), F(100), H(100)

THIS PROGRAM FORMS AN OUTER GRID BOUNDARY USING CONTIGUOUS CUBIC SEGMENTS. NUMBER OF SEGMENTS IS NSEGS. POINT AND SLOPE ARE INPUT AT THE ENDS OF A SEGMENT. SLOPE IS AN ANGLE IN DEGREES. PARAMETRIC CUBICS USED TO PERMIT ANY SLOPE THETA = 90, -90, ET CTC. INITIAL LOGIC DETERMINES CUBIC COEFFICIENTS OF EACH SEGMENT. REMAINING LOGIC DISTRIBUTES POINTS ALONG OUTER BOUNDARY USING ARC LENGTH AS DISTRIBUTION FUNCTION. THUS TWO PARAMETRIC VARIABLES ARE USED. FINDING X, Y SO CUBIC SEGMENTS CAN BE DISJOINT IN SLOPE IS MESSY. A SINGLE SPLINE INTERPOLATION CANNOT BE USED OVER THE COMBINED SEGMENTS BECAUSE OF POSSIBLE SLOPE DISCONTINUITY.

DIMENSION JAO(8), JBO(8)
DIMENSION CAO(8), CA1(8), CA2(8), CB0(8), CB1(8), CB2(8), CA3(8)
DIMENSION NBO(20), JSIS(20), JAO(20), JBO(20), CAO(20), CA1(20), CA2(20), CB0(20), CB1(20), CB2(20)
READ (5, 240) XO, YO, XI, Y1, THO, TH1
WRITE (b, Z50) XA, RA, XI, YI, THO, TH1
RTHO = 0.017 * 3292 * TH0
RTH1 = 0.017 * 3292 * TH1
XI = XI - XO
ETA = YI - Y0
SET MFLAG, LOGIC CHIEFLY USED TO AVOID INFINITE DY/OX, USES DX/DY=0
TA = SIN(RTH0)
TB = SIN(RTH1)
TC = COS(RTH0)
TD = COS(RTH1)
IF (ABS(TA).GT.ABS(TB)) GO TO 13
MFLAG = 1
IF (ABS(TC).GT.ABS(TD)) MFLAG = 2
GO TO 20
10 MFLAG = 3
IF (ABS(TC).GT.ABS(TD)) MFLAG = 4
GO TO 20
20 CONTINUE

DETERMINE COEFFICIENTS FOR PARAMETRIC CUBICS
SET UP LINEAR COEFFS. FIRST. INDEFINITE CUBIC DEFAULTS TO LINEAR
A1 = XI
A2 = 0.
B1 = ETA
B2 = 0.
GO TO (10, 40, 50, 60), MFLAG
30 OYDX = TA, ETA
OYDX = TC
TEST = OYDX - OYDX
IF (ABS(TEST).LT.0.0005) GO TO 80
SM1 = 2. * ETA / XI / TEST
SM2 = SM1
SM3 = SM0 * OYDX
GO TO 70
40 OXOXO=TA/T8
   OX0Y1=TO/TC
   TEST=1-OXOX0*OX0Y1
   IF (ABS likely TEST).LT.0.0005) GO TO 80
   SNO=2.*XI-OX0Y1*ETA/TEST
   SNO=SNO*OXOXO
   GO TO 70
50 OXOXO=TA/T8
   OX0Y1=TO/TC
   TEST=1-OXOX0*OX0Y1
   IF (ABS likely TEST).LT.0.0005) GO TO 80
   SNO=2.*XI-OX0Y1*ETA/TEST
   SNO=SNO*OXOXO
   GO TO 70
60 OXOXO=TA/T8
   OX0Y1=TO/TC
   TEST=1-OXOX0*OX0Y1
   IF (ABS likely TEST).LT.0.0005) GO TO 80
   SNO=2.*XI-OX0Y1*ETA/TEST
   SNO=SNO*OXOXO
   GO TO 70
70 A1=SNO
   A2=XI-SNO
   B1=SNO
   B2=ETA-SNO
   80 CONTINUE
   JNBR=25
   C
   COMPUTE NUMERICAL ARC LENGTH AS A PARAMETER. EXACT ARC LENGTH IS
   POSSIBLE BUT INVERSE PROCESS IS NOT
   GO 90 J=1,JNBR
   TT=JNBR-1
   TT=(J-1)/TT
   XI(J)=XI*TT+(A1+A2*TT)
   Y(J)=Y0*TT+(B1+B2*TT)
   SARC=0.
   C
   NOTE ... COULD USE SUM OF SQUARES AS PARAMETER RATHER THAN ARC LENGTH
   IN THIS WAY WE CAN AVOID SQUARE ROOT CALCULATION MUST USE EVERY
   WHERE
   GO 100 J=2,JNBR
100 SARC=SARC+SQRT((XI(J)-XI(J-1))**2+(Y(J)-Y(J-1))**2)
   C
   DATA FOR EACH CUBIC SEGMENT
   CAL(N)=A1
   CAR(N)=A2
   CB(N)=B1
   CB2(N)=B2
   CA(N)=X0
   CR(N)=Y0
   CARC(N)=SARC
   WRITE (6,200) X0,A1,A2,Y0,B1,B2,SARC
   C
   110 CONTINUE
   C
   CUBICS DETERMINED. NOW DISTRIBUTE POINTS
TOTAL OUTER ARC LENGTH
SARC=0.
00 120 N=1,NSEGS
120 SARC=SARC+CARC(N)
WRITE (6,270) SARC

DEFINE A UNIFORM OUTER DISTRIBUTION ARC LENGTH
RH=1./(JMAX-1)
00 130 J=1,JMAX
130 SSI(J)=(J-1)*RH

OPTIONAL USE OF CUBIC SEGMENTS TO CLUSTER
IF (IOUTD.LE.O) GO TO 190
CALL SODIS (IOUTD,JNAX)
00 140 J=1,JMAX
140 SSI(J)=S(J-1)/S(JMAX)-S(J)
190 CONTINUE
WRITE (6,290) (SSI(J),J=1,JMAX)

NORMALIZE OUTER ARC LENGTH SEGMENTS TO SCALE OF DISTRIBUTION ARC LENGTH
CA=0.
00 160 N=1,NSEGS
CA=CA+CARC(N)/SARC
160 CARC(N)=CA
WRITE (6,290) (CARC(N),N=1,NSEGS)

FORM J INDICES LIMITS WITHIN A SEGMENT
N=1
J(N)=1
00 180 N=1,NSEGS
IF (SSI(J)>CARC(N)) GO TO 170
N=N+1
J(N)=J
170 J(N)=J
180 CONTINUE
OD 190 N=1,NSEGS
WRITE (6,300) J(N),J(N)
190 CONTINUE

FORM PARAMETRIC ARRAYS FROM DISTRIBUTED PARAMETRIC ARRAY,
USE IT TO DETERMINE X,Y WITHIN A OUTER SEGMENT CURVE.
SPLINE REQUIRES ABOUT 5 POINTS IN AN INTERVAL
S(1)=0.
RT=1./(JMAX-1)
00 220 N=1,NSEGS
T(N)=S(N)
IF (N.GT.1) S(N)=CARC(N-1)
X(N)=CA0(N)
Y(N)=CB0(N)
00 240 J=2,JMAX
TT=(J-1)*RT
T(J)=TT
X(J)=CA0(J)+TT*CA0(J)+TT*CA0(J+1)
Y(J)=CB0(J)+TT*CB0(J)+TT*CB0(J+1)
05=SORT( (X(J)-X(J-1))+Y(J)-Y(J-1))
5(J)=5(J-1)+5(J)/SARC
WRITE (6,280) T(J),X(J),Y(J)
J1=JA(N)
J2=JB(N)
CALL CSPLIN (SS,TS,TA,B,C,D,F,H,J1,J2,1,JNBR)
DO 210 J=J1,J2
TT=TS(J)
XS(J)=CAO(N)+TT*(CA1(N)+TT*CA2(N))
YS(J)=CB0(N)+TT*(CB1(N)+TT*CB2(N))
WRITE (6,280) TS(J),XS(J),YS(J)
210 CONTINUE
220 CONTINUE
DO 230 J=1,JMAX
WRITE (6,310) XS(J),YS(J)
230 CONTINUE
RETURN
240 FORMAT (8F10.0)
250 FORMAT (1H0,OUTER 173)
260 FORMAT (1H0,OUTER 174)
270 FORMAT (1H0,OUTER 175)
280 FORMAT (1H0,OUTER 176)
290 FORMAT (1H0,OUTER 177)
300 FORMAT (1H0,OUTER 178)
310 FORMAT (1H0,OUTER 179)
320 FORMAT (1H0,OUTER 180)
330 FORMAT (1H0,OUTER 181)
340 FORMAT (1H0,OUTER 182)
350 FORMAT (1H0,OUTER 183)
360 FORMAT (1H0,OUTER 184)
370 FORMAT (1H0,OUTER 185)
380 FORMAT (1H0,OUTER 186)
390 FORMAT (1H0,OUTER 187)
400 FORMAT (1H0,OUTER 188)
410 FORMAT (1H0,OUTER 189)
420 FORMAT (1H0,OUTER 190)
430 FORMAT (1H0,OUTER 191)
440 FORMAT (1H0,OUTER 192)
450 FORMAT (1H0,OUTER 193)
460 FORMAT (1H0,OUTER 194)
470 FORMAT (1H0,OUTER 195)
480 FORMAT (1H0,OUTER 196)
490 FORMAT (1H0,RETURN 173)
SUBROUTINE RELAX (ITER, IPER, J1, JF, OMEGA)
COMMON JMAX, KM, JM, KM, NB33, JD00
COMMON XR(JB00, YR(JB00), TB00, SB00)
COMMON XR(JB00, YR(JB00), TB00, SB00)
COMMON /GRID/ X(80,80), Y(80,80)
COMMON /ARRAY/ A(1100), B(1100), C(1100), D(1100), F(1100), H(1100)
DIMENSION IP(142), IR(142)
DIMENSION G(100)
COMMON /SOURCE/ P(80,2I, 0(80,2), PFAC(2), OFAC(2))

C SLOR SOLUTION OF ELLIPTIC GRID GENERATION EOS.
C DEL XI AND DEL ETA = 1.0
J1=1
J2=JMAX
IF (IPER .GT. 0) GO TO 10
J1=J1-1
J2=JF-1
10 CONTINUE
CALL INIPO
ITER=0
KK=KMAX-1
C SET PERIODIC INDICES
OD 20 J=1,JMAX
IP(J)=J
20 IR(J)=J-1
IP(JMAX)=1
IR(1)=JMAX
C FORM DIFFERENCE EXPRESSIONS AND TRIDIAGONALS
30 ITER=ITER+1
RSUM=0.
OD 160 KK=2,KM
K=KM-2-KK
KP<K-1
KR<K-1
CPI=EXP(11-K)*PFAC(1)
CP2=EXP((K-KMAX)*PFAC(2))
CG1=EXP((1-K)*PFAC(1))
CG2=EXP((K-KMAX)*PFAC(2))
OD 40 J=J1,J2
JP=JP+1
JR=IR(J)
XXD=X(JP,*)-X(JR,*),5
XYD=Y(JP,*)-Y(JR,*),5
YED=Y(JP,KP)-Y(JR,KP),5
AD=XED**2+YED**2
BD=2.*(XXD*XED+YXD*YED)
GD=XXD**2+YXD**2
XXD=XXD+25.*(X(JP,KP)-X(JR,KP))*X(JR,KP)
YXD=YXD+25.*(Y(JP,KP)-Y(JR,KP))*Y(JR,KP)
AD(J)=AD
BD(J)=BD+AD*G-AD*G
C(J)=AD
F(J)=BD*XXD*GD*(X(JP,KP)-X(JR,KP))
G(J)=BD*XXD*GD*(Y(JP,KP)-Y(JR,KP))
C SOURCE TERMS
40
40 CONTINUE
C
IF (IPER.GT.0) GO TO 130
C
SET B.C. AND INVERT
C
XI MIN AND MAX PLANES MUST BE X OR Y CARTESIAN PLANES
C
OUTFLOW B C ON X
C
TEST WHETHER XI PLANE IS X OR Y = CONSTANT PLANE
IF (ABS(Y(JF,KMAX) - Y(JF,1)).LT.0.001) GO TO 50
C
OUTFLOW XI-PLANE TAKEN AS X=CONSTANT PLANE
A(JF) = 0.
B(JF) = -1.
C(JF) = 0.
F(JF) = X(JF,1)
GO TO 60
C
OUTFLOW XI-PLANE TAKEN AS Y= CONSTANT
50 A(JF) = 1.
B(JF) = -1.
C(JF) = 0.
F(JF) = (-X(JF-1,K1)+X(JF-2,K1))/3.
GO TO 60
C
INFLOW B C ON X
C
IF (ABS(Y(JJ,KMAX) - Y(JJ,1)).LT.0.001) GO TO 70
C
INFLOW XI-PLANE TAKEN AS X = CONSTANT PLANE
A(JJ) = 0.
B(JJ) = -1.
C(JJ) = 0.
F(JJ) = X(JJ,1)
GO TO 80
C
INFLOW XI-PLANE TAKEN AS Y = CONSTANT PLANE
A(JJ) = 0.
B(JJ) = 1.
C(JJ) = 0.
F(JJ) = (-X(JJ+1,K1)+X(JJ+2,K1))/3.
80 CALL TRIB (A+B*C+D+F+JF+JF)
C
Y B.C. AND INVERSION
IF (ABS(Y(JF,KMAX) - Y(JF+1)).LT.0.001) GO TO 90
C
OUTFLOW XI-PLANE TAKEN AS X=CONSTANT PLANE
A(JF) = 1.
B(JF) = -1.
C(JF) = 0.
G(JF) = (-Y(JF-1,K1)+Y(JF-2,K1))/3.
GO TO 100
OUTFLOW XI-PLANE TAKEN AS Y = CONSTANT PLANE

10 A(JF) = 0.
B(JF) = 1.
G(JF) = Y(JF+1).

IF (ABS(Y(JJ+1,KMAX) - Y(JJ+1),LT,0.001) GO TO 110

INFLOW XI-PLANE TAKEN AS X = CONSTANT PLANE

A(JJ) = 0.
B(JJ) = 1.
C(JJ) = Y(JJ+1).
G(JJ) = Y(JJ+1),X(1)-Y(JJ+2,K))/3.
GO TO 120

INFLOW XI-PLANE TAKEN AS Y = CONSTANT PLANE

A(JJ) = 0.
B(JJ) = 1.
C(JJ) = 0.
G(JJ) = Y(JJ+1).
GO TO 120

CALL TRIP (A,B,C,D,G,JF+1)
GO TO 140

CONTINUE

PERIODIC B.C.

CALL TRIP (A,B,C,D+1,JMAX+1)
CALL TRIP (A,B,C+D,JNAX)

RELAXATION UPDATE

GO 150 J=JF+1
X(J) = F(J)-X(J-1)
X(J,J) = X(J,J)+OMEGA*XC
Y(J,J) = X(J,J)+OMEGA*Y(J)
RSUM = RSUM + ABS(XC) + ABS(Y(J))
GO TO 160

IF (ITER/LIM) = 10, LT, ITER) GO TO 170
WRITE (6,180) ITER, RSUM
GO TO 170

CONTINUE

IF (ITER.LT.ITER) GO TO 30
RETURN

FORMAT (1H,26H ITERATION NBR AND RSUM *15*E12.5)
END

RELAX 116
RELAX 117
RELAX 118
RELAX 119
RELAX 120
RELAX 121
RELAX 122
RELAX 123
RELAX 124
RELAX 125
RELAX 126
RELAX 127
RELAX 128
RELAX 129
RELAX 130
RELAX 131
RELAX 132
RELAX 133
RELAX 134
RELAX 135
RELAX 136
RELAX 137
RELAX 138
RELAX 139
RELAX 140
RELAX 141
RELAX 142
RELAX 143
RELAX 144
RELAX 145
RELAX 146
RELAX 147
RELAX 148
RELAX 149
RELAX 150
RELAX 151
RELAX 152
RELAX 153
RELAX 154
RELAX 155
RELAX 156
RELAX 157
RELAX 158
RELAX 159
RELAX 160
SUBROUTINE INIPO
COMMON JMAX, KMAX, JM, KM, N800, J800
COMMON /SOURCE/ P(80,2), Q(80,2), PFAC(2), QFAC(2)
DIMENSION PC(1), QC(1)

C
READ (5,20) PFAC(1), QFAC(1), PFAC(2), QFAC(2)
READ (5,20) PC(1), QC(1), PC(2), QC(2)
WRITE (6,30) PFAC(1), QFAC(1), PFAC(2), QFAC(2)
WRITE (6,40) PC(1), QC(1), PC(2), QC(2)

C
DO 10 N=1,Z
DO 10 J=1,JMAX
P(J,N)=PC(N)
10 Q(J,N)=QC(N)
RETURN

C
20 FORMAT (8F10.0)
30 FORMAT (1H0, ' EXPONENT COEFFICIENTS FOR SOURCE TERMS, PFAC, QFAC
40 FORMAT (1H0,16H PCl, QCl, PC2, QC2, 4F13.5)
END

C
SUBROUTINE CLUST (JI,JF,XI, XF,0X1,0XF,SJ)
DIMENSION S(1)

C
XI=XF-XI
H=1./((JF-JI)
H2=H*H
H3=H2*H
C=(0XF*0X1+2.*H*HFX1)/(H-3.*H2+2.*H3)
B=(0XI-H*HFX1-C+H3-H2)/(H2-H)
A=HFX1-B-C

C
DO 10 J=1, JF
X=J-JI)*H
10 S(J)=X*X*(A*X*(B*C*X))
RETURN
END

RELAX 161
RELAX 162
RELAX 163
RELAX 164
RELAX 165
RELAX 166
RELAX 167
RELAX 168
RELAX 169
RELAX 170
RELAX 171
RELAX 172
RELAX 173
RELAX 174
RELAX 175
RELAX 176
RELAX 177
RELAX 178
RELAX 179
RELAX 180
RELAX 181
RELAX 182
RELAX 183
WORKS 2
WORKS 3
WORKS 4
WORKS 5
WORKS 6
WORKS 7
WORKS 8
WORKS 9
WORKS 10
WORKS 11
WORKS 12
WORKS 13
WORKS 14
WORKS 15
WORKS 16
WORKS 17
FUNCTION EPSIL (FMX,FMIN,DFM,FPT,FPCC,ICCL,NCALL)

THIS SUBROUTINE APPLIES A NEWTON-RAPHSON ROOT-FINDING
TECHNIQUE TO FIND A VALUE OF EPSILON FOR A PARTICULAR USE
OF THE EXPONENTIAL STRETCHING TRANSFORMATION.

FMX IS TOTAL ARC LENGTH ALONG COORDINATE
FMIN IS STARTING VALUE OF ARC LENGTH SUCH AS 0.0
DFM IS SPECIFIED INITIAL INCREMENT OF ARC LENGTH
FPT IS NUMBER OF POINTS ALONG COORDINATE
FPCC IS ITERATIVE ERROR BOUND, E.G.0.000001
ICCL IS MAXIMUM NUMBER OF ITERATIONS
NCALL IF NCALL+1 INITIAL GUESS FOR EPS IS USED
IF NCALL .GT. 1, PREVIOUS EPS USED AS INITIAL GUESS

FMX=FMX
FMIN=FMIN
DFM=DFM
FPCC=FPCC
ICCL=ICCL

FNPTH2=NPT-2
IF (NCALL.EQ.1) EPS=(FMX/DFM)**(1.0/FNPTH2)-1.0

EPSIL=EPS
EP1=EPS**FNPTH2
REPS=1.0/EPS
DFM0E=DFM*REPS
F=FMX-FMIN-DFM0E*(EP1**EP1-1.0)
IF (ABS(F).LT.FPCC) GO TO 20
DFM0E2=DFM0E*REPS
FPN=DFM0E2*(1.0-EP1**EP1)**(EPS**FNPTH2-1.0)
EPSIL=EPS+FPN
EPSIL=EPS
WRITE ((6,30))
RETURN

10 CONTINUE

RETURN

20 EPSIL=EPS
SURPRESS THE EPSIL PRINTOUT
WRITE((6,601)) EPSIL+F*NIT
RETURN

30 FORMAT (1/42M EKEE2E0 MAX. NO. OF ITERATIONS IN EPSIL.)
END

WORKS  18
WORKS  19
WORKS  20
WORKS  21
WORKS  22
WORKS  23
WORKS  24
WORKS  25
WORKS  26
WORKS  27
WORKS  28
WORKS  29
WORKS  30
WORKS  31
WORKS  32
WORKS  33
WORKS  34
WORKS  35
WORKS  36
WORKS  37
WORKS  38
WORKS  39
WORKS  40
WORKS  41
WORKS  42
WORKS  43
WORKS  44
WORKS  45
WORKS  46
WORKS  47
WORKS  48
WORKS  49
WORKS  50
WORKS  51
WORKS  52
WORKS  53
WORKS  54
WORKS  55
WORKS  56
WORKS  57
WORKS  58
WORKS  59
WORKS  60
WORKS  61
WORKS  62
WORKS  63
WORKS  64
WORKS  65
WORKS  66
WORKS  67
WORKS  68
WORKS  69
SUBROUTINE CSPUM (XX, YY, X, Y, A, B, C, D, F, H, M, N, J1, J2)  
DIMENSION XX(I), YY(I), X(I), Y(I), A(I), B(I), C(I), D(I), F(I), H(I)

Cubic spline interpolation
XX and YY arrays are to be interpolated
YY are found interpolates corresponding to XX
J1, J2 are indice limits on XX (also YY)
Dimension of arrays carried in from outside, XX(1) must be monotone

Formula from NUMERICAL METHODS BY AHLQUIST, BJORK, ANDERSON
JLS FEB. '77

Rounding error protection
Caution may mask error in logic

IF (XX(N1).LT.XXX1) XX(N1)=X(J1)
IF (XX(N2).GT.XX(N1)) XX(N2)=X(J2)

First find derivative like terms that are coefficients

J1=J1+1
J2=J2+1
00 10 J=J1+J2
H(J)=X(J)-X(J-1)
10 O(J)=(YY(J)-YY(J-1))/H(J)
00 20 J=J1+J2
A(J)=H(J)+1
B(J)=B(J)+1
20 F(J)=3.*H(J)*O(J)+H(J+1)+H(J-1)+O(J)
B(J)=2.*H(J)+O(J)
M(J)=1.
F(J)=3.*O(J)
A(J)=H(J)
B(J)=2.*O(J)
F(J)=3.*O(J)
CALL TRIB (A+B+C+F+J1+J2)

Interpolation, X(J) array must be monotone
J=J1
I=J1-1
00 80 N=N1+N2
30 IF (X(J).LE.XX(N1).AND.X(J).GE.XX(N2)) GO TO 70
40 IF (J.GT.J2) GO TO 60
50 IF (J.LT.J1) GO TO 60
GO TO 30
60 WRITE (6,90) STOP
70 T=(XX(H)-X(J))/H(J)
T=T-T
YY(H)=YY(J)+T*T*(F(J)-D(I))*T*(F(I)-D(J))*T
80 CONTINUE
RETURN

90 FORMAT (1He,20H ERROR IN CSPUM)
END

WORKS 70
WORKS 71
WORKS 72
WORKS 73
WORKS 74
WORKS 75
WORKS 76
WORKS 77
WORKS 78
WORKS 79
WORKS 80
WORKS 81
WORKS 82
WORKS 83
WORKS 84
WORKS 85
WORKS 86
WORKS 87
WORKS 88
WORKS 89
WORKS 90
WORKS 91
WORKS 92
WORKS 93
WORKS 94
WORKS 95
WORKS 96
WORKS 97
WORKS 98
WORKS 99
WORKS 100
WORKS 101
WORKS 102
WORKS 103
WORKS 104
WORKS 105
WORKS 106
WORKS 107
WORKS 108
WORKS 109
WORKS 110
WORKS 111
WORKS 112
WORKS 113
WORKS 114
WORKS 115
WORKS 116
WORKS 117
WORKS 118
WORKS 119
WORKS 120
WORKS 121
WORKS 122
WORKS 123
WORKS 124
WORKS 125
WORKS 126
WORKS 127
WORKS 128
WORKS 129
WORKS 130

53
SUBROUTINE TRI8(A,B,C,F,NL,NU)
DIMENSION A(2), B(2), C(2), X(2), F(2)
XL1=NL/8(NL)
NL1=NL+1
DO 10 J=NL1,NU
Z=1./(8(J-1)*A(J)*X(J-1))
X(J)=C(J)*Z
10 F(J)=(F(J)-A(J)*F(J-1))*Z
IU=NU+NL
DO 20 J=NL1,NU
J=NL1-J1
20 F(J)=F(J)-X(J)*F(J-1)
RETURN
END

SUBROUTINE TRIP(A,B,C,F,NL,NU)
DIMENSION A(3), B(3), C(3), F(3), O(3), S(3)
J1=J1+1
F(N)*F(J2)
C FORWARD ELIMINATION SWEEP
O(J1)=C(J1)/B(J1)
F(J1)=F(J1)/B(J1)
S(J1)=A(J1)/B(J1)
DO 10 J=J1,J2
P=1./(B(J1)*A(J)*O(J-1))
O(J)=C(J)*P
F(J)=F(J)-A(J)*F(J-1)*P
S(J)=A(J)*S(J-1)*P
10 CONTINUE
C BACKWARD PASS
J1=J1+1
Q(J2)=O(J2)
S(J2)=1.
DO 20 J=J1,J2
J=J-1
20 Q(J)=Q(J)+Q(J-1)
F(J)=F(N)*C(J2)*Q(J)+A(J2)*O(J2-1)/C(J2)*S(J1)+A(J2)*S(J2-1)
I=8(J2)
C BACKWARD ELIMINATION PASS
DO 30 J=J1,J2
30 F(J)=F(J)+S(J)*Q(J)
RETURN
END
SUBROUTINE SCALC

COMMON /CALC/ X0, XATF, BS, ICT, FLAG, POYDX, FXF, RFN
COMMON /INPS/ X1, X2, X3, X4, RAD, DTDX, CHORD, FUSE, AS, RADS

THIS SUBRTN CALCULATES VALUES NEEDED IN THE
EQUATIONS SOLVING FOR Y VALUES IN SUBRTN BODAN

READ (5,10) X1, X2, X3, X4, RAD, THETA, CHORD
READ (5,10) RAD, FUSE, AS
THETA=THETA*0.174533
DTDX=TAN(THETA)
CHORD=ABS(CHORD)
FUSE=FUSE/2

TO FIND THE Y VALUE (BS) OF CIRCLE USED IN
THE SECANT OGIVE CALCULATIONS
XBAR=AS-X2
YR=RAD**2-XBAR**2
YSS=SQRT(YR)
BS=RAD-YSS

TO FIND THE X VALUE (XATF) AT THE FUSE
YR=ABS(BS)+FUSE
YRR=RAD**2-YR**2
XSS=SQRT(YRR)
XATF=AS-XSS

TO FIND THE SLOPE
POYDX=(XATF-AS)/(FUSE-BS)

TO FIND THE X VALUE (XO) AT THE NOSECAP
X0=SQRT(1.0+POYDX)
RFN=FUSE*X0
FXF2=RFN**2-FUSE**2
FXF=SQRT(FXF2)
X0=RFN-XF
XO=XATF-X0
WRITE (6,20) X1, X2, X3, X4, RAD, THETA, CHORD
WRITE (6,30) RAD, FUSE, AS, BS, X0, XATF
RETURN

10 FORMAT (9F10.0)
20 FORMAT (1H0,27HXI,X2,X3,X4,RAD,THETA,CHORD),/9F14.5)
30 FORMAT (1H0,27HRADS,FUSE,AS,BS,X0,XATF),/9F14.5)
END
SUBROUTINE SECANT
COMMON /CALC/ X0, XATF, BS, ICT, FLAG, POYOX, FXF, RFN
COMMON /INPS/ XI, X2, X3, X4, RAD, DYOX, CHORD, FUSE, AS, RADS
COMMON /IMCS/ XI, X2, X3, X*, AS0, OYDX, CHORD, FUSE, AS, RADS
COMMON JHAX, JMAX, JM, KM, NBOO, JB00
COMMON /BOIFOT/ X(100), Y(100), X1(100), X2(100), S(100), T(100)

PROJECTILE WITH NOSECAP
SECANT 3GIVE+CYLINDER+BOATTAIL

RCH=1/CHORD
X1=X1*RCH
X2=X2*RCH
X3=X3*RCH
X4=X4*RCH
RAD=RAD*RCH
FUSE=FUSE*RCH
RADS=RADS*RCH
XATF=XATF*RCH
AS=AS*RCH
BS=BS*RCH
GO 40 J=1,JB00
IF XX(J).GE.XATF GO TO 10

COMPUTE Y VALUES FOR NOSECAP
XBAR=XX(J)-(XATF+FXF)
RADI=RFN**2-XBAR**2
YY(J)=SORT(RADI)
GO TO 40

10 IF XX(J).GE.X2 GO TO 20

COMPUTE Y VALUES FOR OGIVE
XBAR=XX(J)-AS
RADI=RADS**2-XBAR**2
YY(J)=BS+SORT(RADI)
GO TO 40

20 IF XX(J).GE.X3 GO TO 30

COMPUTE Y VALUES FOR CYLINDER
YY(J)=RAD
GO TO 40

COMPUTE Y VALUES FOR BOATTAIL
YY(J)=RAD+(XX(J)-X3)*POYOX
GO CONTINUE
WRITE (6,50) POYOX, RFN, FXF
RETURN

50 FORMAT (1I0,13HPDYOX,RFN,FXF,5X,3F14.5)
END
SUBROUTINE STING (NGRD)
COMMON JMAX, KMAX, J, K, NWDD, JBOO
COMMON IBOO, YX(100), YX(100), YS(100), SS(100), TS(100)
1 = TS(100), TS(100)
STING 2
STING 3
STING 4
STING 5
STING 6
STING 7
STING 8
STING 9
STING 10
STING 11
STING 12
STING 13
STING 14
STING 15
STING 16
STING 17
STING 18
STING 19
STING 20
STING 21
STING 22
STING 23
STING 24
STING 25
STING 26
STING 27
STING 28
STING 29
STING 30
STING 31
STING 32
STING 33
STING 34
STING 35
STING 36
STING 37
STING 38
STING 39
STING 40
STING 41
STING 42
STING 43
STING 44
STING 45
STING 46
STING 47
STING 48
STING 49
STING 50
STING 51
STING 52
STING 53
STING 54
STING 55
STING 56
STING 57
STING 58
APPENDIX B
DEFINITION OF INPUT VALUES

Because the computer code has so many options, input statements are scattered throughout the code. Listed below are the input parameters to the program along with an explanation of each parameter. An input card is indicated below by numbering and underlining. All input formats are either I5 or F10.0 and are so indicated on the right hand side of the list of input parameters. Variables names follow conventional FORTRAN conventions and all integer names begin with I, J, K, L, N, or M. Special instructions as to whether or not a card is read are indicated with $ symbols.
INNER BOUNDARY

1. I3D, ND, LMAX  
   I3D = 1 generates 3-dimensional data, otherwise, 2D data only is generated.
   ND = same as LMAX.
   LMAX = the number of planes in the circumferential direction.

2. NBOD, JBOD, IXORS, ISEGS  
   NBOD = number of ordinates (i.e., x,y data points) used to define body. If NBOD < 0, an analytic body shape is used in subroutine BODAN.
   JBOD = number of points user will distribute on body surface.
   IXORS... use x or s (arc length) as monotone clustering parameter, x used if and only if (iff) IXORS = 0.
   ISEGS = number of contiguous clustering segments along body surface. Each segment requires end points and spacing specification as read in below.

   $\text{Read cards 3, 4, 5 iff NBOD } \geq \text{0}$

3. CHORD  
   All x,y data is normalized (i.e., divided by) CHORD. CHORD may be set to 1.

4. X(J), Y(J)  
   x,y ordinates that define body, J = 1, NBOD data cards are read in. If x,y are correctly normalized, set CHORD = 1.

5. JI, JF, XI, XF, DXI, DXF  
   Data that defines the cubic stretching function, see Eqs. (1) and (2) and Fig. (11).
   There are ISEGS such cards read-in.
In the notation of the text

\[ \begin{align*}
JI &= J_0 \\
JF &= J_f \\
XI &= x_0 \\
XF &= x_f \\
DXI &= \Delta x_0 \\
DXF &= \sqrt{v_x}
\end{align*} \]

Note \( DXI \) and \( DXF \) may both be positive or both be negative as \( x \) is increasing or decreasing

$ \text{Read cards 6, 7, 8 iff NBOD .LT.0 } $ 

6. **TAU, FLAG**

\[ \text{TAU} = \text{parabolic arc thickness ratio} \]

\[ \text{FLAG} = 0 \text{ tangent-ogive cylinder, boattail projectile read in} \]

\[ 1 \text{ secant-ogive cylinder, boattail projectile with nosecap read in} \]

iff FLAG = 1, card 7 goes after card 8b.

7. **JI, JF, XI, XF, DXI, DXF**

\[ \text{see card 5} \]

$ \text{iff FLAG .GE.0} $ 

8a. **X1, X2, X3, X4, RAD, THETA, CHORD**

\[ \begin{align*}
X1 &= \text{value of } x \text{ at nose} \\
X2 &= \text{value of } x \text{ at ogive-cylinder juncture} \\
X3 &= \text{value of } x \text{ at cylinder-boattail juncture} \\
X4 &= \text{value of } x \text{ at boattail base} \\
RAD &= \text{radius of cylinder} \\
\text{THETA} &= \text{angle of degrees that boattail makes with cylinder} \\
&\quad (\text{THETA is negative}) \\
\text{CHORD} &= \text{if CHORD .GE.0, body length normalized to one.}
\end{align*} \]

Note: a spherical cap is added to boattail so the body length is not \( X4-X1 \).
8b.  RADS, FUSE, AS

$ \text{iff } \text{FLAG .EQ.1}$

\begin{align*}
\text{RADS} &= \text{radius of secant} \\
\text{FUSE} &= \text{fuse height (at nosecap)} \\
\text{AS} &= \text{value of } x \text{ at secant-origin}
\end{align*}

9.  NFLAG

Option to exit program after body clustering data is printed out.

\text{NFLAG .LT.0, STOP}

10.  NCGRD, NCART

Parameters that allow addition of sting/rear cut and a front cut.

\begin{align*}
\text{NCGRD} &= \text{.GT.0, NCGRD points added for rear cut or sting} \\
\text{NCART} &= \text{.GT.0, NCART points added for front cut (or lower cut of C-grid)}
\end{align*}

$ \text{Read cards 11, 12, 13 iff } \text{NCGRD .GT.0}$

11.  ISEGS

\text{ISEGS of sting}

12.  XMIN, YMIN, XMAX, YMAX

\begin{align*}
\text{XMIN} &= \text{initial } x \text{ value of sting} \\
\text{YMIN} &= \text{initial } y \text{ value of sting} \\
\text{XMAX} &= \text{final } x \text{ value of sting} \\
\text{YMAX} &= \text{YMIN}
\end{align*}

13.  JI, JF, XI, XF, DXI, DXF

\text{see card 5}

$ \text{Read cards 14, 15, 16 iff } \text{NCART .GT.0}$

14.  ISEGS

\text{ISEGS}

15.  XMIN, YMIN, XMAX, YMAX

\begin{align*}
\text{XMIN} &= \text{XMIN} \\
\text{YMIN} &= \text{YMIN} \\
\text{XMAX} &= \text{XMAX} \\
\text{YMAX} &= \text{YMAX}
\end{align*}

16.  JI, JF, XI, XF, DXI, DXF

Front cut data like sting data
17.  NSEGS, IOUTD *2I5*

NSEGS = number of contiguous cubic segments which are used to form an outer boundary

IOUTD = number of clustering segments along outer boundary (i.e., previous ISEGS)

18.  XO, YO, X1, Y1, TH0, TH1 *6F10.0*

x,y,θ end point values used to define cubic segment according to Eq. (4). There are NSEGS such cards read-in. Here 0 implies initial point, 1 implies final end point. The angle θ is in degrees, and is defined in the usual way. See discussion of Fig. 12 for examples of θ.

$ Read cards 19 iff IOUTD .GT.0$

19.  JI, JF, XI, XF, DXI, DXF *2I5,4F10.0*

See card 5, there are IOUTD such data cards read in.

Arc length clustering used and, as the total arc length is not known on the first run of the program it is output. Use of normalized arc length allows user to cluster without true value of arc length.

GRID GENERATION

20.  KMAX, ITERM, IPER, NCLUS, ISTOR, JELLI *6I5*

KMAX = number of points in n-direction

ITERM = number of iterations used to relax Eq. (9)
   If ITERM .LT.0, straight ray grid is generated

IPER.... set IPER .GT.0 if periodic grid generated

NCLUS... NCLUS .LT.0 means grid is not reclustered using Eq. (8)

ISTOR... store grid on computer disc storage if ISTOR .GT.0

JELLI... If JELLI .GE.1, Limits JI and JF are set on the elliptic grid domain.

21.  DS, OMEGA *2F10.0*

DS = Δs₀ (i.e., Δs in n direction at n = 0 boundary). See Eq. (8). Note Δs₀ used along entire n = 0 boundary.

OMEGA = relaxation factor for SLOR in Subroutine RELAX. Typical safe value is 1.55. 0 < OMEGA < 2.0.
22. JI, JF

*215*

JELLI such cards read in. Limits of $\xi$ min to $\xi$ max over which an elliptic solver is used. JI and JF must correspond to vertical or horizontal rays.

23. BLANK CARD

24. BLANK CARD

$P$ and $Q$ input data, not recommended
APPENDIX C

SAMPLE INPUT AND OUTPUT

The following computer output illustrates the output from a sample grid generation. The tubular projectile illustrated in Figure 15 (the outer boundary is identical to that shown in Figure 13a) was used as a sample case. This particular case is the most difficult to set up as it requires the largest number of special instructions. Input values are printed after they are read in, so the output also supplies the user with an example of the data input cards.
<table>
<thead>
<tr>
<th>JXY, ON Body</th>
<th>LENGTH</th>
<th>INPUT</th>
<th>CHORD</th>
<th>CHORD LENGTH INPUT</th>
<th>2.25000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.25000</td>
<td>4.00000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2.34532</td>
<td>-0.00824</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2.34387</td>
<td>-0.00929</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2.17415</td>
<td>-0.16743</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2.03962</td>
<td>-0.29950</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.93257</td>
<td>-0.59952</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1.71513</td>
<td>-0.82612</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1.61064</td>
<td>-1.08385</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1.44943</td>
<td>-1.33022</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1.28432</td>
<td>-1.56213</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1.12734</td>
<td>-1.76501</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1.06028</td>
<td>-1.89027</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.98576</td>
<td>-2.01166</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0.86046</td>
<td>-2.01016</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.74535</td>
<td>-2.18148</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0.60523</td>
<td>-2.19115</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0.70077</td>
<td>-2.20018</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>0.72471</td>
<td>-2.32968</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>0.64419</td>
<td>-2.36019</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0.56366</td>
<td>-2.38053</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>0.45094</td>
<td>-2.33924</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>0.32909</td>
<td>-2.14127</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>0.24157</td>
<td>-2.12110</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>0.16104</td>
<td>-2.09998</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>0.12078</td>
<td>-2.07593</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>0.06441</td>
<td>-2.02866</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>0.02415</td>
<td>0.02866</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>0.00151</td>
<td>0.07991</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>0.00000</td>
<td>1.00000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>0.00161</td>
<td>1.25465</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>0.02415</td>
<td>1.84472</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>0.04441</td>
<td>2.03494</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>0.12078</td>
<td>2.80846</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>0.16104</td>
<td>3.05476</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>0.24157</td>
<td>3.66839</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>0.32909</td>
<td>3.74608</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>0.45094</td>
<td>4.09709</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>0.56366</td>
<td>4.28705</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>0.64419</td>
<td>4.34500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>0.72471</td>
<td>4.35937</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>0.79079</td>
<td>4.35152</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>0.80523</td>
<td>4.33213</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>0.83595</td>
<td>4.20299</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>0.81089</td>
<td>4.24306</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>0.84576</td>
<td>4.28664</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J &amp; J Y. ON Rody</td>
<td>4</td>
<td>.96829</td>
<td>.42114</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J &amp; J Y. ON Rody</td>
<td>47</td>
<td>1.27734</td>
<td>.60394</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J &amp; J Y. ON Rody</td>
<td>48</td>
<td>1.24438</td>
<td>.36686</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J &amp; J Y. ON Rody</td>
<td>49</td>
<td>1.44943</td>
<td>.32790</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J &amp; J Y. ON Rody</td>
<td>50</td>
<td>1.41044</td>
<td>.26911</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J &amp; J Y. ON Rody</td>
<td>51</td>
<td>1.71153</td>
<td>.20704</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J &amp; J Y. ON Rody</td>
<td>52</td>
<td>1.93257</td>
<td>.14184</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J &amp; J Y. ON Rody</td>
<td>53</td>
<td>2.09362</td>
<td>.07456</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J &amp; J Y. ON Rody</td>
<td>54</td>
<td>2.17146</td>
<td>.04874</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J &amp; J Y. ON Rody</td>
<td>57</td>
<td>2.25000</td>
<td>.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J &amp; J F. XI X F. ON Rody</td>
<td>1</td>
<td>13</td>
<td>1.00000</td>
<td>.40000</td>
<td>-.02000</td>
</tr>
<tr>
<td>J &amp; J F. XI X F. ON Rody</td>
<td>13</td>
<td>21</td>
<td>.40000</td>
<td>0.00000</td>
<td>-.06000</td>
</tr>
<tr>
<td>J &amp; J F. XI X F. ON Rody</td>
<td>21</td>
<td>28</td>
<td>0.00000</td>
<td>.40000</td>
<td>.01500</td>
</tr>
<tr>
<td>J &amp; J F. XI X F. ON Rody</td>
<td>28</td>
<td>40</td>
<td>.40000</td>
<td>1.00000</td>
<td>.06000</td>
</tr>
<tr>
<td>J &amp; J F. XI X F. ON Rody</td>
<td>1</td>
<td>92000</td>
<td>.95091</td>
<td>.31132</td>
<td>.86942</td>
</tr>
<tr>
<td>.52165</td>
<td>.45000</td>
<td>.34000</td>
<td>.27571</td>
<td>.21071</td>
<td>.16617</td>
</tr>
<tr>
<td>.59436</td>
<td>.44455</td>
<td>.70727</td>
<td>.75545</td>
<td>.82600</td>
<td>.86942</td>
</tr>
</tbody>
</table>

<p>| J &amp; J Rody Distributions | J &amp; J Y. ON Rody | 1 | 1.00000 | .00000 |
| J &amp; J Y. ON Rody | 2 | .97991 | -.00754 |
| J &amp; J Y. ON Rody | 3 | .95017 | -.01340 |
| J &amp; J Y. ON Rody | 4 | .91023 | -.01646 |
| J &amp; J Y. ON Rody | 5 | .86370 | -.02311 |
| J &amp; J Y. ON Rody | 6 | .81084 | -.03294 |
| J &amp; J Y. ON Rody | 7 | .75786 | -.04266 |
| J &amp; J Y. ON Rody | 8 | .70101 | -.05293 |
| J &amp; J Y. ON Rody | 9 | .64637 | -.06176 |
| J &amp; J Y. ON Rody | 10 | .59012 | -.07115 |
| J &amp; J Y. ON Rody | 11 | .53300 | -.08078 |
| J &amp; J Y. ON Rody | 12 | .47691 | -.09088 |
| J &amp; J Y. ON Rody | 13 | .42237 | -.10616 |
| J &amp; J Y. ON Rody | 14 | .37019 | -.12324 |
| J &amp; J Y. ON Rody | 15 | .32788 | -.14103 |
| J &amp; J Y. ON Rody | 16 | .28645 | -.16054 |
| J &amp; J Y. ON Rody | 17 | .24598 | -.18398 |
| J &amp; J Y. ON Rody | 18 | .20383 | -.20648 |
| J &amp; J Y. ON Rody | 19 | .16208 | -.23018 |
| J &amp; J Y. ON Rody | 20 | .12071 | -.25456 |
| J &amp; J Y. ON Rody | 21 | .07975 | -.28454 |
| J &amp; J Y. ON Rody | 22 | .03813 | -.31478 |
| J &amp; J Y. ON Rody | 23 | .00657 | -.34493 |
| J &amp; J Y. ON Rody | 24 | .07227 | -.37319 |
| J &amp; J Y. ON Rody | 25 | .16594 | -.40705 |</p>
<table>
<thead>
<tr>
<th>J+X.Y</th>
<th>ON RODY</th>
<th>26</th>
<th>.74450</th>
<th>.19000</th>
</tr>
</thead>
<tbody>
<tr>
<td>J+X.Y</td>
<td>ON RODY</td>
<td>27</td>
<td>.32405</td>
<td>.19331</td>
</tr>
<tr>
<td>J+X.Y</td>
<td>ON RODY</td>
<td>28</td>
<td>.3441</td>
<td>.19860</td>
</tr>
<tr>
<td>J+X.Y</td>
<td>ON RODY</td>
<td>29</td>
<td>.65272</td>
<td>.18402</td>
</tr>
<tr>
<td>J+X.Y</td>
<td>ON RODY</td>
<td>30</td>
<td>.51651</td>
<td>.17500</td>
</tr>
<tr>
<td>J+X.Y</td>
<td>ON RODY</td>
<td>31</td>
<td>.5441A</td>
<td>.16917</td>
</tr>
<tr>
<td>J+X.Y</td>
<td>ON RODY</td>
<td>32</td>
<td>.6442A</td>
<td>.14215</td>
</tr>
<tr>
<td>J+X.Y</td>
<td>ON RODY</td>
<td>33</td>
<td>.71041</td>
<td>.12143</td>
</tr>
<tr>
<td>J+X.Y</td>
<td>ON RODY</td>
<td>34</td>
<td>.76041</td>
<td>.09493</td>
</tr>
<tr>
<td>J+X.Y</td>
<td>ON RODY</td>
<td>35</td>
<td>.87342</td>
<td>.07701</td>
</tr>
<tr>
<td>J+X.Y</td>
<td>ON RODY</td>
<td>36</td>
<td>.8731A</td>
<td>.05706</td>
</tr>
<tr>
<td>J+X.Y</td>
<td>ON RODY</td>
<td>37</td>
<td>.5144A</td>
<td>.03894</td>
</tr>
<tr>
<td>J+X.Y</td>
<td>ON RODY</td>
<td>38</td>
<td>.95373</td>
<td>.02240</td>
</tr>
<tr>
<td>J+X.Y</td>
<td>ON RODY</td>
<td>39</td>
<td>.98202</td>
<td>.01105</td>
</tr>
<tr>
<td>J+X.Y</td>
<td>ON RODY</td>
<td>40</td>
<td>1.00000</td>
<td>0.00000</td>
</tr>
</tbody>
</table>

**CARTA PART OF PROGRAM.....ISEGS = 1**

<table>
<thead>
<tr>
<th>X+MIN , YMIN , XMAX , YMAX</th>
<th>1.00000</th>
<th>0.00000</th>
<th>8.00000</th>
<th>0.00000</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1 , F.X , X1 , X1 , X1 , X1</td>
<td>11</td>
<td>11</td>
<td>0.00000</td>
<td>7.00000</td>
</tr>
<tr>
<td>0.00000</td>
<td>0.02400</td>
<td>0.24111</td>
<td>1.66442</td>
<td>1.27500</td>
</tr>
</tbody>
</table>

**FINAL VALUE OF JMAX**

60

**FINAL VALUES OF J+X.Y ALONG INNER BOUNDARY**

<p>| 1 | 6.00000 | 0.00000 |
| 2 | 6.90000 | 0.00000 |
| 3 | 9.82778 | 0.00000 |
| 4 | 4.86425 | 0.00000 |
| 5 | 3.45433 | 0.00000 |
| 6 | 3.00494 | 0.00000 |
| 7 | 2.77500 | 0.00000 |
| 8 | 1.69417 | 0.00000 |
| 9 | 1.274111 | 0.00000 |</p>
<table>
<thead>
<tr>
<th>J</th>
<th>L = 1</th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>J</th>
<th>L = 2</th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>1</td>
<td>8.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5.42778</td>
<td>0.00000</td>
<td>0.00000</td>
<td>3</td>
<td>5.42778</td>
<td>0.00000</td>
<td>0.00000</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3.85933</td>
<td>0.00000</td>
<td>0.00000</td>
<td>5</td>
<td>3.85933</td>
<td>0.00000</td>
<td>0.00000</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2.27500</td>
<td>0.00000</td>
<td>0.00000</td>
<td>7</td>
<td>2.27500</td>
<td>0.00000</td>
<td>0.00000</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1.26111</td>
<td>0.00000</td>
<td>0.00000</td>
<td>9</td>
<td>1.26111</td>
<td>0.00000</td>
<td>0.00000</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>11</td>
<td>1.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.95017</td>
<td>0.00000</td>
<td>-0.0139</td>
<td>13</td>
<td>0.95017</td>
<td>0.00000</td>
<td>-0.0139</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.86370</td>
<td>0.00000</td>
<td>-0.0243</td>
<td>15</td>
<td>0.86370</td>
<td>0.00000</td>
<td>-0.0243</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0.75244</td>
<td>0.00000</td>
<td>-0.0422</td>
<td>17</td>
<td>0.75244</td>
<td>0.00000</td>
<td>-0.0422</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>0.62437</td>
<td>0.00000</td>
<td>-0.0617</td>
<td>19</td>
<td>0.62437</td>
<td>0.00000</td>
<td>-0.0617</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>0.49330</td>
<td>0.00000</td>
<td>-0.0792</td>
<td>21</td>
<td>0.49330</td>
<td>0.00000</td>
<td>-0.0792</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>0.36237</td>
<td>0.00000</td>
<td>-0.0951</td>
<td>23</td>
<td>0.36237</td>
<td>0.00000</td>
<td>-0.0951</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>0.22886</td>
<td>0.00000</td>
<td>-0.0749</td>
<td>25</td>
<td>0.22886</td>
<td>0.00000</td>
<td>-0.0749</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>0.09491</td>
<td>0.00000</td>
<td>-0.0565</td>
<td>27</td>
<td>0.09491</td>
<td>0.00000</td>
<td>-0.0565</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>0.01038</td>
<td>0.00000</td>
<td>-0.0133</td>
<td>29</td>
<td>0.01038</td>
<td>0.00000</td>
<td>-0.0133</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>0.00188</td>
<td>0.00000</td>
<td>-0.0064</td>
<td>31</td>
<td>0.00188</td>
<td>0.00000</td>
<td>-0.0064</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>0.03662</td>
<td>0.00000</td>
<td>-0.1159</td>
<td>33</td>
<td>0.03662</td>
<td>0.00000</td>
<td>-0.1159</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>0.16595</td>
<td>0.00000</td>
<td>-0.1735</td>
<td>35</td>
<td>0.16595</td>
<td>0.00000</td>
<td>-0.1735</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>0.33045</td>
<td>0.00000</td>
<td>-0.1933</td>
<td>37</td>
<td>0.33045</td>
<td>0.00000</td>
<td>-0.1933</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>0.45272</td>
<td>0.00000</td>
<td>-0.1842</td>
<td>39</td>
<td>0.45272</td>
<td>0.00000</td>
<td>-0.1842</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>0.58418</td>
<td>0.00000</td>
<td>-0.1601</td>
<td>41</td>
<td>0.58418</td>
<td>0.00000</td>
<td>-0.1601</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>0.71661</td>
<td>0.00000</td>
<td>-0.1743</td>
<td>43</td>
<td>0.71661</td>
<td>0.00000</td>
<td>-0.1743</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>0.82373</td>
<td>0.00000</td>
<td>-0.1771</td>
<td>45</td>
<td>0.82373</td>
<td>0.00000</td>
<td>-0.1771</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>0.91688</td>
<td>0.00000</td>
<td>-0.1695</td>
<td>47</td>
<td>0.91688</td>
<td>0.00000</td>
<td>-0.1695</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>0.99202</td>
<td>0.00000</td>
<td>-0.1105</td>
<td>49</td>
<td>0.99202</td>
<td>0.00000</td>
<td>-0.1105</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>1.02500</td>
<td>0.00000</td>
<td>0.00000</td>
<td>51</td>
<td>1.02500</td>
<td>0.00000</td>
<td>0.00000</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>1.68542</td>
<td>0.00000</td>
<td>0.00000</td>
<td>53</td>
<td>1.68542</td>
<td>0.00000</td>
<td>0.00000</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>3.00694</td>
<td>0.00000</td>
<td>0.00000</td>
<td>55</td>
<td>3.00694</td>
<td>0.00000</td>
<td>0.00000</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>4.49625</td>
<td>0.00000</td>
<td>0.00000</td>
<td>57</td>
<td>4.49625</td>
<td>0.00000</td>
<td>0.00000</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>6.90000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>59</td>
<td>6.90000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>-0.0020</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>K = A L = 20</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>J</td>
<td>K = 2 L = 21</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>---</td>
<td>-------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>-------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>1</td>
<td>8,00000</td>
<td>0,00000</td>
<td>-0,05087</td>
<td>1</td>
<td>5,82778</td>
<td>0,00000</td>
<td>-0,06347</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5,82778</td>
<td>0,00000</td>
<td>-0,05087</td>
<td>2</td>
<td>3,85833</td>
<td>0,00000</td>
<td>-0,06347</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2,27500</td>
<td>0,00000</td>
<td>-0,05087</td>
<td>3</td>
<td>2,27500</td>
<td>0,00000</td>
<td>-0,06347</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1,26111</td>
<td>0,00000</td>
<td>-0,05087</td>
<td>4</td>
<td>1,26111</td>
<td>0,00000</td>
<td>-0,06347</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1,00000</td>
<td>0,00000</td>
<td>-0,05087</td>
<td>5</td>
<td>1,00000</td>
<td>0,00000</td>
<td>-0,06347</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0,93289</td>
<td>0,00000</td>
<td>-0,05079</td>
<td>6</td>
<td>0,93289</td>
<td>0,00000</td>
<td>-0,07104</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0,84217</td>
<td>0,00000</td>
<td>-0,0514</td>
<td>7</td>
<td>0,84217</td>
<td>0,00000</td>
<td>-0,08430</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0,72691</td>
<td>0,00000</td>
<td>-0,05345</td>
<td>8</td>
<td>0,72691</td>
<td>0,00000</td>
<td>-0,10675</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0,59315</td>
<td>0,00000</td>
<td>-0,12037</td>
<td>9</td>
<td>0,59315</td>
<td>0,00000</td>
<td>-0,13619</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0,44000</td>
<td>0,00000</td>
<td>-0,1418</td>
<td>10</td>
<td>0,44000</td>
<td>0,00000</td>
<td>-0,15278</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0,29871</td>
<td>0,00000</td>
<td>-0,15127</td>
<td>11</td>
<td>0,29871</td>
<td>0,00000</td>
<td>-0,16219</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0,14290</td>
<td>0,00000</td>
<td>-0,13673</td>
<td>12</td>
<td>0,14290</td>
<td>0,00000</td>
<td>-0,14069</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>-0,01283</td>
<td>0,00000</td>
<td>-0,01977</td>
<td>13</td>
<td>-0,01283</td>
<td>0,00000</td>
<td>-0,09786</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>-0,10366</td>
<td>0,00000</td>
<td>0,00460</td>
<td>14</td>
<td>-0,10366</td>
<td>0,00000</td>
<td>0,00862</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>-0,20415</td>
<td>0,00000</td>
<td>0,00351</td>
<td>15</td>
<td>-0,20415</td>
<td>0,00000</td>
<td>0,00169</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>-0,30493</td>
<td>0,00000</td>
<td>0,00236</td>
<td>16</td>
<td>-0,30493</td>
<td>0,00000</td>
<td>0,00363</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>-0,40703</td>
<td>0,00000</td>
<td>0,00103</td>
<td>17</td>
<td>-0,40703</td>
<td>0,00000</td>
<td>0,00303</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>-0,2976</td>
<td>0,00000</td>
<td>0,00000</td>
<td>18</td>
<td>-0,2976</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>-0,1429</td>
<td>0,00000</td>
<td>0,00000</td>
<td>19</td>
<td>-0,1429</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>-0,0000</td>
<td>0,00000</td>
<td>0,00000</td>
<td>20</td>
<td>-0,0000</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>J</th>
<th>K = 5 L = 1</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>J</th>
<th>K = 6 L = 2</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8,00000</td>
<td>0,00000</td>
<td>0,00000</td>
<td>1</td>
<td>8,00000</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5,82778</td>
<td>0,00000</td>
<td>0,00000</td>
<td>2</td>
<td>5,82778</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3,85833</td>
<td>0,00000</td>
<td>0,00000</td>
<td>3</td>
<td>3,85833</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2,27500</td>
<td>0,00000</td>
<td>0,00000</td>
<td>4</td>
<td>2,27500</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1,26111</td>
<td>0,00000</td>
<td>0,00000</td>
<td>5</td>
<td>1,26111</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1,00000</td>
<td>0,00000</td>
<td>0,00000</td>
<td>6</td>
<td>1,00000</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0,93289</td>
<td>0,00000</td>
<td>0,00000</td>
<td>7</td>
<td>0,93289</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0,84217</td>
<td>0,00000</td>
<td>0,00000</td>
<td>8</td>
<td>0,84217</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0,72691</td>
<td>0,00000</td>
<td>0,00000</td>
<td>9</td>
<td>0,72691</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0,59315</td>
<td>0,00000</td>
<td>0,00000</td>
<td>10</td>
<td>0,59315</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0,44000</td>
<td>0,00000</td>
<td>0,00000</td>
<td>11</td>
<td>0,44000</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0,29871</td>
<td>0,00000</td>
<td>0,00000</td>
<td>12</td>
<td>0,29871</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0,14290</td>
<td>0,00000</td>
<td>0,00000</td>
<td>13</td>
<td>0,14290</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0,01283</td>
<td>0,00000</td>
<td>0,00000</td>
<td>14</td>
<td>0,01283</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0,00362</td>
<td>0,00000</td>
<td>0,00000</td>
<td>15</td>
<td>0,00362</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0,00188</td>
<td>0,00000</td>
<td>0,00000</td>
<td>16</td>
<td>0,00188</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0,00093</td>
<td>0,00000</td>
<td>0,00000</td>
<td>17</td>
<td>0,00093</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>0,00049</td>
<td>0,00000</td>
<td>0,00000</td>
<td>18</td>
<td>0,00049</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>0,00026</td>
<td>0,00000</td>
<td>0,00000</td>
<td>19</td>
<td>0,00026</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0,00013</td>
<td>0,00000</td>
<td>0,00000</td>
<td>20</td>
<td>0,00013</td>
<td>0,00000</td>
<td>0,00000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX D

PLOT PROGRAM LISTING

A listing of the computer code used to generate grid plots is presented in this appendix. The plot program is written in standard FORTRAN IV and uses the Tektronix Plot 10 software package. All plots were produced on the Tektronix 4010-1 display terminal which was connected to the BRL Cyber 173/76. The program is an interactive plotting routine which prompts the user for all requested information.
*PROGRAM IGP 76/76 OPT=1 ROUND***

1

PROGRAM IGP (INPUT,OUTPUT,TAPES=INPUT,TAPE0=OUTPUT,TAPER0,TAP0,KBI
COMMON /GRIDC/ GMXN, GMAX, GMIN, GYMAX, X(100,50), Y(100,50)
COMMON /HDR/ TITLE(5)

C

C*****GRD=PROGRAM TO PLOT THE COMPUTATIONAL GRID ABOUT AN
C AERODYNAMIC BODY OR AIRFOIL SECTION******
C

C

C*****INPUT SECTION*****
C*****CONTROL PARAMETER INPUT
C**NADD=1 SUPPRESSES LISTING OF PLOT DATA; NADD=1 ALLOWS IT
C**JMAX=MAX-NUMBER OF J.K POINTS
C**GMXN=ALPHANUMERIC INFORMATION DESCRIBING THE
C CONFIGURATION BEING PLOTTED
C
CALL CONNEC (5,TAPES)
CALL CONNEC (5,TAPES0)
CALL TERM (1,102)

C**READ HEAD
CALL SETU8F (3)
CALL ANMODE

20

WRITE (6,10)
READ (5,*) IBO
IIO=IBO/10
CALL INITT (IBO)

C**READ HEAD

25

WRITE (6,20)
READ (5,*) JMAX,KMAX

C**READ IN X AND Y VALUES
READ (8) ((X(J,K),J=1,JMAX),K=1,KMAX),((Y(J,K),J=1,JMAX),K=1,KMAX)

C**PLOT THE GRID *****
CALL GDPLT (JMAX,KMAX,IBO)

40

C**TERMINATE PLOTTING
CALL FINITT (0,700)
STOP

C

10 FORMAT (1H WHAT IS BAUD RATE?)
20 FORMAT (2H WHAT ARE JMAX,KMAX?)
30 FORMAT (2H WHAT ARE XMIN,XMAX,YMIN,YMAX?)
40 FORMAT (2H ENTER TITLE - UP TO 50 CHAR)
50 FORMAT (5A10)
SUBROUTINE GROPLT(IJMAX,XMAX,IJMIN,XMIN)
COMMON /GRIDC/ GXMIN, GYMAX, GYMIN, GYMAX, XI100, YI100, XI50, YI50
COMMON /HOR/ TITLE5
DIMENSION GX(128),GY(128)
C
C***READJUST PLOT LIMITS IN ORDER TO AVOID STRETCHED PLOTS
ICOUNT=0
10 XMAX=GXMAX
YMIN=GYMIN
10 YMAX=GYMAX
YMIN=GYMIN
XDIFF=XMAX-XMIN
YDIFF=YMAX-YMIN
IF (XDIFF.LT.YDIFF) GO TO 20
15 XDIFF=XDIFF*.5
YMIN=(YMAX+YMIN)*.5
YMAX=YMIN+XDIFF
YMIN=YMAX-XDIFF
20 GO TO 30
20 YDIFF=YDIFF*.5
XMIN=(XMAX+XMIN)*.5
XMAX=XMIN+YDIFF
XMIN=XMAX-YDIFF
25 CONTINUE
30 CONTINUE
C
C PLOT THE LINES
IF (ICOUNT.GT.0) GO TO 40
AXMIN=XMIN
AXMAX=XMAX
AYMIN=YMIN
AYMAX=YMAX
35 CONTINUE
C
CALL BINITT
CALL NPTS (IJMAX)
CALL ANMODE
WRITE (6,120) TITLE
CALL XFRM (2)
CALL YFRM (2)
C
CALL DLINK (XMIN, XMAX)
CALL DLINK (YMIN, YMAX)
CALL SLINK (150, 800)
CALL SLINY (50, 700)
C
DO 70 J=1,JMAX
DO 50 I=1,IMAX
GX(I,J)=XI(I,J)
GY(I,J)=Y(J)
50 CONTINUE
C
IF (X.GT.1) GO TO 50
CALL CHECK (GX,GY)
50 CONTINUE
CALL DSPLAY (GX,GY)
60 CALL CPLLOT (GX,GY)
70 CONTINUE
CALL NPTS (KMAX)
60 DO 90 J=1,JMAX
60 DO 80 K=1,KMAX
GX(K)=X(J,K)
80 GY(K)=Y(J,K)
CALL CPLLOT (GX,GY)
90 CONTINUE
CALL BELL
CALL TSEND
CALL TINPUT (II)
IF (II.EQ.93) GO TO 110
CALL NEWPAG
CALL TSEND
DO 100 J=1,JMAX
100 GX(J)=X(J,1)
100 GY(J)=Y(J,1)
CALL BINIIT
CALL DLIMX (AXMIN,AXMAX)
CALL DLIMY (AYMIN,AYMAX)
CALL SLIMX (50,700)
CALL NPTS (JMAX)
CALL XFRM (2)
CALL YFRM (2)
CALL CHECK (GX,GY)
CALL DSPLAY (GX,GY)
CALL TSEND
CALL BELL
CALL ANMODE
WRITE (6,130)
CALL TSEND
CALL RECOVR
95 CALL VCURSR (ICH+XX,YY)
GXMIN=XX
GYMIN=YY
CALL ANMODE
WRITE (6,140)
CALL RECOVR
CALL TSEND
CALL ANMODE
CALL VCURSR (ICH+XX,YY)
GXMAX=XX
GYMAX=YY
CALL TSEND
ICOUNT=ICOUNT+1
CALL NEWPAG
GO TO 10
110 CONTINUE
CALL NEWPAG
RETURN
120 FORMAT (5A10)
130 FORMAT (30H POSITION CURSOR FOR XMIN,YMIN)
140 FORMAT (140,30H POSITION CURSOR FOR XMAX,YMAX)
<table>
<thead>
<tr>
<th>No. of Copies</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>Defense Technical Info Center</td>
</tr>
<tr>
<td></td>
<td>ATTN: DDC-DDA</td>
</tr>
<tr>
<td></td>
<td>Cameron Station</td>
</tr>
<tr>
<td></td>
<td>Alexandria, VA 22314</td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>US Army Materiel Development and Readiness Command</td>
</tr>
<tr>
<td></td>
<td>ATTN: DRCMDMD-ST</td>
</tr>
<tr>
<td></td>
<td>5001 Eisenhower Avenue</td>
</tr>
<tr>
<td></td>
<td>Alexandria, VA 22333</td>
</tr>
<tr>
<td>8</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>US Army Armament Research and Development Command</td>
</tr>
<tr>
<td></td>
<td>ATTN: DRDAR-TSS (2 cys)</td>
</tr>
<tr>
<td></td>
<td>DRDAR-LCA-F</td>
</tr>
<tr>
<td></td>
<td>Mr. D. Mertz</td>
</tr>
<tr>
<td></td>
<td>Mr. A. Loeb</td>
</tr>
<tr>
<td></td>
<td>Mr. S. Wasserman</td>
</tr>
<tr>
<td></td>
<td>Mr. H. Hudgins</td>
</tr>
<tr>
<td></td>
<td>Mr. E. Friedman</td>
</tr>
<tr>
<td></td>
<td>Dover, NJ 07801</td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>US Army Armament Materiel Readiness Command</td>
</tr>
<tr>
<td></td>
<td>ATTN: DRSAR-LEP-L, Tech Lib</td>
</tr>
<tr>
<td></td>
<td>Rock Island, IL 61299</td>
</tr>
<tr>
<td>1</td>
<td>Director</td>
</tr>
<tr>
<td></td>
<td>US Army Armament Research and Development Command</td>
</tr>
<tr>
<td></td>
<td>ATTN: DRDAR-LCB-TL</td>
</tr>
<tr>
<td></td>
<td>Watervliet, NY 12189</td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>US Army Aviation Research and Development Command</td>
</tr>
<tr>
<td></td>
<td>ATTN: DRDAV-E</td>
</tr>
<tr>
<td></td>
<td>4300 Goodfellow Boulevard</td>
</tr>
<tr>
<td></td>
<td>St. Louis, MO 63120</td>
</tr>
<tr>
<td>1</td>
<td>Director</td>
</tr>
<tr>
<td></td>
<td>US Army Air Mobility Research and Development Laboratory</td>
</tr>
<tr>
<td></td>
<td>ATTN: DDCDDA</td>
</tr>
<tr>
<td></td>
<td>Ames Research Center</td>
</tr>
<tr>
<td></td>
<td>Moffett Field, CA 94035</td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>US Army Communications Research and Development Command</td>
</tr>
<tr>
<td></td>
<td>ATTN: DRDCO-PPA-SA</td>
</tr>
<tr>
<td></td>
<td>Fort Monmouth, NJ 07703</td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>US Army Electronics Research and Development Command</td>
</tr>
<tr>
<td></td>
<td>Technical Support Activity</td>
</tr>
<tr>
<td></td>
<td>ATTN: DELSD-L</td>
</tr>
<tr>
<td></td>
<td>Fort Monmouth, NJ 07703</td>
</tr>
<tr>
<td>3</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>US Army Missile Command</td>
</tr>
<tr>
<td></td>
<td>ATTN: DRSMI-R</td>
</tr>
<tr>
<td></td>
<td>DRSMI-YDL</td>
</tr>
<tr>
<td></td>
<td>DRSMI-RDK</td>
</tr>
<tr>
<td></td>
<td>Mr. R. Deep</td>
</tr>
<tr>
<td></td>
<td>Redstone Arsenal, AL 35809</td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>US Army Tank Automotive Research and Development Command</td>
</tr>
<tr>
<td></td>
<td>ATTN: DRDTA-UL</td>
</tr>
<tr>
<td></td>
<td>Warren, MI 48090</td>
</tr>
<tr>
<td>1</td>
<td>Director</td>
</tr>
<tr>
<td></td>
<td>US Army TRADOC Systems Analysis Activity</td>
</tr>
<tr>
<td></td>
<td>ATTN: ATAA-SL, Tech Lib</td>
</tr>
<tr>
<td></td>
<td>White Sands Missile Range, NM 88002</td>
</tr>
<tr>
<td>No. of Copies</td>
<td>Organization</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
</tr>
<tr>
<td>1 Commander</td>
<td>US Army Research Office</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 Commander</td>
<td>US Naval Air Systems Command</td>
</tr>
<tr>
<td></td>
<td>ATTN: AIR-604</td>
</tr>
<tr>
<td></td>
<td>Washington, D. C. 20360</td>
</tr>
<tr>
<td>4 Commander</td>
<td>US Naval Surface Weapons Center</td>
</tr>
<tr>
<td></td>
<td>ATTN: Dr. T. Clare, Code DK20</td>
</tr>
<tr>
<td></td>
<td>Dr. P. Daniels</td>
</tr>
<tr>
<td></td>
<td>Mr. D. A. Jones III</td>
</tr>
<tr>
<td></td>
<td>Mr. L. Mason</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 312</td>
</tr>
<tr>
<td></td>
<td>Mr. R. Voisinet</td>
</tr>
<tr>
<td></td>
<td>Mr. R. Driftmeyer</td>
</tr>
<tr>
<td></td>
<td>Mr. J. Knott</td>
</tr>
<tr>
<td></td>
<td>Mr. R. Schlie</td>
</tr>
<tr>
<td></td>
<td>Silver Spring, MD 20910</td>
</tr>
<tr>
<td>1 Commander</td>
<td>US Naval Weapons Center</td>
</tr>
<tr>
<td></td>
<td>ATTN: Code 3431, Tech Lib</td>
</tr>
<tr>
<td></td>
<td>China Lake, CA 93555</td>
</tr>
<tr>
<td>1 Director</td>
<td>NASA Langley Research Center</td>
</tr>
<tr>
<td></td>
<td>ATTN: NS-185, Tech Lib</td>
</tr>
<tr>
<td></td>
<td>Langley Station</td>
</tr>
<tr>
<td></td>
<td>Hampton, VA 23365</td>
</tr>
<tr>
<td>3 Director</td>
<td>NASA Ames Research Center</td>
</tr>
<tr>
<td></td>
<td>ATTN: MS-202A-14</td>
</tr>
<tr>
<td></td>
<td>Dr. P. Kutler</td>
</tr>
<tr>
<td></td>
<td>MS-202-1,</td>
</tr>
<tr>
<td></td>
<td>Dr. T. Pulliam</td>
</tr>
<tr>
<td></td>
<td>MS-227-8,</td>
</tr>
<tr>
<td></td>
<td>Dr. L. Schiff</td>
</tr>
<tr>
<td></td>
<td>Moffett Field, CA 94035</td>
</tr>
<tr>
<td>2 Sandia Laboratories</td>
<td>ATTN: Division No. 1331,</td>
</tr>
<tr>
<td></td>
<td>Mr. H.R. Vaughn</td>
</tr>
<tr>
<td></td>
<td>Mr. G.R. Eisler</td>
</tr>
<tr>
<td></td>
<td>P.O. Box 580</td>
</tr>
<tr>
<td></td>
<td>Albuquerque, NJ 87184</td>
</tr>
<tr>
<td>1 University of California, Davis</td>
<td>Department of Mechanical Engineering</td>
</tr>
<tr>
<td></td>
<td>ATTN: Prof. H.A. Dwyer</td>
</tr>
<tr>
<td></td>
<td>Davis, CA 95616</td>
</tr>
<tr>
<td>1 University of Colorado</td>
<td>Department of Aerospace Engineering</td>
</tr>
<tr>
<td></td>
<td>ATTN: Prof. G. Inger</td>
</tr>
<tr>
<td></td>
<td>Boulder, CO 80309</td>
</tr>
<tr>
<td>1 University of Delaware</td>
<td>Mechanical and Aerospace Engineering Department</td>
</tr>
<tr>
<td></td>
<td>ATTN: Dr. J. E. Danberg</td>
</tr>
<tr>
<td></td>
<td>Newark, DE 19711</td>
</tr>
<tr>
<td>No. of Copies</td>
<td>Organization</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td>Aberdeen Proving Ground</td>
</tr>
<tr>
<td></td>
<td>Dir, USAMSAA</td>
</tr>
<tr>
<td></td>
<td>ATTN: DRXSY-D</td>
</tr>
<tr>
<td></td>
<td>DRXSY-MP, H. Cohen</td>
</tr>
<tr>
<td></td>
<td>Cdr, USATECOM</td>
</tr>
<tr>
<td></td>
<td>ATTN: DRSTE-TO-F</td>
</tr>
<tr>
<td></td>
<td>Dir, USACSL, Bldg. E3516, EA</td>
</tr>
<tr>
<td></td>
<td>ATTN: DRDAR-CLB-PA</td>
</tr>
<tr>
<td></td>
<td>DRDAR-ACW, Mr. M. Miller</td>
</tr>
</tbody>
</table>
USER EVALUATION OF REPORT

Please take a few minutes to answer the questions below; tear out this sheet, fold as indicated, staple or tape closed, and place in the mail. Your comments will provide us with information for improving future reports.

1. BRL Report Number

2. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

3. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.)

4. Has the information in this report led to any quantitative savings as far as man-hours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

5. General Comments (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.)

6. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name:

Telephone Number:

Organization Address: