

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

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 indorsement of any commercial product.*

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM								
1. REPORT NUMBER MEMORANDUM REPORT ARBRL-MR-03142	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER								
4. TITLE (and Subtitle) A GENERAL CURVILINEAR GRID GENERATION PROGRAM FOR PROJECTILE CONFIGURATIONS		5. TYPE OF REPORT & PERIOD COVERED Final								
		6. PERFORMING ORG. REPORT NUMBER								
7. AUTHOR(s) J.L. Steger, C.J. Nietubicz, & K.R. Heavey		8. CONTRACT OR GRANT NUMBER(s)								
9. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. Army Ballistic Research Laboratory (ATTN: DRDAR-BLL) Aberdeen Proving Ground, Maryland 21005		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 1L161102AH43								
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Armament Research & Development Command US Army Ballistic Research Laboratory (DRDAR-BL) Aberdeen Proving Ground, MD 21005		12. REPORT DATE OCTOBER 1981								
		13. NUMBER OF PAGES 83								
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified								
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE								
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release, distribution unlimited.										
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)										
18. SUPPLEMENTARY NOTES										
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) <table border="0" style="width: 100%;"> <tr> <td>Numerical grid generation</td> <td>Finite difference techniques</td> </tr> <tr> <td>Non-Orthogonal grids</td> <td>Projectile shapes</td> </tr> <tr> <td>Axisymmetric bodies</td> <td>Elliptic equations</td> </tr> <tr> <td>Transformed coordinates</td> <td></td> </tr> </table>			Numerical grid generation	Finite difference techniques	Non-Orthogonal grids	Projectile shapes	Axisymmetric bodies	Elliptic equations	Transformed coordinates	
Numerical grid generation	Finite difference techniques									
Non-Orthogonal grids	Projectile shapes									
Axisymmetric bodies	Elliptic equations									
Transformed coordinates										
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) 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										

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. ABSTRACT (Continued)

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

	<u>Page</u>
LIST OF ILLUSTRATIONS.....	5
I. INTRODUCTION.....	7
II. GRID GENERATION.....	8
A. Surface Representation and Grid Point Distribution.....	9
B. Outer Boundary Formation and Grid Point Distribution.....	11
C. Grid Generation and Clustering.....	13
D. Grid Plotting.....	16
III. DISCUSSION OF RESULTS AND CONCLUDING REMARKS.....	16
REFERENCES.....	18
LIST OF SYMBOLS.....	33
APPENDIX A.....	35
APPENDIX B.....	59
APPENDIX C.....	65
APPENDIX D.....	77
DISTRIBUTION LIST.....	81

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	Standard Projectile Grid with Sting.....	19
2a	Cartesian-like Projectile Grid with Sting.....	19
2b	Special Case Isolated Boattail.....	19
3	Standard Projectile Grid with Base, Special Case of O-Grid with Symmetry.....	20
4	Tubular Projectile Grid of C-Grid.....	20
5	Projectile Blunt Body Grid (Fraction of O-Grid).....	21
6	Projectile with Symmetric Nose and Nonsymmetric Afterbody.....	21
7	Projectile Cross Section with Periodic B.C. (O-Grid).....	22
8	Projectile Cross Section with Symmetry Plane (O-Grid).....	22
9	Mapping from Physical Space to Computational Space.....	23
10a	Standard Projectile Grid in the Physical Plane.....	24
10b	Grid Detail near Projectile.....	25
11	Stretching Function, Points j are Specified along with x_0 , x_f , Δx_0 , and ∇x_f	26
12	Outer Boundary Structure and Terminology for Two Classes of Grid.....	27
13a	Overview of Straight Ray Grid for Tubular Projectile.....	28
13b	Grid Detail near Lower Trailing Edge.....	29
14a	Standard Projectile Grid with Controlled Reclustering along Lines of Constant ξ	30
14b	Boattail Base Detail Showing Specified Clustering in n near the Body.....	31
15	Hybrid Straight Ray and Elliptic Solver Grid Detail for a Tubular Projectile.....	32

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 ξ and η in the physical x-y plane as indicated in Figure 9a. Corresponding uniform values of ξ and η 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 ξ_x , ξ_y , η_x , and η_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 η -invariant⁴ flow calculations. A three-dimensional grid can be formed by rotating the grid about the axis and defining constant lines in θ . 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 $\xi = \xi_{\max}$ lines. The inner and outer boundaries correspond to $\eta = 0$ and $\eta = \eta_{\max}$.

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 $\eta = 0$ 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.

1. Steger, J. L., "Implicit Finite-Difference Simulation of Flow About Arbitrary Two-Dimensional Geometries", *AIAA Journal*, Vol. 16, July 1978, pp. 679-686.
2. Pulliam, T. H. and Steger, J. L. "On Implicit Finite-Difference Simulations of Three-Dimensional Flow", *AIAA Paper No. 78-10*, 1978.
3. Schiff, L. B. and Steger, J. L., "Numerical Simulation of Steady Supersonic Viscous Flow", *AIAA Paper No. 79-130*, 1979.
4. Nietubicz, C. J., Pulliam, T. H., and Steger, J. L., "Numerical Solution of the Azimuthal-Invariant Thin-Layer Navier-Stokes Equations", *AIAA Paper No. 79-0010*, 1979.

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 $\eta = 0$ boundary (i.e., specify ξ as a function of x and z along $\eta = 0$). 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 $\eta = \eta_{\max}$ boundary. Here we restrict $\xi = 0$ and $\xi = \xi_{\max}$ to be vertical or horizontal straightline rays in order to simplify programming logic, thus the endpoints of the $\eta_{\max}$ curve must properly align with those of the $\eta = 0$ boundary; (3) once the outer boundaries are defined, they are "connected" by generating the interior grid with appropriate clustering functions in η .

In the remainder of this Section the procedures used to generate the $\eta = 0$ boundary, the $\eta = \eta_{\max}$ 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

$$x_j = x_0 + a\psi_j + b\psi_j^2 + c\psi_j^3 \quad \begin{matrix} x_0 \leq x_j \leq x_f \\ j_0 \leq j \leq j_f \end{matrix} \quad (1)$$

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

$$\Delta x_0 = x_{j_0+1} - x_{j_0} \quad (2a)$$

$$\nabla x_f = x_{j_f} - x_{j_f-1} \quad (2b)$$

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

$$c = \{\nabla x_f + \Delta x_0 - 2h(x_f - x_0)\} / (h - 3h^2 + 2h^3)$$

$$b = \{\Delta x_0 - h(x_f - x_0) - c(h^3 - h)\} / (h^2 - h)$$

$$a = x_f - x_0 - b - c$$

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

The amount of clustering at each point is determined by the specified values of Δx_0 and ∇x_f . Moreover, because Δx_0 and ∇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 Δx_0 is too small and ∇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 interpolation⁵. 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.

5. Dahlquist, G. and Bjorck, A., "Numerical Methods". Prentice-Hall, Englewood Cliffs, New Jersey, 1974.

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 $n = n_{\max}$. Each segment, from a to b for example, is formed by specifying the endpoints x, y , and angle θ , where θ 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° and $\theta_d = 90^\circ$.

The data x, y, θ at each endpoint determines the shape of the parametric curves

$$\begin{aligned} x &= x_0 + \alpha_1 t + \alpha_2 t^2 \\ y &= y_0 + \beta_1 t + \beta_2 t^2 \end{aligned} \quad 0 \leq t \leq 1 \quad (4)$$

which are equivalent to a cubic

$$y = y_0 + \gamma_1(x-x_0) + \gamma_2(x-x_0)^2 + \gamma_3(x-x_0)^3 \quad (5)$$

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 β_2 are given by

$$\begin{aligned} \alpha_1 &= m_0 \\ \alpha_2 &= (x_f - x_0) - \alpha_1 \\ \beta_1 &= n_0 \\ \beta_2 &= (y_f - y_0) - \beta_1 \end{aligned} \quad (6)$$

where

$$\begin{aligned} m_0 &= \left. \frac{dx}{dt} \right|_0 & m_1 &= \left. \frac{dx}{dt} \right|_1 \\ n_0 &= \left. \frac{dy}{dt} \right|_0 & n_1 &= \left. \frac{dy}{dt} \right|_1 \end{aligned}$$

and

$$\begin{aligned} n_0 &= m_0 \left. \frac{dy}{dx} \right|_0 & \text{or} & & m_0 &= n_0 \left. \frac{dx}{dy} \right|_0 \\ n_1 &= m_1 \left. \frac{dy}{dx} \right|_1 & \text{or} & & m_1 &= n_1 \left. \frac{dx}{dy} \right|_1 \end{aligned}$$

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

i) If $y_x|_0 < x_y|_0$, $y_x|_1 < x_y|_1$, and $y_x|_1 \neq y_x|_0$

$$m_1 = 2 [(x_f - y_0) - (x_f - x_0) y_x|_0] / (y_x|_1 - y_x|_0)$$

$$m_0 = 2 (x_f - x_0) - m_1$$

$$n_0 = m_0 y_x|_0$$

$$n_1 = m_1 y_x|_1$$

(7a)

ii) If $y_x|_0 < x_y|_0$, $x_y|_1 < y_x|_1$, and $x_y|_1 y_x|_0 \neq 1$

$$m_0 = 2 [(x_f - x_0) - (y_f - y_0) x_y|_1] / (1 - x_y|_1 y_x|_0)$$

$$n_1 = 2 (y_f - y_0) - m_0 y_x|_0$$

$$n_0 = m_0 y_x|_0$$

$$m_1 = n_1 x_y|_1$$

(7b)

iii) If $x_y|_0 < y_x|_0$, $y_x|_1 < x_y|_1$, and $y_x|_1 x_y|_0 \neq 1$

$$n_0 = 2 [(y_f - y_0) - (x_f - x_0) y_x|_1] / (1 - y_x|_1 x_y|_0)$$

$$m_1 = 2 (x_f - x_0) - n_0 x_y|_0$$

$$m_0 = n_0 x_y|_0$$

$$n_1 = m_1 y_x|_1$$

(7c)

iiii) If $x_y|_0 < y_x|_0$, $x_y|_1 < y_x|_1$, and $x_y|_1 \neq x_y|_0$

$$n_1 = 2 [(x_f - x_0) - (y_f - y_0) x_y|_0] / (x_y|_1 - x_y|_0)$$

$$n_0 = 2 (y_f - y_0) - n_1$$

$$m_0 = n_0 x_y|_0$$

$$m_1 = n_1 x_y|_1$$

(7d)

Whenever the third constraint is violated (for example in case (i), if $y_x|_1 = y_x|_0$) 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 $\eta_{\max}$ boundaries. Two types of grid generation procedures were used and are discussed below.

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

$$\Delta s_k = \Delta s_0 (1 + \epsilon)^{k-1}, \quad k = 1, k_{\max} - 1 \quad (8)$$

Here Δs_0 is the specified constant grid spacing at the inner boundary. The parameter ϵ 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 $\eta = \eta_{\max}$ for points which have the same values of ξ . Figures 13a and 13b illustrate a straight ray grid with clustering in η 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 $y_{j,k}$:

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -J^2 (\bar{P}x_{\xi} + \bar{Q}x_{\eta}) \quad (9a)$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -J^2 (\bar{P}y_{\xi} + \bar{Q}y_{\eta}) \quad (9b)$$

where

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

and

$$\bar{P} = P_0 e^{-a(\eta-\eta_0)} + P_m e^{-a(\eta-\eta_{\max})}$$

$$\bar{Q} = Q_0 e^{-b(\eta-\eta_0)} + Q_m e^{-b(\eta-\eta_{\max})}$$

Here P_0 , Q_0 , P_m , Q_m , a and b are prescribed clustering parameters. Along the $\eta = 0$ and $\eta = \eta_{\max}$ boundaries, $x_{j,k}$ and $y_{j,k}$ have been previously prescribed. Along the $\xi = 0$ and $\xi = \xi_{\max}$, which are either vertical or horizontal lines in the physical space, the following boundary conditions are enforced: either

$$x \text{ is given and } y_{\xi} = 0 \quad (10a)$$

on a vertical boundary, or

$$x_{\xi} = 0 \text{ and } y \text{ is given} \quad (10b)$$

on a horizontal boundary.

-
6. Chu, W. H., "Development of a General Finite Difference Approximation for a General Domain". *Journal of Comp. Physics*, Vol. 8, 1971, pp. 392-408.
 7. Thompson, J. F., Thames, F. C., and Mastin, C. M., "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing any Number of Arbitrary Two-Dimensional Bodies". *Journal of Comp. Physics*, Vol. 15, 1974, pp. 299-319.
 8. Sorenson, R. L. and Steger, J. L., "Simplified Clustering of Nonorthogonal Grids Generated by Elliptic Partial Differential Equations". NASA TM 73252, August 1977.

For periodic grids as indicated in Figure 7, these boundary conditions in ξ 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

$$\begin{aligned}
 x_{\xi} &= (x_{j+1,k} - x_{j-1,k}) / (2\Delta\xi) \\
 x_{\eta} &= (x_{j,k+1} - x_{j,k-1}) / (2\Delta\eta) \\
 x_{\xi\xi} &= (x_{j+1,k} - 2x_{j,k} + x_{j-1,k}) / (\Delta\xi)^2 \\
 x_{\xi\eta} &= (x_{j+1,k+1} - x_{j+1,k-1} - x_{j-1,k+1} + x_{j-1,k-1}) / (4\Delta\xi\Delta\eta) \\
 x_{\eta\eta} &= (x_{j,k+1} - 2x_{j,k} + x_{j,k-1}) / (\Delta\eta)^2
 \end{aligned} \tag{11}$$

while derivatives of y are treated identically. The Jacobian J is formed with central differencing. The right hand side companion terms to $\bar{P}$ and $\bar{Q}$, however, are backward or forward differenced depending on the sign of $\bar{P}$ and $\bar{Q}$. If $\bar{P}$ is positive, x_{ξ} and y_{ξ} are forward differenced. The terms x_{η} , y_{η} 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 $\bar{P}$ and $\bar{Q}$ 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 $\bar{P} = \bar{Q} = 0$. The x, y points along a $\xi = \text{constant}$ 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 ξ . 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 ξ . Figure 15 shows details of such a procedure used in the previous tubular projectile case. Here the ξ -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 η 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 $\xi = \xi_{\max}$ 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 η 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 ξ . 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 Δx (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 O-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

1. Steger, J. L., "Implicit Finite-Difference Simulation of Flow About Arbitrary Two-Dimensional Geometries", AIAA Journal, Vol. 16, July 1978, pp. 679-686.
2. Pulliam, T. H., and Steger, J. L., "On Implicite Finite-Difference Simulations of Three-Dimensional Flow", AIAA Paper No. 78-10, 1978.
3. Schiff, L. B., and Steger, J. L., "Numerical Simulation of Steady Supersonic Viscous Flow", AIAA Paper No. 79-130, 1979.
4. Nietubicz, C. J., Pulliam, T. H., and Steger, J. L., "Numerical Solution of the Azimuthal-Invariant Thin-Layer Navier-Stokes Equations", AIAA Paper No. 79-0010, 1979.
5. Dahlquist, G. and Bjorck, A., "Numerical Methods". Prentice-Hall, Englewood Cliffs, New Jersey, 1974.
6. Chu, W. H., "Development of a General Finite Difference Approximation for a General Domain". Journal of Comp. Physics, Vol. 8, 1971, pp. 392-408.
7. Thompson, J. F., Thames, F. C., and Mastin, C. M., "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing any Number of Arbitrary Two-Dimensional Bodies". Journal of Comp. Physics, Vol. 15, 1974, pp. 299-319.
8. Sorenson, R. L., and Steger, J. L., "Simplified Clustering of Nonorthogonal Grids Generated by Elliptic Partial Differential Equations". NASA TM 73252, August 1977.

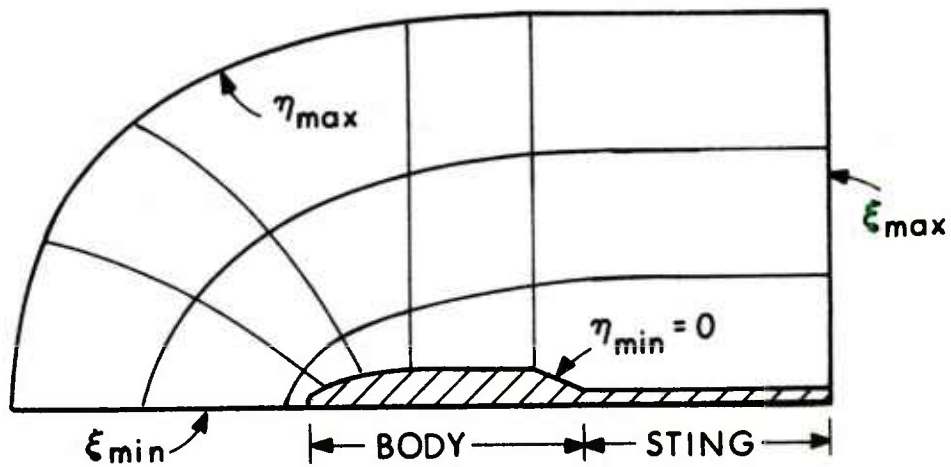


Figure 1. Standard Projectile Grid with Sting

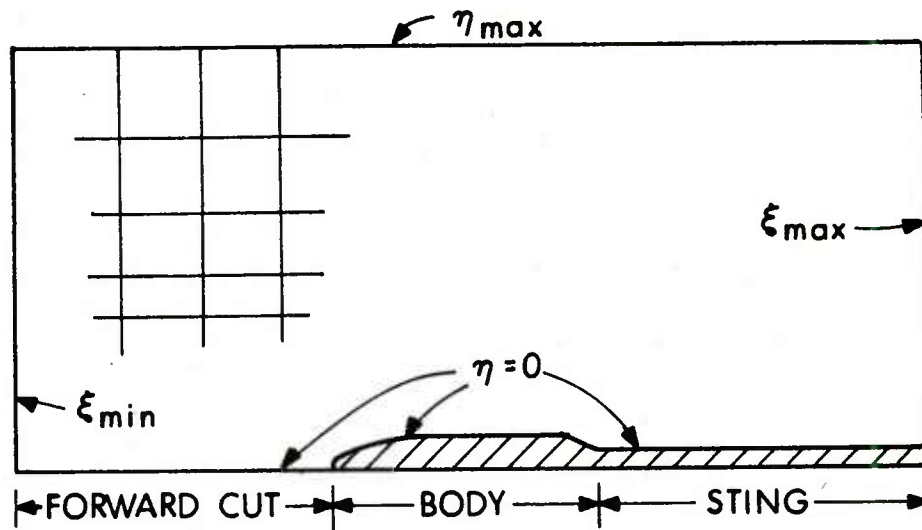


Figure 2a. Cartesian-like Projectile Grid with Sting



Figure 2b. Special Case Isolated Boattail

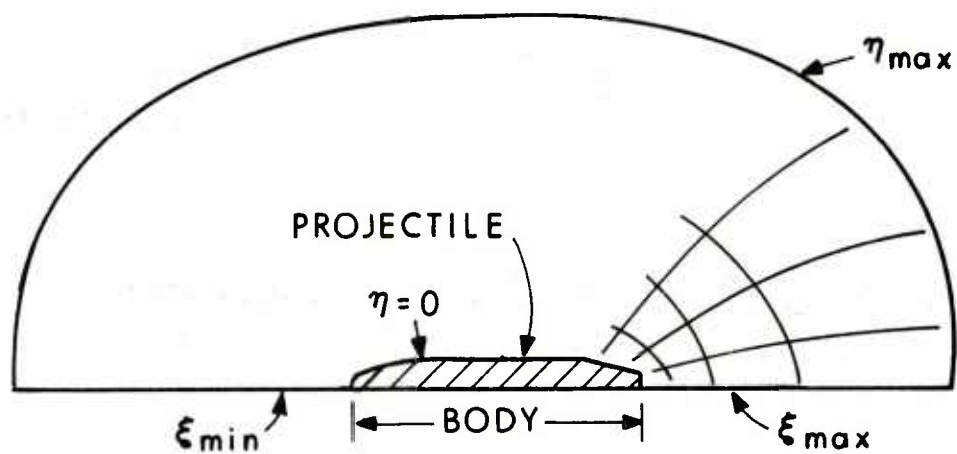


Figure 3. Standard Projectile Grid with Base,
Special Case of O-Grid with Symmetry

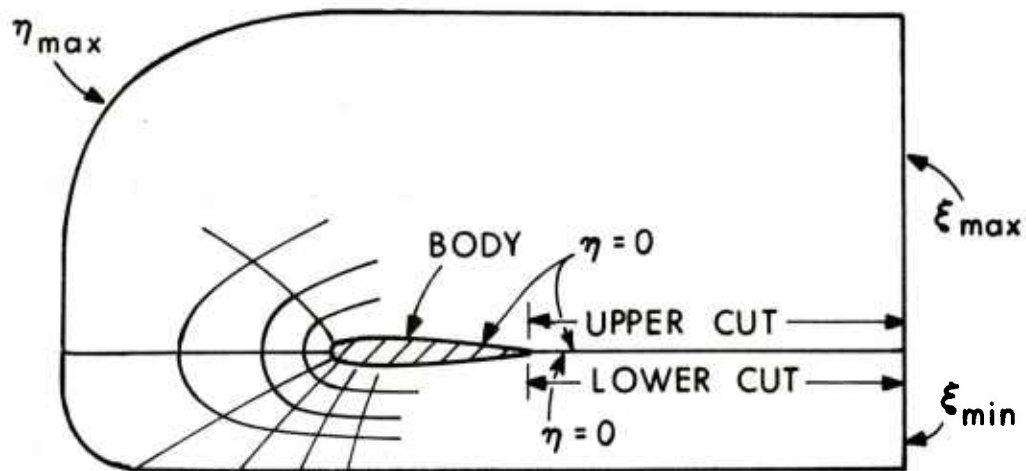


Figure 4. Tubular Projectile Grid or C-Grid

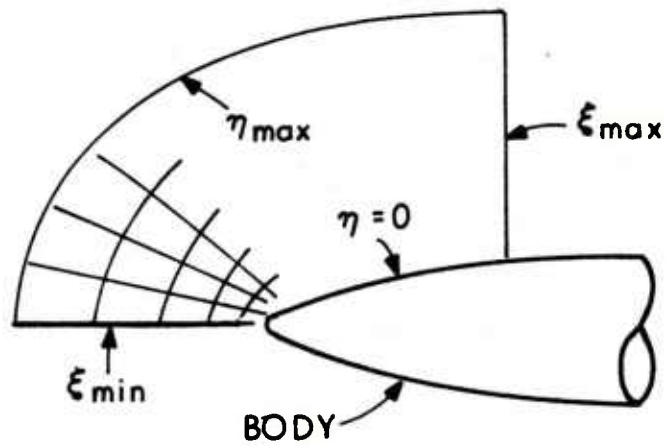


Figure 5. Projectile Blunt Body Grid (Fraction of O-Grid)

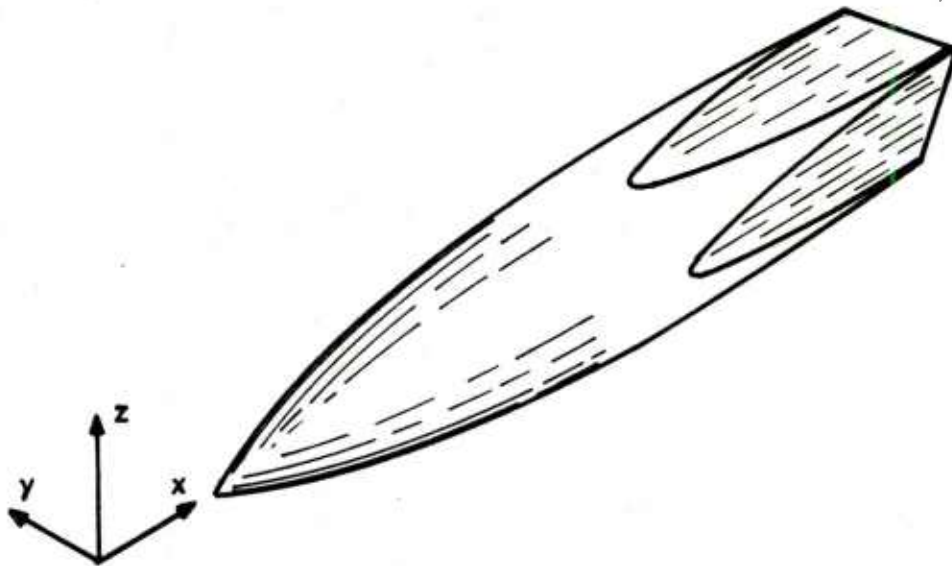


Figure 6. Projectile with Symmetric Nose and Nonsymmetric Afterbody

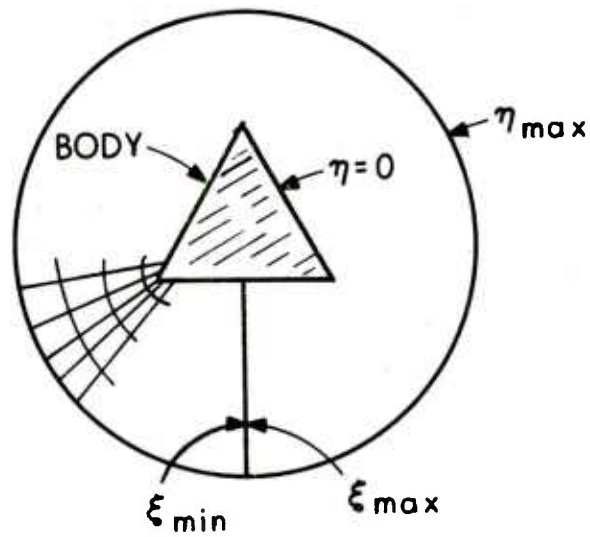


Figure 7. Projectile Cross Section with Periodic B.C. (O-Grid)

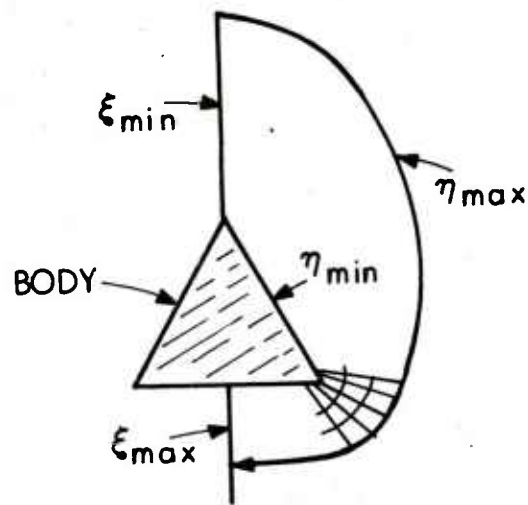


Figure 8. Projectile Cross Section with Symmetry Plane (O-Grid)

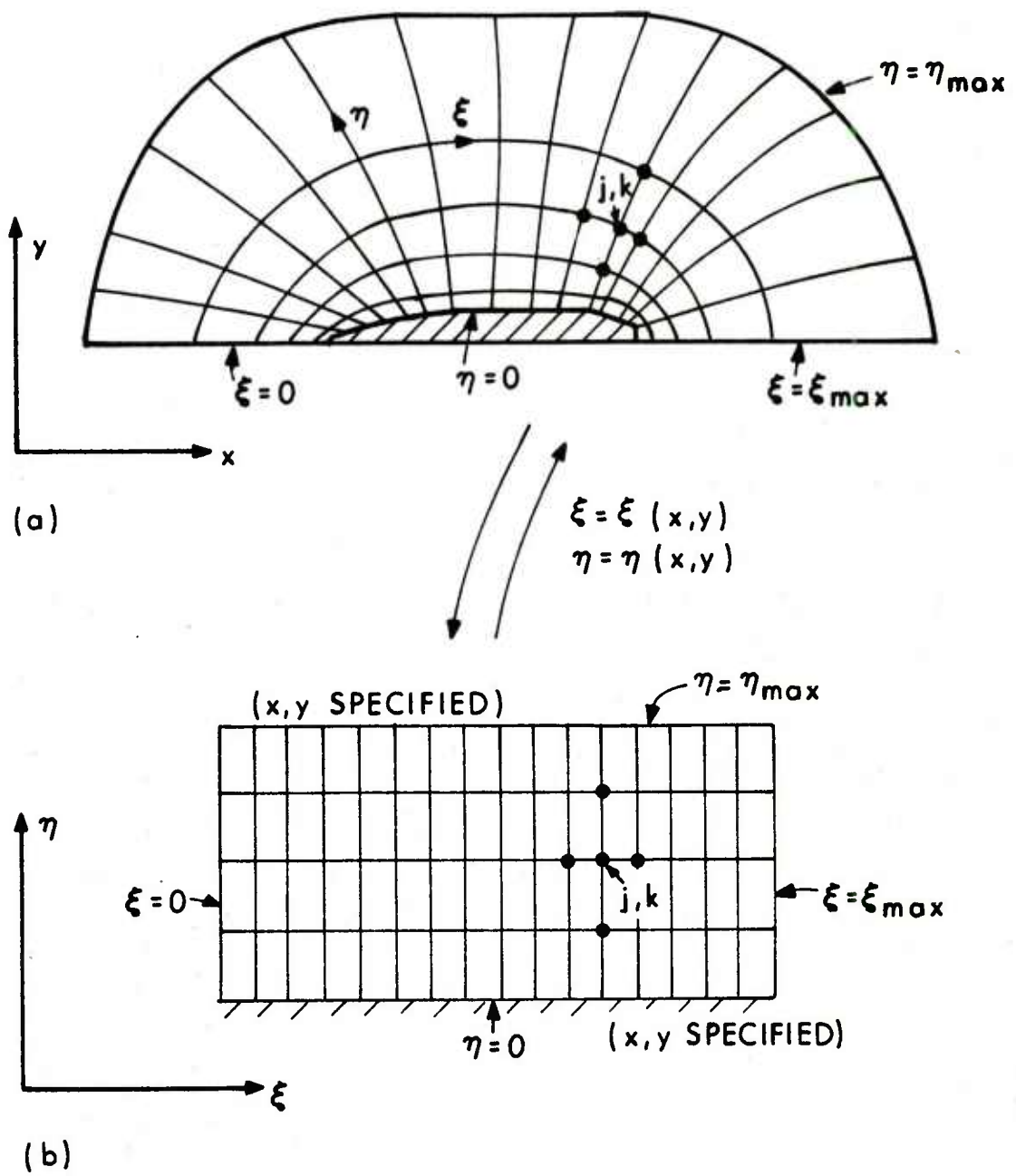


Figure 9. Mapping from Physical Space to Computational Space

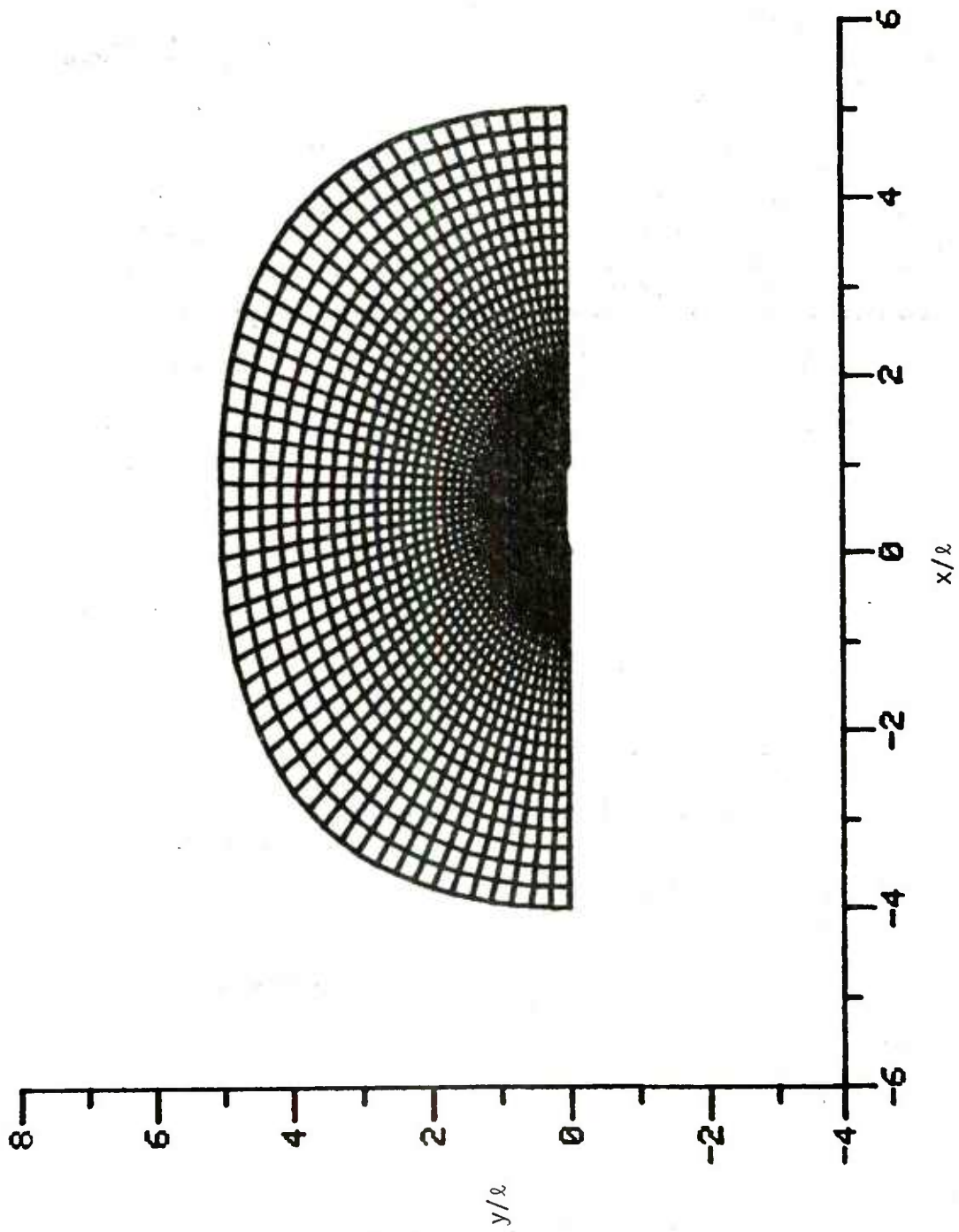


Figure 10a. Standard Projectile Grid in the Physical Plane

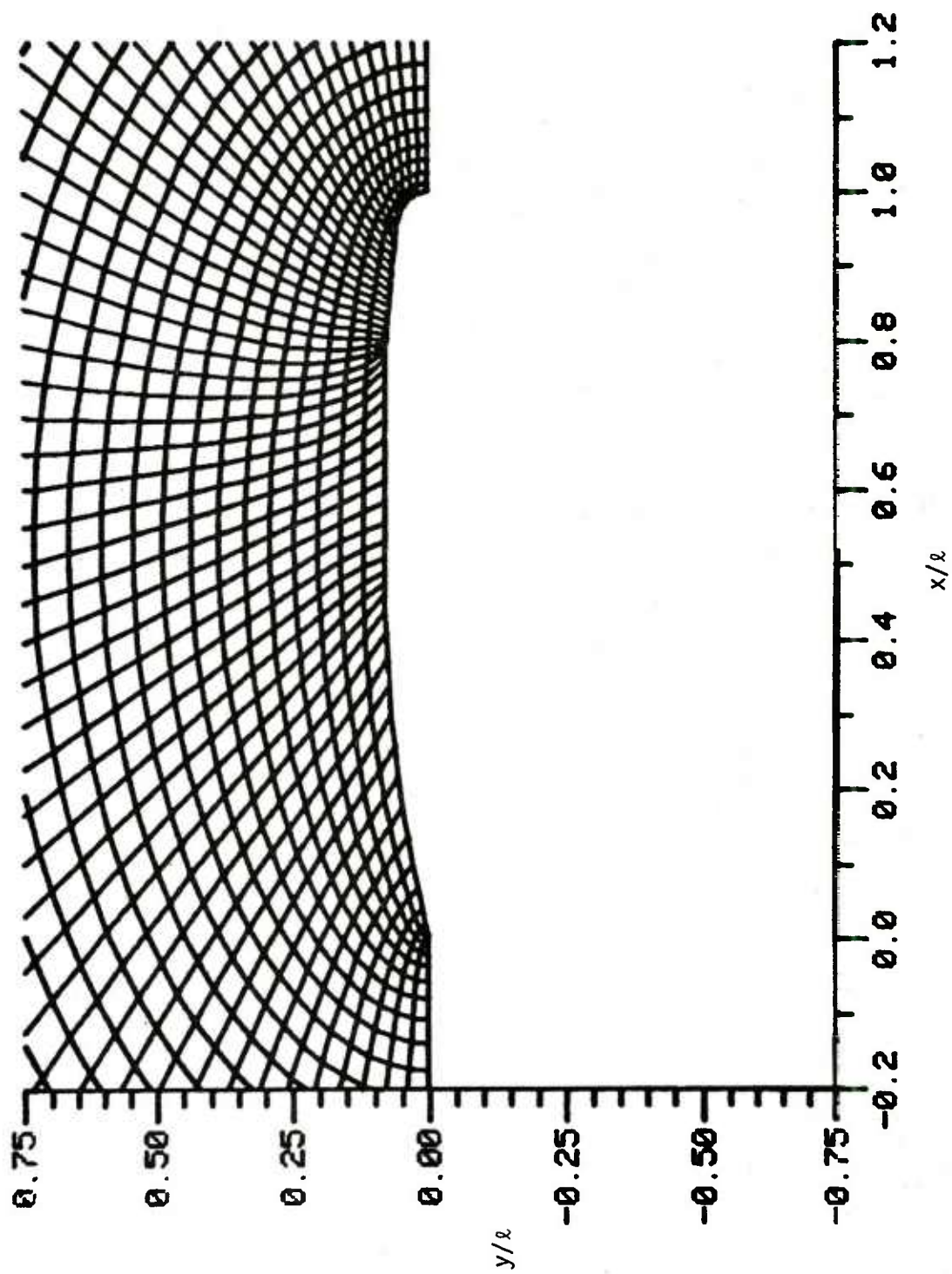


Figure 10b. Grid Detail near Projectile

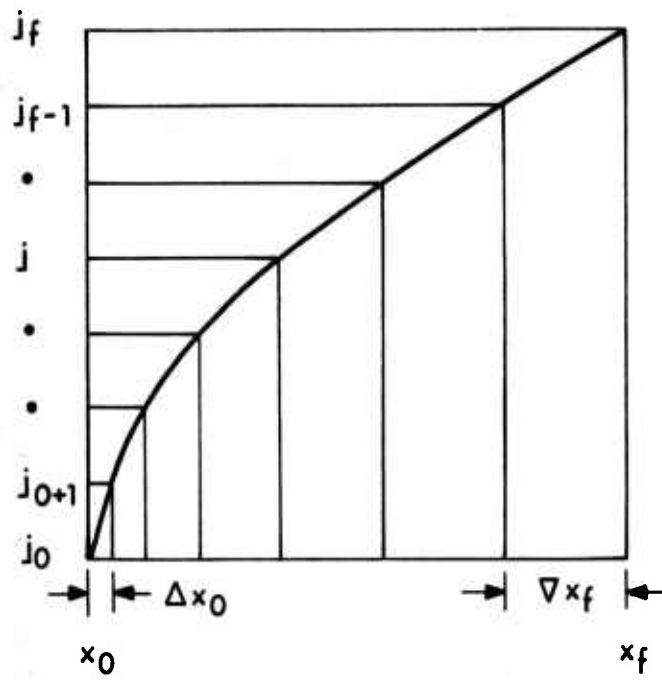
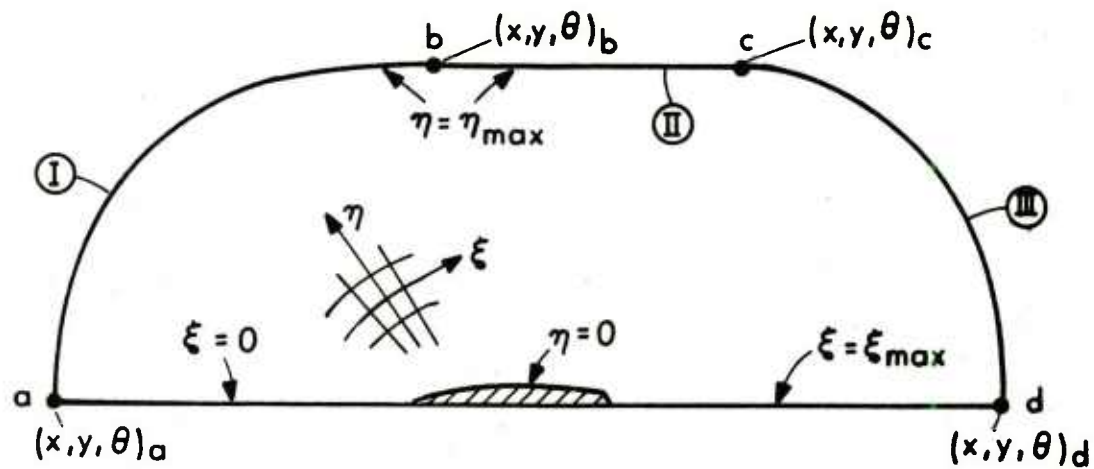
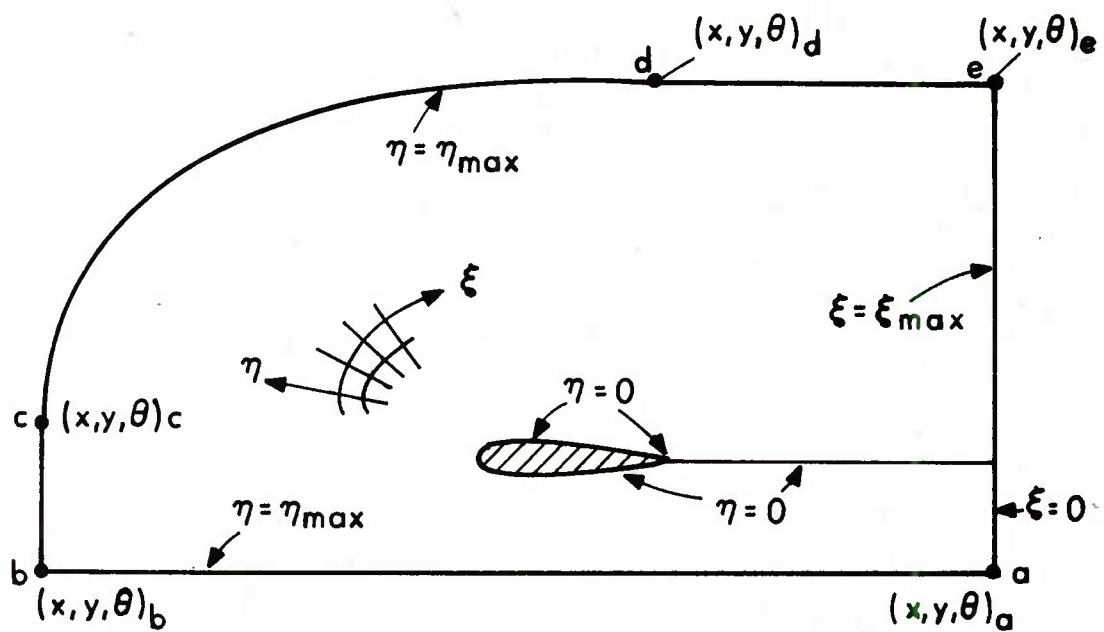


Figure 11. Stretching Function, Points j are Specified along with x_0 , x_f , Δx_0 , and ∇x_f



(a) STANDARD PROJECTILE GRID



(b) C-GRID FOR TUBULAR PROJECTILE

Figure 12. Outer Boundary Structure and Terminology for Two Classes of Grid

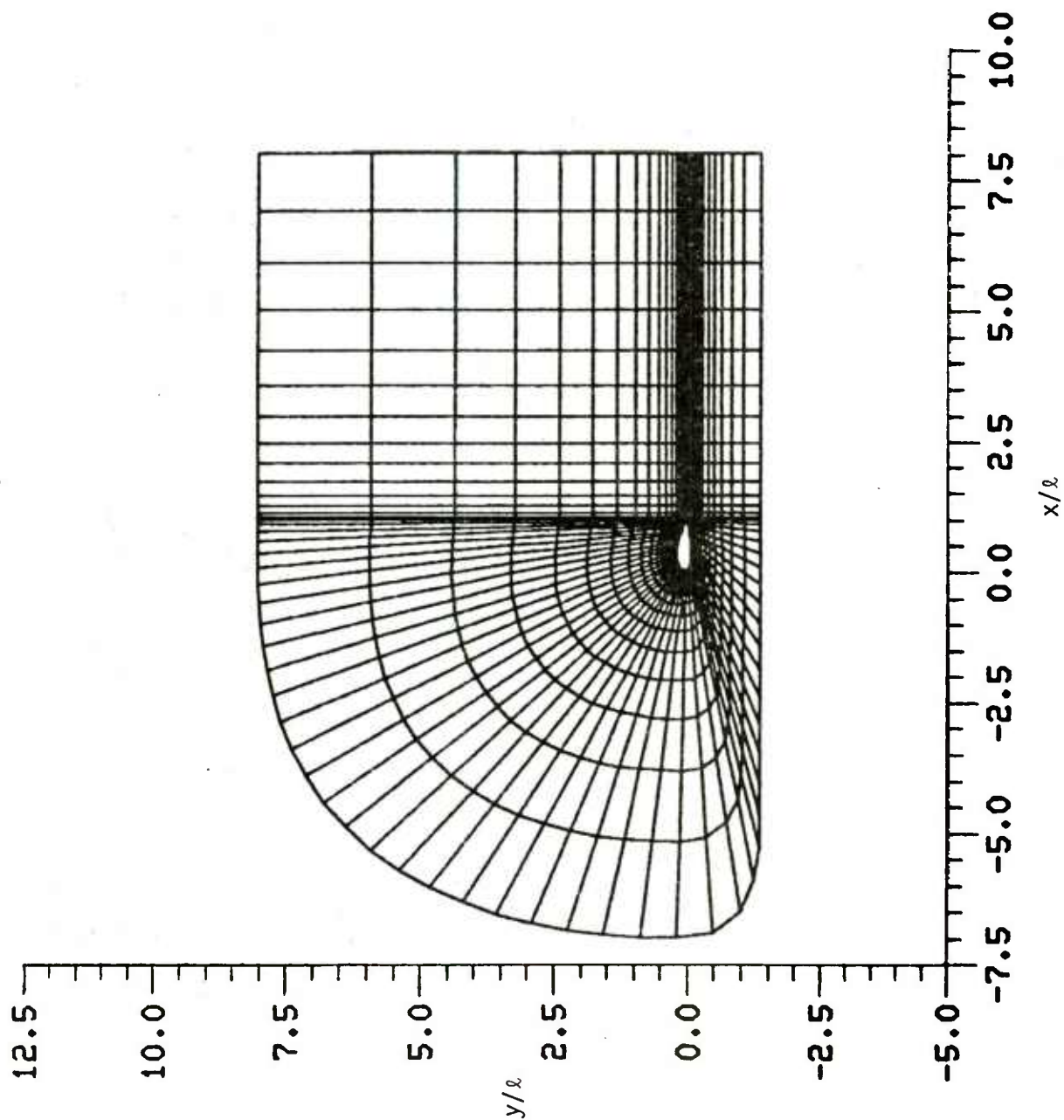


Figure 13a. Overview of Straight Ray Grid for Tubular Projectile

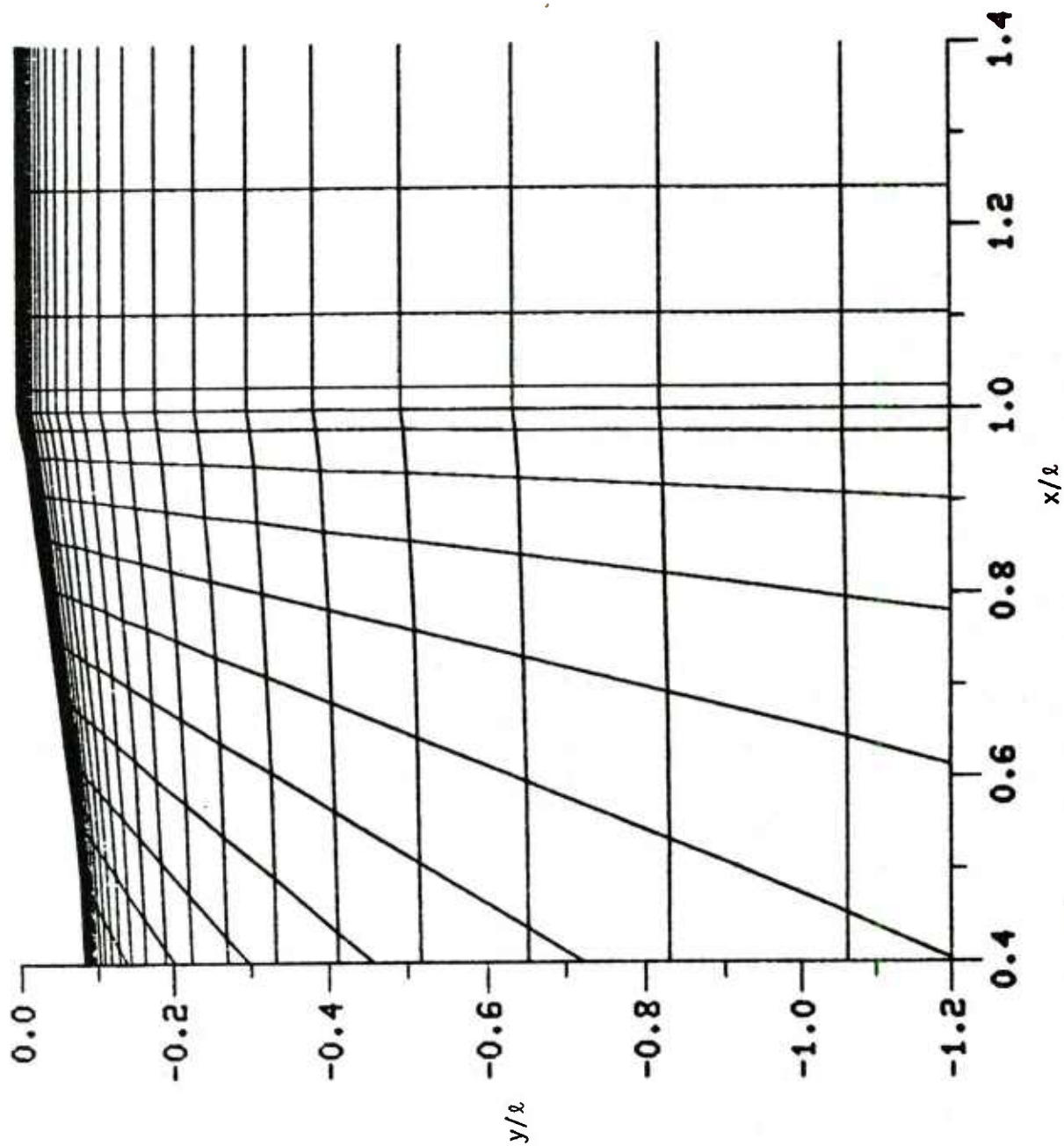


Figure 13b. Grid Detail near Lower Trailing Edge

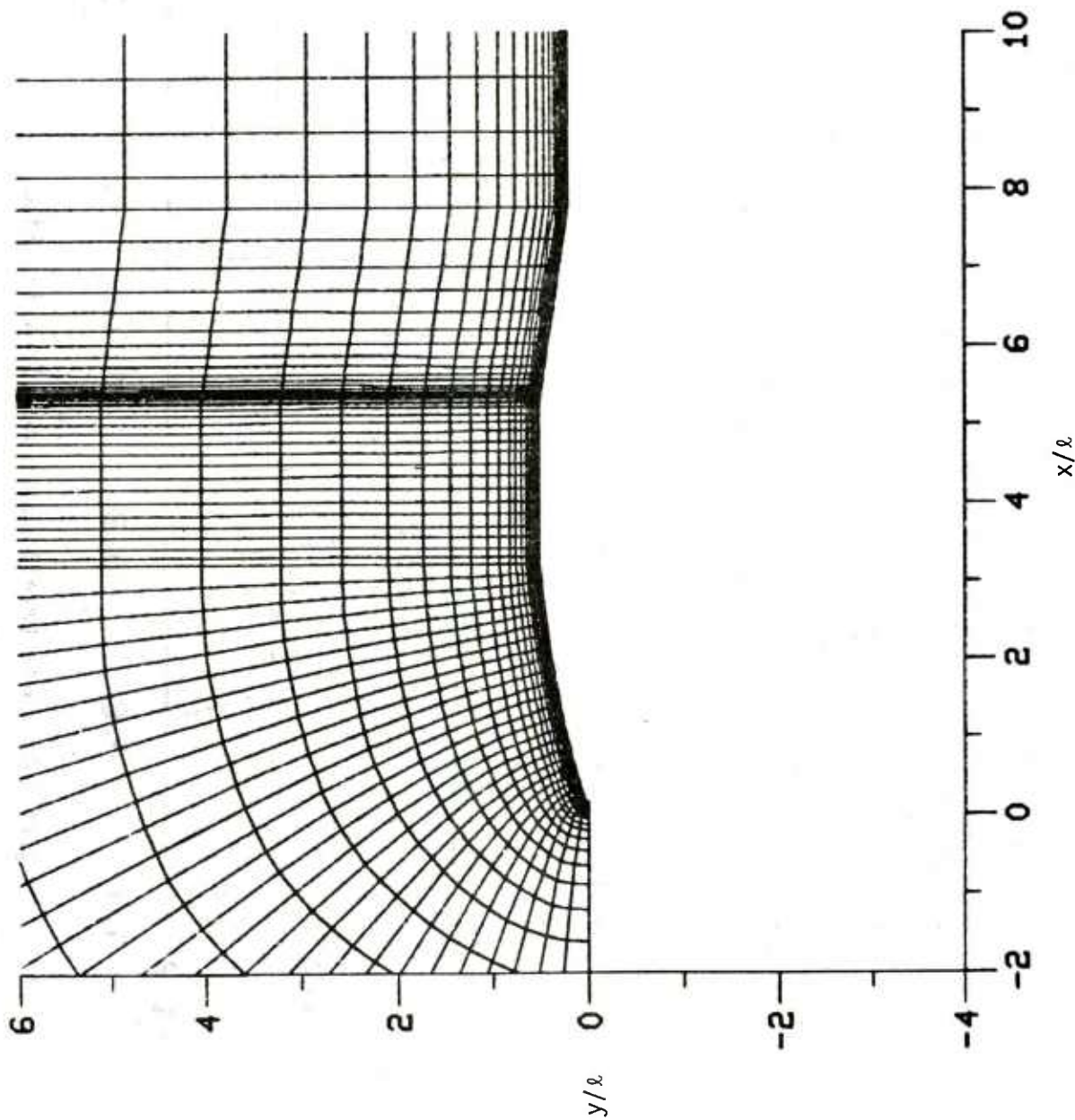


Figure 14a. Standard Projectile Grid with Controlled Reclustering along Lines of Constant x/l

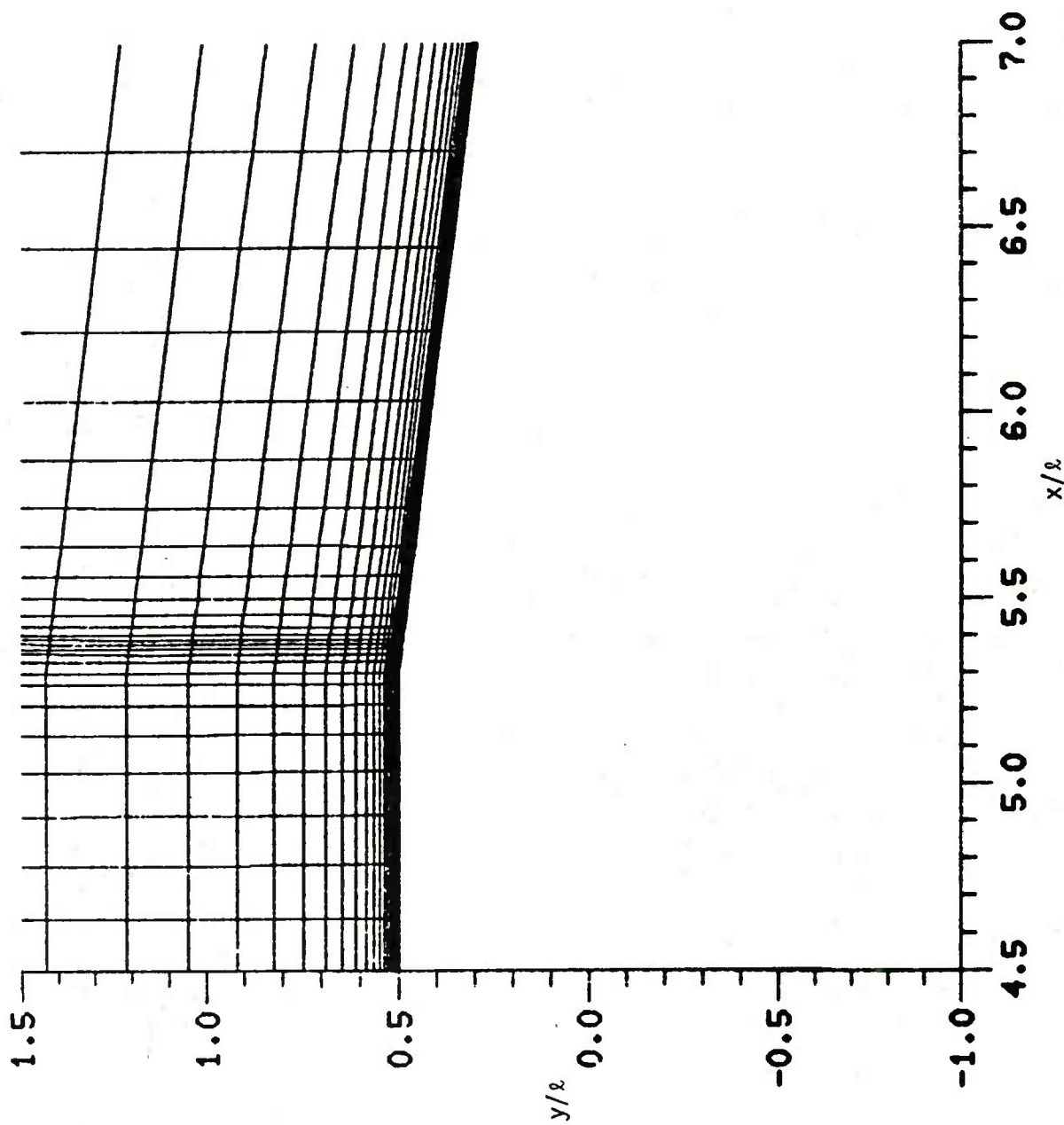


Figure 14b. Boattail Base Detail Showing Specified Clustering in η near the Body

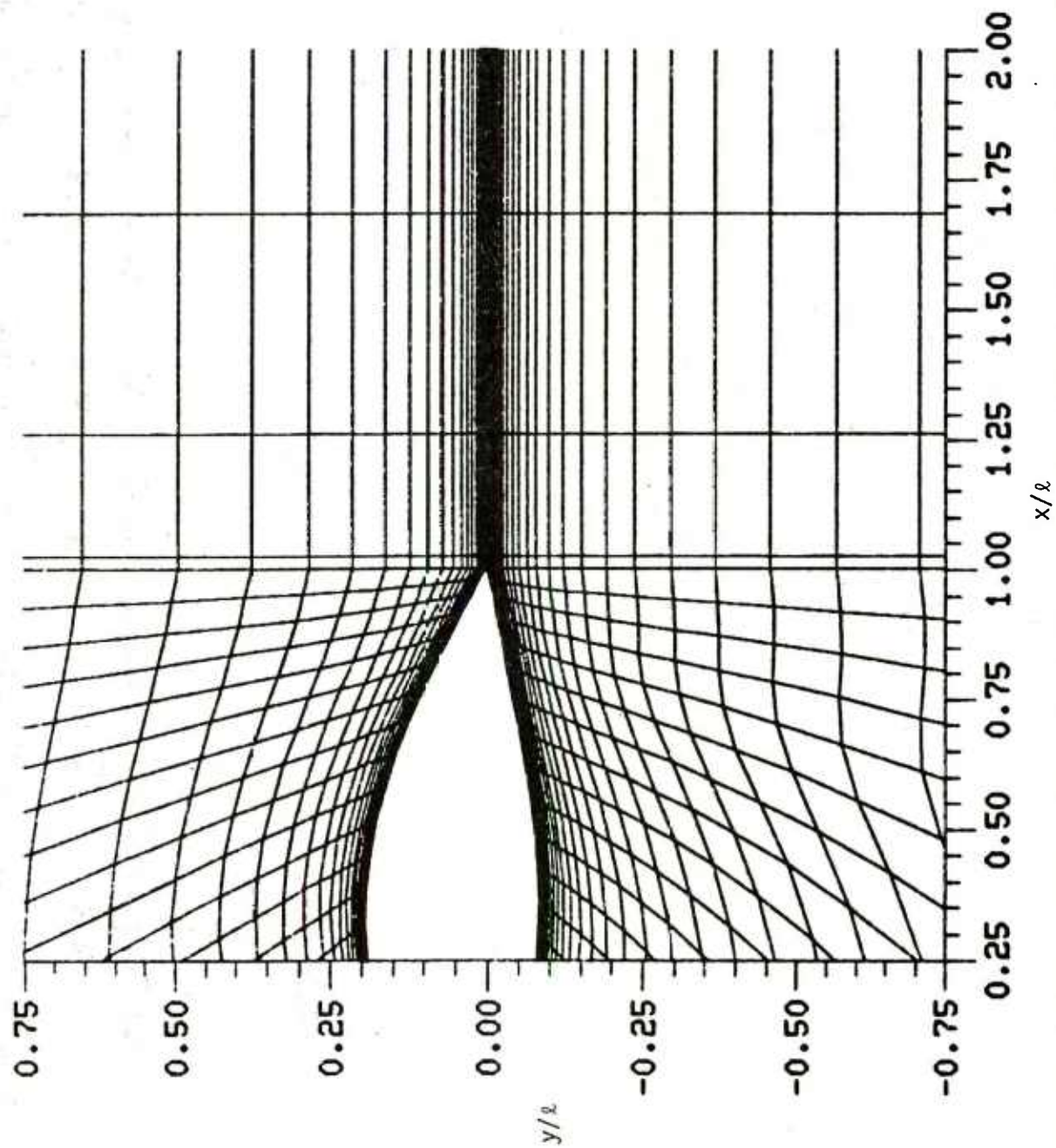


Figure 15. Hybrid Straight Ray and Elliptic Solver Grid Detail for a Tubular Projectile

LIST OF SYMBOLS

j	index value in ξ direction
k	index value in μ direction
s	arc length
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
Δ	forward finite difference
Δs	grid spacing at the inner boundary
∇	backward finite difference
ϵ	clustering parameter in η direction
θ	angle between segments of the outer boundary and the x-axis
ξ,η	computational coordinates in the axial and radial directions
L	model length

Subscripts

f	final value
o	initial value

APPENDIX A

COMPUTER CODE DESCRIPTION AND LISTING

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 $\bar{y}$, as a function of the distribution arc length $\bar{s}$. 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 NCGRD. NCGRD-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 $\bar{P}$ and $\bar{Q}$ 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)	MAIN	2
C		MAIN	3
C	CHECK FOR 30 GRID	MAIN	4
C		MAIN	5
	COMMON JMAX, KMAX, JM, KM, N800, JB00	MAIN	6
	COMMON /8000Y/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)	MAIN	7
	1 , Y(100), TS(100)	MAIN	8
	COMMON /COMP/ X(100), Y(100)	MAIN	9
C		MAIN	10
C		MAIN	11
	READ (5,60) I30,NO,LMAX	MAIN	12
	WRITE (6,50) I30,NO,LMAX	MAIN	13
C		MAIN	14
C		MAIN	15
C	DISTRIBUTE POINTS ALONG INNER BOUNDARY	MAIN	16
	WRITE (6,70)	MAIN	17
	CALL 800Y	MAIN	18
	READ (5,60) NFLAG	MAIN	19
	IF (NFLAG.LT.0) GO TO 40	MAIN	20
	READ (5,60) MCGRO,NCART	MAIN	21
	WRITE (6,80) MCGRO,NCART	MAIN	22
	IF (MCGRO.GT.0) CALL STING (MCGRO)	MAIN	23
	IF (NCART.GT.0) CALL CARTB (NCART)	MAIN	24
	JMAX=JB00	MAIN	25
	JM=JMAX-1	MAIN	26
	WRITE (6,90) JMAX	MAIN	27
	IF (MCGRO.GT.0.OR.NCART.GT.0) GO TO 10	MAIN	28
	GO TO 30	MAIN	29
	10 WRITE (6,100)	MAIN	30
	DO 20 J=1,JMAX	MAIN	31
	WRITE (6,110) J,XX(J),YY(J)	MAIN	32
	20 CONTINUE	MAIN	33
	30 CONTINUE	MAIN	34
C		MAIN	35
C	FORM OUTER BOUNDARY	MAIN	36
	WRITE (6,120)	MAIN	37
	READ (5,60) NSEGS,IOUTO	MAIN	38
	WRITE (6,130) JMAX,NSEGS	MAIN	39
	CALL OUTER (NSEGS,IOUTO)	MAIN	40
C		MAIN	41
C	GRID GENERATION	MAIN	42
	WRITE (6,140)	MAIN	43
	CALL ALGRO (ISTOR)	MAIN	44
C		MAIN	45
C		MAIN	46
C	FORM 30 GRID. LMAX IS CIRCUMFERENTIAL DIRECTION	MAIN	47
	IF (I30.EQ.1) CALL GSPIN (I30,NO,ISTOR,LMAX)	MAIN	48
	40 STOP	MAIN	49
C		MAIN	50
	50 FORMAT (1H0,11H(30,NO,LMAX,3I5)	MAIN	51
	60 FORMAT (8I5)	MAIN	52
	70 FORMAT (1H1,36H+++++ INNER BOUNDARY +++++)	MAIN	53
	80 FORMAT (1H0,13H MCGRO,NCART ,2I5)	MAIN	54
	90 FORMAT (1H0,21H FINAL VALUE OF JMAX ,I5)	MAIN	55
	100 FORMAT (1H0,43H FINAL VALUES OF J,X,Y ALONG INNER BOUNDARY)	MAIN	56
	110 FORMAT (1H ,I5,2F13.6)	MAIN	57
	120 FORMAT (1H1,37H ++++++ OUTER BOUNDARY ++++++)	MAIN	58
	130 FORMAT (1H0,10HJMAX,NSEGS,2I5)	MAIN	59
	140 FORMAT (1H1,39H ++++++ GRID GENERATION ++++++)	MAIN	60
	END	MAIN	61

	SUBROUTINE ALGRD (ISTDR)	ALGRD	2
	COMMON JMAX, KMAX, JM, KM, N800, J800	ALGRD	3
	COMMON /BOUDY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)	ALGRD	4
	1, T(100), TS(100)	ALGRD	5
	COMMON /GRID/ X(80,60), Y(80,60)	ALGRD	6
	COMMON /ARRAY/ A(100), B(100), C(100), D(100), F(100), H(100)	ALGRD	7
C		ALGRD	8
C	FORM ALGEBRAIC GRID OR ELLIPTIC EQ. GENERATED GRID	ALGRD	9
C		ALGRD	10
	READ (5,150) KMAX,ITERM,IPER,NCLUS,ISTDR,JELLI	ALGRD	11
	WRITE (6,160) KMAX,ITERM,IPER,NCLUS,ISTDR,JELLI	ALGRD	12
	KM=KMAX-1	ALGRD	13
	READ (5,170) DS,DMEGA	ALGRD	14
	WRITE (6,180) DS,DMEGA	ALGRD	15
C		ALGRD	16
C	STRAIGHT RAY GRID USED IF ITERM .LE. 0	ALGRD	17
C	OTHERWISE USED AS INITIAL GUESS	ALGRD	18
	DO 40 J=1,JMAX	ALGRD	19
	XO=XX(J)	ALGRD	20
	X1=XS(J)	ALGRD	21
	YO=YY(J)	ALGRD	22
	Y1=YS(J)	ALGRD	23
	R=SQRT((X1-XD)**2+(Y1-YD)**2)	ALGRD	24
	EPS=D.	ALGRD	25
	IF (ITERM.LT.D.) GO TO 10	ALGRD	26
	DD=R/(KMAX-1)	ALGRD	27
	GO TO 20	ALGRD	28
10	EPS=EPSIL(R,D.,DS,<MAX,D.DDD02,20,J)	ALGRD	29
	DD=DS	ALGRD	30
20	X(J,1)=XD	ALGRD	31
	Y(J,1)=YD	ALGRD	32
	TR=0.	ALGRD	33
	DO 30 K=2,<MAX	ALGRD	34
	TR=TR+DD*(1.+EPS)**(K-2)	ALGRD	35
	TT=TR/R	ALGRD	36
	X(J,K)=XO+(X1-XD)*TT	ALGRD	37
	Y(J,K)=YD+(Y1-YD)*TT	ALGRD	38
30	CONTINUE	ALGRD	39
40	CONTINUE	ALGRD	40
C		ALGRD	41
	IF (ITERM.LT.D) GO TO 70	ALGRD	42
C		ALGRD	43
C	ELLIPTIC P.D.E. GRID GENERATION SCHEME	ALGRD	44
	IF (JELLI.GE.1) GO TO 50	ALGRD	45
	CALL RELAX (ITERM,IPER,1,JMAX,JMEGA)	ALGRD	46
	GO TO 70	ALGRD	47
50	DO 60 LL=1,JELLI	ALGRD	48
	READ (5,150) JI,JF	ALGRD	49
	WRITE (6,190) JI,JF	ALGRD	50
60	CALL RELAX (ITERM,IPER,JI,JF,JMEGA)	ALGRD	51
70	CONTINUE	ALGRD	52
C		ALGRD	53
C	CLUSTERING OPTION	ALGRD	54
	IF (NCLUS.LT.D) GO TO 130	ALGRD	55
	DO 120 J=1,JMAX	ALGRD	56
	T(1)=0.	ALGRD	57
	DO 80 K=2,<MAX	ALGRD	58

83	T(K)=T(K-1)+SQRT((X(J,K)-X(J,K-1))**2+(Y(J,K)-Y(J,K-1))**2)	ALGRD	59
	EPS=EPSIL(T(KMAX),D,.DS,KMAX,D.DDD02,2D,J)	ALGRD	60
	S(1)=0.	ALGRD	61
	N=-1	ALGRD	62
	DD 90 K=2,KMAX	ALGRD	63
	N=N+1	ALGRD	64
90	S(K)=S(K-1)+DS*(1.+EPS)**N	ALGRD	65
	DO 100 K=1,KMAX	ALGRD	66
	XX(K)=X(J,K)	ALGRD	67
100	YY(K)=Y(J,K)	ALGRD	68
	CALL CSPLIN (S,XS,T,XX,A,B,C,D,F,H,2,KM,1,KMAX)	ALGRD	69
	CALL CSPLIN (S,YS,T,YY,A,B,C,D,F,H,2,KM,1,KMAX)	ALGRD	70
	DD 110 K=2,KM	ALGRD	71
	X(J,K)=XS(K)	ALGRD	72
110	Y(J,K)=YS(K)	ALGRD	73
120	CONTINUE	ALGRD	74
C		ALGRD	75
130	CONTINUE	ALGRD	76
C		ALGRD	77
C	SUPPRESS THE PRINTOUT	ALGRD	78
C	K = KMAX/2	ALGRD	79
C	DD 20 J=1,JMAX	ALGRD	80
C	WRITE(6,603) X(J,2),Y(J,2),X(J,3),Y(J,3),X(J,4),Y(J,4),	ALGRD	81
C	1 X(J,K),Y(J,K),X(J,KM),Y(J,KM)	ALGRD	82
C	603 FORMAT(1H ,10F12.5)	ALGRD	83
C	20 CONTINUE	ALGRD	84
C		ALGRD	85
C		ALGRD	86
C		ALGRD	87
C	OPTIONAL STORE OF DATA	ALGRD	88
	IF (ISTDR.LE.0) GO TO 140	ALGRD	89
	WRITE (9) ((X(J,K),J=1,JMAX),K=1,KMAX),((Y(J,K),J=1,JMAX),K=1,	ALGRD	90
	1 KMAX)	ALGRD	91
140	CONTINUE	ALGRD	92
	RETURN	ALGRD	93
C		ALGRD	94
150	FORMAT (8I5)	ALGRD	95
160	FORMAT (1H,' SUB. ALGRD PRINTOUT.... KMAX,ITERM,IPER,NCLUS,ISTOR,	ALGRD	96
	1 JELLI',/,8I5)	ALGRD	97
170	FORMAT (8F10.0)	ALGRD	98
180	FORMAT (1H,11H DS, DMEGA ,2F13.5)	ALGRD	99
190	FORMAT (1H,26H ELLIPTIC GRID FORMED FROM ,13,5H TO ,13)	ALGRD	100
	END		

SUBROUTINE BDDAN I1SEGS)	BDDAN	2
COMMON /CALC/ X0, XATF, BS, ICT, FLAG, POYDX, FFX, RFM	BDDAN	3
COMMON /IMPS/ X1, X2, X3, X4, RAD, DYDX, CHORD, FUSE, AS, RAD5	BDDAN	4
COMMON JMAX, KMAX, JM, KM, NRDO, JBDD	BDDAN	5
COMMON /BDDY/ XX(100), YY(100), XS(100), SS(100), S(100)	BDDAN	6
1 , T(100), TS(100)	BDDAN	7
C	BDDAN	8
READ (5,100) TAU,FLAG	BDDAN	9
WRITE (6,110) TAU,FLAG	BDDAN	10
JWRIT=JBDD	BDDAN	11
ICT=0	BDDAN	12
IF (FLAG.GE.1.) CALL SCALC	BDDAN	13
CALL BDDIS (ISEGS,JWRIT)	BDDAN	14
DO 10 J=1,JBDD	BDDAN	15
10 XX(J)=S(J)	BDDAN	16
C	BDDAN	17
C ANALYTIC BODY SHAPE	BDDAN	18
IF (FLAG.GE.0.) GO TO 30	BDDAN	19
DO 20 J=1,JBDD	BDDAN	20
XX(J)=S(J)/S(JBDD)	BDDAN	21
C PARABOLIC ARC	BDDAN	22
20 YY(J)=2.*TAU*XX(J)*(1.-XX(J))	BDDAN	23
GO TO 90	BDDAN	24
30 CONTINUE	BDDAN	25
IF (FLAG.LE.0.) GO TO 40	BDDAN	26
CALL SECANT	BDDAN	27
GO TO 90	BDDAN	28
C	BDDAN	29
C PROJECTILE WITH TANGENT OGIVE,CYLINDER,BDATTAIL, CIRCULAR CAP	BDDAN	30
40 READ (5,100) X1,X2,X3,X4,RAD,THETA,CHORD	BDDAN	31
THETA=THETA*.0174533	BDDAN	32
DYDX=TAN(THETA)	BDDAN	33
BASE=RAD*(X4-X3)*DYDX	BDDAN	34
XE=BASE*(DYDX*SQRT(1.+DYDX**2))	BDDAN	35
X5=X4+XE	BDDAN	36
C NDIMENSIONAL OPTION	BDDAN	37
IF (CHORD.GE.0.) CHORD=X5-X1	BDDAN	38
CHORD=ABS(CHORD)	BDDAN	39
WRITE (6,120) X1,X2,X3,X4,RAD,THETA,CHORD	BDDAN	40
RCH=1./CHORD	BDDAN	41
X1=X1*RCH	BDDAN	42
X2=X2*RCH	BDDAN	43
X3=X3*RCH	BDDAN	44
X4=X4*RCH	BDDAN	45
X5=X5*RCH	BDDAN	46
BASE=BASE*RCH	BDDAN	47
XE=XE*RCH	BDDAN	48
RAD=RAD*RCH	BDDAN	49
DO 80 J=1,JBDD	BDDAN	50
IF (XX(J).GE.X2) GO TO 50	BDDAN	51
XDG=X2-X1	BDDAN	52
XBAR=(XX(J)-X1)/XDG	BDDAN	53
VLAM=XDG/RAD	BDDAN	54
VSQ=VLAM**2	BDDAN	55
RBAR=VSQ*.25	BDDAN	56
RADI=1.-VSQ*(1.-XBAR)**2/(RBAR**2)	BDDAN	57
YY(J)=(1.-2.*RBAR*(1.-SQRT(RADI)))*RAD	BDDAN	58
GO TO 80	BDDAN	59
50 IF (XX(J).GE.X3) GO TO 60	BDDAN	60
YY(J)=RAD	BDDAN	61
GO TO 80	BDDAN	62
60 IF (XX(J).GE.X4) GO TO 70	BDDAN	63
YY(J)=RAD*(XX(J)-X3)*DYDX	BDDAN	64
GO TO 80	BDDAN	65
70 RS=(BASE**2)*(1.+DYDX**2)	BDDAN	66
XBAR=(XX(J)-X4)	BDDAN	67
RADI=RS-(XBAR-BASE*DYDX)**2	BDDAN	68
YY(J)=SQRT(RADI)	BDDAN	69
80 CONTINUE	BDDAN	70
C END OF PROJECTILE	BDDAN	71
90 CONTINUE	BDDAN	72
RETURN	BDDAN	73
C	BDDAN	74
100 FORMAT (8F10.0)	BDDAN	75
110 FORMAT (1H0,8HTAU,FLAG,2F14.5)	BDDAN	76
120 FORMAT (1H0,29H X1,X2,X3,X4,RAD,THETA,CHORD ,/,7F14.5)	BDDAN	77
END	BDDAN	78

	SUBROUTINE BODIS (ISEGS, JWRIT)	BODIS	2
	COMMON /CALC/ XO, XATF, BS, ICT, FLAG, POYDX, FXF, RFM	BODIS	3
	COMMON JMAX, KMAX, JM, KM, NBDJ, J80D	BODIS	4
	COMMON /BOUDY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)	BODIS	5
	1 , T(100), TS(100)	BODIS	6
C		BODIS	7
C	5 DISTRIBUTION ON BODY	BODIS	8
C		BODIS	9
	00 10 I=1, ISEGS	BODIS	10
	READ (5,20) JI, JF, XI, XF, OXI, DXF	BODIS	11
	WRITE (6,30) JI, JF, XI, XF, OXI, DXF	BODIS	12
	CALL CLUST (JI, JF, XI, XF, OXI, DXF, S)	BODIS	13
10	CONTINUE	BODIS	14
	WRITE (6,40) (S(J), J=1, JWRIT)	BODIS	15
	RETURN	BODIS	16
C		BODIS	17
	20 FORMAT (2I5, 6F10.0)	BODIS	18
	30 FORMAT (1H0, 21H JI, JF, XI, XF, OXI, DXF , 2I5, 4F12.5)	BODIS	19
	40 FORMAT (1H , 10F11.5)	BODIS	20
	END	BODIS	21

SUBROUTINE BODY	800Y	2
COMMON JMAX, KMAX, JM, KM, NB00, JB00	800Y	3
COMMON /BOUDY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)	800Y	4
1 , Y(100), TS(100)	800Y	5
COMMON /ARRAY/ A(100), B(100), C(100), D(100), F(100), H(100)	800Y	6
COMMON /COMP/ X(100), Y(100)	800Y	7
C	800Y	8
READ (5,110) NB00,JB00,IXORS,ISEGS	800Y	9
WRITE (6,120) NB00,JB00,IXORS,ISEGS	800Y	10
C IF NB00 IS NEGATIVE, ANALYTIC SHAPE IS USED	800Y	11
IF (NB00.GE.0) GO TO 10	800Y	12
NB00=-NB00	800Y	13
CALL B00AM (ISEGS)	800Y	14
GO TO 90	800Y	15
C	800Y	16
10 CONTINUE	800Y	17
WRITE (6,130)	800Y	18
READ (5,150) CHORD	800Y	19
WRITE (6,140) CHORD	800Y	20
C READ CAROS IN CLOCKWISE NOSE TO TAIL	800Y	21
00 20 J=1,NB00	800Y	22
READ (5,150) X(J),Y(J)	800Y	23
WRITE (6,160) J,X(J),Y(J)	800Y	24
X(J)=X(J)/CHORD	800Y	25
Y(J)=Y(J)/CHORD	800Y	26
20 CONTINUE	800Y	27
C COMPUTE NORMALIZED ARC LENGTH TO USE AS A MONOTONE PARAMETER	800Y	28
SS(1)=0.	800Y	29
00 30 J=2,NB00	800Y	30
30 SS(J)=SS(J-1)+SQRT((X(J)-X(J-1))**2+(Y(J)-Y(J-1))**2)	800Y	31
00 40 J=2,NB00	800Y	32
40 SS(J)=SS(J)/SS(NB00)	800Y	33
C	800Y	34
C COMPUTE A NORMALIZED CLUSTERED PARAMETRIC FUNCTION FOR OISTRIBUTUON	800Y	35
C OF BODY POINTS	800Y	36
JWRIT=JB00	800Y	37
CALL B00IS (ISEGS,JWRIT)	800Y	38
SAV=S(1)	800Y	39
S(1)=0.	800Y	40
00 50 J=2,JB00	800Y	41
SOEL=ABS(S(J)-SAV)	800Y	42
SAV=S(J)	800Y	43
50 S(J)=S(J-1)+SOEL	800Y	44
00 60 J=1,JB00	800Y	45
60 S(J)=S(J)/S(JB00)	800Y	46
C OPTION WHETHER TO SET X EQUAL TO S OISTRIBUTUON	800Y	47
IF (IXORS.LT.0) GO TO 70	800Y	48
CALL CSPLIN (S,XX,SS,X,A,B,C,D,F,H,1,JB00,1,NB00)	800Y	49
CALL CSPLIN (S,YY,SS,Y,A,B,C,D,F,H,1,JB00,1,NB00)	800Y	50
GO TO 90	800Y	51
70 CONTINUE	800Y	52
00 80 J=1,JB00	800Y	53
80 XX(J)=X(1)+S(J)*(X(NB00)-X(1))	800Y	54
CALL CSPLIN (XX,YY,X,Y,A,B,C,D,F,H,1,JB00,1,NB00)	800Y	55
C	800Y	56
90 CONTINUE	800Y	57
WRITE (6,170)	800Y	58
00 100 J=1,JB00	800Y	59
WRITE (6,160) J,XX(J),YY(J)	800Y	60
100 CONTINUE	800Y	61
RETURN	800Y	62
C	800Y	63
110 FORMAT (4I5)	800Y	64
120 FORMAT (1H0,22HNB00,JB00,IXORS,ISEGS ,4I5)	800Y	65
130 FORMAT (1H0,38H X,Y INPUT DEFINING BODY... SUB. BODY)	800Y	66
140 FORMAT (1H0,32H NORMALIZING CHORD LENGTH INPUT ,F13.5)	800Y	67
150 FORMAT (2F10.0)	800Y	68
160 FORMAT (1H ,16H J,X,Y, ON BODY ,15,2F14.5)	800Y	69
170 FORMAT (1H1,32H J AND BODY OISTRIBUTED X AND Y)	800Y	70
END	800Y	--

SUBROUTINE GSPIN (I3D,ND,ISTOR,LMAX)	GSPIN	2
COMMON JMAX, KMAX, JM, KM, N8DD, J8DD	GSPIN	3
COMMON /GRIO/ X(80,60), Y(80,60)	GSPIN	4
COMMON /GRIO30/ X3(420,60), Y3(420,60), Z3(420,60)	GSPIN	5
LEVEL 2,X3,Y3,Z3	GSPIN	6
PI=4.*ARCTAN(1.)	GSPIN	7
DS=PI/(LMAX-3)	GSPIN	8
N1=KMAX*LMAX	GSPIN	9
DO 10 K=1,KMAX	GSPIN	10
OT=-2.*OS	GSPIN	11
DO 10 L=1,LMAX	GSPIN	12
OT=OT+DS	GSPIN	13
KL=(K-1)*ND+L	GSPIN	14
DO 10 J=1,JMAX	GSPIN	15
Y3(KL,J)=Y(J,K)*SIN(OT)	GSPIN	16
Z3(KL,J)=Y(J,K)*COS(OT)	GSPIN	17
X3(KL,J)=X(J,K)	GSPIN	18
10 CONTINUE	GSPIN	19
IF (ISTOR.E.0) GO TO 40	GSPIN	20
REWIND 9	GSPIN	21
WRITE (9) ((X3(KL,J),KL=1,N1),J=1,JMAX),((Y3(KL,J),KL=1,N1),J=1,	GSPIN	22
1 JMAX),((Z3(KL,J),KL=1,N1),J=1,JMAX)	GSPIN	23
C	GSPIN	24
C	GSPIN	25
C	GSPIN	26
REWRITE 30 DATA FOR 20 PLOTTING	GSPIN	27
DO 30 J=1,JMAX	GSPIN	28
DO 20 KL=2,N1,LMAX	GSPIN	29
N=1+(KL-2)/ND	GSPIN	30
X(J,N)=X3(KL,J)	GSPIN	31
Y(J,N)=Z3(KL,J)	GSPIN	32
20 CONTINUE	GSPIN	33
30 CONTINUE	GSPIN	34
WRITE (10) ((X(J,N),J=1,JMAX),N=1,KMAX),((Y(J,N),J=1,JMAX),N=1,	GSPIN	35
1 KMAX)	GSPIN	36
C	GSPIN	37
40 CONTINUE	GSPIN	38
WRITE (6,90)	GSPIN	39
DO 60 L=2,LMAX,3	GSPIN	40
DO 60 K=1,KMAX,4	GSPIN	41
KL=(K-1)*ND+L	GSPIN	42
WRITE (6,70) L,K,L,K+1	GSPIN	43
KL2=K*ND+L	GSPIN	44
DO 50 J=1,JMAX,2	GSPIN	45
50 WRITE (6,80) J,X3(KL,J),Y3(KL,J),Z3(KL,J),J,X3(KL2,J),Y3(KL2,J),	GSPIN	46
1 Z3(KL2,J)	GSPIN	47
60 CONTINUE	GSPIN	48
RETURN	GSPIN	49
70 FORMAT (1H3,5H J ,2HK=,I2,3H L=,I2,9X,1HX,10X,1HY,12X,1HZ,13X,	GSPIN	50
1 3HJ ,2HK=,I2,3H L=,I2,5X,1HX,10X,1HY,11X,1HZ)	GSPIN	51
80 FORMAT (1H ,I3,14X,F10.5,3X,F10.5,3X,F10.5,9X,I2,14X,F10.5,2X,F10.	GSPIN	52
15,2X,F10.5)	GSPIN	53
90 FORMAT (1H1)	GSPIN	54
END	GSPIN	55

	SUBROUTINE OUTER (NSEGS, IOJTO)	OUTER	2
	COMMON JMAX, KMAX, JM, KM, NBOJ, JBOJ	OUTER	3
	COMMON /BOUOY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)	OUTER	4
	1, T(100), TS(100)	OUTER	5
	COMMON /COMP/ X(100), Y(100)	OUTER	6
	COMMON /ARRAY/ A(100), B(100), C(100), D(100), F(100), H(100)	OUTER	7
		OUTER	8
C	THIS PROGRAM FORMS AN OUTER GRID BOUNDARY USING CONTIGUOUS CUBIC	OUTER	9
C	SEGMENTS. NUMBER OF SEGMENTS IS NSEGS. POINT AND SLOPE ARE INPUT AT	OUTER	10
C	THE ENDS OF A SEGMENT. SLOPE IS AN ANGLE IN DEGREES. PARAMETRIC	OUTER	11
C	CUBICS USED TO PERMIT ANY SLOPE (THETA = 90, -90, ETC). INITIAL	OUTER	12
C	LOGIC DETERMINES CUBIC COEFFICIENTS OF EACH SEGMENT. REMAINING	OUTER	13
C	LOGIC DISTRIBUTES POINTS ALONG OUTER BOUNDARY USING ARC LENGTH	OUTER	14
C	AS DISTRIBUTION FUNCTION. THUS TWO PARAMETRIC VARIABLES ARE USED.	OUTER	15
C	FINDING X,Y SO CUBIC SEGMENTS CAN BE DISJOINT IN SLOPE IS MESSY	OUTER	16
C	A SINGLE SPLINE INTERPOLATION CANNOT BE USED OVER THE COMBINED	OUTER	17
C	SEGMENTS BECAUSE OF POSSIBLE SLOPE DISCONTINUITY.	OUTER	18
C		OUTER	19
	DIMENSION JA(8), JB(8)	OUTER	20
	DIMENSION CAO(8), CA1(8), CA2(8), CBO(8), CB1(8), CB2(8), CARC(8)	OUTER	21
	DO 110 N=1, NSEGS	OUTER	22
C	POINTS AND SLOPES, -90 .LE. THETA .LE. 90 DEGREES USED	OUTER	23
	READ (5,240) XO,YO,X1,Y1,THO,TH1	OUTER	24
	WRITE (6,250) XO,YO,X1,Y1,THO,TH1	OUTER	25
	RTHO=0.017453292*THO	OUTER	26
	RTH1=0.017453292*TH1	OUTER	27
	XI=X1-XO	OUTER	28
	ETA=Y1-YO	OUTER	29
C		OUTER	30
C	SET MFLAG, LOGIC CHIEFLY USED TO AVOID INFINITE OY/OX, USES DX/OY=0	OUTER	31
	TA=SIN(RTHO)	OUTER	32
	TB=COS(RTHO)	OUTER	33
	TC=SIN(RTH1)	OUTER	34
	TD=COS(RTH1)	OUTER	35
	IF (ABS(TA).GT.ABS(TB)) GO TO 13	OUTER	36
	MFLAG=1	OUTER	37
	IF (ABS(TC).GT.ABS(TD)) MFLAG=2	OUTER	38
	GO TO 20	OUTER	39
10	MFLAG=3	OUTER	40
	IF (ABS(TC).GT.ABS(TD)) MFLAG=4	OUTER	41
20	CONTINUE	OUTER	42
C	DETERMINE COEFFICIENTS FOR PARAMETRIC CUBICS	OUTER	43
C	SET UP LINEAR COEFFS. FIRST, INDEFINITE CUBIC DEFAULTS TO LINEAR	OUTER	44
	A1=XI	OUTER	45
	A2=0.	OUTER	46
	B1=ETA	OUTER	47
	B2=0.	OUTER	48
C		OUTER	49
	GO TO (30,40,50,60), MFLAG	OUTER	50
30	OYOXO=TA/TB	OUTER	51
	OYOX1=TC/TD	OUTER	52
	TEST=OYOX1-OYOXO	OUTER	53
	IF (ABS(TEST).LT.0.0005) GO TO 80	OUTER	54
	SM1=2.*(ETA-OYOXO*X1)/TEST	OUTER	55
	SMO=2.*XI-SM1	OUTER	56
	SNO=SNO+OYOXO	OUTER	57
	GO TO 70	OUTER	58

40 OYOXO=TA/T8	OUTER	59
OXOY1=TO/TC	OUTER	60
TEST=1.-OYOXO*OXOY1	OUTER	61
IF (ABS(TEST).LT.0.0005) GO TO 80	OUTER	62
SNO=2.*(XI-OXOY1*ETA)/TEST	OUTER	63
SNO=SNO*OYOXO	OUTER	64
GO TO 70	OUTER	65
50 OXOYO=T8/TA	OUTER	66
OYOX1=TC/TO	OUTER	67
TEST=1.-OYOX1*OXOY0	OUTER	68
IF (ABS(TEST).LT.0.0005) GO TO 80	OUTER	69
SNO=2.*(ETA-OYOX1*XI)/TEST	OUTER	70
SNO=SNO*OXOY0	OUTER	71
GO TO 70	OUTER	72
60 OXOYO=T8/TA	OUTER	73
OXOY1=TO/TC	OUTER	74
TEST=OXOY1-OXOY0	OUTER	75
IF (ABS(TEST).LT.0.0005) GO TO 80	OUTER	76
SNI=2.*(XI-OXOY0*ETA)/TEST	OUTER	77
SNO=2.*ETA-SNI	OUTER	78
SNO=SNO*OXOY0	OUTER	79
C	OUTER	80
70 A1=SNO	OUTER	81
A2=XI-SNO	OUTER	82
B1=SNO	OUTER	83
B2=ETA-SNO	OUTER	84
80 CONTINUE	OUTER	85
JNBR=25	OUTER	86
C	OUTER	87
C COMPUTE NUMERICAL ARC LENGTH AS A PARAMETER(EXACT ARC LENGTH IS	OUTER	88
POSSIBLE BUT INVERSE PROCESS IS NOT	OUTER	89
00 90 J=1,JNBR	OUTER	90
TT=JNBR-1	OUTER	91
TT=(J-1)/TT	OUTER	92
X(J)=XO+TT*(A1+A2*TT)	OUTER	93
90 Y(J)=YO+TT*(B1+B2*TT)	OUTER	94
SARC=0.	OUTER	95
C	OUTER	96
C NOTE ...COULD USE SUM OF SQUARES AS PARAMETER RATHER THAN ARC LENGTH	OUTER	97
C IN THIS WAY ONE CAN AVOID SQUARE ROOT CALCULATION MUST USE EVERY	OUTER	98
C WHERE	OUTER	99
C	OUTER	100
00 100 J=2,JNBR	OUTER	101
100 SARC=SARC+SQRT((X(J)-X(J-1))**2+(Y(J)-Y(J-1))**2)	OUTER	102
C DATA FOR EACH CUBIC SEGMENT	OUTER	103
CA1(N)=A1	OUTER	104
CA2(N)=A2	OUTER	105
CB1(N)=B1	OUTER	106
CB2(N)=B2	OUTER	107
CAO(N)=XO	OUTER	108
CBO(N)=YO	OUTER	109
CARC(N)=SARC	OUTER	110
WRITE (6,260) XO,A1,A2,YO,B1,B2,SARC	OUTER	111
C	OUTER	112
110 CONTINUE	OUTER	113
C	OUTER	114
C CUBICS DETERMINED, NOW DISTRIBUTE POINTS	OUTER	115

C		OUTER	116
C	TOTAL OUTER ARC LENGTH	OUTER	117
	SARC=0.	OUTER	118
	DO 120 N=1,NSEGS	OUTER	119
120	SARC=SARC+CARC(N)	OUTER	120
	WRITE (6,270) SARC	OUTER	121
C		OUTER	122
C	DEFINE A UNIFORM OUTER DISTRIBUTION ARC LENGTH	OUTER	123
	RH=1./(JMAX-1)	OUTER	124
	DO 130 J=1,JMAX	OUTER	125
130	SS(J)=(J-1)*RH	OUTER	126
C		OUTER	127
C	OPTIONAL USE OF CUBIC SEGMENTS TO CLUSTER	OUTER	128
	IF (IOUTO.LE.0) GO TO 150	OUTER	129
	CALL BOOIS (IOUTO,JMAX)	OUTER	130
	DO 140 J=1,JMAX	OUTER	131
140	SS(J)=(S(J)-S(1))/(S(JMAX)-S(1))	OUTER	132
150	CONTINUE	OUTER	133
	WRITE (6,280) (SS(J),J=1,JMAX)	OUTER	134
C	NORMALIZE OUTER ARC LENGTH SEGMENTS TO SCALE OF DISTRIBUTION ARC LEN	OUTER	135
	CA=0.	OUTER	136
	DO 160 N=1,NSEGS	OUTER	137
	CA=CA+CARC(N)/SARC	OUTER	138
160	CARC(N)=CA	OUTER	139
	WRITE (6,290) (CARC(N),N=1,NSEGS)	OUTER	140
C	FIND J INDICES LIMITS WITHIN A SEGMENT	OUTER	141
	N=1	OUTER	142
	JA(N)=1	OUTER	143
	DO 180 J=2,JMAX	OUTER	144
	IF (SS(J).LE.CARC(N)) GO TO 170	OUTER	145
	N=N+1	OUTER	146
	JA(N)=J	OUTER	147
170	JB(N)=J	OUTER	148
180	CONTINUE	OUTER	149
	DO 190 N=1,NSEGS	OUTER	150
	WRITE (6,300) JA(N),JB(N)	OUTER	151
190	CONTINUE	OUTER	152
C		OUTER	153
C	FORM PARAMETRIC ARRAYS FROM DISTRIBUTED PARAMETRIC ARRAY,	OUTER	154
C	USE IT TO DETERMINE X,Y WITHIN A OUTER SEGMENT CURVE.	OUTER	155
C	SPLINE REQUIRES ABOUT 5 POINTS IN AN INTERVAL	OUTER	156
	S(1)=0.	OUTER	157
	RT=1./(JNBR-1)	OUTER	158
	DO 220 N=1,NSEGS	OUTER	159
	T(1)=0.	OUTER	160
	IF (N.GT.1) S(1)=CARC(N-1)	OUTER	161
	X(1)=CAO(N)	OUTER	162
	Y(1)=CBO(N)	OUTER	163
	DO 200 J=2,JNBR	OUTER	164
	TT=(J-1)*RT	OUTER	165
	T(J)=TT	OUTER	166
	X(J)=CAO(N)+TT*(CA1(N)+TT*CA2(N))	OUTER	167
	Y(J)=CBO(N)+TT*(CB1(N)+TT*CB2(N))	OUTER	168
	OS=SQRT((X(J)-X(J-1))**2+(Y(J)-Y(J-1))**2)	OUTER	169
	S(J)=S(J-1)+OS/SARC	OUTER	170
	WRITE (6,280) T(J),X(J),Y(J),S(J)	OUTER	171
200	CONTINUE	OUTER	172

C		OUTER	173
	J1=JA(N)	OUTER	174
	J2=JB(N)	OUTER	175
	CALL CSPLIN (SS,TS,S,T,A,B,C,D,F,H,J1,J2,1,JNBR)	OUTER	176
	DO 210 J=J1,J2	OUTER	177
	TT=TS(J)	OUTER	178
	XS(J)=CA0(N)+TT*(CA1(N)+TT*CA2(N))	OUTER	179
	YS(J)=CB0(N)+TT*(CB1(N)+TT*CB2(N))	OUTER	180
	WRITE (6,280) TS(J),XS(J),YS(J)	OUTER	181
210	CONTINUE	OUTER	182
220	CONTINUE	OUTER	183
	DO 230 J=1,JMAX	OUTER	184
	WRITE (6,310) XS(J),YS(J)	OUTER	185
230	CONTINUE	OUTER	186
	RETURN	OUTER	187
		OUTER	188
240	FORMAT (8F10.0)	OUTER	189
250	FORMAT (1H0,21H X0,Y0,X1,Y1,TH0,TH1 ,6F13.5)	OUTER	190
260	FORMAT (1H0,23HX0,A1,A2,Y0,B1,B2,SARC ,/,7F13.5)	OUTER	191
270	FORMAT (1H0,42H OUTER BOUNDARY NONDIMENSIONAL ARC LENGTH ,F14.6)	OUTER	192
280	FORMAT (1H ,8F12.5)	OUTER	193
290	FORMAT (1H0,6H CARC ,6F13.5)	OUTER	194
300	FORMAT (1H ,7H JA,JB ,2I5)	OUTER	195
310	FORMAT (1H ,6HXS,YS ,2F15.6)	OUTER	196
	END	OUTER	197

SUBROUTINE RELAX (ITERM,IPER,JI,JF,OMEGA)	RELAX	2
COMMON JMAX, KMAX, JM, KM, N800, J800	RELAX	3
COMMON /GRID/ X(80,60), Y(80,60)	RELAX	4
COMMON /ARRAY/ A(100), B(100), C(100), D(100), F(100), H(100)	RELAX	5
DIMENSION IP(142), IR(142)	RELAX	6
DIMENSION G(100)	RELAX	7
COMMON /SOURCE/ P(80,2), D(80,2), PFAC(2), OFAC(2)	RELAX	8
C	RELAX	9
SLDR SOLUTION OF ELLIPTIC GRID GENERATION EOS.	RELAX	10
C DEL XI AND DEL ETA = 1.0	RELAX	11
C	RELAX	12
J1=1	RELAX	13
J2=JMAX	RELAX	14
IF (IPER.GT.0) GO TO 10	RELAX	15
J1=J1+1	RELAX	16
J2=JF-1	RELAX	17
10 CONTINUE	RELAX	18
CALL INIPO	RELAX	19
C	RELAX	20
ITER=0	RELAX	21
KM=KMAX-1	RELAX	22
C SET PERIODIC INDICES	RELAX	23
DO 20 J=1,JMAX	RELAX	24
IP(J)=J+1	RELAX	25
20 IR(J)=J-1	RELAX	26
IP(JMAX)=1	RELAX	27
IR(1)=JMAX	RELAX	28
C	RELAX	29
C FORM DIFFERENCE EXPRESSIONS AND TRIDIAGONALS	RELAX	30
30 ITER=ITER+1	RELAX	31
RSUM=0.	RELAX	32
DO 160 KK=2,KM	RELAX	33
K=KM+2-KK	RELAX	34
KP=K+1	RELAX	35
KR=K-1	RELAX	36
CP1=EXP((1-K)*PFAC(1))	RELAX	37
CP2=EXP((K-KMAX)*PFAC(2))	RELAX	38
CQ1=EXP((1-K)*OFAC(1))	RELAX	39
CQ2=EXP((K-KMAX)*OFAC(2))	RELAX	40
DO 40 J=J1,J2	RELAX	41
JP=IP(J)	RELAX	42
JR=IR(J)	RELAX	43
XXD=(X(JP,K)-X(JR,K))*0.5	RELAX	44
XED=(X(J,KP)-X(J,KR))*0.5	RELAX	45
YXD=(Y(JP,K)-Y(JR,K))*0.5	RELAX	46
YED=(Y(J,KP)-Y(J,KR))*0.5	RELAX	47
AD=XED**2+YED**2	RELAX	48
BD=-2.*(XXD*XED+YXD*YED)	RELAX	49
GD=XXD**2+YXD**2	RELAX	50
XXED=-.25*(X(JP,KP)-X(JP,KR)-X(JR,KP)+X(JR,KR))	RELAX	51
YXED=-.25*(Y(JP,KP)-Y(JP,KR)-Y(JR,KP)+Y(JR,KR))	RELAX	52
A(J)=AD	RELAX	53
B(J)=-AD-AD-GD-GD	RELAX	54
C(J)=AD	RELAX	55
F(J)=-BD*XXED-GD*(X(J,KP)+X(J,KR))	RELAX	56
G(J)=-BD*YXED-GD*(Y(J,KP)+Y(J,KR))	RELAX	57
C SOURCE TERMS	RELAX	58

OJAC=(XXO*YEO-XEO*YXO)	RELAX	59
OSQ=OJAC**2	RELAX	60
COFA=OSQ*(CP1*P(J,1)+CP2*P(J,2))	RELAX	61
COFB=OSQ*(CQ1*Q(J,1)+CQ2*Q(J,2))	RELAX	62
F0BX=SIGN(.5,COFA)	RELAX	63
F0BE=SIGN(.5,COFB)	RELAX	64
A(J)=A(J)-COFA*(.5-F0BX)	RELAX	65
B(J)=B(J)-2.*(COFA*F0BX+COFB*F0BE)	RELAX	66
C(J)=C(J)+COFA*(.5+F0BX)	RELAX	67
F(J)=F(J)-COFB*(.5+F0BE)*X(J,XP)-(.5-F0BE)*X(J,KR)	RELAX	68
G(J)=G(J)-COFB*(.5+F0BE)*Y(J,XP)-(.5-F0BE)*Y(J,KR)	RELAX	69
40 CONTINUE	RELAX	70
C IF (IPER.GT.0) GO TO 130	RELAX	71
C SET B.C. AND INVERT	RELAX	72
C	RELAX	73
C XI MIN AND MAX PLANES MUST BE X OR Y CARTESIAN PLANES	RELAX	74
C	RELAX	75
C OUTFLOW B.C. ON X	RELAX	76
C TEST WHETHER XI PLANE IS X OR Y = CONSTANT PLANE	RELAX	77
C IF (ABS(Y(JF,KMAX)-Y(JF,1)).LT.0.001) GO TO 50	RELAX	78
C OUTFLOW XI-PLANE TAKEN AS X=CONSTANT PLANE	RELAX	79
A(JF)=0.	RELAX	80
B(JF)=-1.	RELAX	81
C(JF)=0.	RELAX	82
F(JF)=-X(JF,1)	RELAX	83
GO TO 60	RELAX	84
C OUTFLOW XI-PLANE TAKEN AS Y= CONSTANT	RELAX	85
50 A(JF)=1.	RELAX	86
B(JF)=-1.	RELAX	87
C(JF)=0.	RELAX	88
F(JF)=(-X(JF-1,K)+X(JF-2,K))/3.	RELAX	89
60 CONTINUE	RELAX	90
C INFLOW B.C. ON X	RELAX	91
C IF (ABS(Y(JI,KMAX)-Y(JI,1)).LT.0.001) GO TO 70	RELAX	92
C INFLOW XI-PLANE TAKEN AS X = CONSTANT PLANE	RELAX	93
A(JI)=0.	RELAX	94
B(JI)=-1.	RELAX	95
C(JI)=0.	RELAX	96
F(JI)=-X(JI,1)	RELAX	97
GO TO 80	RELAX	98
70 CONTINUE	RELAX	99
C INFLOW XI-PLANE TAKEN AS Y = CONSTANT PLANE	RELAX	100
A(JI)=0.	RELAX	101
B(JI)=-1.	RELAX	102
C(JI)=1.	RELAX	103
F(JI)=(-X(JI+1,K)+X(JI+2,K))/3.	RELAX	104
80 CALL TRIB (A,B,C,O,F,JI,JF)	RELAX	105
C	RELAX	106
C Y B.C. AND INVERSION	RELAX	107
C IF (ABS(Y(JF,KMAX)-Y(JF,1)).LT.0.001) GO TO 90	RELAX	108
C OUTFLOW XI-PLANE TAKEN AS X=CONSTANT PLANE	RELAX	109
A(JF)=1.	RELAX	110
B(JF)=-1.	RELAX	111
C(JF)=0.	RELAX	112
G(JF)=(-Y(JF-1,K)+Y(JF-2,K))/3.	RELAX	113
GO TO 100	RELAX	114
	RELAX	115

C	OUTFLOW XI-PLANE TAKEN AS Y = CONSTANT PLANE	RELAX	116
	90 A(JF)=0.	RELAX	117
	B(JF)=-1.	RELAX	118
	C(JF)=0.	RELAX	119
	G(JF)=-Y(JF,1)	RELAX	120
C	INFLOW	RELAX	121
	100 IF (ABS(Y(JI,KMAX)-Y(JI,1)).LT.0.001) GO TO 110	RELAX	122
C	INFLOW XI-PLANE TAKEN AS X = CONSTANT PLANE	RELAX	123
	A(JI)=0.	RELAX	124
	B(JI)=-1.	RELAX	125
	C(JI)=1.	RELAX	126
	G(JI)=-((Y(JI+1,K)-Y(JI+2,K))/3.	RELAX	127
	GO TO 120	RELAX	128
C	INFLOW XI-PLANE TAKEN AS Y = CONSTANT PLANE	RELAX	129
	110 A(JI)=0.	RELAX	130
	B(JI)=-1.	RELAX	131
	C(JI)=0.	RELAX	132
	G(JI)=-Y(JI,1)	RELAX	133
	120 CALL TR18 (A,B,C,D,G,J1,JF)	RELAX	134
	GO TO 140	RELAX	135
C		RELAX	136
	130 CONTINUE	RELAX	137
C		RELAX	138
C	PERIODIC B.C.	RELAX	139
	CALL TRIP (A,B,C,F,D,M,1,JMAX)	RELAX	140
	CALL TRIP (A,B,C,G,D,M,1,JMAX)	RELAX	141
	140 CONTINUE	RELAX	142
C		RELAX	143
C	RELAXATION UPDATE	RELAX	144
	DO 150 J=J1,JF	RELAX	145
	YC=G(J)-Y(J,X)	RELAX	146
	XC=F(J)-X(J,K)	RELAX	147
	X(J,K)=X(J,X)+OMEGA*XC	RELAX	148
	Y(J,K)=Y(J,K)+OMEGA*YC	RELAX	149
	150 RSUM=RSUM+ABS(XC)+ABS(YC)	RELAX	150
	160 CONTINUE	RELAX	151
C		RELAX	152
	IF ((ITER/10)*10.LT.ITER) GO TO 170	RELAX	153
	WRITE (6,180) ITER,RSUM	RELAX	154
	170 CONTINUE	RELAX	155
	IF (ITER.LT.ITERM) GO TO 30	RELAX	156
	RETURN	RELAX	157
C		RELAX	158
	180 FORMAT (1H ,26H ITERATION NBR AND RSUM ,15,E12.5)	RELAX	159
	END	RELAX	160

	SUBROUTINE INIPQ	RELAX	161
	COMMON JMAX, KMAX, JM, KM, N800, J800	RELAX	162
	COMMON /SOURCE/ P(80,2), Q(80,2), PFAC(2), QFAC(2)	RELAX	163
	DIMENSION PC(2), QC(2)	RELAX	164
C	READ (5,20) PFAC(1),QFAC(1),PFAC(2),QFAC(2)	RELAX	165
	READ (5,20) PC(1),QC(1),PC(2),QC(2)	RELAX	166
	WRITE (6,30) PFAC(1),QFAC(1),PFAC(2),QFAC(2)	RELAX	167
	WRITE (6,40) PC(1),QC(1),PC(2),QC(2)	RELAX	168
C	DO 10 N=1,2	RELAX	169
	DO 10 J=1,JMAX	RELAX	170
	P(J,N)=PC(N)	RELAX	171
10	Q(J,N)=QC(N)	RELAX	172
	RETURN	RELAX	173
C	20 FORMAT (8F10.0)	RELAX	174
	30 FORMAT (1H0,' EXPONENT COEFFICIENTS FOR SOURCE TERMS,PFAC,QFAC	RELAX	175
	1 ',/,4F13.5)	RELAX	176
	40 FORMAT (1H0,18H PC1,QC1,PC2,QC2 ,4F13.5)	RELAX	177
	END	RELAX	178
C		RELAX	179
C	SUBROUTINE CLUST (JI,JF,XI,XF,DXI,DXF,S)	RELAX	180
	DIMENSION S(1)	RELAX	181
C	XF XI=XF-XI	RELAX	182
	H=1./((JF-JI)	RELAX	183
	H2=H*H	WORKS	2
	H3=H2*H	WORKS	3
	C=(DXF+OXI-2.*H*XF XI)/(H-3.*H2+2.*H3)	WORKS	4
	B=(OXI-H*XF XI-C*(H3-H))/(H2-H)	WORKS	5
	A=XF XI-B-C	WORKS	6
C	DO 10 J=JI,JF	WORKS	7
	X=(J-JI)*H	WORKS	8
10	S(J)=XI+X*(A+X*(B+C*X))	WORKS	9
	RETURN	WORKS	10
	END	WORKS	11
		WORKS	12
		WORKS	13
		WORKS	14
		WORKS	15
		WORKS	16
		WORKS	17

C	FUNCTION EPSIL (FMX,FMIN,OFM,NPT,FPCC,ICC,NCALL)	WORKS	18
C		WORKS	19
C	THIS SUBROUTINE APPLIES A NEWTON-RAPHSON ROOT-FINDING	WORKS	20
C	TECHNIQUE TO FIND A VALUE OF EPSILON FOR A PARTICULAR USE	WORKS	21
C	OF THE EXPONENTIAL STRECHING TRANSFORMATION.	WORKS	22
C		WORKS	23
C	FMX IS TOTAL ARC LENGTH ALONG COORDINATE	WORKS	24
C	FMIN IS STARTING VALUE OF ARC LENGTH SUCH AS 0.0	WORKS	25
C	OFM IS SPECIFIED INITIAL INCREMENT OF ARC LENGTH	WORKS	26
C	NPT IS NUMBER OF POINTS ALONG COORDINATE	WORKS	27
C	FPCC IS ITERATIVE ERROR BOUND, E.G.01 0.000021	WORKS	28
C	ICC IS MAXIMUM NUMBER OF ITERATIONS	WORKS	29
C	NCALL IF NCALL=1 INITIAL GUESS FOR EPS IS USED	WORKS	30
C	IF NCALL .GT. 1, PREVIOUS EPS USED AS INITIAL GUESS	WORKS	31
C		WORKS	32
C		WORKS	33
C		WORKS	34
C	FMXL=FMX	WORKS	35
C	FMINL=FMIN	WORKS	36
C	OFML=OFM	WORKS	37
C	FPCL=FPCC	WORKS	38
C	ICCL=ICC	WORKS	39
C		WORKS	40
C		WORKS	41
C	FNPTM2=NPT-2	WORKS	42
C	IF (NCALL.EQ.1) EPS=(FMXL/OFML)**(1.0/FNPTM2)-1.0	WORKS	43
C		WORKS	44
C	00 10 NIT=1,ICCL	WORKS	45
C	EP1=EPS+1.0	WORKS	46
C	EP1TN=EP1*FNPTM2	WORKS	47
C	REPS=1.0/EP1	WORKS	48
C	DFMOE=OFML*REPS	WORKS	49
C	F=FMXL-FMINL-OFMOE*(EP1TN*EP1-1.0)	WORKS	50
C	IF (ABS(F).LT.FPCL) GO TO 20	WORKS	51
C	OFMOE2=OFMOE*REPS	WORKS	52
C	FPN=OFMOE2*(1.0+EP1TN*(EPS*FNPTM2-1.0))	WORKS	53
C	EPS=EPS+F/FPN	WORKS	54
C	10 CONTINUE	WORKS	55
C		WORKS	56
C	EPSIL=EPS	WORKS	57
C	WRITE (6,30)	WORKS	58
C	RETURN	WORKS	59
C		WORKS	60
C	20 EPSIL=EPS	WORKS	61
C	SUPPRESS THE EPSIL PRINTOUT	WORKS	62
C	WRITE(6,601) EPSIL,F,NIT	WORKS	63
C		WORKS	64
C	RETURN	WORKS	65
C		WORKS	66
C		WORKS	67
C	30 FORMAT (/42H EXCEEDED MAX. NO. OF ITERATIONS IN EPSIL.)	WORKS	68
C	END	WORKS	69

C	SUBROUTINE CSPLIN (XX,YY,X,Y,A,8,C,O,F,H,N1,N2,J1,J2)	WORKS	70
	DIMENSION XX(1), YY(1), X(1), Y(1), A(1), B(1), C(1), D(1), F(1),	WORKS	71
	1 4(1)	WORKS	72
C	CUBIC SPLINE INTERPOLATION	WORKS	73
C	X,Y ARRAYS ARE TO BE INTERPOLATED	WORKS	74
C	YY ARE FOUND INTERPOLATES CORRESPONDING TO XX	WORKS	75
C	J1,J2, ARE INVOICE LIMITS ON X,Y	WORKS	76
C	N1, N2 ARE INVOICE LIMITS ON XX (ALSO YY)	WORKS	77
C	DIMENSION OF ARRAYS CARRIED IN FROM OUTSIDE, X(J) MUST BE MONOTONE	WORKS	78
C	FORMULA FROM NUMERICAL METHODS BY JAMLIQUIST, BJORCK, ANDERSON	WORKS	79
C	JLS FEB. 77	WORKS	80
C	ROUNDING ERROR PROTECTION CAUTION MAY MASK ERROR IN LOGIC	WORKS	81
C	IF (XX(N1).LT.X(J1)) XX(N1)=X(J1)	WORKS	82
C	IF (XX(N2).GT.X(J2)) XX(N2)=X(J2)	WORKS	83
C	FIRST FIND DERIVATIVE LIKE TERMS THAT ARE COEFFICIENTS	WORKS	84
C	JA=J1+1	WORKS	85
	J8=J2-1	WORKS	86
	DO 10 J=JA,J2	WORKS	87
	H(J)=X(J)-X(J-1)	WORKS	88
10	O(J)=(Y(J)-Y(J-1))/H(J)	WORKS	89
	DO 20 J=JA,J8	WORKS	90
	A(J)=H(J+1)	WORKS	91
	B(J)=2.*(H(J)+H(J+1))	WORKS	92
20	F(J)=3.*(H(J)*O(J+1)+H(J+1)*O(J))	WORKS	93
	B(J1)=2.	WORKS	94
	H(J1)=1.	WORKS	95
	F(J1)=3.*O(JA)	WORKS	96
	A(J2)=1.	WORKS	97
	B(J2)=2.	WORKS	98
	F(J2)=3.*O(J2)	WORKS	99
	CALL TRI8 (A,8,H,C,F,J1,J2)	WORKS	100
C	INTERPOLATION , X(J) ARRAY MUST BE MONOTONE	WORKS	101
	J=J1	WORKS	102
	I=J1+1	WORKS	103
	DO 80 N=N1,N2	WORKS	104
30	IF (X(J).LE.XX(N).AND.X(I).GE.XX(N)) GO TO 70	WORKS	105
	IF (X(I)-XX(N)) 40,40,50	WORKS	106
40	J=J+1	WORKS	107
	I=I+1	WORKS	108
	IF (I.GT.J2) GO TO 60	WORKS	109
	GO TO 30	WORKS	110
50	J=J-1	WORKS	111
	I=I-1	WORKS	112
	IF (J.LT.J1) GO TO 60	WORKS	113
	GO TO 30	WORKS	114
60	WRITE (6,90)	WORKS	115
	STOP	WORKS	116
70	T=(XX(N)-X(J))/H(I)	WORKS	117
	TT=1.-T	WORKS	118
	YY(N)=T*Y(I)+TT*Y(J)+H(I)*T*TT*((F(J)-O(I))*TT-(F(I)-O(I))*T)	WORKS	119
80	CONTINUE	WORKS	120
	RETURN	WORKS	121
C	90 FORMAT (1H0,20H ERROR IN CSPLIN)	WORKS	122
	END	WORKS	123
		WORKS	124
		WORKS	125
		WORKS	126
		WORKS	127
		WORKS	128
		WORKS	129
		WORKS	130

C	SUBROUTINE TRIB (A,B,C,X,F,NL,NU)	WORKS	131
	DIMENSION A(2), B(2), C(2), X(2), F(2)	WORKS	132
	X(NL)=C(NL)/B(NL)	WORKS	133
	F(NL)=F(NL)/B(NL)	WORKS	134
	NLP1=NL+1	WORKS	135
	DO 10 J=NLP1,NU	WORKS	136
	Z=1./(B(J)-A(J)*X(J-1))	WORKS	137
	X(J)=C(J)*Z	WORKS	138
10	F(J)=(F(J)-A(J)*F(J-1))*Z	WORKS	139
	NUPNL=NU+NL	WORKS	140
	DO 20 J1=NLP1,NU	WORKS	141
	J=NUPNL-J1	WORKS	142
20	F(J)=F(J)-X(J)*F(J+1)	WORKS	143
	RETURN	WORKS	144
	END	WORKS	145
C	SUBROUTINE TRIP (A,B,C,F,Q,S,J1,J2)	WORKS	146
	DIMENSION A(3), B(3), C(3), F(3), Q(3), S(3)	WORKS	147
	JA=J1+1	WORKS	148
	FN=F(J2)	WORKS	149
C	FORWARD ELIMINATION SWEEP	WORKS	150
	Q(J1)=-C(J1)/B(J1)	WORKS	151
	F(J1)=F(J1)/B(J1)	WORKS	152
	S(J1)=-A(J1)/B(J1)	WORKS	153
	DO 10 J=JA,J2	WORKS	154
	P=1./(B(J)+A(J)*Q(J-1))	WORKS	155
	Q(J)=-C(J)*P	WORKS	156
	F(J)=(F(J)-A(J)*F(J-1))*P	WORKS	157
	S(J)=-A(J)*S(J-1)*P	WORKS	158
10	CONTINUE	WORKS	159
C	BACKWARD PASS	WORKS	160
	JJ=J1+J2	WORKS	161
	Q(J2)=0.	WORKS	162
	S(J2)=1.	WORKS	163
	DO 20 I=JA,J2	WORKS	164
	J=JJ-I	WORKS	165
	S(J)=S(J)+C(J)*S(J+1)	WORKS	166
20	Q(J)=F(J)+Q(J)*Q(J+1)	WORKS	167
	F(J2)=(FN-C(J2)*Q(J1)-A(J2)*Q(J2-1))/(C(J2)*S(J1)+A(J2)*S(J2-1)	WORKS	168
	+B(J2))	WORKS	169
C	BACKWARD ELIMINATION PASS	WORKS	170
	DO 30 I=JA,J2	WORKS	171
	J=JJ-I	WORKS	172
30	F(J)=F(J2)*S(J)+Q(J)	WORKS	173
	RETURN	WORKS	174
	END	WORKS	175
		WORKS	176
		WORKS	177

	SUBROUTINE SCALC	SCALC	2
	COMMON /CALC/ X0, XATF, BS, ICT, FLAG, POYDX, FXF, RFN	SCALC	3
	COMMON /INPS/ X1, X2, X3, X4, RAD, DYDX, CHORD, FUSE, AS, RADS	SCALC	4
C		SCALC	5
C	THIS SUBRTN CALCULATES VALUES NEEDED IN THE	SCALC	6
C	EQUATIONS SOLVING FOR Y VALUES IN SUBRTN BODAN	SCALC	7
C		SCALC	8
	READ (5,10) X1,X2,X3,X4,RAD,THETA,CHORD	SCALC	9
	READ (5,10) RADS,FJSE,AS	SCALC	10
	THETA=THETA*.0174533	SCALC	11
	DYDX=TAN(THETA)	SCALC	12
	CHORD=ABS(CHORD)	SCALC	13
	FUSE=FUSE/2	SCALC	14
C		SCALC	15
C	TO FIND THE Y VALUE (BS) OF CIRCLE USED IN	SCALC	16
C	THE SECANT GIVE CALCULATIONS	SCALC	17
C		SCALC	18
	XBAR=AS-X2	SCALC	19
	YS=RADS**2-XBAR**2	SCALC	20
	YSS=SQRT(YS)	SCALC	21
	BS=RAD-YSS	SCALC	22
C		SCALC	23
C	TO FIND THE X VALUE (XATF) AT THE FUSE	SCALC	24
C		SCALC	25
	YR=ABS(BS)+FUSE	SCALC	26
	YRR=RADS**2-YR**2	SCALC	27
	XSS=SQRT(YRR)	SCALC	28
	XATF=AS-XSS	SCALC	29
C		SCALC	30
C	TO FIND THE SLOPE	SCALC	31
C		SCALC	32
	POYDX=(XATF-AS)/(FUSE-BS)	SCALC	33
	POYDX=-POYDX	SCALC	34
C		SCALC	35
C	TO FIND THE X VALUE (X0) AT THE NOSECAP	SCALC	36
C		SCALC	37
	XSQ=SQRT(1.+POYDX)	SCALC	38
	RFN=FUSE*XSQ	SCALC	39
	FXF2=RFN**2-FUSE**2	SCALC	40
	FXF=SQRT(FXF2)	SCALC	41
	XS=RFN-FXF	SCALC	42
	X0=XATF-XS	SCALC	43
	WRITE (6,20) X1,X2,X3,X4,RAD,THETA,CHORD	SCALC	44
	WRITE (6,30) RADS,FUSE,AS,BS,X0,XATF	SCALC	45
	RETURN	SCALC	46
C		SCALC	47
	10 FORMAT (8F10.0)	SCALC	48
	20 FORMAT (1H0,27HX1,X2,X3,X4,RAD,THETA,CHORD,/,8F14.5)	SCALC	49
	30 FORMAT (1H0,23HRADS,FUSE,AS,BS,X0,XATF,/,6F14.5)	SCALC	50
	END	SCALC	51

SUBROUTINE SECANT	SECANT	2
COMMON /CALC/ XD, XATF, BS, ICT, FLAG, POYOX, FXF, RFN	SECANT	3
COMMON /INPS/ X1, X2, X3, X4, RAD, OYOX, CHORD, FUSE, AS, RAOS	SECANT	4
COMMON JMAX, KMAX, JM, KM, NBDD, JBDD	SECANT	5
COMMON /BOUDY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)	SECANT	6
1, T(100), TS(100)	SECANT	7
C PROJECTILE WITH NOSECAP, SECANT DGIVE, CYLINDER, BOATTAIL	SECANT	8
C	SECANT	9
RCH=1./CHORD	SECANT	10
X1=X1*RCH	SECANT	11
X2=X2*RCH	SECANT	12
X3=X3*RCH	SECANT	13
X4=X4*RCH	SECANT	14
RAD=RAD*RCH	SECANT	15
FUSE=FUSE*RCH	SECANT	16
RAOS=RAOS*RCH	SECANT	17
XATF=XATF*RCH	SECANT	18
AS=AS*RCH	SECANT	19
BS=BS*RCH	SECANT	20
DD 40 J=1, JBDD	SECANT	21
IF (XX(J).GE.XATF) GO TO 10	SECANT	22
C	SECANT	23
C COMPUTE Y VALUES FOR NOSECAP	SECANT	24
C	SECANT	25
XBAR=XX(J)-(XATF+FXF)	SECANT	26
RADI=RFN**2-XBAR**2	SECANT	27
YY(J)=SQRT(RADI)	SECANT	28
GO TO 40	SECANT	29
10 IF (XX(J).GE.X2) GO TO 20	SECANT	30
C	SECANT	31
C COMPUTE Y VALUES FOR DGIVE	SECANT	32
C	SECANT	33
XBAR=XX(J)-AS	SECANT	34
RADIS=RAOS**2-XBAR**2	SECANT	35
YY(J)=BS+SQRT(RADIS)	SECANT	36
GO TO 40	SECANT	37
20 IF (XX(J).GE.X3) GO TO 30	SECANT	38
C	SECANT	39
C COMPUTE Y VALUES FOR CYLINDER	SECANT	40
C	SECANT	41
YY(J)=RAD	SECANT	42
GO TO 40	SECANT	43
C	SECANT	44
C COMPUTE Y VALUES FOR BOATTAIL	SECANT	45
C	SECANT	46
30 YY(J)=RAD+(XX(J)-X3)*OYOX	SECANT	47
40 CONTINUE	SECANT	48
WRITE (6,50) POYOX, RFN, FXF	SECANT	49
RETURN	SECANT	50
C	SECANT	51
50 FORMAT (1H0,13HPOYOX,RFN,FXF,5X,3F14.5)	SECANT	52
END	SECANT	53
	SECANT	54

	SUBROUTINE STING (NCGRD)	STING	2
	COMMON JMAX, KMAX, JM, KM, N800, JB00	STING	3
	COMMON /80UQY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)	STING	4
	1 , T(100), TS(100)	STING	5
C	STING ASSUMED ALIGNED WITH X COORDINATE	STING	6
C	STING INPUT AS A SEPARATE BODY, J=1,NCGRD...800IS DATA STARTS AT 1	STING	7
	READ (5,20) ISEGS	SYING	9
	WRITE (6,30) ISEGS	STING	10
	READ (5,40) XMIN,YMIN,XMAX,YMAX	STING	11
	WRITE (6,50) XMIN,YMIN,XMAX,YMAX	STING	12
	CALL 800IS (ISEGS,NCGRD)	STING	13
	DO 10 JJ=2,NCGRD	STING	14
	J=JB00+JJ-1	STING	15
	S(JJ)=(S(JJ)-S(1))/(S(NCGRD)-S(1))	STING	16
	XX(J)=XMIN+S(JJ)*(XMAX-XMIN)	STING	17
10	YY(J)=YMIN+S(JJ)*(YMAX-YMIN)	STING	18
	JB00=JB00+NCGRD-1	STING	19
	WRITE (6,60) (XX(J),J=1,JB00)	STING	20
	RETURN	STING	21
C	20 FORMAT (I5)	STING	22
	30 FORMAT (1H0,35H STING PART OF PROGRAM.....ISEGS =,I5)	STING	23
	40 FORMAT (8F10.0)	STING	24
	50 FORMAT (1H0,21H XMIN,YMIN,XMAX,YMAX ,4F13.5)	STING	25
	60 FORMAT (1H ,8F12.5)	STING	26
	ENO	STING	27
C	SUBROUTINE CARTB (NCART)	STING	28
	COMMON JMAX, KMAX, JM, KM, N800, JB00	STING	29
	COMMON /80JOY/ XX(100), YY(100), XS(100), YS(100), SS(100), S(100)	STING	30
	1 , T(100), TS(100)	STING	31
C	READ (5,30) ISEGS	STING	32
	WRITE (6,40) ISEGS	STING	33
	READ (5,50) XMIN,YMIN,XMAX,YMAX	STING	34
	WRITE (6,60) XMIN,YMIN,XMAX,YMAX	STING	35
C	SHIFT POINTS	STING	36
	DO 10 JS=1,JB00	STING	37
	J=JB00+NCART-JS	STING	38
	JJ=JB00+1-JS	STING	39
	XX(J)=XX(JJ)	STING	40
10	YY(J)=YY(JJ)	STING	41
	JB00=JB00+NCART-1	STING	42
	CALL 800IS (ISEGS,NCART)	STING	43
	NCM=NCART-1	STING	44
	DO 20 J=1,NCM	STING	45
	S(J)=(S(J)-S(1))/(S(NCART)-S(1))	STING	46
	XX(J)=XMIN+S(J)*(XMAX-XMIN)	STING	47
20	YY(J)=YMIN+S(J)*(YMAX-YMIN)	STING	48
	RETURN	STING	49
C	30 FORMAT (I5)	STING	50
	40 FORMAT (1H0,33H CARTB PART OF PROGRAM.....ISEGS=,I5)	STING	51
	50 FORMAT (8F10.0)	STING	52
	60 FORMAT (1H0,21H XMIN,YMIN,XMAX,YMAX ,4F13.5)	STING	53
	ENO	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 *3I5*

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 *4I5*

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.

\$ Read cards 3, 4, 5 iff NBOD .GE.0\$
3. CHORD *F10.0*

All x,y data is normalized (i.e., divided by) CHORD. CHORD may be set to 1.
4. X(J), Y(J) *2F10.0*

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 *2I5,4F10.0*

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

$$JI = j_0$$

$$JF = j_f$$

$$XI = x_0$$

$$XF = x_f$$

$$DXI = \Delta x_0$$

$$DXF = \nabla x_f$$

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

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

6. TAU, FLAG *2F10.0*

TAU = parabolic arc thickness ratio

FLAG = 0 tangent-ogive cylinder, boattail projectile read in
1 secant-ogive cylinder, boattail projectile with nose cap read in

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

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

see card 5

\$ iff FLAG .GE.0\$

8a. X1, X2, X3, X4, RAD, THETA, CHORD *6F10.0*

X1 = value of x at nose

X2 = value of x at ogive-cylinder juncture

X3 = value of x at cylinder-boattail juncture

X4 = value of x at boattail base

RAD = radius of cylinder

THETA = angle of degrees that boattail makes with cylinder
(THETA is negative)

CHORD = if CHORD .GE.0, body length normalized to one.

Note: a spherical cap is added to boattail so the body length is not X4-X1.

- 8b. RADS, FUSE, AS *3F10.0*
- \$ iff FLAG .EQ.1\$
- RADS = radius of secant
FUSE = fuse height (at nosecap)
AS = value of x at secant-origin
9. NFLAG *I5*
- Option to exit program after body clustering data is printed out.
- NFLAG .LT.0, STOP
10. NCGRD, NCART *2I5*
- Parameters that allow addition of sting/rear cut and a front cut.
- NCGRD .GT.0, NCGRD points added for rear cut or sting
- NCART .GT.0, NCART points added for front cut (or lower cut of C-grid)
- \$ Read cards 11, 12, 13 iff NCGRD .GT.0\$
11. ISEGS *I5*
- ISEGS of sting
12. XMIN, YMIN, XMAX, YMAX *4F10.0*
- XMIN = initial x value of sting
- YMIN = initial y value of sting
- XMAX = final x value of sting
- YMAX = YMIN
13. JI, JF, XI, XF, DXI, DXF *2I5,4F10.0*
- see card 5
- \$Read cards 14, 15, 16 iff NCART .GT.0\$
14. ISEGS *I5*
15. XMIN, YMIN, XMAX, YMAX *4F10.0*
16. JI, JF, XI, XF, DXI, DXF *2I5,4F10.0*
- 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. X0, Y0, 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, ISTORE, JELLI *6I5*

KMAX = number of points in η -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 ISTORE .GT.0

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

21. DS, OMEGA *2F10.0*

DS = Δs_0 (i.e., Δs in η direction at $\eta = 0$ boundary). See Eq. (8). Note Δs_0 used along entire $\eta = 0$ boundary.

OMEGA = relaxation factor for SLOR in Subroutine RELAX. Typical safe value is 1.55. $0 < \text{OMEGA} < 2.0$.

22. JI, JF

2I5

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

23. BLANK CARD

24. BLANK CARD

$\bar{P}$ and $\bar{Q}$ 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.

130.N0.LMAX 1 10 10

***** INNER ROUNDOFF *****

NR00.JR00.IX0R5.I5F6S 57 40 1

X.Y INPUT OFFINING, H00Y... SUB, R00Y

NORMALIZING CHORD LENGTH INPUT		2.25000
J.X.Y. ON R00Y	1	0.00000
J.X.Y. ON R00Y	2	-0.00628
J.X.Y. ON R00Y	3	-0.00789
J.X.Y. ON R00Y	4	-0.01674
J.X.Y. ON R00Y	5	-0.02995
J.X.Y. ON R00Y	6	-0.05652
J.X.Y. ON R00Y	7	-0.08261
J.X.Y. ON R00Y	8	-0.10838
J.X.Y. ON R00Y	9	-0.13302
J.X.Y. ON R00Y	10	-0.15621
J.X.Y. ON R00Y	11	-0.17650
J.X.Y. ON R00Y	12	-0.18842
J.X.Y. ON R00Y	13	-0.19116
J.X.Y. ON R00Y	14	-0.19164
J.X.Y. ON R00Y	15	-0.19144
J.X.Y. ON R00Y	16	-0.19116
J.X.Y. ON R00Y	17	-0.19068
J.X.Y. ON R00Y	18	-0.18890
J.X.Y. ON R00Y	19	-0.18601
J.X.Y. ON R00Y	20	-0.18053
J.X.Y. ON R00Y	21	-0.16684
J.X.Y. ON R00Y	22	-0.14123
J.X.Y. ON R00Y	23	-0.12110
J.X.Y. ON R00Y	24	-0.09598
J.X.Y. ON R00Y	25	-0.07553
J.X.Y. ON R00Y	26	-0.04663
J.X.Y. ON R00Y	27	-0.02866
J.X.Y. ON R00Y	28	-0.0161
J.X.Y. ON R00Y	29	-0.00000
J.X.Y. ON R00Y	30	0.0161
J.X.Y. ON R00Y	31	0.02415
J.X.Y. ON R00Y	32	0.0441
J.X.Y. ON R00Y	33	0.12078
J.X.Y. ON R00Y	34	0.16104
J.X.Y. ON R00Y	35	0.24157
J.X.Y. ON R00Y	36	0.32200
J.X.Y. ON R00Y	37	0.40993
J.X.Y. ON R00Y	38	0.50366
J.X.Y. ON R00Y	39	0.6419
J.X.Y. ON R00Y	40	0.72471
J.X.Y. ON R00Y	41	0.79879
J.X.Y. ON R00Y	42	0.8523
J.X.Y. ON R00Y	43	0.89355
J.X.Y. ON R00Y	44	0.92665
J.X.Y. ON R00Y	45	0.94574

J.X.Y. ON BODY	46	.96628	.42114			
J.X.Y. ON BODY	47	1.12734	.40084			
J.X.Y. ON BODY	48	1.28838	.36886			
J.X.Y. ON BODY	49	1.44943	.32290			
J.X.Y. ON BODY	50	1.61048	.26911			
J.X.Y. ON BODY	51	1.77153	.20904			
J.X.Y. ON BODY	52	1.93257	.14164			
J.X.Y. ON BODY	53	2.09362	.07456			
J.X.Y. ON BODY	54	2.17145	.04074			
J.X.Y. ON BODY	57	2.25000	0.00000			
J1.JF.X1.XF.OX1.OXF	1	13	1.00000	.40000	-.02000	-.06000
J1.JF.X1.XF.OX1.OXF	13	21	.40000	0.00000	-.06000	-.01500
J1.JF.X1.XF.OX1.OXF	21	28	0.00000	.40000	.01500	.06000
J1.JF.X1.XF.OX1.OXF	28	40	.40000	1.00000	.06000	.02000
1.00000	.98000	.95091	.91382	.86982	.82000	.76545
.52182	.40000	.40000	.34000	.27571	.21071	.14857
0.00000	.01500	.05714	.11857	.19143	.26786	.34000
.58436	.64455	.70727	.75545	.82000	.86982	.91382
						.95091
						.64655
						.04714
						.46000
						.52192
						1.00000

J AND BODY DISTRIBUTED X AND Y

J.X.Y. ON BODY	1	1.00000	0.00000			
J.X.Y. ON BODY	2	.97991	-.00454			
J.X.Y. ON BODY	3	.95017	-.01039			
J.X.Y. ON BODY	4	.91023	-.01646			
J.X.Y. ON BODY	5	.86370	-.02431			
J.X.Y. ON BODY	6	.81084	-.03296			
J.X.Y. ON BODY	7	.75284	-.04226			
J.X.Y. ON BODY	8	.69101	-.05203			
J.X.Y. ON BODY	9	.62637	-.06175			
J.X.Y. ON BODY	10	.56012	-.07115			
J.X.Y. ON BODY	11	.49330	-.07923			
J.X.Y. ON BODY	12	.42693	-.08304			
J.X.Y. ON BODY	13	.36237	-.08516			
J.X.Y. ON BODY	14	.29783	-.08324			
J.X.Y. ON BODY	15	.22886	-.07804			
J.X.Y. ON BODY	16	.15986	-.06554			
J.X.Y. ON BODY	17	.09491	-.05050			
J.X.Y. ON BODY	18	.04103	-.02484			
J.X.Y. ON BODY	19	.01038	.01330			
J.X.Y. ON BODY	20	.00002	.04566			
J.X.Y. ON BODY	21	.00188	.06164			
J.X.Y. ON BODY	22	.00775	.07678			
J.X.Y. ON BODY	23	.03682	.11159			
J.X.Y. ON BODY	24	.09222	.14710			
J.X.Y. ON BODY	25	.16595	.17365			

NC390.NCART 11 11

ENDING PART OF PROGRAM.....ISEGS = 1

[illegible]

CARRY PART OF PROGRAM.....1SEGS= 1

XMIN, YMIN, XMAX, YMAX	R.00000	0.00000	1.00000	0.00000

FINAL VALUE OF JMAX	60
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1
17	1
18	1
19	1
20	1
21	1
22	1
23	1
24	1
25	1
26	1
27	1
28	1
29	1
30	1
31	1
32	1
33	1
34	1
35	1
36	1
37	1
38	1
39	1
40	1
41	1
42	1
43	1
44	1
45	1
46	1
47	1
48	1
49	1
50	1
51	1
52	1
53	1
54	1
55	1
56	1
57	1
58	1
59	1
60	1

FINAL VALUES OF J, x, y ALONG INNER BOUNDARY

1	8.000000	0.000000
2	6.900000	0.000000
3	5.827778	0.000000
4	4.806250	0.000000
5	3.858333	0.000000
6	3.006964	0.000000
7	2.275000	0.000000
8	1.685417	0.000000
9	1.261111	0.000000

10	1.025000	0.000000
11	1.000000	0.000000
12	.979906	-.004536
13	.950165	-.010391
14	.910228	-.016457
15	.863697	-.024312
16	.810845	-.032953
17	.752835	-.042259
18	.691006	-.052031
19	.626374	-.061748
20	.560119	-.071150
21	.493296	-.079234
22	.426927	-.083841
23	.362371	-.085156
24	.297834	-.083235
25	.228856	-.078041
26	.159862	-.066536
27	.094908	-.050501
28	.041031	-.024843
29	.010175	.013305
30	.000018	.045659
31	.001885	.061642
32	.007740	.076784
33	.036619	.111588
34	.092225	.147097
35	.165945	.173648
36	.246503	.190027
37	.324049	.193306
38	.388408	.190598
39	.452717	.184418
40	.514500	.174999
41	.584179	.160173
42	.648621	.142148
43	.710605	.121428
44	.769414	.099933
45	.823815	.077812
46	.873175	.057050
47	.916884	.038946
48	.953733	.022902
49	.982822	.011045
50	1.000000	0.000000
51	1.025000	0.000000
52	1.241111	0.000000
53	1.685417	0.000000
54	2.275000	0.000000
55	3.008944	0.000000
56	3.858333	0.000000
57	4.806250	0.000000
58	5.827778	0.000000
59	6.900000	0.000000

JMAX.NSEGS 40 4									
X0.Y0.X1.Y1.T00.T01	0.00000	-1.30000	-5.00000	-1.10000	190.00000	180.00000			
X0.A1.A2.Y0.A1.R2.SARC	0.00000	-1.30000	0.00000	0.00000	13.00000				
X0.Y0.X1.Y1.T00.T01	-5.00000	-1.30000	-7.00000	0.00000	180.00000	90.00000			
X0.A1.A2.Y0.A1.R2.SARC	-5.00000	-1.30000	0.00000	1.35000	2.74604				
X0.Y0.X1.Y1.T00.T01	-7.00000	0.00000	0.00000	8.00000	90.00000	0.00000			
X0.A1.A2.Y0.A1.R2.SARC	-7.00000	0.00000	16.00000	-8.00000	12.18265				
X0.Y0.X1.Y1.T00.T01	0.00000	0.00000	0.00000	8.00000	0.00000	0.00000			
X0.A1.A2.Y0.A1.R2.SARC	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000			
OUTER BOUNDARY NONDIMENSIONAL ARC LENGTH 35.928688									
J1.JF.X1.XF.DX1.DXF	1	11	0.00000	7.00000	1.10000	0.2500			
J1.JF.X1.XF.DX1.DXF	11	50	7.00000	28.92869	0.2500	0.2500			
J1.JF.X1.XF.DX1.DXF	50	50	28.92869	35.92869	0.2500	1.10000			
0.00000	1.10000	2.17222	3.19375	4.14167	4.99306	5.72500	6.31458	6.73899	6.97500
7.00000	7.02500	7.13483	7.32491	7.59065	7.92747	8.33078	8.79599	9.31853	9.89379
10.51721	11.14418	11.89014	12.63049	13.40064	14.19601	15.01202	15.84408	16.69760	17.53900
18.39069	19.24109	20.09461	20.91657	21.73268	22.52905	23.29820	24.03855	24.74451	25.41148
26.03490	26.61016	27.13270	27.59791	28.00122	28.33804	28.60378	28.79386	29.0369	29.28869
29.49369	29.18480	29.61411	30.20369	30.93563	31.78702	32.73494	33.75847	34.82859	35.92869
0.00000	0.3062	0.6046	0.9889	1.11527	1.13897	1.15934	1.17575	1.1875	1.1934
1.18756	1.19413	1.19483	1.19553	1.19558	1.1958	1.19587	1.1959	1.196	1.196
1.19617	1.19622	1.19627	1.19632	1.19637	1.19642	1.19647	1.19652	1.19657	1.19662
1.19667	1.19672	1.19677	1.19682	1.19687	1.19692	1.19697	1.19702	1.19707	1.19712
1.19717	1.19722	1.19727	1.19732	1.19737	1.19742	1.19747	1.19752	1.19757	1.19762
1.19767	1.19772	1.19777	1.19782	1.19787	1.19792	1.19797	1.19802	1.19807	1.19812
1.19817	1.19822	1.19827	1.19832	1.19837	1.19842	1.19847	1.19852	1.19857	1.19862
1.19867	1.19872	1.19877	1.19882	1.19887	1.19892	1.19897	1.19902	1.19907	1.19912
1.19917	1.19922	1.19927	1.19932	1.19937	1.19942	1.19947	1.19952	1.19957	1.19962
1.19967	1.19972	1.19977	1.19982	1.19987	1.19992	1.19997	1.20002	1.20007	1.20012
1.20017	1.20022	1.20027	1.20032	1.20037	1.20042	1.20047	1.20052	1.20057	1.20062
1.20067	1.20072	1.20077	1.20082	1.20087	1.20092	1.20097	1.20102	1.20107	1.20112
1.20117	1.20122	1.20127	1.20132	1.20137	1.20142	1.20147	1.20152	1.20157	1.20162
1.20167	1.20172	1.20177	1.20182	1.20187	1.20192	1.20197	1.20202	1.20207	1

XS.YS	1.261112	-1.360000
XS.YS	1.025000	-1.360000
XS.YS	1.000000	-1.360000
XS.YS	.075000	-1.360000
XS.YS	.865168	-1.360000
XS.YS	.675088	-1.360000
XS.YS	.409746	-1.360000
XS.YS	.072529	-1.360000
XS.YS	-.130780	-1.360000
XS.YS	-.795003	-1.360000
XS.YS	-2.517207	-1.360000
XS.YS	-3.184104	-1.360000
XS.YS	-3.890139	-1.360000
XS.YS	-4.630485	-1.360000
XS.YS	-5.400276	-1.344820
XS.YS	-6.176874	-1.185241
XS.YS	-6.815752	-.700284
XS.YS	-6.999731	-.098904
XS.YS	-6.974258	.940842
XS.YS	-6.901152	1.788348
XS.YS	-6.771242	2.630966
XS.YS	-6.574505	3.458143
XS.YS	-6.301497	4.255967
XS.YS	-5.944205	5.007065
XS.YS	-5.500658	5.691402
XS.YS	-4.976997	6.289398
XS.YS	-4.389910	6.787133
XS.YS	-3.763302	7.180785
XS.YS	-3.122659	7.476729
XS.YS	-2.490312	7.688412
XS.YS	-1.883623	7.831645
XS.YS	-1.315570	7.921821
XS.YS	-.795170	7.972613
XS.YS	-.331686	7.995400
XS.YS	.072529	8.000000
XS.YS	.409346	8.000000
XS.YS	.675088	8.000000
XS.YS	.865168	8.000000
XS.YS	.975000	8.000000
XS.YS	1.000000	8.000000
XS.YS	1.025000	8.000000
XS.YS	1.261112	8.000000
XS.YS	1.645417	8.000000
XS.YS	2.275000	8.000000
XS.YS	3.006945	8.000000
XS.YS	3.859334	8.000000
XS.YS	4.806250	8.000000
XS.YS	5.827779	8.000000
XS.YS	6.900000	8.000000
XS.YS	8.000000	8.000000

```

***** GRID GENERATION *****
SUR. ALGRD PRINTOUT..... KMAX,ITERM,1PER,NCLUS,1STOR, JELL
35 40 0 1 1 1
NS. OMEGA .00020 1.55000
ELLIPTIC GRID FORMED FROM 11 TO 50
EXPONENT COEFFICIENTS FOR SOURCE TERMS.PFAC.OFAC
0.00000 0.00000 0.00000 0.00000
PC1,PC1,PC2,OC2 0.00000 0.00000 0.00000 0.00000
ITERATION NRR AND P5UM 10 .32950E+02
ITERATION NRR AND P5UM 20 .24893E+02
ITERATION NRR AND P5UM 30 .25234E+02
ITERATION NRR AND P5UM 40 .18947E+02

```

J K= 2 L= 1				J K= 2 L= 2				J K= 2 L= 2			
J	K	L	1	J	K	L	2	J	K	L	2
1	1	1	0.00000	1	1	1	0.00000	1	1	1	0.00000
1	1	1	5.42778	1	1	1	5.42778	1	1	1	5.42778
3	3	3	3.85433	3	3	3	3.85433	3	3	3	3.85433
5	5	5	2.27500	5	5	5	2.27500	5	5	5	2.27500
7	7	7	1.26111	7	7	7	1.26111	7	7	7	1.26111
9	9	9	1.00000	9	9	9	1.00000	9	9	9	1.00000
11	11	11	0.00000	11	11	11	0.00000	11	11	11	0.00000
13	13	13	0.00000	13	13	13	0.00000	13	13	13	0.00000
15	15	15	0.00000	15	15	15	0.00000	15	15	15	0.00000
17	17	17	0.00000	17	17	17	0.00000	17	17	17	0.00000
19	19	19	0.00000	19	19	19	0.00000	19	19	19	0.00000
21	21	21	0.00000	21	21	21	0.00000	21	21	21	0.00000
23	23	23	0.00000	23	23	23	0.00000	23	23	23	0.00000
25	25	25	0.00000	25	25	25	0.00000	25	25	25	0.00000
27	27	27	0.00000	27	27	27	0.00000	27	27	27	0.00000
29	29	29	0.00000	29	29	29	0.00000	29	29	29	0.00000
31	31	31	0.00000	31	31	31	0.00000	31	31	31	0.00000
33	33	33	0.00000	33	33	33	0.00000	33	33	33	0.00000
35	35	35	0.00000	35	35	35	0.00000	35	35	35	0.00000
37	37	37	0.00000	37	37	37	0.00000	37	37	37	0.00000
39	39	39	0.00000	39	39	39	0.00000	39	39	39	0.00000
41	41	41	0.00000	41	41	41	0.00000	41	41	41	0.00000
43	43	43	0.00000	43	43	43	0.00000	43	43	43	0.00000
45	45	45	0.00000	45	45	45	0.00000	45	45	45	0.00000
47	47	47	0.00000	47	47	47	0.00000	47	47	47	0.00000
49	49	49	0.00000	49	49	49	0.00000	49	49	49	0.00000
51	51	51	0.00000	51	51	51	0.00000	51	51	51	0.00000
53	53	53	0.00000	53	53	53	0.00000	53	53	53	0.00000
55	55	55	0.00000	55	55	55	0.00000	55	55	55	0.00000
57	57	57	0.00000	57	57	57	0.00000	57	57	57	0.00000
59	59	59	0.00000	59	59	59	0.00000	59	59	59	0.00000

J	K = 2	L = 20	X	Y	Z	J	K = 2	L = 21	X	Y	Z
1			8.00000	0.00000	-0.05087	1			8.00000	0.00000	-0.06347
3			5.82778	0.00000	-0.05087	3			5.82778	0.00000	-0.06347
5			3.85833	0.00000	-0.05087	5			3.85833	0.00000	-0.06347
7			2.27500	0.00000	-0.05087	7			2.27500	0.00000	-0.06347
9			1.26111	0.00000	-0.05087	9			1.26111	0.00000	-0.06347
11			1.00000	0.00000	-0.05087	11			1.00000	0.00000	-0.06347
13			.93289	0.00000	-0.05879	13			.92932	0.00000	-0.07104
15			.84217	0.00000	-0.07144	15			.83722	0.00000	-0.04340
17			.72691	0.00000	-0.09346	17			.72043	0.00000	-0.10675
19			.59315	0.00000	-0.12037	19			.58399	0.00000	-0.13619
21			.44800	0.00000	-0.14418	21			.43441	0.00000	-0.16219
23			.29871	0.00000	-0.15127	23			.27854	0.00000	-0.16969
25			.14209	0.00000	-0.13673	25			.11398	0.00000	-0.15278
27			-.01283	0.00000	-0.0977	27			-.04831	0.00000	-0.09786
29			-.10367	0.00000	.00440	29			-.14003	0.00000	.00802
31			-.10416	0.00000	.11023	31			-.13833	0.00000	.12469
33			-.04096	0.00000	.20107	33			-.07122	0.00000	.22447
35			.08073	0.00000	.26135	35			.05300	0.00000	.28935
37			.23074	0.00000	.24444	37			.21321	0.00000	.31460
39			.38626	0.00000	.24032	39			.36369	0.00000	.32382
41			.52775	0.00000	.27132	41			.50926	0.00000	.30722
43			.66148	0.00000	.23578	43			.64617	0.00000	.27304
45			.78153	0.00000	.19451	45			.76494	0.00000	.23264
47			.88350	0.00000	.15796	47			.87411	0.00000	.19675
49			.96589	0.00000	.13278	49			.96163	0.00000	.17215
51			1.02500	0.00000	.12258	51			1.02500	0.00000	.16211
53			1.68542	0.00000	.12258	53			1.68542	0.00000	.16211
55			3.00694	0.00000	.12258	55			3.00694	0.00000	.16211

J	K = 6	L = 1	X	Y	Z	J	K = 6	L = 2	X	Y	Z
1			8.00000	0.00000	0.00000	1			8.00000	-0.00019	.00004
3			5.82778	0.00000	0.00000	3			5.82778	-0.00019	.00004
5			3.85833	0.00000	0.00000	5			3.85833	-0.00019	.00004
7			2.27500	0.00000	0.00000	7			2.27500	-0.00019	.00004
9			1.26111	0.00000	0.00000	9			1.26111	-0.00019	.00004
11			1.00000	0.00000	0.00000	11			1.00000	-0.00019	.00004
13			.95017	-0.01013	.00231	13			.95009	-0.01031	.00235
15			.88370	-0.02370	.00541	15			.86361	-0.02388	.00545
17			.75284	-0.04120	.00940	17			.75274	-0.04137	.00944
19			.62637	-0.06070	.01374	19			.62628	-0.05037	.01378
21			.49330	-0.07725	.01763	21			.49318	-0.07741	.01767
23			.36237	-0.08302	.01895	23			.36224	-0.08317	.01898
25			.22886	-0.07608	.01737	25			.22870	-0.07620	.01739
27			.09491	-0.04923	.01124	27			.09473	-0.04932	.01126
29			.01038	.01297	-.00296	29			.01018	.01291	-.00295
31			.00188	.06010	-.01372	31			.00170	.06018	-.01373
33			.03662	.10879	-.02483	33			.03653	.10896	-.02487
35			.16595	.16929	-.03864	35			.16580	.16943	-.03867
37			.32405	.18846	-.04301	37			.32391	.18860	-.04305
39			.45272	.18018	-.04113	39			.45261	.18035	-.04116
41			.58418	.15616	-.03564	41			.58409	.15633	-.03568

J	K=6 L=20	X	Y	Z	J	K=6 L=21	X	Y	Z
43		.71061	.11838	-.02702	43		.71053	.11856	-.02706
45		.82382	.07586	-.01731	45		.82375	.07504	-.01736
47		.91684	.03797	-.00867	47		.91683	.03816	-.00871
49		.98202	.01077	-.00246	49		.98199	.01096	-.00250
51		1.02500	0.00000	0.00000	51		1.02500	0.00019	-.00004
53		1.68542	0.00000	0.00000	53		1.68542	0.00019	-.00004
55		3.00694	0.00000	0.00000	55		3.00694	0.00019	-.00004
57		4.80625	0.00000	0.00000	57		4.80625	0.00019	-.00004
59		6.90000	0.00000	0.00000	59		6.90000	0.00019	-.00004

J	K=6 L=20	X	Y	Z	J	K=6 L=21	X	Y	Z
1		8.00000	-.04960	-.01132	1		8.00000	-.06187	.01412
3		5.82778	-.04960	-.01132	3		5.82778	-.06187	.01412
5		3.85833	-.04960	-.01132	5		3.85833	-.06187	.01412
7		2.27500	-.04960	-.01132	7		2.27500	-.06187	.01412
9		1.26111	-.04960	-.01132	9		1.26111	-.06187	.01412
11		1.00000	-.04960	-.01132	11		1.00000	-.06187	.01412
13		.93249	-.05731	-.01308	13		.92932	-.06926	.01581
15		.84217	-.06868	-.01590	15		.83722	-.08131	.01856
17		.72691	-.09111	-.02080	17		.72043	-.10408	.02375
19		.59315	-.11736	-.02679	19		.58399	-.13278	.03031
21		.44800	-.14057	-.03208	21		.43441	-.15813	.03609
23		.29871	-.16747	-.03366	23		.27854	-.16543	.03776
25		.14209	-.13330	-.03043	25		.11398	-.14895	.03400
27		-.01283	-.08752	-.01998	27		-.04831	-.09541	.02178
29		-.10367	-.00448	-.00102	29		-.14003	.00782	-.00179
31		-.10416	.10746	-.02453	31		-.13833	.12156	-.02775
33		-.04096	.19603	-.04474	33		-.07122	.21884	-.04995
35		.09073	.25480	-.05816	35		.05300	.24210	-.06439
37		.23974	.27731	-.06329	37		.21321	.30572	-.07001
39		.38626	.28304	-.06440	39		.36369	.31570	-.07206
41		.52775	.26452	-.06038	41		.50926	.29951	-.06836
43		.66148	.22987	-.05247	43		.64617	.26620	-.06076
45		.78153	.18963	-.04328	45		.76898	.22681	-.05177
47		.88350	.15400	-.03515	47		.87411	.19182	-.04378
49		.96589	.12944	-.02955	49		.96163	.16783	-.03831
51		1.02500	.11951	-.02728	51		1.02500	.15804	-.03607
53		1.68542	.11951	-.02728	53		1.68542	.15804	-.03607
55		3.00694	.11951	-.02728	55		3.00694	.15804	-.03607
57		4.80625	.11951	-.02728	57		4.80625	.15804	-.03607
59		6.90000	.11951	-.02728	59		6.90000	.15804	-.03607

J	K=10 L=1	X	Y	Z	J	K=10 L=2	X	Y	Z
1		8.00000	0.00000	0.00000	1		8.00000	.00009	.00018
3		5.82778	0.00000	0.00000	3		5.82778	.00009	.00018
5		3.85833	0.00000	0.00000	5		3.85833	.00009	.00018
7		2.27500	0.00000	0.00000	7		2.27500	.00009	.00018
9		1.26111	0.00000	0.00000	9		1.26111	.00009	.00018
11		1.00000	0.00000	0.00000	11		1.00000	.00009	.00018
13		.95017	.00451	.00935	13		.95009	.00459	.00953
15		.86370	.01055	.02190	15		.86361	.01063	.02207
17		.75284	.01834	.03807	17		.75274	.01841	.03823
19		.62637	.02679	.05563	19		.62628	.02687	.05574

K=10 L=20				K=10 L=21			
J	X	Y	Z	J	X	Y	Z
21	.49330	.03438	.07139	21	.49318	.03445	.07154
23	.36237	.03695	.07672	23	.36224	.03701	.07696
25	.22886	.03386	.07031	25	.22870	.03391	.07042
27	.09491	.02191	.04550	27	.09473	.02195	.04558
29	.01038	-.00577	-.01199	29	.01018	-.00575	-.01193
31	.00188	-.02675	-.05554	31	.00170	-.02678	-.05561
33	.03662	-.04842	-.10054	33	.03653	-.04849	-.10070
35	.16595	-.07534	-.15645	35	.16580	-.07540	-.15658
37	.32405	-.08387	-.17416	37	.32391	-.08394	-.17429
39	.45272	-.08019	-.16651	39	.45261	-.08026	-.16667
41	.58418	-.06950	-.14431	41	.58409	-.06957	-.14447
43	.71061	-.05269	-.10940	43	.71053	-.05277	-.10957
45	.82382	-.03376	-.07011	45	.82375	-.03384	-.07027
47	.91688	-.01690	-.03509	47	.91683	-.01698	-.03526
49	.98202	-.00479	-.00995	49	.98199	-.00488	-.01013
51	1.02500	0.00000	0.00000	51	1.02500	-.00009	-.00018
53	1.68562	0.00000	0.00000	53	1.68542	-.00009	-.00018
55	3.00694	0.00000	0.00000	55	3.00694	-.00009	-.00018
57	4.80625	0.00000	0.00000	57	4.80625	-.00009	-.00018
59	6.90000	0.00000	0.00000	59	6.90000	-.00009	-.00018

K=10 L=20				K=10 L=21			
J	X	Y	Z	J	X	Y	Z
1	8.00000	.02207	.04583	1	8.00000	.02754	.05718
3	5.82778	.02207	.04583	3	5.82778	.02754	.05718
5	3.85833	.02207	.04583	5	3.85833	.02754	.05718
7	2.27500	.02207	.04583	7	2.27500	.02754	.05718
9	1.26111	.02207	.04583	9	1.26111	.02754	.05718
11	1.00000	.02207	.04583	11	1.00000	.02754	.05718
13	.93289	.02551	.05297	13	.92932	.03082	.06400
15	.84217	.03101	.06440	15	.83722	.03619	.07514
17	.72691	.04055	.08420	17	.72043	.04632	.09618
19	.59315	.05223	.10845	19	.58399	.05909	.12271
21	.44800	.06256	.12991	21	.43441	.07037	.14613
23	.29871	.06563	.13629	23	.27854	.07362	.15288
25	.14209	.05933	.12319	25	.11348	.06524	.13765
27	-.01283	.03895	.08088	27	-.04831	.04246	.08817
29	-.10367	-.00199	-.00414	29	-.14003	-.00748	-.00723
31	-.10416	-.04783	-.09931	31	-.13833	-.05410	-.11234
33	-.04096	-.08724	-.18116	33	-.07122	-.09739	-.20224
35	.08073	-.11340	-.23547	35	.05300	-.12555	-.26070
37	.23974	-.12341	-.25627	37	.21321	-.13550	-.28345
39	.38626	-.12597	-.26157	39	.36369	-.14050	-.29175
41	.52775	-.11772	-.24445	41	.50926	-.13330	-.27679
43	.66148	-.10230	-.21243	43	.64617	-.11847	-.24600
45	.78153	-.08439	-.17524	45	.76898	-.10094	-.20960
47	.88350	-.06854	-.14231	47	.87411	-.08537	-.17727
49	.96589	-.05761	-.11963	49	.96163	-.07469	-.15510
51	1.02500	-.05319	-.11044	51	1.02500	-.07034	-.14605
53	1.68542	-.05319	-.11044	53	1.68542	-.07034	-.14605
55	3.00694	-.05319	-.11044	55	3.00694	-.07034	-.14605
57	4.80625	-.05319	-.11044	57	4.80625	-.07034	-.14605
59	6.90000	-.05319	-.11044	59	6.90000	-.07034	-.14605

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.

```

1      PROGRAM IGP (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE8,PL3,KBI
      COMMON /GRIDC/ GXMIN, GXMAX, GYMIN, GYMAX, X(100,50), Y(100,50)
      COMMON /HOR/ TITLE(5)
      C
5      C*****GRIP-PROGRAM TO PLOT THE COMPUTATIONAL GRID ABOUT AN
      C      AERODYNAMIC BODY OR AIRFOIL SECTION*****
      C
      C*****INPUT SECTION*****
      C*****CONTROL PARAMETER INPUT
10     C**NADD=-1 SUPPRESSES LISTING OF PLOT DATA,NADD=1 ALLOWS IT
      C**JMAX,KMAX=NUMBER OF J,K POINTS
      C**HEAD-ALPHANUMERIC INFORMATION DESCRIBING THE
      C      CONFIGURATION BEING PLOTTED
15     CALL CONNEC (5,TAPE5)
      CALL CONNEC (5,TAPE6)
      CALL TERM (1,1024)
      CALL SETBUF (3)
      CALL ANMODE
20     C      READ(5,10)HEAD
      C
      WRITE (6,10)
      READ (5,*) IBO
      IBO=IBO/10
      CALL INITT (IBO)
25     CALL BINITT
      C
      WRITE (6,20)
      READ (5,*) JMAX,KMAX
      C
30     WRITE (6,30)
      READ (5,*) GXMIN,GXMAX,GYMIN,GYMAX
      WRITE (6,40)
      READ (5,50) TITLE
      CALL NEWPAG
35     C
      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,KMA
      C***** PLOT THE GRID *****
      CALL GROPLT (JMAX,KMAX,IBO)
40     C***TERMINATE PLOTTING
      CALL FINITT (0.700)
      STOP
      C
45     10 FORMAT (19H WHAT IS BAUD RATE?)
      20 FORMAT (20H WHAT ARE JMAX,KMAX?)
      30 FORMAT (30H WHAT ARE XMIN,XMAX,YMIN,YMAX?)
      40 FORMAT (28H ENTER TITLE - UP TO 50 CHAR)
      50 FORMAT (5A10)
      END

```

```

1      SUBROUTINE GROPLT (JMAX,KMAX,I3D)
      COMMON /GRIDC/ GXMIN, GXMAX, GYMIN, GYMAX, XI100,50, YI100,50)
      COMMON /HDR/ TITLE(5)
      DIMENSION GX(128), GY(128)

5      C
      C***READJUST PLOT LIMITS IN ORDER TO AVOID STRETCHED PLOTS
      ICOUNT=0
      10  XMAX=GXMAX
      XMIN=GXMIN
10      YMAX=GYMAX
      YMIN=GYMIN
      XDIF=XMAX-XMIN
      YDIF=YMAX-YMIN
      IF (XDIF.LT.YDIF) GO TO 20
15      XDIFH=XDIF*0.5
      YMID=(YMAX+YMIN)*0.5
      YMX=YMID+XDIFH
      YMN=YMID-XDIFH
      YMAX=YMX
      YMIN=YMN
      GO TO 30
      20  YDIFH=YDIF*0.5
      XMID=(XMAX+XMIN)*0.5
      XMX=XMID+YDIFH
      XMN=XMID-YDIFH
      XMAX=XMX
      XMIN=XMN
25      30  CONTINUE

      C
30      C      PLOT THE LINES
      IF (ICOUNT.GT.0) GO TO 40
      AXMIN=XMIN
      AXMAX=XMAX
      AYMIN=YMIN
      AYMAX=YMAX
35      40  CONTINUE

      C
      CALL BINITT
      CALL NPTS (JMAX)
40      CALL ANMODE
      WRITE (6,120) TITLE
      CALL XFRM (2)
      CALL YFRM (2)

      C
45      C
      CALL DLIMX (XMIN,XMAX)
      CALL DLIMY (YMIN,YMAX)
      CALL SLIMX (150,800)
      CALL SLIMY (150,700)

50      C
      DO 70 K=1,KMAX
      DO 50 J=1,JMAX
      GX(IJ)=X(IJ,K)
55      50  GY(IJ)=Y(IJ,K)

      C
      IF (K.GT.1) GO TO 50
      CALL CHECK (GX,GY)

```

```

      CALL DSPLAY (GX,GY)
60    CALL CPLDT (GX,GY)
      70 CONTINUE
      CALL NPTS (KMAX)
      DO 90 J=1,JMAX
      DO 80 K=1,KMAX
      GX(K)=X(J,K)
65    80 GY(K)=Y(J,K)
      CALL CPLDT (GX,GY)
      90 CONTINUE
      CALL BELL
      CALL TSEND
70    CALL TINPUT (II)
      C
      IF (II.EQ.83) GO TO 110
      C
      CALL NEWPAG
      CALL TSEND
75    DO 100 J=1,JMAX
      GX(J)=X(J,1)
      100 GY(J)=Y(J,1)
      CALL BINITT
80    CALL DLIMX (AXM(N,AXMAX)
      CALL DLIMY (AYMIN,AYMAX)
      CALL SLIMX (150,800)
      CALL SLIMY (50,700)
      CALL NPTS (JMAX)
85    CALL XFRM (2)
      CALL YFRM (2)
      CALL CHECK (GX,GY)
      CALL DSPLAY (GX,GY)
      CALL TSEND
90    CALL BELL
      CALL ANMODE
      WRITE (6,130)
      CALL TSEND
      CALL RECOVR
95    CALL VCUSR (ICH,XX,YY)
      GXMIN=XX
      GYMIN=YY
      CALL ANMODE
      WRITE (6,140)
100    CALL RECOVR
      CALL TSEND
      CALL ANMODE
      CALL VCUSR (ICH,XX,YY)
      GXMAX=XX
      GYMAX=YY
105    CALL TSEND
      ICOUNT=ICOUNT+1
      CALL NEWPAG
      GO TO 10
110    110 CONTINUE
      CALL NEWPAG
      RETURN
      C
120 FORMAT (5A10)

.15 130 FORMAT (30H POSITION CURSOR FOR XMIN,YMIN)
      140 FORMAT (140,30H POSITION CURSOR FOR XMAX,YMAX)
      END

```

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
12	Commander Defense Technical Info Center ATTN: DDC-DDA Cameron Station Alexandria, VA 22314	1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDMD-ST 5001 Eisenhower Avenue Alexandria, VA 22333	1	Commander US Army Communications Research and Development Command ATTN: DRDCO-PPA-SA Fort Monmouth, NJ 07703
8	Commander US Army Armament Research and Development Command ATTN: DRDAR-TSS (2 cys) DRDAR-LCA-F Mr. D. Mertz Mr. A. Loeb Mr. S. Wasserman Mr. H. Hudgins Mr. E. Friedman Dover, NJ 07801	1	Commander US Army Electronics Research and Development Command Technical Support Activity ATTN: DELSD-L Fort Monmouth, NJ 07703
1	Commander US Army Armament Materiel Readiness Command ATTN: DRSAR-LEP-L, Tech Lib Rock Island, IL 61299	3	Commander US Army Missile Command ATTN: DRSMI-R DRSMI-YDL DRSMI-RDK Mr. R. Deep Redstone Arsenal, AL 35809
1	Director US Army Armament Research and Development Command ATTN: DRDAR-LCB-TL Watervliet, NY 12189	1	Commander US Army Tank Automotive Research and Development Command ATTN: DRDTA-UL Warren, MI 48090
1	Commander US Army Aviation Research and Development Command ATTN: DRDAV-E 4300 Goodfellow Boulevard St. Louis, MO 63120	1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL, Tech Lib White Sands Missile Range, NM 88002

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
1	Commander US Army Research Office P. O. Box 12211 Research Triangle Park NC 27709	3	Director NASA Ames Research Center ATTN: MS-202A-14 Dr. P. Kutler MS-202-1, Dr. T. Pulliam MS-227-8, Dr. L. Schiff Moffett Field, CA 94035
1	Commander US Naval Air Systems Command ATTN: AIR-604 Washington, D. C. 20360		
4	Commander US Naval Surface Weapons Center ATTN: Dr. T. Clare, Code DK20 Dr. P. Daniels Mr. D. A. Jones III Mr. L. Mason Dahlgren, VA 22448	2	Sandia Laboratories ATTN: Division No. 1331, Mr. H.R. Vaughn Mr. G.R. Eisler P.O. Box 580 Albuquerque, NJ 87184
4	Commander US Naval Surface Weapons Center ATTN: Code 312 Mr. R. Voisinnet Mr. R. Driftmeyer Mr. J. Knott Mr. R. Schlie Silver Spring, MD 20910	1	University of California, Davis Department of Mechanical Engineering ATTN: Prof. H.A. Dwyer Davis, CA 95616
1	Commander US Naval Weapons Center ATTN: Code 3431, Tech Lib China Lake, CA 93555	1	University of Colorado Department of Aerospace Engineering ATTN: Prof. G. Inger Boulder, CO 80309
1	Director NASA Langley Research Center ATTN: NS-185, Tech Lib Langley Station Hampton, VA 23365	1	University of Delaware Mechanical and Aerospace Engineering Department ATTN: Dr. J. E. Danberg Newark, DE 19711

DISTRIBUTION LIST

No. of
Copies

Organization

Aberdeen Proving Ground

Dir, USAMSAA

ATTN: DRXSY-D

DRXSY-MP, H. Cohen

Cdr, USATECOM

ATTN: DRSTE-TO-F

Dir, USACSL, Bldg. E3516, EA

ATTN: DRDAR-CLB-PA

DRDAR-ACW, Mr. M. Miller

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: _____

