

NASA Contractor Report 3222

NASA
CR
3222
c.1

DUPLICATE COPY: RET
NEWL TECHNICAL LIBRA
KIRTLAND AFB, N. M

0062032



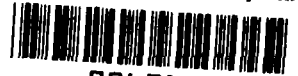
TECH LIBRARY KAFB, NM

Calculation of Two-Dimensional Inlet Flow Fields in a Supersonic Free Stream by an Implicit Marching Code With Nonorthogonal Mesh Generation - User's Manual

S. H. Biringen and O. J. McMillan

CONTRACT NAS1-15305
JANUARY 1980





NASA Contractor Report 3222

Calculation of Two-Dimensional Inlet
Flow Fields in a Supersonic Free
Stream by an Implicit Marching Code
With Nonorthogonal Mesh Generation -
User's Manual

S. H. Biringen and O. J. McMillan
Nielsen Engineering & Research, Inc.
Mountain View, California

Prepared for
Langley Research Center
under Contract NAS1-15305



National Aeronautics
and Space Administration

**Scientific and Technical
Information Office**



TABLE OF CONTENTS

<u>Section</u>	<u>Page No.</u>
1. INTRODUCTION	1
2. CALCULATION OF TWO-DIMENSIONAL INLET FLOW FIELDS (THE EULER CODE)	1
2.1 Formulation of the Problem	2
2.1.1 Governing Equations	2
2.1.2 Boundary and Initial Conditions	4
2.1.3 Solution Procedure and the Numerical Scheme	6
2.1.4 Numerical Stability	8
2.2 Overall Program Logic	9
2.3 Program Use and Operation	11
2.4 Program Accuracy and Limitations	14
2.4.1 Accuracy	14
2.4.2 Limitations	14
3. GENERATION OF NONORTHOGONAL GRIDS FOR TWO-DIMENSIONAL INLET FLOW FIELDS	15
3.1 Problem Formulation	15
3.2 Overall Program Logic	17
3.3 Program Use and Operation	18
REFERENCES	21
FIGURE 1	23

1. INTRODUCTION

This report consists of a technical explanation of the two computer codes that are employed to calculate two-dimensional inlet flow fields in a supersonic free stream. The first code (the Euler code) is a modified version of the program that was developed and used in reference 1 to calculate external flows about two-dimensional airfoils. The second is a modified version of a nonorthogonal grid-generation routine which was developed in reference 2 to generate body fitted coordinate systems for airfoils.

In the next section of this report the Euler code is explained in detail. Mathematical formulation of the problem is given. The computational algorithm and program logic are explained. The third section of the report consists of a detailed explanation of the grid-generation routine.

2. CALCULATION OF TWO-DIMENSIONAL INLET FLOW FIELDS (THE EULER CODE)

In this section the technical approach employed for the calculation of two-dimensional inlet flow fields in a uniform supersonic free stream is presented. In Section 2.1 the mathematical formulation of the problem is given. Section 2.2 consists of the overall logic of the computer code that is used for the numerical integration procedure. Program usage and operation instructions are given in Section 2.3 and an explanation of program accuracy and limitations is given in Section 2.4.

2.1. Formulation of the Problem

In this section the governing equations (Euler equations) for inviscid flow fields in two-dimensional inlets are given, boundary and initial conditions are discussed and the numerical solution procedure and its stability are described. Our description follows closely that of reference 1.

2.1.1 Governing Equations.- The Euler equations in Cartesian coordinates for two-dimensional or axisymmetric flows can be written in dimensionless, conservation-law form for a perfect gas without external forces as ^{3,4,5}

$$\partial_t \vec{q} + \partial_x \vec{E} + \partial_y \vec{F} + j \frac{\vec{H}}{Y} = 0 \quad (2.1)$$

$$\vec{q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix}, \quad \vec{E} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(e + p) \end{pmatrix}, \quad \vec{F} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(e + p) \end{pmatrix}, \quad \vec{H} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 \\ (e + p)v \end{pmatrix}$$

and

$$p = (\gamma - 1) \left[e - 0.5 \rho (u^2 + v^2) \right] \quad (2.2)$$

Also, $j = 0$ for plane flows and $j = 1$ for axisymmetric flows.

In these equations u and v are the velocities along the x and y coordinates respectively, p is the pressure, ρ is the density and e is the total energy per unit mass. The sound speed, a , is given by $a^2 = \gamma(\gamma - 1) \left[\frac{e}{\rho} - 0.5(u^2 + v^2) \right]$. In order to use a body-fitted coordinate system, the governing equations are re-written subject to the general transformation

$$\left. \begin{aligned} \xi &= \xi(x, y, t) \\ \eta &= \eta(x, y, t) \\ \tau &= t \end{aligned} \right\} \quad (2.3)$$

where ξ is the coordinate along the body and η is the coordinate perpendicular to the body.

In terms of the new independent variables ξ , η and τ which map the original x, y, t plane into a rectangular domain, the conservation form of the governing equations is

$$\partial_{\tau} \hat{q} + \partial_{\xi} \hat{E} + \partial_{\eta} \hat{F} + j\hat{H} = 0 \quad (2.4)$$

where

$$\begin{aligned} \hat{q} &= \frac{\vec{q}}{J}, \quad \hat{E} = \left(\xi_t \vec{q} + \xi_x \vec{E} + \xi_y \vec{F} \right) / J, \\ \hat{F} &= \left(\eta_t \vec{q} + \eta_x \vec{E} + \eta_y \vec{F} \right) / J, \quad \hat{H} = \frac{\vec{H}}{YJ} \end{aligned}$$

Here J is the transformation Jacobian

$$J = \xi_x \eta_y - \xi_y \eta_x = 1 / \left(x_{\xi} y_{\eta} - x_{\eta} y_{\xi} \right) \quad (2.5)$$

The metrics ξ_t , ξ_x etc. are formed from the derivatives of x_{τ} , x_{ξ} , etc., using the relations

$$\left\{ \begin{aligned} \xi_x &= J y_{\eta} & \eta_x &= -J y_{\xi} \\ \xi_y &= -J x_{\eta} & \eta_y &= J x_{\xi} \\ \xi_t &= -x_{\tau} \xi_x - y_{\tau} \xi_y & \eta_t &= -x_{\tau} \eta_x - y_{\tau} \eta_y \end{aligned} \right\} \quad (2.6)$$

We now define velocities along the ξ and η coordinates

$$\begin{aligned}
 U &= \xi_t + \xi_x u + \xi_y v \\
 V &= \eta_t + \eta_x u + \eta_y v
 \end{aligned}
 \tag{2.7}$$

Using these defined velocities, \hat{E} and \hat{F} can be written as

$$\hat{E} = J^{-1} \begin{pmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ (e + p)U - \xi_t p \end{pmatrix}, \hat{F} = J^{-1} \begin{pmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ (e + p)V - \eta_t p \end{pmatrix}$$

(2.8)

$$\hat{H} = \frac{J}{Y}^{-1} \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 \\ (e + p)v \end{pmatrix}$$

The inlet flow solution is obtained as the steady state solution of the time-marching method employed to solve equation (2.4).

2.1.2 Boundary and Initial Conditions.- In the problem under consideration, four types of boundaries are considered (figure 1). These are solid boundaries, inflow boundaries, outer boundaries and outflow boundaries.

Along the cowl surface and the ramp surface the tangency condition is satisfied, i.e. $V = 0$. This gives

$$\begin{pmatrix} u \\ v \end{pmatrix} = J^{-1} \begin{bmatrix} \eta_y & \xi_y \\ \eta_x & \xi_x \end{bmatrix} \begin{pmatrix} U - \xi_t \\ -\eta_t \end{pmatrix} \quad (2.9)$$

The pressure on the body surface is found from a relation which is obtained by adding η_x times the ξ -momentum equation to η_y times the η -momentum equation. This gives

$$\begin{aligned} \rho \left(\partial_\tau \eta_t + u \partial_\tau \eta_x + v \partial_\tau \eta_y \right) - \rho U \left(\eta_x u_\xi + \eta_y v_\xi \right) - \frac{\rho v}{Y} \left(u \eta_x + v \eta_y \right) = \\ p_\xi \left(\eta_x \xi_x + \eta_y \xi_y \right) + p_\eta \left(\eta_x^2 + \eta_y^2 \right) \end{aligned} \quad (2.10)$$

In this way the values of the flow field quantities on the wall are calculated by extrapolation from the inner field.

Free stream values are specified at the inflow and outer boundaries. For economy in the use of mesh points in the work done to date, the inflow boundary was located a short distance downstream of the leading edge of the ramp. This requires special treatment for the mesh points on the inflow boundary which are downstream of the ramp-leading-edge shock. This treatment is discussed below under the paragraph dealing with initial conditions.

The outflow boundary conditions on the portion of the outflow boundary external to the inlet are calculated by zeroth-order extrapolation from the interior. On that portion inside the inlet duct, the boundary conditions are specified according to whether the condition being calculated is supercritical or subcritical. In the case of supercritical operation, the flow field variables are again calculated by zeroth-order extrapolation from the interior. For subcritical operation (subsonic outflow) a set of boundary conditions are incorporated which are

obtained from the steady-state forms of the governing equations assuming uniform parallel outflow. In accord with this assumption, a constant-area section is added to the downstream portion of the inlet duct and the axial velocity u , at the end of this duct extension is fixed as constant at its value calculated from one-dimensional compressible flow equations. The velocity component parallel to the outflow boundary, v , is set to zero; pressure, p , and density, ρ , are calculated by zeroth-order extrapolation from the interior. Energy is calculated from these values of u , v , ρ , and p . For compatibility with the normal momentum equation, the pressure gradient in the direction perpendicular to the main flow direction, η , is set to zero at both walls.

The initial conditions are specified by using either impulsive initial condition, i.e. freestream conditions or the final solution of a previously calculated flow field. When starting from freestream conditions, values calculated from 2-D shock theory are used at the 3-5 points on the inflow boundary that are downstream of the ramp shock.

2.1.3 Solution Procedure and the Numerical Scheme.- In the solution procedure employed to integrate equation (2.4) a temporal linearization process is employed. In order to employ this on the vector terms of equation (2.4) one needs to evaluate the Jacobian matrices $\hat{A} \equiv \partial \hat{E} / \partial \hat{q}$ and $\hat{B} \equiv \partial \hat{F} / \partial \hat{q}$. The flux vectors \hat{E} and \hat{F} are both linear combinations of \vec{q} , \vec{E} , and \vec{F} (reference 1).

In the solution procedure the linearized equations are cast into delta-form algorithm by approximate factorization^{6,7} (AF). The algorithm is non-iterative, and requires the inversion of two block-tridiagonal (4x4) coefficient matrices at each time step in the integration procedure at only the interior points. References 6, 7, 8 and 9 contain detailed descriptions and various applications of the delta-form AF algorithm, hence here we outline it briefly. The delta-form AF algorithm can be used

either with trapezoidal or Euler temporal implicit differencing and reads

$$\begin{aligned}
 (\mathbf{I} + h\delta_{\xi}\hat{\mathbf{A}}^n)(\mathbf{I} + h\delta_{\eta}\hat{\mathbf{B}}^n)(\hat{\mathbf{q}}^{n+1} - \hat{\mathbf{q}}^n) = & -\Delta t(\delta_{\xi}\hat{\mathbf{E}}^n + \delta_{\xi}\hat{\mathbf{F}}^n + \eta\hat{\mathbf{H}}^n) \\
 & -\alpha\Delta t(\delta_{\xi\xi\xi\xi} + \delta_{\eta\eta\eta\eta})\hat{\mathbf{q}}^n
 \end{aligned} \tag{2.11}$$

Here δ_{ξ} and δ_{η} are second-order central-difference operators, Δt is the integration step size, $h = \Delta t$ or $\Delta t/2$ for first-order or second-order two-level time differencing, respectively. The metric coefficients are evaluated at time-level $n + 1/2$ for second-order accuracy. The fourth-order smoothing term is added to the RHS of the difference equation both to overcome non-linear instability and to damp short wave lengths (references 6 and 7). It is generally desirable to add the smoothing term both in terms of explicit and implicit portions. The explicit part suppresses non-linear instabilities whereas the implicit part enables the use of $\alpha \sim O(\Delta t)$ for very large Δt , where α is the artificial viscosity coefficient. In the current work (reference 1) these terms have been included in a pseudo-implicit manner by inverting a product of scalar pentadiagonals

$$-\alpha\Delta t(\delta_{\xi\xi\xi\xi} + \delta_{\eta\eta\eta\eta})\hat{\mathbf{q}}^n \equiv \hat{\mathbf{q}}^* - \hat{\mathbf{q}}^n \tag{2.12}$$

where

$$\left[\mathbf{I} + \alpha\Delta t\mathbf{J}^{-1}(\nabla_{\xi}\Delta_{\xi})^2\mathbf{J} \right] \left[\mathbf{I} + \alpha\Delta t\mathbf{J}^{-1}(\nabla_{\eta}\Delta_{\eta})^2\mathbf{J} \right] \hat{\mathbf{q}}^* - \hat{\mathbf{q}}^n = 0$$

It should be noted that for compatibility with the boundary conditions, a second-order smoothing is used at points adjacent to the boundary, whereas no smoothing is used at the boundary points themselves.

2.1.4 Numerical Stability.- In explicit methods the step size in the marching direction is bounded by the Courant-Friedrics-Lewy (CFL) condition. According to this, for the AF scheme in the $t - \xi$ sweep

$$\frac{\Delta t}{\Delta \xi} (\hat{\sigma}_A)_{\max} \leq 1 \quad (2.13a)$$

Where $(\hat{\sigma}_A)_{\max}$ is the maximum spectral radius of the local eigenvalues of \hat{A} . Similarly in the $t-\eta$ sweep, one could write the CFL condition as

$$\frac{\Delta t}{\Delta \eta} (\hat{\sigma}_B)_{\max} \leq 1 \quad (2.13b)$$

where $(\hat{\sigma}_B)_{\max}$ is the maximum spectral radius of the local eigenvalues of \hat{B} . A practical constraint in a two-dimensional problem can be written as

$$\Delta t \left[\max \text{ of } \left(\frac{\hat{\sigma}_A}{\Delta \xi}, \frac{\hat{\sigma}_B}{\Delta \eta} \right) \right] \leq 1 \quad (2.14)$$

In implicit schemes usually this condition is relaxed so that the right-hand side constant can assume values much larger than one. In the present case typical values were around 5 to 10. But since the present scheme is only neutrally stable, its convergence cannot be improved by the use of very large time steps (reference 1).

It should be noted that in most of the calculations done to date (reference 13), a Courant number of about 10 was used without any stability problems. However the following precautionary remarks are in order:

(1) The cowl had to be introduced into the calculation over some 50 time steps by slowly inflating it (i.e. by slowly applying the wall boundary conditions).

(2) In the case of an inlet with an isentropic-compression ramp, calculations had to be made for about 500 time steps at a Courant number of one to capture the ramp shock alone. The cowl boundary condition was then introduced slowly as above and the Courant number was increased to 5 and then to 10.

(3) In the case of subcritical operation, the calculations produced instabilities after the shock was expelled from the inlet. Stability was restored by keeping the Courant number equal to one.

2.2 Overall Program Logic

In this section the overall logic of the computer program used to solve equation (2.11) is summarized. The details are given in reference 1. The computational procedure starts by forming the right-hand side terms (RHS) in equation (2.11), first the smoothing operator then the steady part. Then, the computational algorithm forms Δq^* by block tridiagonal matrix inversion in ξ

$$(I + h\delta_{\xi} \hat{A}^n) \Delta q^* = \text{RHS} \quad (2.15)$$

Δq^* is stored temporarily in \hat{q}^{n+1} . The procedure continues with block tridiagonal matrix inversion in η

$$(I + h\delta_{\eta} \hat{B}^n) \Delta q^n = \Delta q^* \quad (2.16)$$

Finally \hat{q}^{n+1} is obtained from

$$\hat{q}^{n+1} = \hat{q}^n + \Delta \hat{q}^n \quad (2.17)$$

Note that for the two-level scheme used here, the algorithm requires two levels of data storage.

In the computer program used in this work the execution of the above outlined algorithm is performed by PROGRAM MAIN. It starts the computation by calling SUB.INITIA. This sub-routine defines the physical and mathematical constants that are to be used in the computation procedure. The initial field is also specified in this subprogram. An option is provided for starting capabilities from a previously calculated (converged) field. This subroutine also calls SUB.GRID by which either a simple-stretched grid for a flat plate is computed or a previously calculated general grid is read from a disc. Then SUB.XYMETS is called to evaluate the metric coefficients for the grid system.

The main program then calls SUB.EIGEN, which calculates the eigenvalues of the coefficient matrices \hat{A} and \hat{B} , it finds the spectral radius and finally computes the time-step from the prescribed Courant number. The optional SUB.MAP maps the P-matrix; this can either be the Jacobian of transformation or the calculated pressure field. The normalized variables and P/P_∞ of the initial field, are then printed by SUB.OUTPT. Free stream quantities are given or computed by SUB.HARVIO which is a library subroutine.

After the initial field is prescribed and printed, the main program starts to execute the integration loop. This is done by calling SUB.STEP. This subroutine in turn calls SUB.BC to calculate data at the flow field boundaries. SUB.STEP then calls SUB.RHS which computes the steady part of the difference algorithm and SUB.SMOOTH which forms the smoothing operator. The temporal differences are evaluated by SUB.DIFFER. SUB.STEP then calls

SUB.FILTRX which performs the block tridiagonal matrix inversion in the ξ -direction and SUB.FILTRY which performs the block tridiagonal matrix inversion in the η -direction. Final output is furnished by subroutine SUB.HARVIO at the end of the iteration loop.

It should be noted that the computer code employed here is an adaptation of a code which has been applied effectively to external flows over airfoils (reference 1). The original code contains options for viscous effects (laminar and turbulent), however since no attempt has been made to include viscous effects in the present calculations, the pertaining subroutines have not been included into the above described overall flow logic. The original program also contains options for upwind differencing in conservative form (references 1, 5, and 6) to be used in transonic flow calculations. These subroutines have also not been used in the present work.

2.3 Program Use and Operation

In this section we give a detailed description of the input information required to run a typical case. As described in Section 2.2, the initial field as well as values of various constants are evaluated in SUB.INITIA. This subroutine also reads the input data except for the last card, which specifies geometry (planar or axisymmetric) and type of outflow boundary. This last card is read by SUB.BC. We now list the input data cards in order:

Card No.	Format	Variables
1	8I5	NMAX: Maximum number of time steps JMAX: Maximum number of points along ξ KMAX: Maximum number of points along η NP: Number of time steps for calling SUB.MAP.

Card No.	Format	Variables
1 (continued)	8I5	<p>METH: Flag for upwind differencing. If METH > 0, program skips upwind differencing.</p> <p>IREAD: Option for grid system. If IREAD > 0, reads grid from disc.</p> <p>INVIS: If INVIS > 0, program accounts for viscous effects; If INVIS = 0, the calculation is inviscid.</p>
2	7I5	<p>IREGO: If IREGO > 0, reads initial field from disc, otherwise starts from free stream conditions.</p> <p>ISTORE: If ISTORE > 0, solution data stored on disc.</p> <p>JTAIL1: First ξ-interior point, set to 1.</p> <p>JTAIL2: Last ξ-interior point, set to JMAX-1.</p> <p>IPLLOT: Index for calling plot routine.</p>
3	3I5	<p>IUPWIND: If IUPWIND > 0, skips upwind differencing.</p> <p>IOSCIL: For stationary airfoil set IOSCIL = 0.</p> <p>LAMIN: If LAMIN \geq 0, calculates turbulent viscosity.</p>
4	8F10.0	<p>CNBR: Courant number; set to about 10.</p> <p>DX: x-increment used in SUB.GRID. For grid read off the disc set DX = 0.</p> <p>DY: y-increment, as above.</p> <p>FSMACH: Free stream Mach number.</p> <p>SMU: Pseudo viscosity coefficient. Set to about 10 times the time step.</p>

Card No.	Format	Variables
4 (continued)	8F10.0	EPS: Used in SUB.GRID. Here set EPS = 0. RE: Reynolds number ALPHA: Angle of attack
5	4F10.0	XOSCIL: } Variables required for oscillating VARA: } airfoil. For the inlet flow set VARB: } to zero. VARC: }
6	2I2	LFAC: If LFAC = 1 outflow is subsonic, otherwise it is supersonic JAXI: If JAXI = 0 flow is plane 2-D, If JAXI = 1 flow is axisymmetric

In starting from free stream conditions, the number of points on the inflow boundary to be placed downstream of the ramp shock are prescribed in SUB.INITIA. Values for the variables are also prescribed in this subroutine using the 2-D wedge theory.

The typical output of the program involves the following (described in more detail in reference 13):

- (1) Tabulation of the input parameters
- (2) Printout of the Jacobian matrix as calculated from the generated grid
- (3) A map of the transformation Jacobian
- (4) Printout of the flow field variables on the cowl surface
- (5) Free stream quantities given by SUB.HARVIO. Note that this is a library subroutine
- (6) Iteration index, physical time and time step
- (7) Maximum ERRORS occurring within the field
- (8) Values of LFAC (if LFAC > 1 outflow is subsonic) and JAXI (if JAXI = 0 flow is plane 2-D, if JAXI = 1 flow is axisymmetric)

- (9) Mass flow rate evaluated at various ξ -stations.
- (10) Residual quantity; must converge to a small number such as $O(10^{-1}) - O(10^{-2})$ for converged solution
- (11) Line-printer plot of pressure in the computational plane
- (12) Line-printer plot of C_p on the ramp surface
- (13) Line-printer plot of C_p on the cowl surface
- (14) At time step equal to NMAX total output as furnished by HARVIO

2.4 Program Accuracy and Limitations

2.4.1 Accuracy.- The local accuracy of the finite difference scheme that is employed in this program is second-order both in time and space. It should be noted that the method is a conservative difference formulation, therefore its global numerical accuracy is also given by the local accuracy and is second-order in time and space. There are, however, other points that contribute to the accuracy of the numerical solution:

(a) The grid system generated has a very important effect on the overall solution. With a course grid the shocks (discontinuities) are not adequately resolved, and in certain cases the resulting field is totally erroneous. Near the solid walls the grid system must be properly clustered. Grids also should vary smoothly in the flow field.

(b) The overall accuracy also depends on the accuracy at the boundaries. In the present program at the flow field boundaries it is assumed that \hat{q}^{n+1} equals \hat{q}^n . Therefore on the boundaries the formal accuracy is first-order in time. Second order accuracy in space, however, is retained all through the flow field.

2.4.2 Limitations.- The main limitations of the program are

- (a) The flow field analyzed is two-dimensional.

- (b) Owing to the neutral stability criteria, convergence to steady state cannot be accelerated by using large time-steps.
- (c) As currently implemented, the flow field analyzed is inviscid.

3. GENERATION OF NONORTHOGONAL GRIDS FOR TWO-DIMENSIONAL INLET FLOW FIELDS

In this section we present the method that is used to create nonorthogonal computational grids for the calculation of two-dimensional inlet flow fields. The computer program used is an adapted version of that described in detail in reference 2. Therefore here, we briefly present the outline of the mathematical formulation and the structure of the program. Detailed information is given only about the changes that were employed for the present purposes.

3.1 Problem Formulation

The computer program is based on the method developed in references 10 and 11 (TTM-method) to generate two-dimensional computational grids about arbitrary bodies with coordinate lines coincident with all boundaries. A provision is included for clustering about any grid line.

The TTM method solves Laplace equations in the physical plane

$$\xi_{xx} + \xi_{yy} = 0 \quad (3.1)$$

$$\eta_{xx} + \eta_{yy} = 0 \quad (3.2)$$

to generate curvilinear coordinates $\xi(x,y)$ and $\eta(x,y)$, which

form a rectangular grid in the transformed plane. In the computational plane, equations (3.1) and (3.2) transform as

$$ax_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = 0 \quad (3.3)$$

$$ay_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = 0 \quad (3.4)$$

where

$$\alpha = x_{\eta}^2 + y_{\eta}^2, \quad \beta = x_{\xi}x_{\eta} + y_{\xi}y_{\eta} \quad \text{and} \quad \gamma = x_{\xi}^2 + y_{\xi}^2 \quad (3.5)$$

The solution of equations (3.3) and (3.4) for x and y on the rectangular ξ, η computational grid yields the solution for ξ, η in the physical plane if the transformation Jacobian is everywhere nonzero. The most attractive aspect of the TTM method is the ability to arbitrarily locate boundary points by specifying their locations in the x, y plane.

The program has the option of providing grid clustering along the y -direction to increase computational accuracy in regions where large gradients of the flow variables occur. This is especially important in the present problem due to the existence of compression and expansion corners internal and external to the inlet. In this work, it was necessary to have the grid points clustered near the cowl surface and the ramp surface to insure accuracy. To this end the clustering transformation used in reference 12 was employed; according to this transformation

$$z = \alpha + (1 - \alpha) \frac{\ln \left[\frac{\beta + \bar{z}(2\alpha + 1) - 2\alpha}{\beta - \bar{z}(2\alpha + 1) + 2\alpha} \right]}{\ln \left[\frac{\beta + 1}{\beta - 1} \right]} \quad (3.6)$$

permits the mesh to be clustered both near the cowl and the ramp surfaces when $\alpha = 1/2$. Note that in equation (3.6) z is the spacing of constant ξ lines in the computational plane, whereas \bar{z} is the spacing of constant ξ lines in the physical plane. Values of β are chosen for each constant- ξ line from the requirement that $\bar{z}_{\max} = z_{\max}$ using a Newton-Raphson iterative procedure. With β_n defined as the value of β after the n th iteration

$$\beta_n = \beta_{n-1} - \frac{F(\beta_{n-1})}{F'(\beta_{n-1})} \quad (3.7)$$

where

$$F(\beta) = z_{\max} - \bar{z}_{\max} \quad (3.8)$$

Note that this procedure requires an initial guess for β_0 on the first constant ξ -line.

3.2 Overall Program Logic

The execution of the foregoing calculation procedure is performed by PROGRAM MAIN. It starts the calculation by reading input data cards. SUB.INNER is called to distribute points on the rear boundaries and SUB.OUTER locates points on the bottom-front-top boundaries. Note that in our modified program all the bottom-front-top boundaries are read as input in SUB.OUTER. Initial conditions for the TTM are provided by equally spacing grid nodes along straight lines of constant ξ between corresponding boundary points. The TTM method is applied by SUB.RELAX, stretching is applied by SUB.CLUSTR. Newton-Raphson scheme for finding β is applied by SUB.EPSIL, whereas SUB.TAINT, which is a system supplied subroutine, is used for polynomial interpolation of the tabulated values of the x_k, y_k coordinates as functions of z_k . Note that the clustering procedure permits one to discard

the lines of constant η obtained from the TTM method and to recalculate them according to the desired clustering near the boundaries.

Subroutines PLAWT, AXIS, LABEL, PLOT, and TITLE are used to illustrate the grids on the SC4020 plotter. They contain calls to the standard SC4020 software subroutines; thus, the SC4020 software subroutines must be made available to the loader. If the user's computer installation does not include an SC4020, the five subroutines READIN, PLAWT, and EOF-TV should be removed from the main program, and data card 7 should be deleted.

3.3 Program Use and Operation

In this section we give a description of the input information required to run a typical test case. Various subroutines have been modified to accommodate a flexibility for input data for the present calculations. We first list the input cards in order and then summarize the modifications.

Input data is read by PROGRAM MAIN in the following order:

Card No.	Format	Variables
1	3I5	<p>JWAKE: Number of grid points in the x-direction in the wake including the trailing edge point. Set to 1 for the present calculations.</p> <p>KMAX: Number of grid points in the y-direction.</p> <p>MAXIT: Number of iterations to be used in creating the unclustered grid.</p>
2	8F10.0	<p>XGMX: x-direction coordinate of rearward boundary.</p> <p>XGMN: x-direction coordinate of front boundary.</p>

Card No.	Format	Variables
2 (continued)	8F10.0	<p>YMAX: y-direction coordinate of front boundary.</p> <p>YMIN: y-direction coordinate of bottom boundary.</p> <p>XNOSE: x-direction coordinate of leading edge of cowl.</p> <p>XTAIL: x-direction coordinate of end of cowl.</p> <p>XRAMP: x-direction coordinate of leading edge of ramp.</p>
3	F10.0	DY1: Minimum y-increment on rearward boundary for initial conditions.
4	4F10.0	<p>XORG: Location on the x-axis of the special clustering origin.</p> <p>YORG: Location on the y-axis of the special clustering origin.</p> <p>ETAC: Angle (in degrees), about which angular clustering is done.</p> <p>BETA: Parameter determining the strength of angular distribution set to zero. Otherwise values in the range 1 to 5.</p>
5	F10.0	OMEGA: Parameter for Thompson solver values between 0 to 2 recommended.
6	F10.0	DY2: Minimum y-direction spacing for final clustering.
7	A80	A80: Computer card for the title frame of SC4020 plots. To be deleted if the user's

Card No.	Format	Variables
7 (continued)	A80	installation does not include an SC4020.
8	A71	Title on each SC4020 plots re- resulting from each run.
9	A80	Description of the cowl.
10	2F10.0	All cards following card 9 con- tain in columns 1 to 10 and 11 to 20 two floating-point numbers, x and y, respectively, which are coordinates of points on the cowl surface defining the cowl shape. The points should proceed counter- clockwise.

It should be noted that the number of points in the ξ -direction is specified in PROGRAM.MAIN by assigning an integer value to the variable NBOD. We have further modified SUB.OUTER such that it allows the boundary points along the ramp inflow boundary and the outer boundary to be specified point by point. For each different geometry, these boundary points must be prescribed by making the required changes in SUB.OUTER.

This completes of discussions on the overall aspects of the grid generation program that was employed to generate non-orthogonal grids for two-dimensional inlet flow fields. Example calculations to illustrate the use of the program are reported separately (reference 13).

REFERENCES

1. Steger, J. L.: Implicit Finite Difference Simulation of Flow About Arbitrary Geometries with Application to Airfoils, AIAA Paper No. 77-665, June 1977.
2. Sorenson, R. L.; and Steger, J. L.: Simplified Clustering of Nonorthogonal Grids Generated by Elliptic Partial Differential Equations, NASA TM 73252, August 1977.
3. Viviand, H.: Conservative Forms of Gas Dynamic Equations, La Recherche Aerospatiale, No. 1, Jan.-Feb. 1974, pp. 65-68.
4. Lapidus, A.: A Detached Shock Calculation by Second-Order Finite Differences, J. of Comp. Physics, Vol. 2, 1967, pp. 154-177.
5. Vinokur, M.: Conservation Equations of Gas-dynamics in Curvilinear Coordinate Systems, J. of Comp. Physics, Vol. 14, Feb. 1974, pp. 105-125.
6. Beam, R.; and Warming, R. F.: An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation-Law-Form, J. Comp. Physics, Vol. 22, Sept. 1976, pp. 87-110.
7. Beam, R.; and Warming, R. F.: An Implicit Factored Scheme for the Compressible Navier-Stokes Equations, AIAA Paper No. 77-645, June 1977.
8. Steger, J. L.; and Bailey, H. E.: Calculation of Aileron Buzz. AIAA Paper No. 79-0134, Jan. 1979.
9. Steger, J. L.; and Kutler, P. K.: Implicit Finite-Difference Procedures for the Computation of Vortex Waves. AIAA Paper No. 76-385, July 1976.
10. Thompson, J. F.; Thames, F. C.; and Mastin, C. W.: Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies. J. Comp. Phys., Vol. 15, no. 3, July 1974, pp. 299-319.
11. Thames, F. C.; Thompson, J. F.; and Mastin, C. W.: Numerical Solution of the Navier-Stokes Equations for Arbitrary Two-Dimensional Airfoils. NASA SP-347, 1975, Pt. 1, pp. 469-530.

12. Tannehill, J. C.; and Holst, T. L.: Numerical Computation of Two-Dimensional Viscous Blunt Body Flows with an Impinging Shock. AIAA Journal, Vol. 14, No. 2, February 1976, pp. 204-211.
13. Biringen, S. H. and McMillan, O. J.: Calculation of Two-Dimensional Inlet Flow Fields in a Supersonic Free Stream - Program Documentation and Test Cases. NASA CR-3221, 1980.

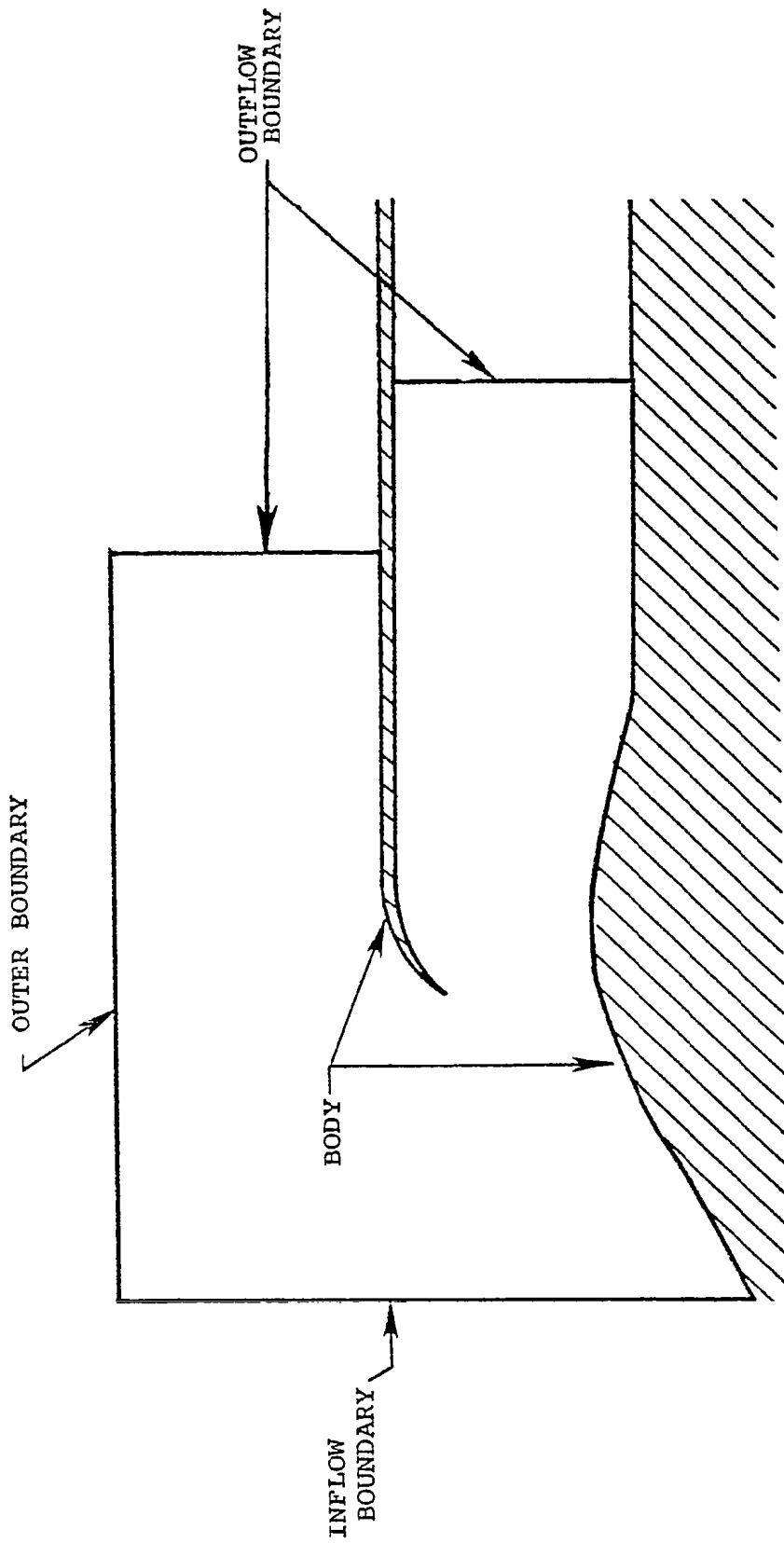


Figure 1.- Computational Domain and Flow Field Boundaries.

1. Report No. NASA CR-3222	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle CALCULATION OF TWO-DIMENSIONAL INLET FLOW FIELDS IN A SUPERSONIC FREE STREAM BY AN IMPLICIT MARCHING CODE WITH NONORTHOGONAL MESH GENERATION - USER'S MANUAL		5. Report Date January 1980	
		6. Performing Organization Code 512/C	
7. Author(s) S. H. Biringen and O. J. McMillan		8. Performing Organization Report No. NEAR TR 193	
		10. Work Unit No.	
9. Performing Organization Name and Address Nielsen Engineering & Research, Inc. 510 Clyde Avenue Mountain View, CA 94043		11. Contract or Grant No. NAS1-15305	
		13. Type of Report and Period Covered Contractor Report 3/1/78 to 4/30/79	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code	
		15. Supplementary Notes Langley Technical Monitors: Wallace C. Sawyer and Charlie M. Jackson, Jr. <p style="text-align: right;">Topical Report</p>	
16. Abstract An implicit, shock-capturing finite-difference code which is used to calculate two-dimensional inlet flow fields in a supersonic free stream is explained. The Euler equations are subjected to general nonorthogonal transformation and a body-fitted coordinate system is employed. The mathematical formulation of the problem is given along with the numerical algorithm. Initial and boundary conditions, numerical stability, program limitations, and accuracy is discussed. An overall program logic as well as instructions for program use and operation are also furnished.			
17. Key Words (Suggested by Author(s)) calculation method inlets supersonic flow inviscid flow		18. Distribution Statement Unlimited - Unclassified Subject Category 02	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 25	22. Price* \$4.00