

AD-A137 935

COMPUTATION OF VISCOUS FLOW ABOUT ADVANCED PROJECTILES
(U) STANFORD UNIV CA DEPT OF AERONAUTICS AND
ASTRONAUTICS J L STEGER 09 SEP 83 SUDAA-CFD-83-2

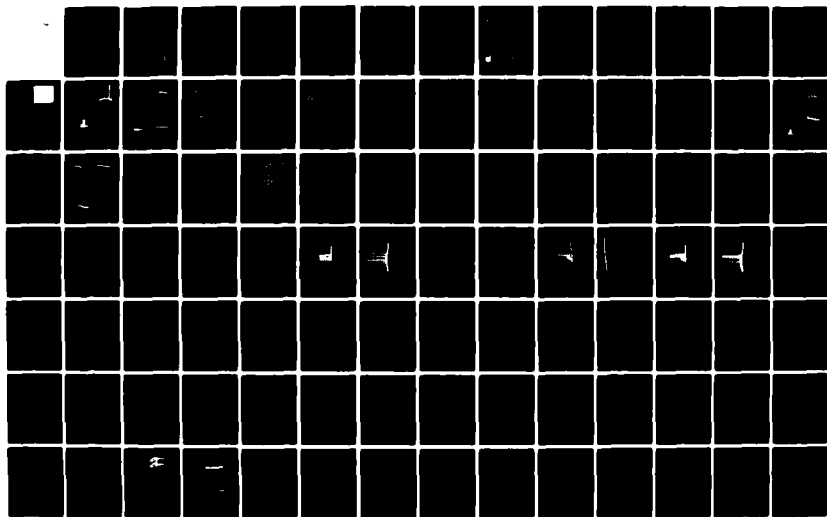
1/2

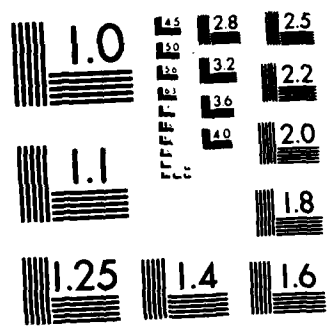
UNCLASSIFIED

ARO-18000.3-EG DAAG29-81-K-0013

F/G 20/4

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

ARO 18000.3-EG

①



**Department of AERONAUTICS and ASTRONAUTICS
STANFORD UNIVERSITY**

AA CFD 83-2

**COMPUTATION OF VISCOUS FLOW
ABOUT ADVANCED PROJECTILES**

Final Report

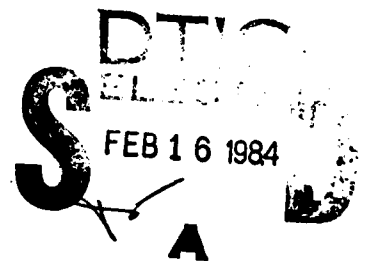
Joseph L. Steger

September 9, 1983

U. S. Army Research Office

Contract DAA 629-81-K-0013, RS-465

**Stanford University
Stanford, California 94305**



APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

DTIC FILE COPY

84 02 16 070

AA CFD 83-2
COMPUTATION OF VISCOUS FLOW
ABOUT ADVANCED PROJECTILES

Final Report

Joseph L. Steger

September 9, 1983

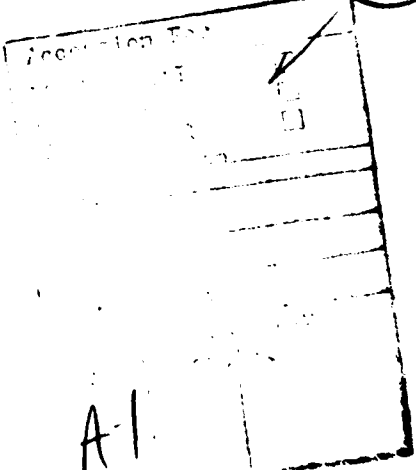
U. S. Army Research Office

Contract DAA 629-81-K-0013, RS-465

Stanford University
Stanford, California 94305

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

Accession For

A-1

I. FOREWARD

The research program has been aimed at developing computational procedures for simulation of flow about projectiles. During the three year effort a variety of ideas were pursued and implemented. As a result this final report will include summaries on numerical algorithm developments, methods of grid generation, and descriptions of flow field solution codes for projectiles in the transonic range. Some five technical papers have been published which describe much of the research effort. These are included in this final report and they constitute the technical content of this report. Some ongoing work which has not yet reached fruition is also described.

The research effort was initially directed toward two tasks: 1) basic algorithm developments; and 2) development of a parabolized Navier-Stokes (PNS) computer code to solve finned projectiles in supersonic viscous flow. The second task was later redirected to the development of computer codes for computing the transonic flow about projectiles with base. This effort was undertaken in collaboration with Messrs. Nietubicz and Sahu of the Ballistics Research Laboratory (BRL). (The original task of computing the supersonic flow about finned projectiles using a PNS code was subsequently accomplished by Mann Mohan Raj, et. al., under joint BRL and NASA Ames Research Center sponsorship.)

The research effort has greatly benefited by a sustained collaboration with researchers at BRL. This collaboration properly focused the work on realistic problems, help stimulate new concepts, and provided necessary stimulate new concepts, and provided necessary critiques.

The report is divided into four main areas. Section II describes the algorithms and methodology for computing transonic flow about projectiles with base. Section III describes some work in grid generation, while Section IV contains a potpourri

of ideas in numerical algorithm development. Ongoing projects initiated, but not concluded, are described in Section V.

II. TRANSONIC PROJECTILES WITH BASE FLOW

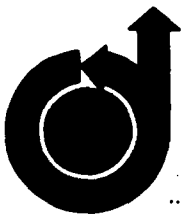
A major accomplishment of the research program has been the development of computer codes to simulate transonic projectiles with base flow. This project was carried out in collaboration with J. Sahu and C. J. Nietubicz at BRL with Sahu being responsible for the code implementation.

A full description of the numerical procedures is given in the two appended AIAA papers. A key feature of the projectile base flow code is its segmentation concept. Beginning with a basic projectile and sting code due to Nietubicz, a simple way of dividing up the computational domain was devised which maintained the simplicity of the implicit numerical algorithm. The sketches shown in Figure 2 of the second attached AIAA paper (83-0224) illustrate the segmentation process. The idea here is that we add grids as needed yet solve the flow field implicitly as one large grid in computational space. Various flags are used to properly turn-off or connect the domains together. This concept can be extended to more complex geometries, and currently, Sahu is treating a projectile with a cut-out in the base by using three grid segments.

AIAA-82-1358

**Numerical Computation of Base Flow for a
Projectile at Transonic Speeds**

J. Sahu and C.J. Nietubicz, Ballistic
Research Lab, Aberdeen Proving Ground,
MD; and J.L. Steger, Stanford Univ.,
Palo Alto, CA



**AIAA 9th Atmospheric Flight
Mechanics Conference**

August 9-11, 1982/San Diego California

**For permission to copy or republish, contact the American Institute of Aeronautics and Astronautics
1290 Avenue of the Americas, New York, NY 10104**

Numerical Computation of Base Flow for a Projectile at Transonic Speeds

Jubaraj Sahu* and Charles J. Nietubicz*
U.S. Army Ballistic Research Laboratory, ARRADCOM,
Aberdeen Proving Ground, MD 21005

and

Joseph L. Steger**
Department of Aeronautics and Astronautics,
Stanford University,
Palo Alto, CA 94305

Abstract

The Azimuthal-Invariant Thin-Layer Navier-Stokes computational technique has been modified for projectile base flow analysis. The resulting new numerical capability is used to compute the entire projectile flow field including the recirculatory base flow. Computed results show the qualitative and quantitative details of the overall base flow structure. Base drag is computed for a secant-ogive-cylinder projectile and compared with limited experimental and semi-empirical data. Results are also presented which show the variation of pressure drag, skin friction drag and the total aerodynamic drag for Mach numbers $.9 < M < 1.2$.

Nomenclature

a	speed of sound
A	cross sectional area at the base
C_{D_b}	base drag coefficient, $2 D_b / \rho_\infty u_\infty^2 A$
c_p	specific heat at constant pressure
C_p	pressure coefficient, $2(p - p_\infty) / \rho_\infty a_\infty^2$
D	body diameter (57.15mm)
D_b	base drag
e	total energy per unit volume / $\rho_\infty a_\infty^2$
$\hat{q}$	vector of dependant variables in transformed equations
$\hat{E}, \hat{F}$	flux vector of transformed Navier-Stokes equations
H	n-invariant source vector
J	Jacobian of transformation
M	Mach number
p	pressure / $\rho_\infty a_\infty^2$
Pr	Prandtl number, $\mu_\infty C_p / \kappa_\infty$
R	body radius
Re	Reynolds number, $\rho_\infty a_\infty D / \mu_\infty$
$\hat{S}$	viscous flux vector
t	physical time
u, v, w	Cartesian velocity components / a
U, V, W	Contravariant velocity components / a
x, y, z	physical Cartesian coordinates
α	angle of attack
γ	ratio of specific heats
κ	coefficient of thermal conductivity
μ	coefficient of viscosity
ξ, η, ζ	transformed coordinates in axial, circumferential and radial directions
ρ	density / ρ_∞
τ	transformed time

ψ circumferential angle

Superscript

* critical value

Subscript

b base
p pressure
v viscous
 ∞ free stream conditions

1. Introduction

The rising costs of experimental measurements has resulted in alternate means of determining the aerodynamics of shells. Because of the recent advances in computer processors, numerical computational capabilities have been developed to predict the aerodynamic behavior of artillery shells.

Recent papers^{1,2} have reported the development and application of the Azimuthal-Invariant Thin-Layer Navier-Stokes computational technique to predict the flow about slender bodies of revolution at transonic speeds. References 1 and 2 showed the technique to be a viable computational tool for predicting both external and internal flows for spinning and non-spinning bodies of various geometric shapes. The base flow of the projectiles however, was not computed. Instead the projectile base was modeled by an extended sting. Experimental base flow data is difficult to obtain and therefore only limited data is available. No sophisticated numerical techniques have yet been utilized for base flow of projectiles at transonic speeds. The objective of this research was to develop a new numerical capability to compute the flow field in the base region of projectiles at transonic speeds and be able to compute the total aerodynamic drag.

The total drag for projectiles can be divided into three components: (1) pressure (wave) drag; (2) viscous (skin friction) drag; and (3) base drag. For a typical shell at $M = .9$, the relative magnitudes of the aerodynamic drag components are: (1) pressure drag, 20%; (2) viscous drag, 30%; and (3) base drag, 50%. In order to predict the total drag for projectiles, computation of the full flow field (including the base flow) must be made. Computation of base flow is especially important at transonic speeds.

*Aerospace Engineer, Member AIAA
**Associate Professor, Member AIAA

The critical aerodynamic behavior of projectiles occurs in the transonic speed regime. This can be attributed to the complex shock structure which exists for the projectiles at transonic speeds. Figure 1 is a spark shadowgraph which shows the shock structure for a typical projectile at $M = .95$, $\alpha = 0$. It also shows a clearly defined wake behind the base of the projectile devoid of any vortex shedding. Primary emphasis is focused on the base region flow field computations; however, the technique used computes the full flow field over the projectile (including the base region). Therefore, all three components of the drag are computed.

A brief description of the governing equations and the method of solution are given in Sections II and III. A unique flow field segmentation procedure and the implementation of boundary conditions are discussed in Section III. In Section IV computed results are given for transonic flow about a 6-caliber secant-ogive-cylinder shape for $.9 < M < 1.2$, $\alpha = 0$. Velocity vector plots and stream function contour plots are presented to show the qualitative features of the flow field in the base region. All three components of drag are obtained. Base drag is compared with experimental and semi-empirical data while the total drag is compared with the only available semi-empirical data. The encouraging results show that the present computational technique can be successfully used to predict the base region flow field of projectiles. Although results here are reported for transonic speeds, future computational efforts will be directed at supersonic velocities.

II. Governing Equations

The Azimuthal Invariant (or Generalized Axisymmetric) thin-layer Navier-Stokes equations for general spatial coordinates ξ , η , ζ can be written as¹

$$\partial_{\tau} \hat{q} + \partial_{\xi} \hat{E} + \partial_{\zeta} \hat{G} + \hat{H} = \text{Re}^{-1} \partial_{\zeta} \hat{S} \quad (1)$$

where $\xi = \xi(x, y, z, t)$ is the longitudinal coordinate

$\eta = \eta(y, z, t)$ is the circumferential coordinate

$\zeta = \zeta(x, y, z, t)$ is the near normal coordinate

$\tau = t$ is the time

The notation for the physical coordinates x , y , z , and the transformed coordinates ξ , η , ζ are shown in Figure 2. The vector of dependent variables $\hat{q}$ and the flux vectors $\hat{E}$, $\hat{G}$, $\hat{H}$ are given as

$$\hat{q} = J^{-1} \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}, \quad \hat{E} = \begin{bmatrix} \rho u \\ \rho u^2 + \xi_x p \\ \rho v u + \xi_y p \\ \rho w u + \xi_z p \\ (e+p)u - \xi_t p \end{bmatrix}$$

$$\hat{G} = J^{-1} \begin{bmatrix} \rho w \\ \rho v w + \zeta_y p \\ \rho w^2 + \zeta_z p \\ (e+p)w - \zeta_t p \end{bmatrix}$$

$$\hat{H} = J^{-1} \begin{bmatrix} 0 \\ 0 \\ \rho V [R_{\xi} (U - \xi_t) + R_{\zeta} (W - \zeta_t)] \\ -\rho V R_{\eta} (V - \eta_t) - p / (R \phi_{\eta}) \\ 0 \end{bmatrix}$$

The thin layer viscous terms valid for high Reynolds number flow are contained in the vector $\hat{S}$, where

$$\hat{S} = \begin{bmatrix} 0 \\ \mu (\zeta_x^2 + \zeta_y^2 + \zeta_z^2) u_{\zeta} + (\mu/3) (\zeta_x u_{\zeta} + \zeta_y v_{\zeta} + \zeta_z w_{\zeta}) \zeta_x \\ \mu (\zeta_x^2 + \zeta_y^2 + \zeta_z^2) v_{\zeta} + (\mu/3) (\zeta_x u_{\zeta} + \zeta_y v_{\zeta} + \zeta_z w_{\zeta}) \zeta_y \\ \mu (\zeta_x^2 + \zeta_y^2 + \zeta_z^2) w_{\zeta} + (\mu/3) (\zeta_x u_{\zeta} + \zeta_y v_{\zeta} + \zeta_z w_{\zeta}) \zeta_z \\ \{ (\zeta_x^2 + \zeta_y^2 + \zeta_z^2) [0.5 \mu (u^2 + v^2 + w^2) + \kappa \text{Pr}^{-1}] \\ (\gamma-1)^{-1} (a^2)_{\zeta} \} + (\mu/3) (\zeta_x u + \zeta_y v + \zeta_z w) \\ (\zeta_x u_{\zeta} + \zeta_y v_{\zeta} + \zeta_z w_{\zeta}) \} \end{bmatrix}$$

The velocities

$$\begin{aligned} U &= \xi_t + \xi_x u + \xi_y v + \xi_z w \\ V &= \eta_t + \eta_x u + \eta_y v + \eta_z w \\ W &= \zeta_t + \zeta_x u + \zeta_y v + \zeta_z w \end{aligned} \quad (2)$$

represent the contravariant velocity components.

The Cartesian velocity components (u, v, w) are nondimensionalized with respect to a_{∞} (the free stream speed of sound). The density (ρ) is referenced to ρ_{∞} and total energy (e) to $\rho_{\infty} a_{\infty}^2$. The local pressure is determined using the equation of state,

$$p = (\gamma-1)[e - 0.5\rho(u^2 + v^2 + w^2)] \quad (3)$$

where γ is the ratio of specific heats.

In high Reynolds number flows the thin-layer approximation is often used because, due to computer speed and storage limitations, fine grid spacing can only be provided in one coordinate direction. The grid spacing available in other directions is usually too coarse to resolve the

viscous terms. Essentially, all the viscous terms in the coordinate direction ξ and n are neglected while terms in the near normal direction to the body ζ are retained. The thin-layer generalized axisymmetric equations (1) are obtained from the three dimensional equations by making use of two restrictions: (i) all body geometries are of an axisymmetric type; and (ii) the state variables and the contravariant velocities do not vary in the circumferential direction (n). Essentially, the n -derivative term in the three dimensional equations is replaced by a source term $\hat{H}$ as it appears in equation (1). The details can be found in Reference 1 and 2.

Equation (1) contains only two spatial derivatives; however it retains all three momentum equations thus allowing a degree of generality over the standard axisymmetric equations. In particular, the circumferential velocity is not assumed to be zero thus allowing computations for spinning projectiles or swirl flow to be accomplished.

For the computation of turbulent flows a turbulence model must be supplied. In the present calculations a Cebeci-type two layer algebraic eddy viscosity model as modified by Baldwin and Lomax³ is used. In their two layer model the inner region follows the Prandtl-Van Driest formulation. Their outer formulation can be used in wakes as well as in attached and separated boundary layers. In both the inner and outer formulations the distribution of vorticity is used to determine length scales thereby avoiding the necessity of finding the outer edge of the boundary layer (or wake). The magnitude of the local vorticity for the axisymmetric formulation is given by

$$|\omega| = \sqrt{\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial z} - \frac{\partial w}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial x} - \frac{\partial u}{\partial z}\right)^2} \quad (4)$$

It should be noted that the turbulence model has not been tailored for use in base flow regions. Moreover, the no slip boundary condition is not applied at the projectile base and slip is allowed along the base (inviscid boundary condition). The velocity component normal to the base is however set to zero.

III. Numerical Method

a. Computational Algorithm

An implicit approximate factorization finite-difference scheme in delta form is used as described by Beam and Warming⁴. An implicit method was chosen because it permits a time step much greater than that allowed by explicit schemes. For problems in which the transient solution is of no interest, this offers the possible advantage of being able to reach the steady state solution faster than existing explicit schemes.

The Beam-Warming implicit algorithm has been used in various applications¹⁻⁶. The algorithm can be first or second order accurate in time and second or fourth order accurate in space. The equations are factored (spatially split) which

reduces the solution process to one-dimensional problems at a given time level. Central difference operators are employed and the algorithm produces block tridiagonal systems for each space coordinate. The main computational work is contained in the solution of these block tridiagonal systems of equations.

b. Finite Difference Equations

The resulting finite difference equations, written in delta form are

$$\begin{aligned} & (1 + h\delta_\xi \hat{A}^n - \epsilon_1 J^{-1} V_\xi \Delta_\xi J) (1 + h\delta_\xi \hat{C}^n - \epsilon_1 J^{-1} V_\xi \Delta_\xi J) \\ & \cdot h R E^{-1} \delta_\xi J^{-1} \hat{M}^n J \times (q^{n+1} - q^n) = -\Delta t (\delta_\xi \hat{E}^n + \delta_\xi \hat{G}^n) \quad (5) \\ & - R E^{-1} \delta_\xi \hat{S}^n - \Delta t \hat{H}^n - \epsilon_E J^{-1} [(V_\xi \Delta_\xi)^2 + (V_\zeta \Delta_\zeta)^2] J q^n \end{aligned}$$

Here $h = \Delta t$ because only first order accuracy in the time differencing is needed for the steady state flows which are considered here. This choice corresponds to the Euler implicit time differencing. The δ 's represent central difference operators, Δ and V are forward and backward difference operators respectively. The Jacobian

matrices $\hat{A} = \frac{\partial \hat{E}}{\partial q}$, $\hat{C} = \frac{\partial \hat{G}}{\partial q}$ along with the coefficient

matrix $\hat{M}$ obtained from the local time linearization of $\hat{S}$ are described in detail in Reference 6. Fourth order explicit (ϵ_E) and implicit (ϵ_I) numerical dissipation terms are incorporated into the differencing scheme to damp high frequency growth and thus to control the nonlinear instabilities. A typical range for the smoothing coefficients is $\epsilon_E = (1 \text{ to } 5) \Delta t$ with $\epsilon_I = 3\epsilon_E$.

c. Flow Field Segmentation

The objective is to compute the full flow field (including the base region) of a projectile at transonic speeds. Figure 3 shows a schematic illustration of the flow field segmentation used in this study for computational purposes. The hatched region represents the projectile. The region ABCD includes the projectile base and the wake and will be referred to as the base region.

The curvilinear coordinates used for the longitudinal and normal directions are represented by their indices J and L . The line $J = 1$ starts at the downstream boundary (line CD) in the base region. J is incremented until the line $J = JB$ is reached which represents the base of the projectile. The line $J = JB+1$ is at the nose of the projectile and J is then incremented until the line $J = JMAX$ is reached which is the downstream boundary in the outer region.

As for the other coordinate L , in the base region $L = 1$ starts at line AC which is a computational cut through the physical wake region. L is incremented until $L = LMAX$ which is the line of symmetry (line AB). In the outer region $L = 1$ starts out from the projectile surface and L is

incremented all the way to the outer boundary where $L = LMAX$.

Implicit integration is carried out from $J = 2$ to $J = JB-1$ and from $J = JB+1$ to $J = JMAX-1$ in the longitudinal direction and from $L = 2$ to $L = LMAX-1$ in the normal direction.

d. Implementation of Boundary Conditions

The no slip boundary conditions for viscous flow is enforced by setting

$$U = V = W = 0 \quad (6)$$

on the projectile surface except for the base. At the projectile base ($J = JB$) the velocity component normal to the base is set to zero, i.e. $U = 0$, while other flow variables are set to be equal to those at $J = JB-1$. In other words, slip is allowed along the base (inviscid boundary condition). Future work will be directed at the implementation of viscous boundary condition at the base to further access this approximation.

Care must be taken in the implementation of the boundary conditions along line AC which is the computational cut. After trial and error the flow variables above and below the cut were simply averaged to determine the boundary conditions on the cut. This procedure proved to work well. On the centerline of the wake region, a symmetry condition is imposed.

$$\begin{aligned} \frac{\partial u}{\partial z} &= 0 \\ \frac{\partial v}{\partial z} &= 0 \\ w &= 0 \\ \frac{\partial p}{\partial z} &= 0 \end{aligned} \quad (7)$$

Free stream conditions are used at the outer boundary. Simple extrapolation for all flow variables is used at the downstream boundary (lines $J = 1, JMAX$). During transient calculations, use of a specified outflow pressure can give rise to numerical oscillations in the base region flow field. Eventually, these grow and swamp the solution. This difficulty is avoided by simply extrapolating pressure to the downstream boundary which is the procedure always used with supersonic outflow. A combination of extrapolation and symmetry is used at $J = JB+1$.

As a result of the flow field segmentation procedure described in Section III b, the block tridiagonal matrix in the ξ direction has elements at $J = JB, JB+1$ which are treated as internal boundaries in the computational domain. The block tridiagonal matrix in the ξ direction takes the following form (after setting $c_1 = 0$ to simplify the illustration)

$$\begin{bmatrix} 1 & A_3 & & & \\ -A_2 & 1 & A_4 & & \\ & & & & \\ & & & & \\ -A_{JB-2} & 1 & A_{JB} & & \\ & 0 & 1 & 0 & \\ & & 0 & 1 & 0 \\ & & & & \\ -A_{JB+1} & 1 & A_{JB+3} & & \\ & & & & \\ & & & & \\ -A_{JMAX-2} & 1 & & & \end{bmatrix} \begin{bmatrix} \Delta q_2 \\ \Delta q_3 \\ \vdots \\ \Delta q_{JB} \\ \Delta q_{JB+1} \\ \vdots \\ \Delta q_{JMAX-1} \end{bmatrix} = \begin{bmatrix} RHS_2 \\ RHS_3 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ RHS_{JMAX-1} \end{bmatrix} \quad (8)$$

Here A's denote the quantity $\frac{\Delta t}{2\Delta \xi} \hat{A}$ and I is a 5×5 identity matrix. Note the appearance of two uncoupled block tridiagonal matrices. The rows at JB and $JB+1$ are particularly simple because boundary conditions are updated explicitly at the end of inversions. These changes were easily implemented in a modular fashion into an existing code for projectile base flow computations. One simply fills the block tridiagonal matrix ignoring the base JB and the nose axis $JB+1$. Elements in these rows are then overloaded as shown above. The flow field segmentation does not affect the block tridiagonal matrix in the ξ direction.

IV. Results

A series of computations have been made for the 3 caliber (1 caliber = 1 max. body diameter) secant-ogive nose and 3 caliber cylinder shape shown in Figure 4. All the computations were obtained for Mach numbers $.9 < M < 1.2$ and $\alpha = 0$. Limited experimental base pressure measurements have been made by Kayser⁷ for this projectile shape and compared with the computed results. The projectile base was supported by a sting attached to it and measurements of base pressure were made at only one location along the base. These experiments were conducted at Langley Research Center 8-foot Transonic Pressure Tunnel. Computational base pressure results are also compared with available semi-empirical⁸ data. The results are presented in the form of surface pressure distribution, contour plots and velocity vector plots.

The computational grid used for the numerical computations was obtained from a versatile grid generator developed in Reference 9. This program allows arbitrary grid point clustering, thus enabling grid points for the projectile shapes to be clustered in the vicinity of the body surface. The grid consists of 108 points in the longitudinal direction and 50 points in the radial direction. The full grid is shown in Figure 5 while Figure 6 shows an expanded view of the grid in the vicinity of the projectile. The computational domain extended to 4 body lengths in front, 4 body lengths in the radial direction and 4 body lengths behind the base of the projectile. The grid

points in the normal direction were exponentially stretched away from the surface with the minimum spacing at the wall of .00002D. This spacing locates at least two points within the laminar sublayer.

The grid shown in Figure 6 was generated in two segments. First, the grid in the outer region is obtained using an elliptic solver⁹ for the ogive portion and straight-line rays for the remaining portion which runs all the way to downstream boundary. Second, the grid in the base region is obtained simply by extending the straight lines perpendicular to line AC down to the center line of symmetry (line BD). An exponential stretching with the minimum spacing of .00002D at line AC is used. It should be noted that the same minimum spacing .00002D is specified on both sides of the cut (line AC) thus maintaining a smooth variation of grid across the cut. This spacing could, of course, be increased downstream of the base. The number of grid points above and below line AC is the same (50 points) which means that an adequate number of points are located in the base region. As can be seen in Figure 6, the grid points are clustered near the nose-cylinder junction and at the projectile base where appreciable changes in flow variables are expected.

The free stream Reynolds number for the series of computations was fixed at 4.5×10^6 based on the total model length. The computations are started from free stream conditions and marched in time to obtain the steady state solution. The initial calculation was made for $M = 0.9$. Previous converged solutions were then used as starting conditions for additional Mach number runs to achieve faster convergence.

Figures 7 and 8 show the distribution of the surface pressure coefficient, C_p , as a function of axial position, x/D . Figure 8 shows the overall view whereas Figure 7 shows the distribution in the near wake region of the base. The distribution over the projectile surface itself is shown in both these figures. The value of C_p beyond $x/D = 6$ is the value of pressure coefficient along the cut AC. Both these figures indicate the shock waves near the nose-cylinder junction and near the blunt base. Although not shown in these figures, the pressure along the base remains fairly constant (within ± 0.005 variation in C_p values).

The series of Figures 9, 10 and 11 show the velocity vector field in the base region for $M = 0.9$ and $\alpha = 0$. Each vector shows the magnitude and the direction of the velocity at that point. Figure 9 shows the velocity field in the entire base region. One can see the expected velocity defect in the far wake region. Figures 10 and 11 show the velocity field in the vicinity of the base (near wake region). The difference between these two plots being that the former one is stretched (not drawn to same scale) while the latter is drawn with the same scale in x and y directions. Both Figures clearly show the recirculation region of flow in the base region and indicate a strong shear layer as well.

The next two Figures 12 and 13 are stream function contour plots in the near wake region, again for $M = 0.9$ and $\alpha = 0$. Figure 13 is drawn

to the same scale in x and y while Figure 12 is not. However, both of these figures are drawn to show the recirculation region and the position of the dividing stream line as clearly as possible. They also show the reattachment point which for this case is about 2 calibers down from the base.

A more critical check of the computational results is presented in Figure 14 where the base drag is plotted as a function of Mach number. Computational results are indicated by circles, experimental results⁷ by triangles and the squares indicate the results obtained using a semi-empirical technique developed by McCoy⁸. Base drag, as expected, increases as the Mach number increases from 0.9 to 1.2. The semi-empirical technique shows generally higher base drag when compared with computational and experimental results. The computational results predict the expected drag rise that occurs for $0.9 < M < 1.2$. The computational results, however, indicate a greater increase in drag than predicted by either the semi-empirical code or the experimental measurements. The discrepancy between the numerical and the experimental results can partly be attributed to the fact that the experimental data was obtained with a sting attached to the base. The sting has an effect of weakening the recirculatory flow in the base region and leads to higher base pressure and hence, lower base drag.

Figures 15, 16 and 17 show the variation of pressure drag, skin friction drag and the total drag with Mach number respectively. The rise in the pressure drag with Mach number is predicted correctly. Skin friction drag decreases as Mach number increases. The total drag, as expected, increases as Mach number increases from 0.9 to 1.2. The computational results are compared with the results obtained by semi-empirical technique and the agreement is considered good.

V. Summary

A procedure has been described in which the Azimuthal-Invariant (generalized axisymmetric) thin-layer Navier-Stokes code is modified in such a way as to compute the base flow field of projectiles at transonic speeds.

The computed results show the qualitative features of the flow field in the base region, namely the recirculation region, dividing stream line, reattachment point, etc. Quantitative comparisons of the base drag have been made with other available data for various Mach numbers in the transonic speed range. These results indicate that the present numerical technique can be used successfully to predict the base drag of projectiles at transonic speeds.

The computed results for this paper represent the first application of thin-layer Navier-Stokes computational technique to predict projectile base flow at transonic velocity using the flow field segmentation described above. The results indicate that this technique shows good promise of providing a useful new computational capability for exterior ballistics of shells.

Future computational efforts will investigate the implementation of viscous boundary condition

on the projectile base, improved grid resolution, and alternate turbulence models.

References

1. Nietubicz, C.J., Pulliam, T.H., and Steger, J.L., "Numerical Solution of the Azimuthal-Invariant Thin-Layer Navier-Stokes Equations", ARBRL-TR-02227, U.S. Army Ballistic Research Laboratory, ARRA DCOM, Aberdeen Proving Ground, MD 21005, March 1980.
2. Nietubicz, C.J., "Navier-Stokes Computations for Conventional and Hollow Projectile Shapes at Transonic Velocities", AIAA Paper No. 81-1262, June 1981.
3. Baldwin, B.S., and Lomax, H., "Thin-Layer Approximation and Algebraic Model for Separated Turbulent Flows", AIAA Paper No. 78-257, 1978.
4. Beam, R., and Warming, R.F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations", AIAA Paper No. 77-645, June 1977.
5. Steger, J.L., "Implicit Finite Difference Simulation of Flow About Arbitrary Geometries with Application to Airfoils", AIAA Journal, Vol 16, No. 4, July 1978, pp. 679-686.
6. Pulliam, T.H., and Steger, J.L., "On Implicit Finite-Difference Simulations of Three-Dimensional Flow," AIAA Journal, Vol. 18, No. 2, February 1980, pp. 159-167.
7. Kayser, L.D., "private communications", Ballistic Research Laboratory, Aberdeen Proving Ground, MD 21005.
8. McCoy, R.L., "McDrag - A Computer Program for Estimating the Drag Coefficients of Projectiles", ARBRL-TR-02293, U.S. Army Ballistic Research Laboratory, ARRA DCOM, Aberdeen Proving Ground, MD 21005, February 1981.
9. Steger, J.L., Nietubicz, C.J., and Heavey, K.R., "A General Curvilinear Grid Generation Program for Projectile Configurations", ARBRL-MR-03142, U.S. Army Ballistic Research Laboratory, ARRA DCOM, Aberdeen Proving Ground, MD 21005, October 1981.
10. Kayser, L.D., and Whiton, F., "Surface Pressure Measurements on a Boattailed Projectile Shape at Transonic Speeds", ARBRL-MR-03161, U.S. Army Ballistic Research Laboratory/ARRA DCOM, Aberdeen Proving Ground, MD 21005, March 1982.



Figure 1. Spark Shadowgraph of Secant-Ogive-Cylinder-Boattail Projectile, $M = 0.95$, $\alpha = 0.0$

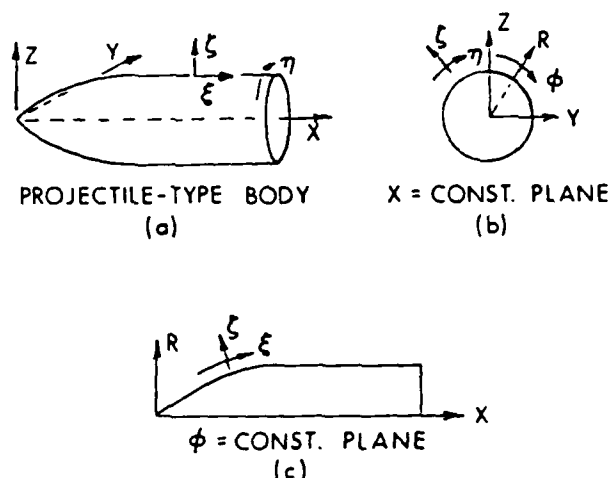


Figure 2. Axisymmetric Body and Coordinate System

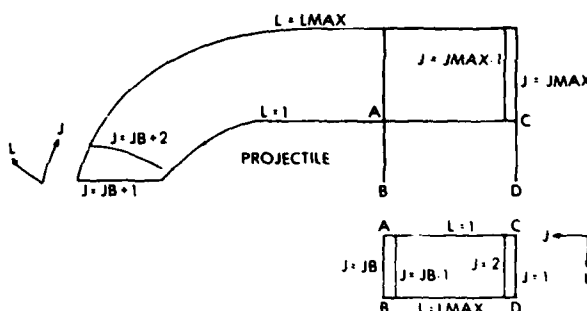


Figure 3. Schematic Illustration of Flow Field Segmentation

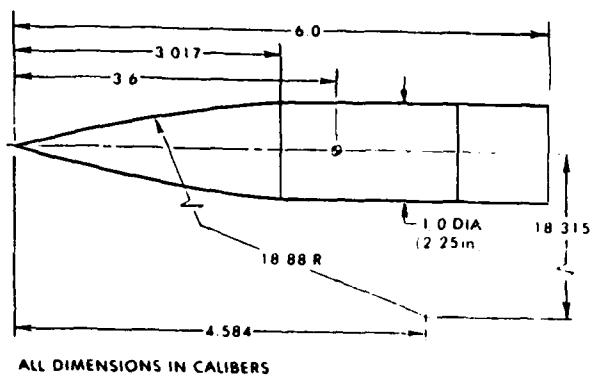


Figure 4. Model Geometry

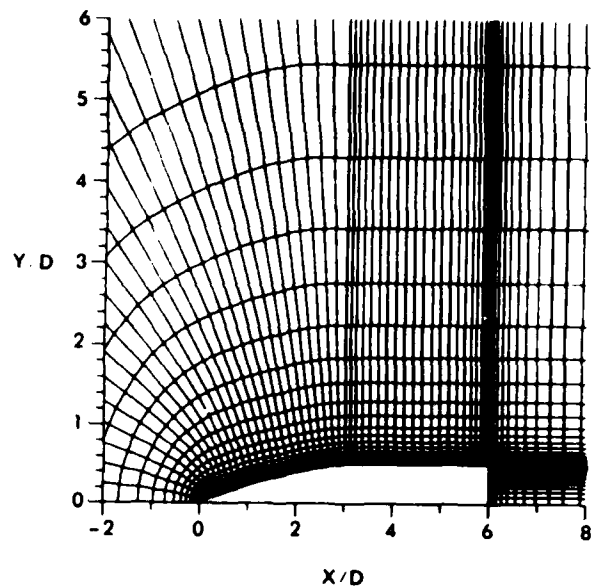


Figure 6. Expanded Grid

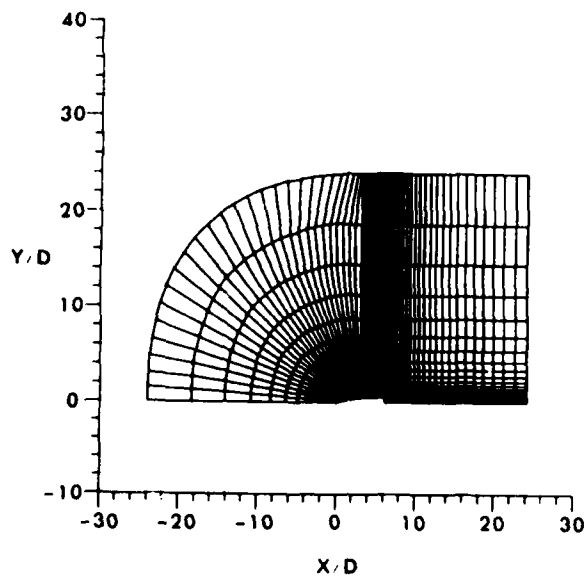


Figure 5. Computational Grid for Flow Field Computations

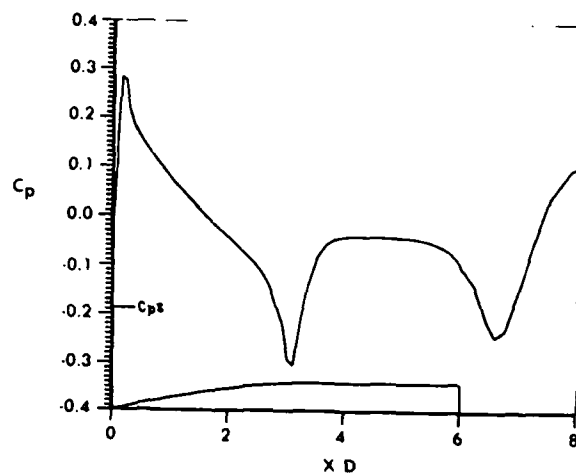


Figure 7. Longitudinal Surface Pressure Distribution, $M = 0.9$, $\alpha = 0$

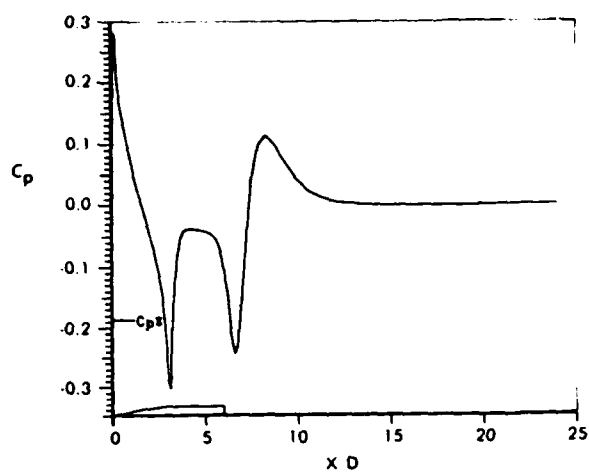


Figure 8. Longitudinal Surface Pressure Distribution, $M = 0.9$, $\alpha = 0$

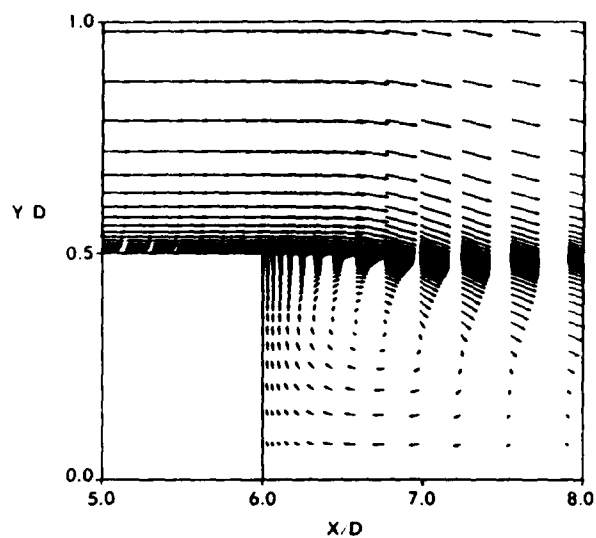


Figure 10. Velocity Vector Field, $M = 0.9$, $\alpha = 0$

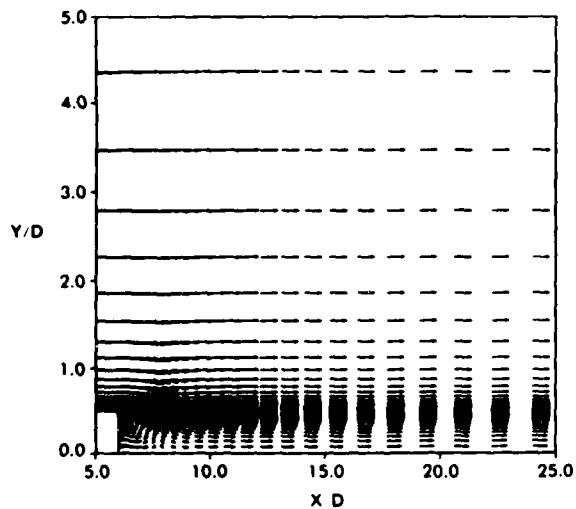


Figure 9. Velocity Vector Field, $M = 0.9$, $\alpha = 0$

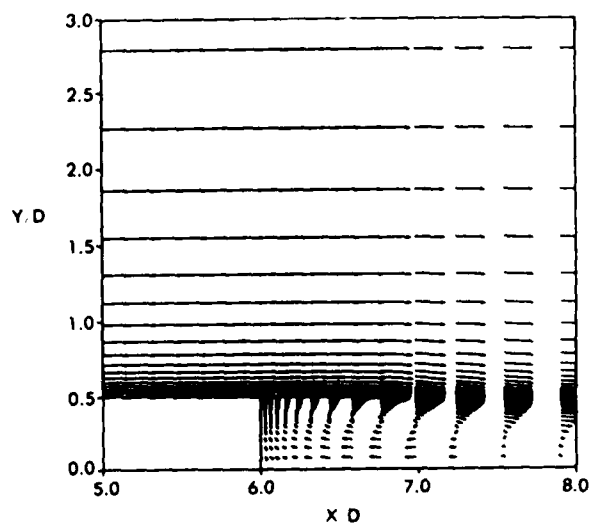


Figure 11. Velocity Vector Field, $M = 0.9$, $\alpha = 0$

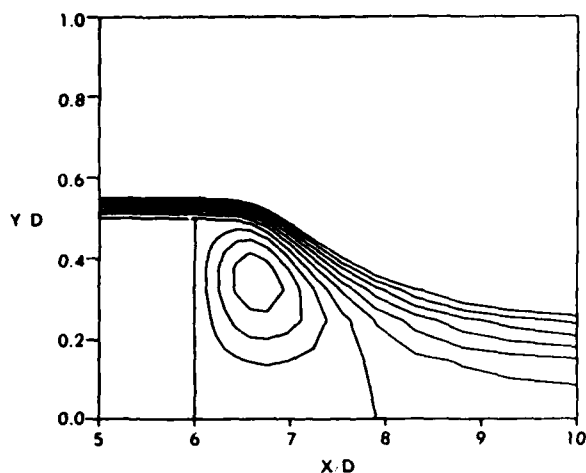


Figure 12. Stream Function Contours, $M = 0.9$, $\alpha = 0$

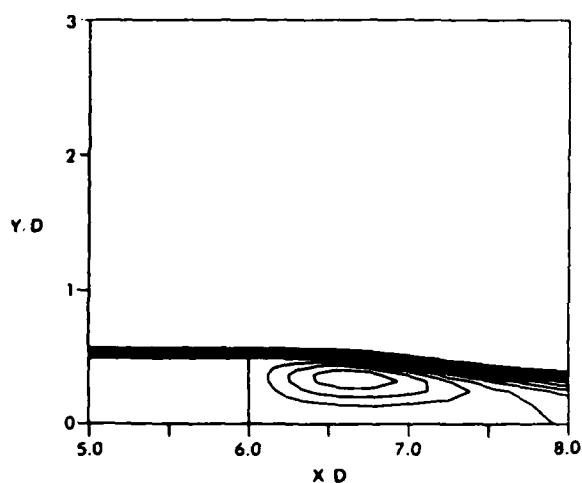


Figure 13. Stream Function Contours, $M = 0.9$, $\alpha = 0$

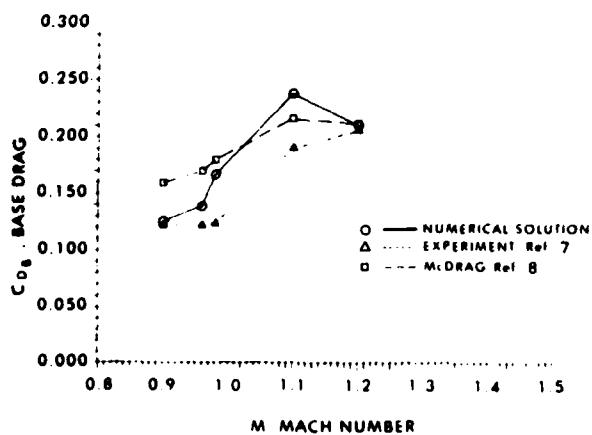


Figure 14. Variation of Base Drag Coefficient with Mach Number, $\alpha = 0$

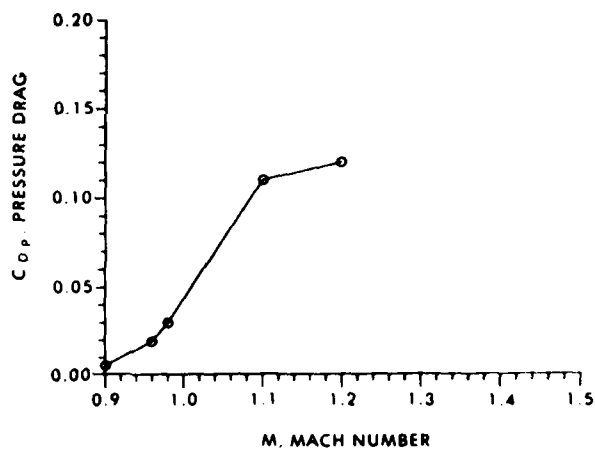


Figure 15. Variation of Pressure Drag Coefficient with Mach Number, $\alpha = 0$

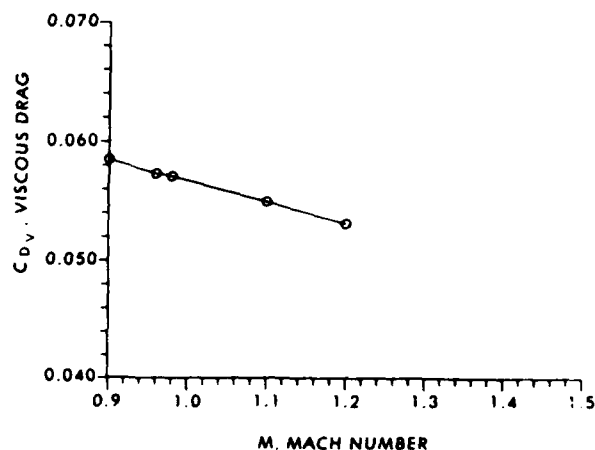


Figure 16. Variation of Viscous Drag Coefficient with Mach Number, $\alpha = 0$

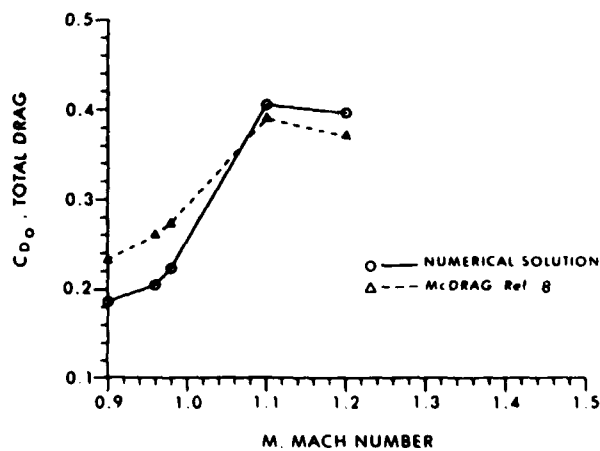


Figure 17. Variation of Total Drag Coefficient with Mach Number, $\alpha = 0$

.. NOTES ..

AIAA'83

AIAA-83-0224

**Navier-Stokes Computations of Projectile Base
Flow with and without Base Injection**

J. Sahu and C.J. Nietubicz, U.S. Army
Ballistic Research Lab., Aberdeen Proving
Ground, MD; and J.L. Steger, Stanford
Univ., Stanford, CA

AIAA 21st Aerospace Sciences Meeting

January 10-13, 1983/Reno, Nevada

For permission to copy or republish, contact the American Institute of Aeronautics and Astronautics
1290 Avenue of the Americas, New York, NY 10104

Navier-Stokes Computations of Projectile Base Flow with and without Base Injection

Jubaraj Sahu* and Charles J. Nietubicz*
U.S. Army Ballistic Research Laboratory, ARRADCOM,
Aberdeen Proving Ground, MD 21005

and

Joseph L. Steger**
Department of Aeronautics and Astronautics,
Stanford University,
Palo Alto, CA 94305

Abstract

A computational capability has been developed for predicting the flow field about the entire projectile, including the recirculatory base flow, at transonic speeds. Additionally, the computer code allows mass injection at the projectile base and hence is used to show the effects of base bleed on base drag. Computations have been made for a secant-ogive-cylinder projectile for a series of Mach numbers in the transonic flow regime. Computed results show the qualitative and quantitative nature of base flow with and without base bleed. The reduction in base drag with base bleed is clearly predicted for various mass injection rates and for Mach numbers $.9 < M < 1.2$. The encouraging results obtained indicate that this computational technique may provide useful design guidance for shells with base bleed.

Nomenclature

a	speed of sound
A	cross sectional area at the base
A_j	injection area for base bleed
C_{D_b}	base drag coefficient, $2 D_b / \rho_\infty u_\infty^2 A$
C_p	specific heat at constant pressure
C_p	pressure coefficient, $2(p - p_\infty) / \rho_\infty u_\infty^2$
D	body diameter (57.15mm)
D_b	base drag
e	total energy per unit volume / $\rho_\infty a_\infty^2$
$\hat{E}, \hat{F}, \hat{q}$	flux vector of transformed Navier-Stokes equations
$\hat{H}$	n-invariant source vector
I	mass injection parameter, $\dot{m}_j / \rho_\infty u_\infty A$
J	Jacobian of transformation
$\dot{m}_j$	mass flow rate for air injection at the base, $\rho_j u_j A_j$
M	Mach number
p	pressure / $\rho_\infty a_\infty^2$
Pr	Prandtl number, $\mu_\infty C_p / \kappa_\infty$
R	body radius
Re	Reynolds number, $\rho_\infty a_\infty D / \mu_\infty$
$\hat{S}$	viscous flux vector
t	physical time
u, v, w	Cartesian velocity components / a_∞
U, V, W	Contravariant velocity components / a_∞
x, y, z	physical Cartesian coordinates

α	angle of attack
γ	ratio of specific heats
κ	coefficient of thermal conductivity
μ	coefficient of viscosity
ξ, η, ζ	transformed coordinates in axial, circumferential and radial directions
ρ	density / ρ_∞
τ	transformed time
ϕ	circumferential angle

Superscript

*	critical value
---	----------------

Subscript

b	base
j	jet conditions
J	longitudinal direction
L	normal direction
o	total conditions
st	stagnation conditions
∞	free stream conditions

I. Introduction

A major area of concern in shell design is in the total aerodynamic drag. The designer, ever desirous of increasing the range and/or terminal velocity of projectiles, is eager to decrease the aerodynamic drag.

The total drag for projectiles can be divided into three components: (1) pressure (wave) drag; (2) viscous (skin friction) drag; and (3) base drag. For a typical shell at $M = .90$ the relative magnitudes of the aerodynamic drag components are: (1) pressure drag, 20%; (2) viscous drag, 30%, and (3) base drag, 50%. The pressure and viscous components generally cannot be reduced significantly without adversely affecting the stability of shell. Recent attempts to reduce the total drag are therefore directed at reducing the base drag.

A number of studies have been made to examine the drag reduction due to the addition of a boat-tail. Although this is very effective in reducing the drag, it has a negative impact on the aerodynamic stability of shell especially at transonic velocities. An excellent review of base drag and the effect of boattailing is presented in Reference 1.

*Aerospace Engineer, Member AIAA

**Associate Professor, Member AIAA

Another effective means of reducing the base drag is that of 'base bleed' or 'base injection'. In this method a small amount of mass is injected into the base region which increases the base pressure and thus reduces the base drag. Recent range and precision tests² of a 155mm projectile with and without base bleed have been conducted and an 85% reduction in base drag was obtained. Presently the XM864 is an active projectile design which is attempting to use the base bleed concept for increased range. This concept of mass injection at the projectile base has been widely studied for supersonic flows and much of the work has been reported in Reference 3. One limited study at supersonic speeds was made at BRL and the results were reported by Dickinson⁴. A limited study made in the transonic flow regime has been reported in Reference 5. The supersonic regime has typically been the area where increased range due to drag reduction has been studied. Thus, only limited attention has been focussed on the 'base bleed' problem in transonic flow.

Most of the work using the 'base bleed' concept has been either experimental or semi-empirical in nature. Sophisticated numerical techniques have not yet been utilized to predict the effects of base bleed on the base drag reduction. Limited computational work has been reported recently by Sullins, et al⁶. Their work dealt with the numerical computation of the base region flow of a supersonic combustion ramjet engine using two-dimensional Navier-Stokes equations. They computed the flow field in the vicinity of the base with parallel gas injection and established the effect of base injection on such flows. Because of the recent advances in computer technology, numerical computational capabilities have been developed to predict the aerodynamic behavior of artillery shells. Recent papers^{7,8} have reported the development and application of the Azimuthal-Invariant Thin-Layer Navier-Stokes computational technique to predict the flow about slender bodies of revolution at transonic speeds. This technique has been modified for base flow analysis and the resulting new numerical capability⁹ is used here to predict the base pressure of shell at transonic speeds including the effect of base bleed. Computed results show the quantitative and qualitative details of the base flow structure. The technique used computes the full flow field over the projectile at transonic speeds; therefore, all three components of the total drag (pressure, viscous, and base drag) are computed. This computational technique is then applied to predict the effects of base bleed on the base drag reduction at transonic speeds.

A brief description of the physical problem and the governing equations is given in Sections II and III. The computational technique and the method of solution are discussed in Section IV. In Section V results are shown for transonic base pressure computations for a 6-caliber secant-ogive-cylinder shape for $.9 < M < 1.2$ with and without base bleed. Velocity vector plots, stream function contour plots and density contour plots are presented to show the qualitative features of the flow field in the base region. Quantitative comparisons of base drag and the total drag both with and without base injection have been made. The encouraging results show that the present computational technique can be used to study the effects of base bleed on base drag and can possi-

bly have a positive impact on the XM864 development. Although results here are presented for transonic speeds, current computational efforts are directed at supersonic velocities.

II. Physical Problem

The physical problem deals with the transonic flow over a projectile shape including the base region. Although the entire projectile flow is computed, the emphasis here is on the flow field in the base region of the projectile. A small amount of air is injected at the projectile base in the direction parallel to the primary flow. The injection at the base can be concentrated at the center of the base or spread throughout the entire base. In the present work, however, the injection takes place over 90% of the base. Figure 1 shows a schematic illustration of the base region flow field with base injection. The dividing streamline separates the recirculatory base flow from the primary external flow. The flow field is dominated by separation and mixed regions of locally supersonic and subsonic flows.

The complete set of time-dependent generalized axisymmetric thin-layer Navier-Stokes equations is solved to obtain a numerical solution to this problem. The numerical technique used is an implicit finite-difference scheme. Although time-dependent calculations are made, the transient flow is not of primary interest at the present time. The steady flow is the desired result which is obtained in a time asymptotic fashion.

III. Governing Equations

The Azimuthal Invariant (or Generalized Axisymmetric) thin-layer Navier-Stokes equations for general spatial coordinates ξ , η , ζ can be written as⁷

$$\partial_{\tau} \hat{q} + \partial_{\xi} \hat{E} + \partial_{\zeta} \hat{G} + \hat{H} = \text{Re}^{-1} \partial_{\zeta} \hat{S} \quad (1)$$

where $\xi = \xi(x, y, z, t)$ is the longitudinal coordinate

$\eta = \eta(y, z, t)$ is the circumferential coordinate

$\zeta = \zeta(x, y, z, t)$ is the near normal coordinate

$\tau = t$ is the time

The vector of dependent variables $\hat{q}$ and the flux vectors $\hat{E}$, $\hat{G}$, $\hat{H}$ are given as

$$\hat{q} = J^{-1} \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}, \quad \hat{E} = \begin{bmatrix} \rho u \\ \rho u^2 + \xi_x p \\ \rho u v + \xi_y p \\ \rho u w + \xi_z p \\ (e + p)u - \xi_x p \end{bmatrix}$$

$$\hat{G} = J^{-1} \begin{bmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ (e+p)W - \zeta_t p \end{bmatrix}$$

$$\hat{H} = J^{-1} \phi_n \begin{bmatrix} 0 \\ 0 \\ \rho V [R_\xi (U - \xi_t) + R_\zeta (W - \zeta_t)] \\ -\rho V R_\eta (V - \eta_t) - p / (R \phi_n) \\ 0 \end{bmatrix}$$

The thin layer viscous terms valid for high Reynolds number flow are contained in the vector $\hat{S}$, where

$$\hat{S} = \begin{bmatrix} 0 \\ \mu(\zeta_x^2 + \zeta_y^2 + \zeta_z^2)u_\zeta + (\mu/3)(\zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta)\zeta_x \\ \mu(\zeta_x^2 + \zeta_y^2 + \zeta_z^2)v_\zeta + (\mu/3)(\zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta)\zeta_y \\ \mu(\zeta_x^2 + \zeta_y^2 + \zeta_z^2)w_\zeta + (\mu/3)(\zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta)\zeta_z \\ \{(\zeta_x^2 + \zeta_y^2 + \zeta_z^2)[0.5\mu(u^2 + v^2 + w^2) + \kappa Pr^{-1} \\ (\gamma-1)(a^2)\zeta_t] + (\mu/3)(\zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta) \\ \times (\zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta)\} \end{bmatrix}$$

The velocities

$$\begin{aligned} U &= \xi_t + \xi_x u + \xi_y v + \xi_z w \\ V &= \eta_t + \eta_x u + \eta_y v + \eta_z w \\ W &= \zeta_t + \zeta_x u + \zeta_y v + \zeta_z w \end{aligned} \quad (2)$$

represent the contravariant velocity components.

The Cartesian velocity components (u, v, w) are nondimensionalized with respect to a_∞ (the free stream speed of sound). The density (ρ) is referenced to ρ_∞ and total energy (e) to $\rho_\infty a_\infty^2$. The local pressure is determined using the equation of state,

$$p = (\gamma-1)[e - 0.5\rho(u^2 + v^2 + w^2)] \quad (3)$$

where γ is the ratio of specific heats.

In Equation (1) the thin-layer approximation is used and the restrictions for axisymmetric flow are imposed. The details can be found in References 8 and 9 and are not discussed here. Equation (1) contains only two spatial derivatives; however it retains all three momentum equations thus allowing a degree of generality over the standard axisymmetric equations. In particular, the circumferential velocity is not assumed to be zero thus allowing computations for spinning

projectiles or swirl flow to be accomplished. There is some evidence which indicates that base pressure can change due to the spin of a projectile. Although the present work considers base flow with no spin, base flow with spin is of interest and can be studied using the present technique.

For the computation of turbulent flows a turbulence model must be supplied. In the present calculations a Cebeci-type two layer algebraic eddy viscosity model as modified by Baldwin and Lomax¹⁰ is used. In their two layer model the inner region follows the Prandtl-Van Driest formulation. Their outer formulation can be used in wakes as well as in attached and separated boundary layers. In both the inner and outer formulations the distribution of vorticity is used to determine length scales thereby avoiding the necessity of finding the outer edge of the boundary layer (or wake). The magnitude of the local vorticity for the axisymmetric formulation is given by

$$|\omega| = \sqrt{\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial z} - \frac{\partial w}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial x} - \frac{\partial u}{\partial z}\right)^2} \quad (4)$$

In determining the outer length scale a function¹⁰

$$F(y) = y|\omega| [1 - \exp(-y^+/A^+)] \quad (5)$$

is used where y^+ and A^+ are the conventional boundary layer terms. For the base flow (or wake flow) the exponential term of Equation (5) is set equal to zero.

IV. Numerical Method

a. Computational Algorithm

An implicit approximate factorization finite-difference scheme in delta form is used as described by Beam and Warming¹¹. An implicit method was chosen because it permits a time step much greater than that allowed by explicit schemes. For problems in which the transient solution is of no interest, this offers the possible advantage of being able to reach the steady state solution faster than existing explicit schemes.

The Beam-Warming implicit algorithm has been used in various applications⁷⁻¹³. The algorithm can be first or second order accurate in time and second or fourth order accurate in space. The equations are factored (spatially split) which reduces the solution process to one-dimensional problems at a given time level. Central difference operators are employed and the algorithm produces block tridiagonal systems for each space coordinate. The main computational work is contained in the solution of these block tridiagonal systems of equations.

b. Finite Difference Equations

The resulting finite difference equations, written in delta form are

$$\begin{aligned}
& (1 + h\delta_c \hat{A}^n - \epsilon_1 J^{-1} \nabla_{\xi} \Delta_{\xi} J) (1 + h\delta_c \hat{C}^n - \epsilon_1 J^{-1} \nabla_{\zeta} \Delta_{\zeta} J) \\
& - h \text{Re}^{-1} \delta_c J^{-1} \hat{M}^n J \times (\hat{q}^{n+1} - \hat{q}^n) = -\Delta t (\delta_c \hat{E}^n + \delta_c \hat{G}^n) \quad (6) \\
& - \text{Re}^{-1} \delta_c \hat{S}^n - \Delta t \hat{H}^n - \epsilon_1 J^{-1} [(\nabla_{\xi} \Delta_{\xi})^2 + (\nabla_{\zeta} \Delta_{\zeta})^2] J \hat{q}^n
\end{aligned}$$

Here $h = \Delta t$ because only first order accuracy in the time differencing is needed for the steady state flows which are considered here. This choice corresponds to the Euler implicit time differencing. The δ 's represent central difference operators, Δ and ∇ are forward and backward difference operators respectively. The Jacobian matrices $\hat{A} = \frac{\partial \hat{E}}{\partial \hat{q}}$, $\hat{C} = \frac{\partial \hat{G}}{\partial \hat{q}}$ along with the coefficient matrix $\hat{M}$ obtained from the local time linearization of $\hat{S}$ are described in detail in Reference 6. Fourth order explicit (ϵ_E) and implicit (ϵ_I) numerical dissipation terms are incorporated into the differencing scheme to damp high frequency growth and thus to control the non-linear instabilities. A typical range for the smoothing coefficients is $\epsilon_E = (1 \text{ to } 5) \Delta t$ with $\epsilon_I = 3\epsilon_E$.

c. Flow Field Segmentation

Figure 2 is a schematic illustration of the flow field segmentation that is used to compute the entire projectile flow field including the base flow. It shows the transformation of the physical domain into the computational domain and the details of the flow field segmentation procedure in both the domains.

The cross hatched region represents the projectile. The line BC is the projectile base and the region ABCD is the base region or the wake. The line AB is a computational cut through the physical wake region which acts as a repetitive boundary in the computational domain. Implicit integration is carried out in both ξ and ζ directions (see Figure 2). Note the presence of the lines BC (the base) and EF (nose axis) in the computational domain. They both, however, act as boundaries in the computational domain and special care must be taken in inverting the block tridiagonal matrix in the ξ direction. The details are presented in the next section.

d. Implementation of Boundary Conditions

1. Base Flow Without Base Injection

The no slip boundary conditions for viscous flow is enforced by setting

$$U = V = W = 0 \quad (7)$$

on the projectile surface except for the base. At the projectile base the velocity component normal to the base is set to zero, i.e. $U = 0$, while other flow variables are set equal to those at grid point next to the base. In other words, slip

is allowed along the base (inviscid boundary condition). Future work will be directed at the implementation of viscous boundary condition at the base to further access this approximation.

Along the computational cut (line AB) the flow variables above and below the cut were simply averaged to determine the boundary conditions on the cut. On the centerline of the wake region, a symmetry condition is imposed.

$$\begin{aligned}
\frac{\partial u}{\partial z} &= 0 \\
\frac{\partial v}{\partial z} &= 0 \\
w &= 0 \\
\frac{\partial p}{\partial z} &= 0
\end{aligned} \quad (8)$$

Free stream conditions are used at the outer boundary. Simple extrapolation for all flow variables is used at the downstream boundary (lines AD and AG). During transient calculations, use of a specified outflow pressure can give rise to numerical oscillations in the base region flow field. Eventually, these grow and swamp the solution. This difficulty is avoided by simply extrapolating pressure to the downstream boundary which is the procedure always used with supersonic outflow. A combination of extrapolation and symmetry is used at on the nose axis (line EF).

As a result of the flow field segmentation procedure described in Section IV c, the block tridiagonal matrix in the ξ direction has elements at $J = JB, JB+1$ which are treated as internal boundaries in the computational domain ($J = JB$ represents the projectile base and $J = JB+1$ is the nose axis). The block tridiagonal matrix in the ξ direction takes the following form (after setting $\epsilon_I = 0$ to simplify the illustration)

$$\begin{bmatrix}
1 & A_3 & & & & \\
-A_2 & 1 & A_4 & & & \\
& \ddots & \ddots & \ddots & & \\
& & \ddots & \ddots & \ddots & \\
-A_{JB-2} & & 1 & A_{JB} & & \\
& & 0 & 1 & 0 & \\
& & & 0 & 1 & 0 \\
& & & & -A_{JB+1} & 1 & A_{JB+3} \\
& & & & \ddots & \ddots & \ddots \\
& & & & & -A_{JMAX-2} & 1
\end{bmatrix}
\begin{bmatrix}
\Delta q_2 \\
\Delta q_3 \\
\vdots \\
\vdots \\
\Delta q_{JB} \\
\Delta q_{JB+1} \\
\vdots \\
\vdots \\
\Delta q_{JMAX-1}
\end{bmatrix}
=
\begin{bmatrix}
RHS_2 \\
RHS_3 \\
\vdots \\
\vdots \\
0 \\
0 \\
\vdots \\
\vdots \\
RHS_{JMAX-1}
\end{bmatrix} \quad (9)$$

Here A's denote the quantity $\frac{\Delta t}{\Delta \xi} \hat{A}$ and I is a 5x5 identity matrix. Note the appearance of two uncoupled block tridiagonals. The rows at JB and JB+1 are particularly simple because boundary conditions are updated explicitly at the end of inversions. These changes were easily implemented in a modular fashion into an existing code for projectile base flow computations. One simply fills the block tridiagonal matrix ignoring the base JB and the nose axis JB+1. Elements in these rows are then overloaded as shown above. The flow field segmentation does not affect the block tridiagonal matrix in the ξ direction.

2. Base Flow with Base Injection

The boundary conditions used for base flow with mass addition are presented here. The boundary conditions along the projectile surface, at the cut and downstream boundary all remain the same as previously described. Along the base boundary the following boundary conditions are imposed.

$$\begin{aligned} u &= u_j \\ v &= v_j = 0 \\ w &= w_j = w_{JB-1} \text{ (grid point next to the base)} \\ \rho &= \rho_j = \rho_{st} \end{aligned}$$

The stagnation density is obtained from the following relation.

$$\frac{\rho_{st}}{\rho_\infty} = \left(1 + \frac{\gamma-1}{2} M_\infty^2\right)^{\frac{1}{\gamma-1}} \quad (10)$$

The amount of air injected into the base region can be specified by the mass flow rate, $\dot{m}_j$. Since ρ_j and A_j are known, u_j can be calculated for any given mass flow rate. Rather than specifying $\dot{m}_j$ however, it is customary to specify a mass injection parameter, I where $I = \dot{m}_j / \rho_\infty u_\infty A_\infty$.

e. Computational Grid

The finite difference grid used for the numerical computations was obtained from a grid generator developed in Reference 14. This program allows arbitrary grid point clustering, thus enabling grid points for the projectile shapes to be clustered in the vicinity of the body surface. The grid consists of 108 points in the longitudinal direction and 50 points in the radial direction. The full grid is shown in Figure 3 while Figure 4 shows an expanded view of the grid in the vicinity of the projectile. The computational domain extended to 4 body lengths in front, 4 body lengths in the radial direction and 4 body lengths behind the base of the projectile. The grid points in the normal direction where exponentially stretched away from the surface with the minimum spacing at the wall of .00002D. This spacing locates at least two points within the laminar sublayer.

The grid shown in Figure 4 was generated in two segments. First, the grid in the outer region is obtained using an elliptic solver¹⁴ for the ogive portion and straight-line rays for the remaining portion which runs all the way to downstream boundary. Second, the grid in the base region is obtained simply by extending the straight lines perpendicular to line AB down to the center line of symmetry (line CD). An exponential stretching with the minimum spacing of .00002D at line AB is used. It should be noted that the same minimum spacing .00002D is specified on both sides of the cut thus maintaining a smooth variation of grid across the cut. This spacing could, of course, be increased downstream of the base. The number of grid points above and below line AB is the same (50 points) which means that an adequate number of points are located in the base region. As can be seen in Figure 4, the grid points are clustered near the nose-cylinder junction and at the projectile base where appreciable changes in flow variables are expected.

As indicated in Figure 4, the fine viscous grid follows the cut labeled as AB in Figure 2. In so far as the viscous shear layer begins to neck-down shortly behind the base, much of this fine grid resolution is wasted. As a consequence logic has been implemented to adjust the grid cut AB to the viscous shear layer. Such a grid is shown in Figure 5 in which the height of the cut is determined from a moment of shear subject to various constraints and averaging. Specifically, the cut height, $\bar{z}_j$ at each J-location is determined by the relation

$$\bar{z}_j = \frac{\sum (\delta_z u_{jL})^2 z_{jL} + \epsilon D/2}{\sum (\delta_z u_{jL})^2 + \epsilon} \quad (11)$$

where the summation is carried out only for those points within an interval $.2D < z_{jL} < D/2$. Here D is the base diameter, δ_z is a central difference operator and ϵ is a positive parameter which ensures a standard grid if all $\delta_z u_{jL}$ are zero or if ϵ is very large. Additional averaging is used in the x-direction (longitudinal direction). Preliminary results have been obtained using the grid shown in Figure 5 and further computations are underway.

V. Results

The model geometry used in the present study is shown in Figure 6. The model consists of a 3 caliber secant-ogive nose and a 3 caliber cylinder.

The free stream Reynolds number for the series of computations was fixed at 4.5×10^6 based on the total model length. The computations are started from free stream conditions and marched in time to obtain the steady state solution. The initial calculation was made for $M = 0.9$. Previous converged solutions were then used as starting conditions for additional Mach number runs to achieve faster convergence. The results are now presented for both cases, (i) base flow

without injection and (ii) base flow with injection.

Figures 7 and 8 show the distribution of the surface pressure coefficient, C_p as a function of axial position respectively, without and with mass injection at the base. The value of C_p beyond $X/D = 6$ is the value of pressure coefficient along the line extending from the cylinder portion straight to the downstream boundary. In Figure 7 there is no mass injection at the base. The pressure distribution reflects the shock pattern that typically occurs on shell at transonic velocities, the rapid expansion which occurs at the blunt base and the recompression that occurs downstream of the base. Although not shown in these figures, the pressure along the base remains fairly constant (within ± 0.005 variation in C_p values). The pressure coefficient distribution for a case with large mass addition is shown in Figure 8. The previous rapid expansion at the base and recompression downstream of the base are seen to be virtually eliminated.

Figure 9 shows the velocity vector field in the base region for $M = 0.9$, $\alpha = 0$ and $I = 0$. Each vector shows the magnitude and the direction of the velocity at that point. The figure shows the velocity field when there is no base bleed and the recirculatory flow in the base region is clearly evident.

The velocity vector plots in Figures 10, 11 and 12 show the effect of base bleed on the near wake flow field. Figure 10 shows the effect of base bleed for the case when the mass injection parameter is very small ($I = .01$). The change in the flow field is not very dramatic. In Figure 11 the mass injection parameter is increased to .07, and the effect of base bleed can be clearly seen. The near wake flow field has changed considerably. Figure 12 shows the effect of base bleed for a still higher mass injection parameter, $I = .13$. The flow field in the base region has now been dramatically altered. The recirculation pattern has been totally swept downstream.

The next four Figures 13, 14, 15 and 16 are stream function contour plots in the wake region, again for $M = 0.9$ and $\alpha = 0$. All these figures are deliberately stretched in y direction (not drawn to the same scale in x and y) to show the flow pattern in the base region as clearly as possible. Figure 13 is for the case of base flow with no mass injection at the base. It clearly shows the recirculation region and the position of the dividing streamline which separates the recirculatory base flow from the main flow. The reattachment point is about 2 calibers down from the base. Note the strong shear layer in the base region.

Figures 14, 15 and 16 show the flow pattern in the base region with mass injection allowed at the base. Figure 14 shows the effect of base bleed when the mass injection parameter is very small ($I = .01$). The reattachment point remains at about the same place as with no injection at the base. The flow pattern has changed slightly as can be seen by the dividing streamline, however the recirculation region has not changed dramatically. In Figure 15, the mass injection parameter,

I , has been increased to .07 and now the effect of mass injection can be clearly seen. The reattachment point has moved further down stream. The flow pattern in the near wake flow field has changed considerably and the separation bubble is reduced in size. When the mass injection parameter is increased further, $I = .13$, its effect on the flow field in the base region is apparent. Figure 16 shows that dramatic change in the flow field. It indicates no presence of any recirculation region and shows how the shear layer has been displaced markedly.

A more critical look at the computational results is presented in Figures 17 through 20. These figures show the quantitative details of projectile flow field. Figure 17 shows the variation of base drag with mass injection rates for $M = 0.9$ and $\alpha = 0$. The reduction in base drag with base injection can be seen clearly. The percent reduction in base drag increases with the an increase in the injection rate.

Since the entire projectile flow field, including the base flow, has been computed, all three drag components have been computed and thus the total drag determined. Figure 18 shows the variation of the total drag with varying mass injection rates. Again, the reduction in the total drag is apparent. As the injection rate is increased, the percent reduction in total drag increases.

Figures 19 and 20 show respectively, the variation of base drag and the total drag with Mach number both with and without base injection. In both these figures the computational results without injection at the base are shown by the solid line whereas the dotted line represent the computational results obtained with injection at the base. The reduction in base drag and also the total drag with base injection can be clearly seen. Figure 19 indicates that the percent reduction in base drag has increased with an increase in Mach number from .9 to .98. In both the figures the expected drag rise in the transonic speed regime is well predicted for $.9 < M < 1.2$ and the reduction in base drag and the total drag, due to base bleed has been clearly demonstrated.

V. Summary

A promising computational capability has been developed which computes the full projectile flow field including the recirculatory base flow at transonic speeds both with and without base injection.

Numerical computations have been made for Mach numbers $.9 < M < 1.2$ to predict the base drag and the total drag with and without base bleed. Computed results show the qualitative features of the flow field in the near wake for both cases. The effect of base injection on the qualitative nature of base flow has been clearly shown. Quantitative comparisons of base drag and the total drag both with and without base injection have been made. For $M = 0.9$ and $\alpha = 0$ the computational results show the reduction in base drag and the total drag for several mass injection parameters. Results are also presented for $.9 < M < 1.2$ for a given mass injection rate and the reduction in

base drag and the total drag has been demonstrated for this range of transonic speeds.

Current computational efforts are directed at the numerical computation of base flow at supersonic speeds. The encouraging results obtained thus far at transonic speeds indicate that the computational technique presented here shows the promise of providing the capability to predict the base drag and hence the total drag both with and without base injection.

References

1. Sedney, R., "Review of Base Drag", U.S. Army Ballistic Research Laboratory/ARRADCOM, Report No. 1337, Aberdeen Proving Ground, MD 21005, October 1966.
2. "155mm ERFB Base Bleed Range and Precision Tests", Conducted at Proof and Experimental Test Establishment, Nicolet, Quebec for Space Research Corporation, January 11, 1978.
3. Murthy, S.N.B. (Ed.), "Progress in Astronautics and Aeronautics: Aerodynamics of Base Combustion", Vol. 40, AIAA, New York, 1976.
4. Dickinson, E.R., "The Effectiveness of Base-Bleed in Reducing Drag of Boattailed Bodies at Supersonic Velocities", U.S. Army Ballistic Research Laboratory/ARRADCOM, Memorandum Report No. 1244, Aberdeen Proving Ground, MD 21005, 1960.
5. Sykes, D.M., "Cylindrical and Boattailed Afterbodies in Transonic Flow with Gas Ejection", *AIAA Journal*, Vol. 8, No. 3, March 1970, pp. 588-589.
6. Sullins, G.A., Anderson, J.D., and Drummond, J.P., "Numerical Investigation of Supersonic Base Flow with Parallel Injection", *AIAA Paper No. 82-1001*, June 1982.
7. Nietubicz, C.J., Pulliam, T.H., and Steger, J.L., "Numerical Solution of the Azimuthal-Invariant Thin-Layer Navier-Stokes Equations", ARBRL-TR-02227, U.S. Army Ballistic Research Laboratory, ARRADCOM, Aberdeen Proving Ground, MD 21005, March 1980.
8. Nietubicz, C.J., "Navier-Stokes Computations for Conventional and Hollow Projectile Shapes at Transonic Velocities", *AIAA Paper No. 81-1262*, June 1981. Also being published as a BRL report.
9. Sahu, J., Nietubicz, C.J., and Steger, J.L., "Numerical Computation of Base Flow for a Projectile at Transonic Speeds", *AIAA Paper No. 82-1358*, August 1982.
10. Baldwin, B.S., and Lomax, H., "Thin-Layer Approximation and Algebraic Model for Separated Turbulent Flows", *AIAA Paper No. 78-257*, 1978.
11. Beam, R., and Warming, R.F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations", *AIAA Paper No. 77-645*, June 1977.
12. Steger, J.L., "Implicit Finite Difference Simulation of Flow About Arbitrary Geometries with Application to Airfoils", *AIAA Journal*, Vol 16, No. 4, July 1978, pp. 679-686.
13. Pulliam, T.H., and Steger, J.L., "On Implicit Finite-Difference Simulations of Three-Dimensional Flow," *AIAA Journal*, Vol. 18, No. 2, February 1980, pp. 159-167.
14. Steger, J.L., Nietubicz, C.J., and Heavey, K.R., "A General Curvilinear Grid Generation Program for Projectile Configurations", ARBRL-MR-03142, U.S. Army Ballistic Research Laboratory, ARRADCOM, Aberdeen Proving Ground, MD 21005, October 1981.

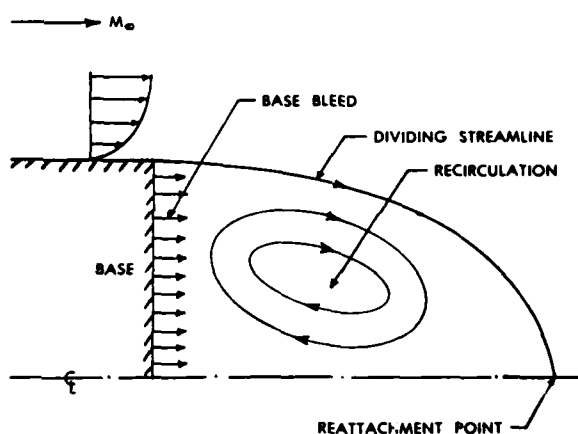


Figure 1. Schematic Illustration of Base Region Flow Field with Base Bleed

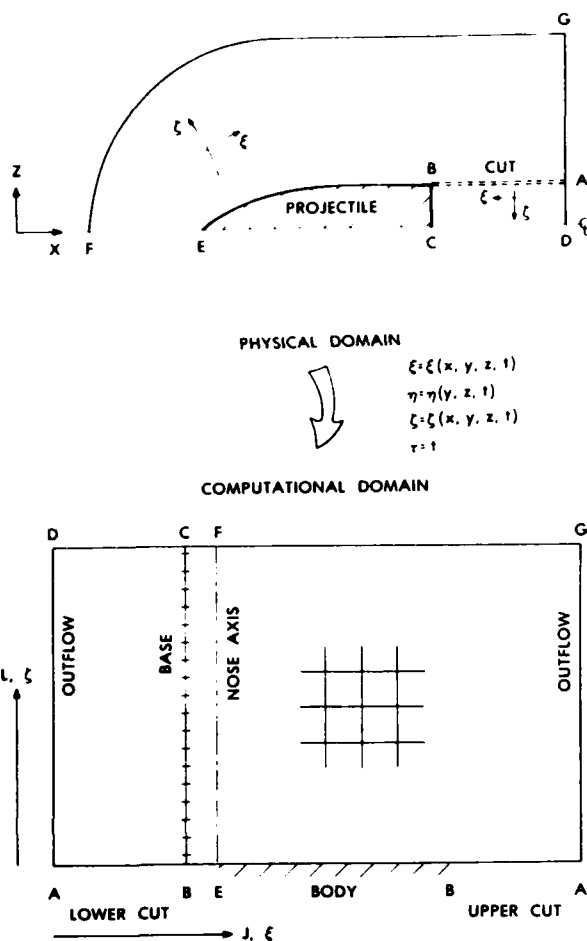


Figure 2. Schematic Illustration of Flow Field Segmentation

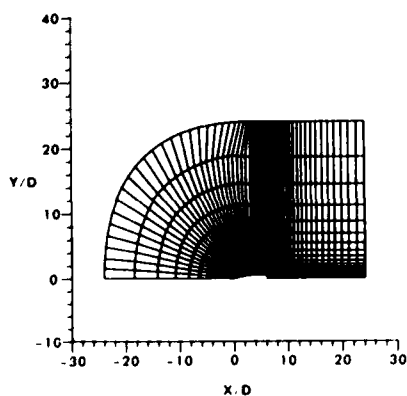


Figure 3. Computational Grid for Flow Field Computations

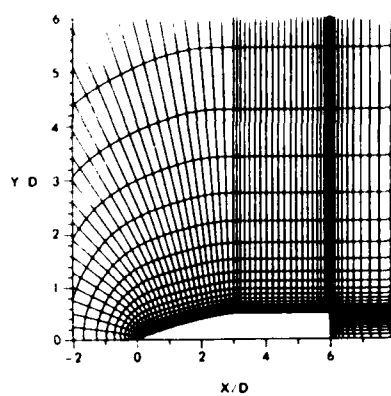


Figure 4. Expanded Grid in the Vicinity of the Projectile

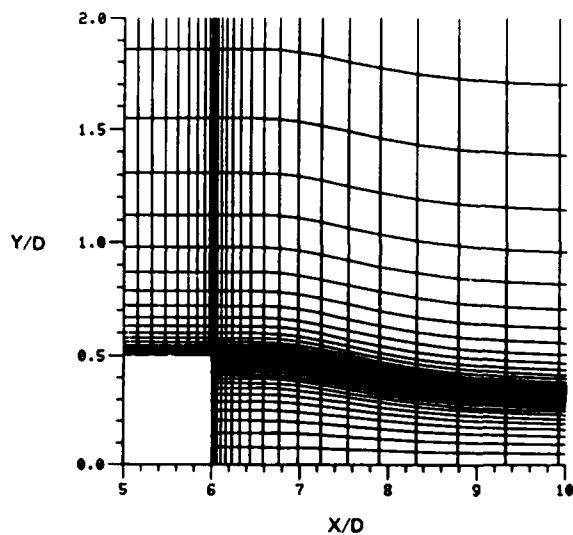


Figure 5. Grid Adapted to the Shear Layer

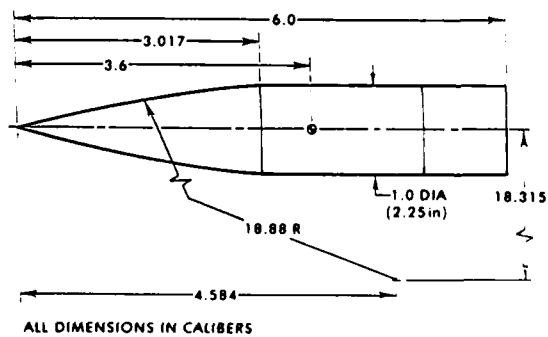


Figure 6. Model Geometry

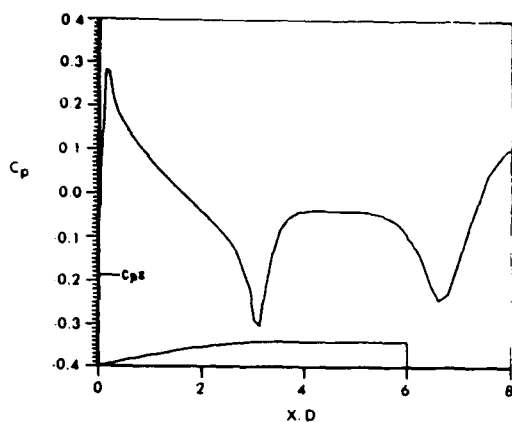


Figure 7. Longitudinal Surface Pressure Distribution, $M = 0.9$, $\alpha = 0$, $I = 0$ (without Base Bleed)

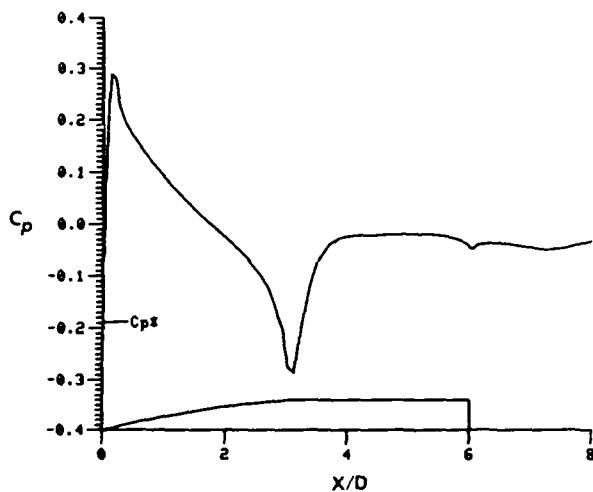


Figure 8. Longitudinal Surface Pressure Distribution, $M = 0.9$, $\alpha = 0$, $I = .13$ (with Base Bleed)

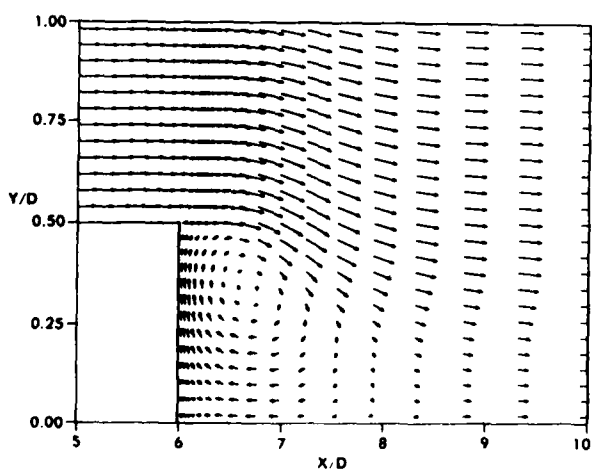


Figure 9. Velocity Vector Field, $M = 0.9$, $\alpha = 0$, $I = 0$

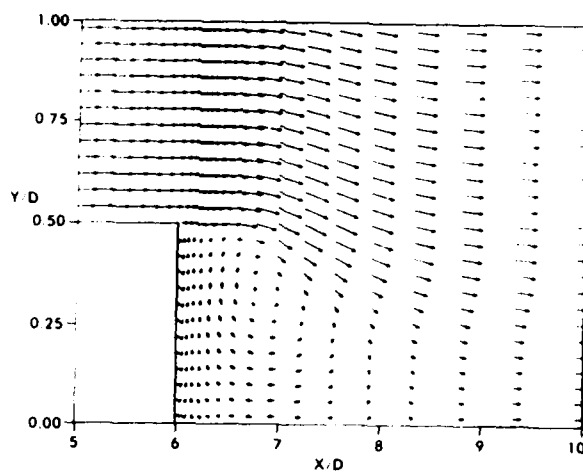


Figure 10. Velocity Vector Field, $M = 0.9$, $\alpha = 0$, $I = .01$

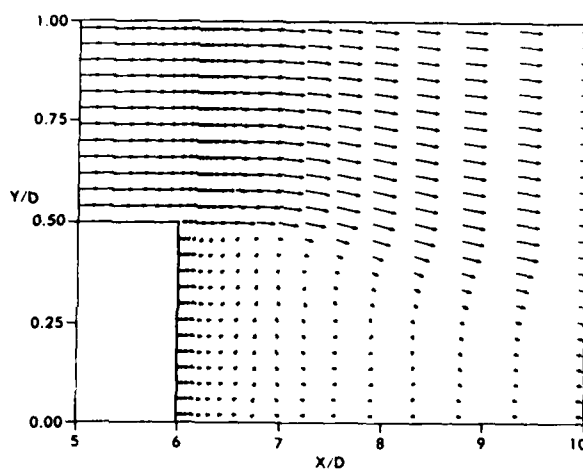


Figure 11. Velocity Vector Field, $M = 0.9$, $\alpha = 0$, $I = .07$

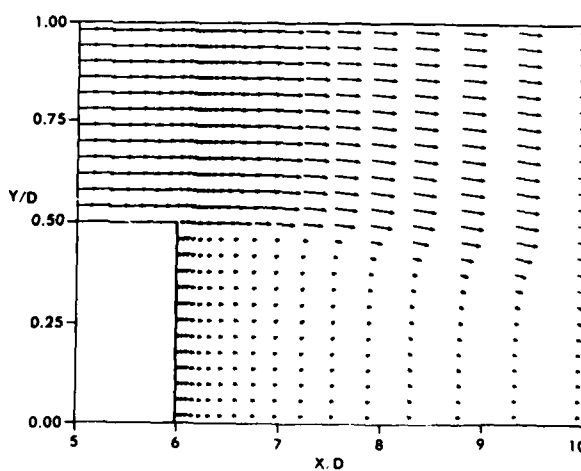


Figure 12. Velocity Vector Field, $M = 0.9$, $\alpha = 0$, $I = .13$

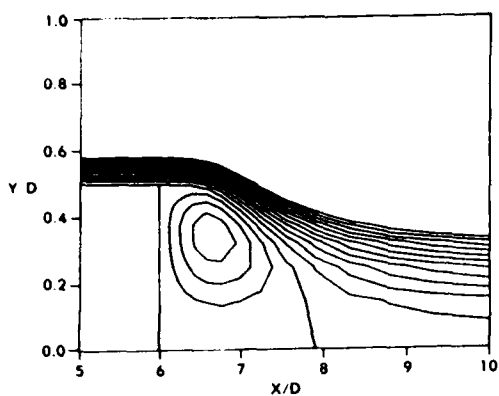


Figure 13. Stream Function Contours, $M = 0.9$,
 $\alpha = 0$, $I = 0$

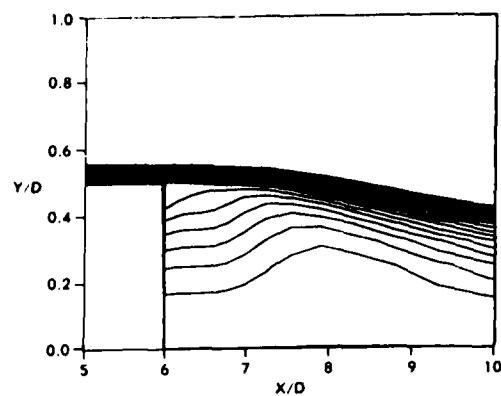


Figure 16. Stream Function Contours, $M = 0.9$,
 $\alpha = 0$, $I = .13$

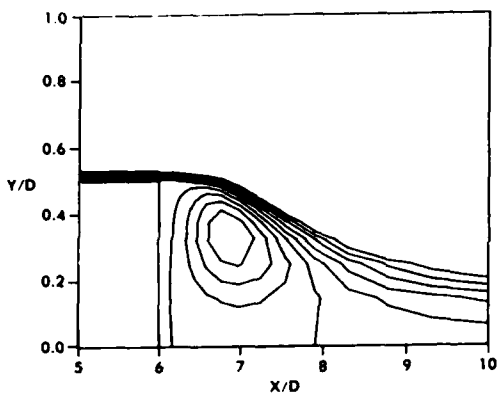


Figure 14. Stream Function Contours, $M = 0.9$,
 $\alpha = 0$, $I = .01$

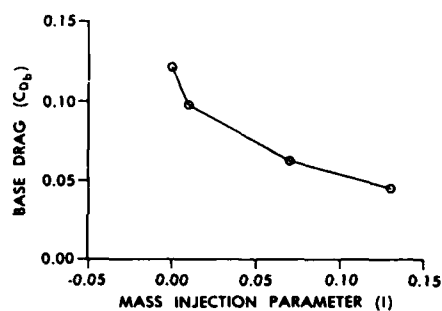


Figure 17. Variation of Base Drag Coefficient
with Base Bleed, $M = 0.9$, $\alpha = 0$

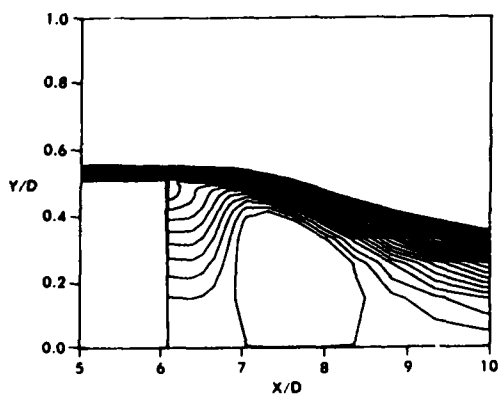


Figure 15. Stream Function Contours, $M = 0.9$,
 $\alpha = 0$, $I = .07$

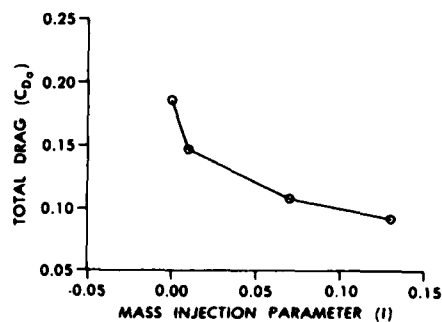


Figure 18. Variation of Total Drag Coefficient
with Base Bleed, $M = 0.9$, $\alpha = 0$

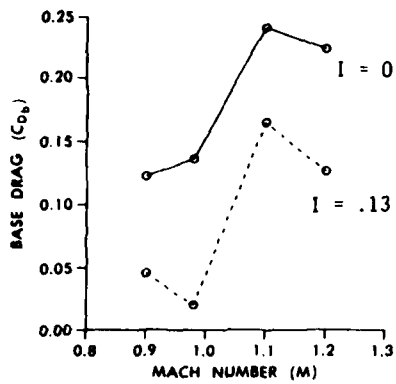


Figure 19. Variation of Base Drag Coefficient with Mach number, $\alpha = 0$ (with and without Base Bleed)

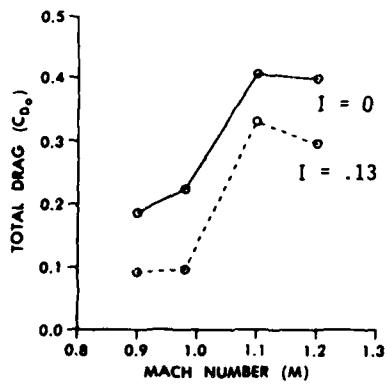


Figure 20. Variation of Total Drag Coefficient with Mach Number, $\alpha = 0$ (with and without Base Bleed)

III. GRIDS

Good solution accuracy depends on having properly spaced grids that are smoothly varying and not overly skewed. During the research program considerable effort was therefore devoted to the task of grid generation. A projectile grid generation program that can use either elliptic or hyperbolic grid generation procedures was devised. The attached paper, presented at the 1982 Army Numerical Analysis and Computers Conference, describes our basic grid generation solver.

ARO Report 82-3
**PROCEEDINGS OF THE 1982 ARMY NUMERICAL
ANALYSIS AND COMPUTERS CONFERENCE**



Approved for public release; distribution unlimited. 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.

SPONSORED BY
THE ARMY MATHEMATICS STEERING COMMITTEE ON BEHALF OF

THE OFFICE OF
THE CHIEF OF RESEARCH, DEVELOPMENT AND
ACQUISITION

Grid Generation Techniques for Projectile Configurations

Charles J. Nietubicz
Karen R. Heavey

Launch and Flight Division
U.S. Army Ballistic Research Laboratory
U.S. Army Armament Research and Development Command
Aberdeen Proving Ground, Maryland 21005

Joseph L. Steger

Department of Aeronautics and Astronautics
Stanford University
Palo, Alto, California 94305

ABSTRACT. The determination of accurate projectile aerodynamics is a major area of concern for shell designers involved with new shapes and Ballisticians concerned with developing artillery aiming data. To achieve the desired goals a research effort has been on going within the Aerodynamics Research Branch/BRL to establish a predictive capability for determining projectile aerodynamics. Modern finite difference codes have been applied to the projectile problem and encouraging results have been obtained in transonic^{1,2} and supersonic³ flow. The generation of good computational grids has been a prerequisite for achieving these flow field solutions.

This paper describes a versatile grid generation program which has been developed for standard, hollow and non-axisymmetric projectile shapes. The grid generator makes use of both elliptic and hyperbolic type partial differential equation solvers. The code allows arbitrary grid point clustering along the body surface in areas of anticipated flow field gradients. The outer boundary can also be arbitrarily defined with its own clustering distribution. The grid is then generated between these two boundaries with either straight rays or by use of an elliptic solver. For those cases when the outer boundary is not restricted, the grid can be generated using a hyperbolic solver which adds the additional benefit of an orthogonal mesh.

The mathematical development of the clustering functions and partial differential equation solvers are described and a series of grids are presented which show the versatility of the grid generation program. Grids for ogive-cylinder-boattail configurations, hollow ring airfoil projectiles and non-axisymmetric projectiles are discussed.

1. INTRODUCTION. The numerical solution of the Navier-Stokes^{4,5,6} equations has been successfully applied to a wide variety of problems. The versatility of these methods is in part attributed to the solution of the transformed set of differential equations. Using transformed equations the physical space can be mapped onto a regularly spaced rectangular region for two dimensional flow. This mapping allows for a wide variety of projectile

configurations to be solved using the same basic numerical technique. An example of some characteristic projectile shapes are shown in Figure 1. A standard projectile shape which consists of an ogive cylinder boattail is shown in 1a; a more non-conventional shape but one of considerable interest, the triangular boattail configuration in 1b; and a tubular projectile configuration which has been type classified and is currently in the Army inventory, in 1c. To calculate the flow field for any one of these shapes the first requirement is to develop a suitable finite difference grid for use with the equation solver. The grid generator described in this paper addresses this problem.

Grid generation routines are employed to generate a network of constant ξ and η lines in the physical x-y plane as indicated in Figure 2a. 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 as shown in Figure 2b. 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 $y_{j,k}$. 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 4-6).

The grid generation program presented here describes earlier work done by the authors⁷ as well as extensions which include a hyperbolic solver and the addition of more general projectile shapes. The grid generator is modular and begins with a determination of the body shape. The inner body clustering routine is then called to distribute points in the vicinity of previously determined flow field gradients. The next option allows for the insertion of stings for wake modeling, a rear cut or forward cut. If the outer boundary is free or unconstrained as is the case for conventional projectiles, the hyperbolic solver, which generates a smoothly varying orthogonal grid, is called. For those cases where the outer boundary is constrained, as is the case for tubular projectile shapes, the outer boundary clustering routine is called. Once the outer boundary is specified the elliptic solver is called. The grids generated up to this point would be planar and sufficient for axi-symmetric calculations. However for three dimensional flow fields a periodic or non-periodic grid is generated by spinning the planar grid about the symmetry axis. A flow chart of the overall grid program is shown in Figure 3.

The following sections of the paper will present some of the details used for the inner boundary clustering the outer boundary description and interior grid generation.

2. INNER BOUNDARY DESCRIPTION. The body shape can be input to the program by cards, file specification or as a set of x,y ordinates. The data is assumed to be non-dimensional with respect to the diameter or cord depending on the projectile configuration. Additionally, the code can generate a parabolic arc or standard class of projectiles such as sharp or blunt, tangent or secant ogive-nose, cylindrical body, boattail, or spherical cap. Once the body shape is determined the values of x along the body axis are distributed by contiguously combining segments of the clustering function

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

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 . 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. 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.

At this point a sting or forward cut can be added to the previously described body as shown in Figures 4a and 4b. Again the clustering function of Equation (1) is used to distribute points along these new boundaries.

3. GRID GENERATION USING A HYPERBOLIC SOLVER. For most projectile applications the outer boundary is unconstrained and simply needs to be placed far enough away from the projectile body so as not to adversely affect the flow field solution. This situation represents an ideal case for a hyperbolic grid generation scheme.

Once the body points have been redistributed and the sting or cut has been determined, a grid can be generated using a hyperbolic solver similar to that described in Reference 8. Before the actual solver can be implemented however, the distance to the outer boundary must be specified and either constant spacing in n or some type of stretching function is required. The n stretching used here is determined by the following relationship

$$\Delta s_k = \Delta s_0 (1 + \epsilon)^{k-1}, \quad k = 1, k_{\max} - 1 \quad (3)$$

Here Δs_0 is the minimum specified grid spacing desired at the wall or 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 $n = 0$ and $n = n_{\max}$ for points which have the same value of ξ .

The governing equations for the hyperbolic solver are obtained by requiring: (1) the coordinate lines ξ and n to be orthogonal; and (2) the specification of a cell volume or area for the two dimensional case. The condition of orthogonality requires

$$\Delta \xi \cdot \Delta n = 0 \quad (4)$$

The second equation is obtained by specifying a grid cell volume (or area in two dimensions). Since the grid cell volume is finite the transformation Jacobian will be greater than one, i.e.,

$$dx dy = |x_{\xi} y_n - x_n y_{\xi}| d\xi dn \quad (5)$$

The set of grid generation equations are therefore given in the physical plane by

$$\xi_x n_x + \xi_y n_y = 0$$

$$\xi_x n_y - \xi_y n_x = J$$

or in the transformed plane by

(6)

$$x_{\xi} x_n + y_{\xi} y_n = 0$$

$$x_{\xi} y_n - x_n y_{\xi} = 1/J \equiv V$$

Using local linearization for this set of non-linear differential equations, the resulting system is shown to be hyperbolic⁸ and can therefore be marched in the n direction.

The linearized set of differential equations to be solved numerically is written in vector form as

$$A \vec{r}_\xi + B \vec{r}_\eta = \vec{f} \quad (7)$$

where

$$A = \begin{bmatrix} x_\eta^0 & y_\eta^0 \\ y_\eta^0 - x_\eta^0 \end{bmatrix}, \quad B = \begin{bmatrix} x_\xi^0 & y_\xi^0 \\ -y_\xi^0 & x_\xi^0 \end{bmatrix}$$

$$\vec{f} = \begin{bmatrix} 0 \\ V + V^0 \end{bmatrix}, \quad \vec{r} = \begin{bmatrix} x \\ y \end{bmatrix}$$

where x_η^0, y_η^0 , etc., refers to known conditions.

The set of Equations (7) are solved with an implicit finite difference scheme which is first order accurate in the η direction(k) and where central differencing is used in the ξ direction(j). The resulting set of finite difference equations becomes

$$A \frac{(\vec{r}_{j+1,k+1} - \vec{r}_{j-1,k+1})}{2\Delta\xi} + B \frac{(\vec{r}_{j,k+1} - \vec{r}_{j,k})}{\Delta\eta} = \vec{f}_{j,k+1} \quad (8)$$

Rearranging Eq. (8) and setting $\Delta\eta = \Delta\xi = 1$ results in

$$\frac{A}{2} \vec{r}_{j+1,k+1} + B \vec{r}_{j,k+1} - \frac{A}{2} \vec{r}_{j-1,k+1} = \vec{f}_{j,k+1} + B \vec{r}_{j,k} = \vec{d}_{j,k+1} \quad (9)$$

where

$$\vec{d}_{j,k+1} = \begin{bmatrix} (x_\xi^0 x^0 + y_\xi^0 y^0)_{j,k} \\ (-y_\xi^0 x^0 + x_\xi^0 y^0)_{j,k} + V + V^0 \end{bmatrix}$$

Equation (9) is now in a form which can be easily solved by inverting a block tridiagonal matrix with 2×2 blocks. The terms x_ξ^0 and y_ξ^0 are central differenced as

$$\begin{aligned}
 x_{\xi,j,k}^0 &= \frac{x_{j+1,k} - x_{j-1,k}}{2} \\
 y_{\xi,j,k}^0 &= \frac{y_{j+1,k} - y_{j-1,k}}{2}
 \end{aligned}
 \tag{10}$$

The terms x_n^0 and y_n^0 are obtained from Equation (6) evaluated at the old station(o). That is

$$\begin{aligned}
 x_{\xi}^0 x_n^0 + y_{\xi}^0 y_n^0 &= 0 \\
 x_{\xi}^0 y_n^0 - x_n^0 y_{\xi}^0 &= V^0
 \end{aligned}
 \tag{11}$$

Solving for x_n^0 and y_n^0 with x_{ξ}^0 and y_{ξ}^0 given in (10) yields

$$\begin{aligned}
 x_n^0 &= \frac{-y_{\xi}^0 V^0}{(x_{\xi}^0 + y_{\xi}^0)} & y_n^0 &= \frac{x_{\xi}^0 V^0}{(x_{\xi}^0 + y_{\xi}^0)}
 \end{aligned}
 \tag{12}$$

The cell volume remains to be specified. This specification is important since it has the effect of controlling the grid evolution as the solution is being marched out from the body. The method chosen here is straight forward and uses the stretching function given by Equation (3). Specifying the minimum spacing at the wall Δs_0 and the total number of points, $j_{\max}$, in the n direction an array of arc lengths Δs_k is determined. Since the Δx is known along the j line, the volumes are calculated by

$$V = (\Delta s_k) (x_{j+1,k} - x_{j,k})
 \tag{13}$$

This specification of cell volumes yields smoothly varying grids in the n direction. Grid volume control is obtained by varying the arc length distribution Δs_k and/or surface point distribution. An additional volume specification approach can be found in Reference 8. A grid generated using this technique is shown in Figure 5a and 5b for a standard projectile configuration with sting.

4. OUTER BOUNDARY DEFINITION. For those cases where the outer boundary is constrained or specified a grid point distribution along the outer boundary is required. An example is shown in Figure 6. A part of the grid generation problem then is the formation of an arbitrary outer boundary. Here this boundary is built up by connecting contiguous cubic segments, which in the degenerate case can be straight lines. Figures 7a and 7b illustrate two

typical outer boundary curves. In Figure 7a three cubic segments make up the boundary, $\eta = \eta_{\max}$. Each segment is formed by specifying the values of x, y , and angle θ , at the endpoints, where θ is the angle between the curve and the x axis. In the example, Figure 7a, $\theta_a = 90^\circ$, $\theta_b = \theta_c = 0^\circ$, or 180° and $\theta_d = 90^\circ$.

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

$$\begin{aligned} x &= x_0 + \alpha_1 t + \alpha_2 t^2 \\ y &= y_0 + \beta_1 t + \beta_2 t^2 \end{aligned} \quad 0 \leq t \leq 1 \quad (14)$$

which are equivalent to a cubic

$$y = y_0 + \gamma_1(x-x_0) + \gamma_2(x-x_0)^2 + \gamma_3(x-x_0)^3 \quad (15)$$

The parametric cubic is used because the condition $\frac{dy}{dx} = \infty$ can be specified (segment bc of Figure 7b has this constraint at both endpoints).

The solution for the parameters $\alpha_1, \alpha_2, \beta_1$, and β_2 can be found in Reference 7.

The outer boundary curve is thus 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 determined only after the cubic segments are specified and the arc length is computed.

5. STRAIGHT RAY AND ELLIPTIC GRID GENERATION. Once the boundary curves have been specified and points are distributed on the $\eta = 0$ and $\eta_{\max}$ boundaries, two types of grid generation procedures can be used.

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 stretching relationship given by Equation (3). Figures 8a and 8b 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 9, 10, and 11. The grid generating equations are solved on the specified computational space for unknowns $x_{j,k}$ and $y_{j,k}$:

$$\begin{aligned}\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} &= -J^2 (\bar{P}x_{\xi} + \bar{Q}x_{\eta}) \\ \alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} &= -J^2 (\bar{P}y_{\xi} + \bar{Q}y_{\eta})\end{aligned}\quad (16)$$

where

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

and

$$\begin{aligned}\bar{P} &= P_0 e^{-a(\eta-\eta_0)} + P_m e^{-a(\eta-\eta_{\max})} \\ \bar{Q} &= Q_0 e^{-b(\eta-\eta_0)} + Q_m e^{-b(\eta-\eta_{\max})}\end{aligned}$$

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

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

on a vertical boundary, or

(17)

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

on a horizontal boundary.

The difference equations to Eq. (16) (see Reference 7) are solved with a successive line over relaxation (SLOR) procedure. As an initial guess for the relaxation procedure the straight line ray procedure previously described is used. For the most part, if coefficients $\bar{P}$ and $\bar{Q}$ are large, the SLOR procedure is very difficult to converge. Consequently, the algebraic clustering function, Eq. (3) is recommended.

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. (3) 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 7.

The elliptic solver need not be used over the entire range in ξ . Because of the boundary condition, Eq. (17), the elliptic equations can be joined to a straight ray along any vertical or horizontal boundary line in ξ . Figure 9 shows details of such a procedure used for a secant-ogive-cylinder boattail projectile which also includes a sting. Here the ξ -region over the secant-ogive nose is generated using the elliptic equations while the remainder is meshed with straight rays. After the basic grid is formed, the entire grid is clustered in n using Eq. (3).

6. 3D GRIDS. The final option available in the code is the ability to generate three dimensional grids. At present the grids are formed in a two dimensional plane and then rotated about a symmetry axis. The rotation is either periodic or non-periodic depending on the grid desired. For cases where the flow field has planar symmetry, such as a projectile at angle of attack, without spin, a non-periodic grid is generated.

The generation of grids for projectile shapes, with non-axisymmetric sections (Figure 1b) is accomplished with a series of planar grids. Planes are generated normal to the projectile axis at incremental values of Δx . For each of these planes a grid is generated using an O type grid (Figure 10). These grids are then combined to form a three dimensional mesh making sure that continuity in the x direction is maintained.

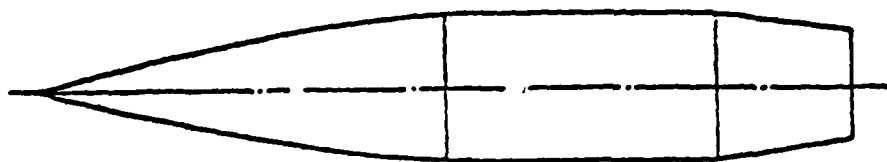
7. SUMMARY. A versatile grid generation program has been described which utilizes general elliptic and hyperbolic equation solvers for internal grid generation. The flexibility of longitudinal grid point distribution is obtained with the general clustering functions allowing points to be placed in the vicinity of flow field gradients. Grid clustering is also obtained near the body surface for viscous flow field calculations.

A series of grids have been presented which show the versatility of the code. Grids for secant-ogive-cylinder boattails have been shown using an elliptic solver, hyperbolic solver and a hybrid elliptic/straight ray solver. The generation of a grid for a non-conventional hollow projectile shape has been demonstrated.

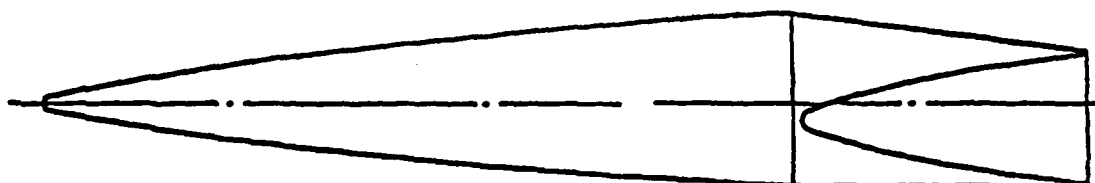
8. ACKNOWLEDGEMENTS. The authors would like to thank Dr. Harry Dwyer for his helpful discussions concerning the hyperbolic solver.

References

1. Nietubicz, C. J., "Navier-Stokes Computations for Conventional and Hollow Projectile Shapes at Transonic Velocities", AIAA Paper No. 81-1262, AIAA 14th Fluid and Plasma Dynamics Conference, Palo Alto, CA, 1981.
2. Nietubicz, C. J., Inger, G. R., and Danberg, J. E., "A Theoretical and Experimental Investigation of a Transonic Projectile Flow Field", AIAA Paper No. 82-0101, AIAA 20th Aerospace Sciences Meeting, Orlando, FL, January 1982.
3. Schiff, L. B., and Sturek, W. B., "Numerical Simulation of Steady Supersonic Flow Over Cone Ogive-Cylinder-Boattail Body", AIAA Paper No. 80-0066, January 14-16, 1980.
4. Steger, J. L., "Implicit Finite-Difference Simulation of Flow About Arbitrary Two-Dimensional Geometries", AIAA Journal, Vol. 16, July 1978, pp. 679-686.
5. Pulliam, T. H., and Steger, J. L., "On Implicit Finite-Difference Simulations of Three-Dimensional Flow", AIAA Paper No. 78-10, 1978.
6. 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.
7. Steger, J. L., Nietubicz, C. J., and Heavey, K. R., "A General Curvilinear Grid Generation Program for Projectile Configurations", BRL Memorandum Report-MR-03142, October 1981.
8. Steger, J. L., and Chaussee, D. S., "Generation of Body Fitted Coordinates Using Hyperbolic Differential Equations". Flow Simulations Report 80-1, January 1980.
9. Chu, W. H., "Development of a General Finite Difference Approximation for a General Domain". Journal of Comp. Physics, Vol. 8, 1971, pp. 392-408.
10. 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.
11. Sorenson, R. L., and Steger, J. L., "Simplified Clustering of Nonorthogonal Grids Generated by Elliptic Partial Differential Equations". NASA TM 73252, August 1977.

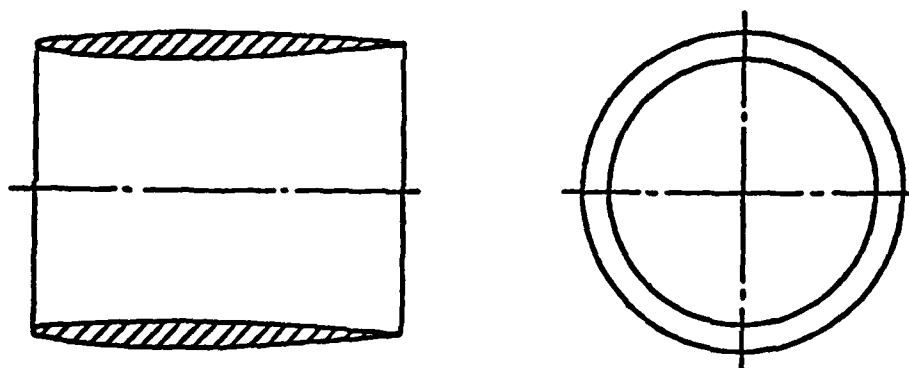


a. Conventional Secant-Ogive Cylinder Boattail (SOCBT) Projectile



7° TRIANGULAR BOATTAIL

b. Secant-Ogive Triangular Boattail Projectile



c. Tubular Projectile

Figure 1. Projectile Configurations

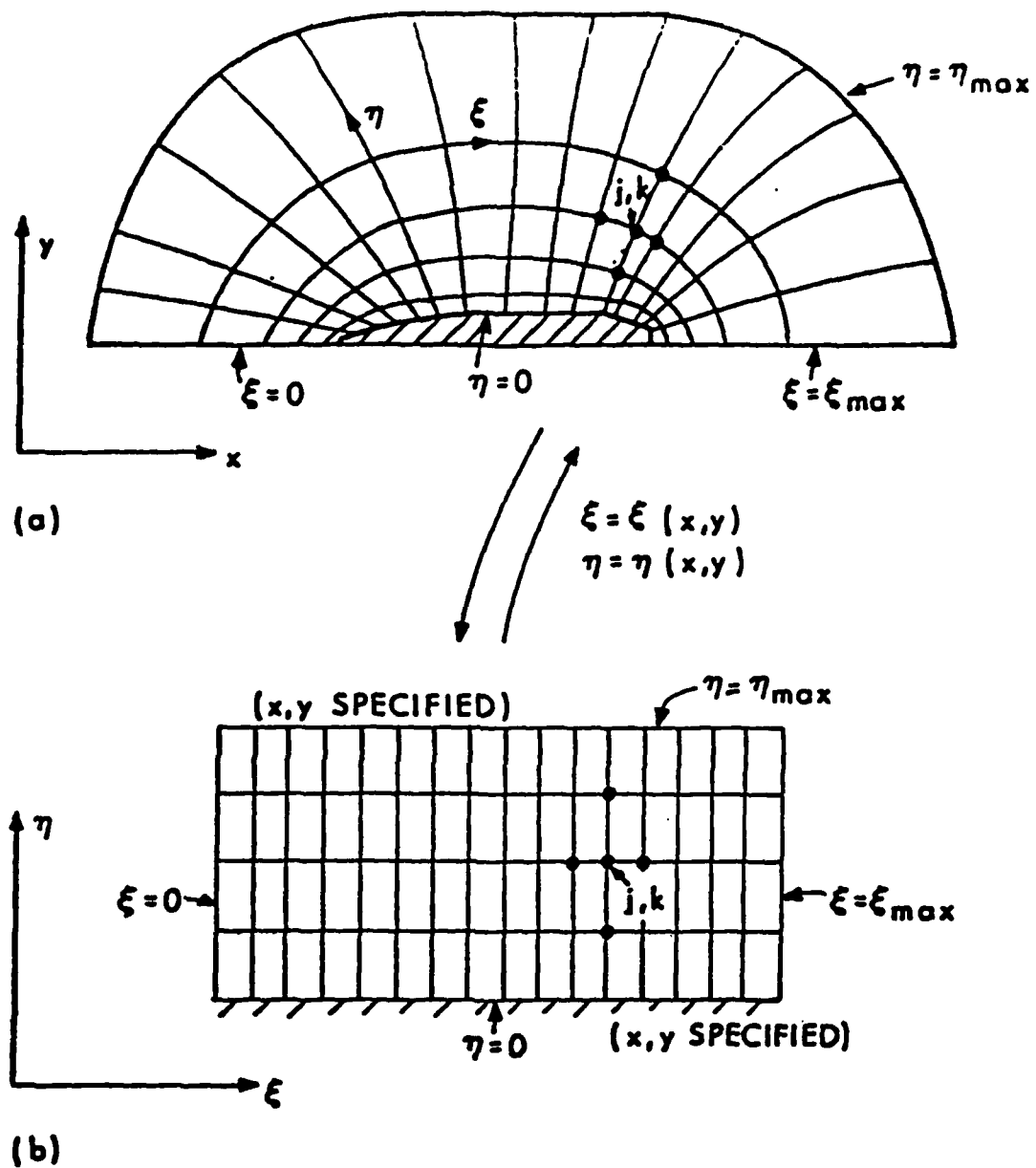


Figure 2. Mapping from Physical Space to Computational Space

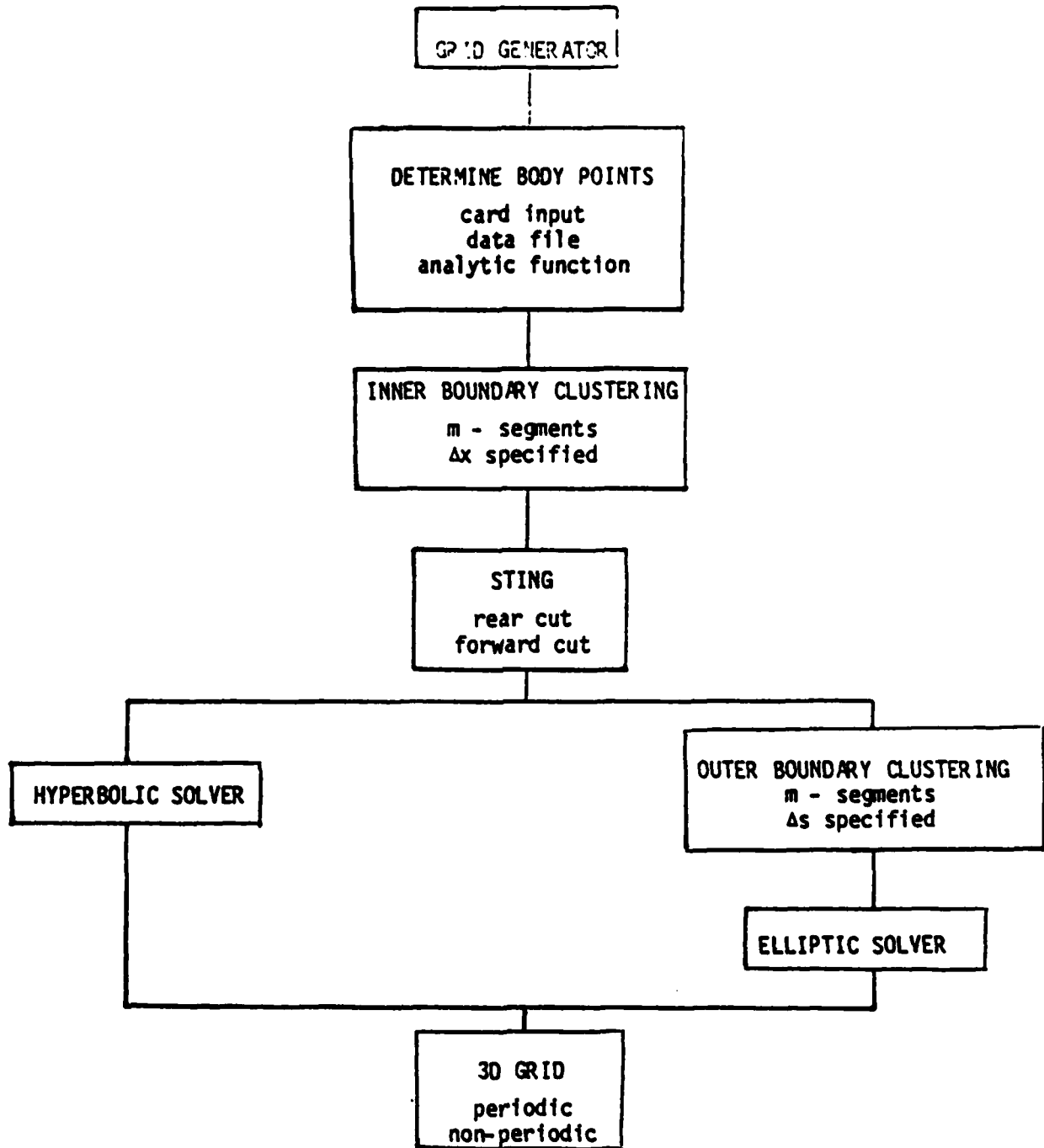


Figure 3. Flow Chart of Grid Generator Programs

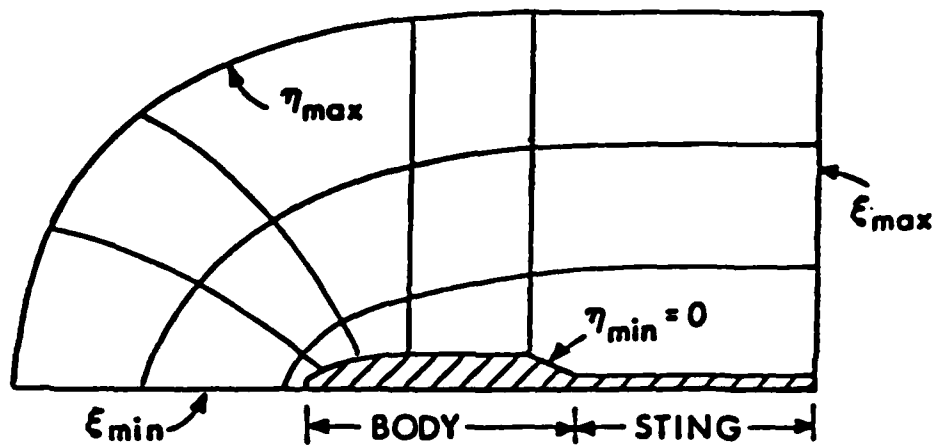


Figure 4a. Standard Projectile Grid with Sting

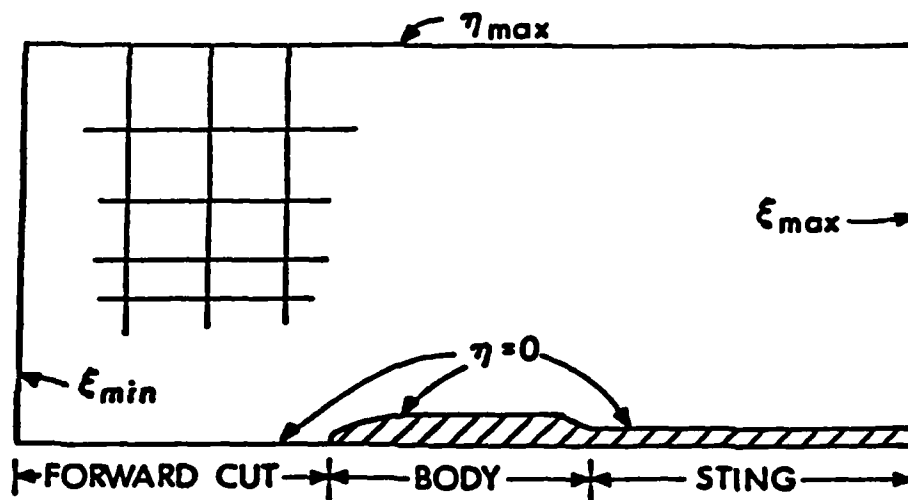


Figure 4b. Cartesian-like Projectile Grid with Sting

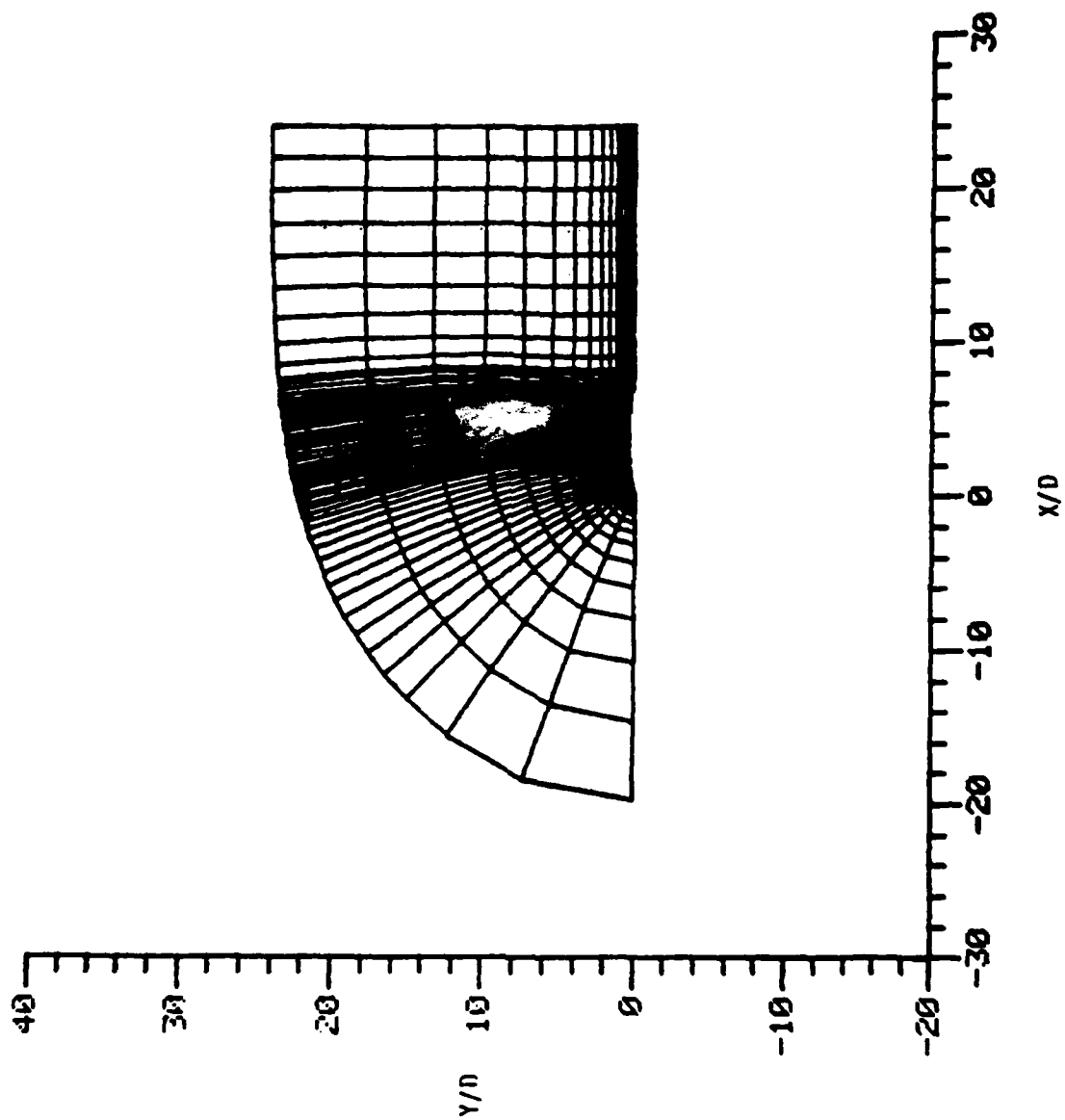


Figure 5a. Overview of Hyperbolic Grid for SOCBT With Sting

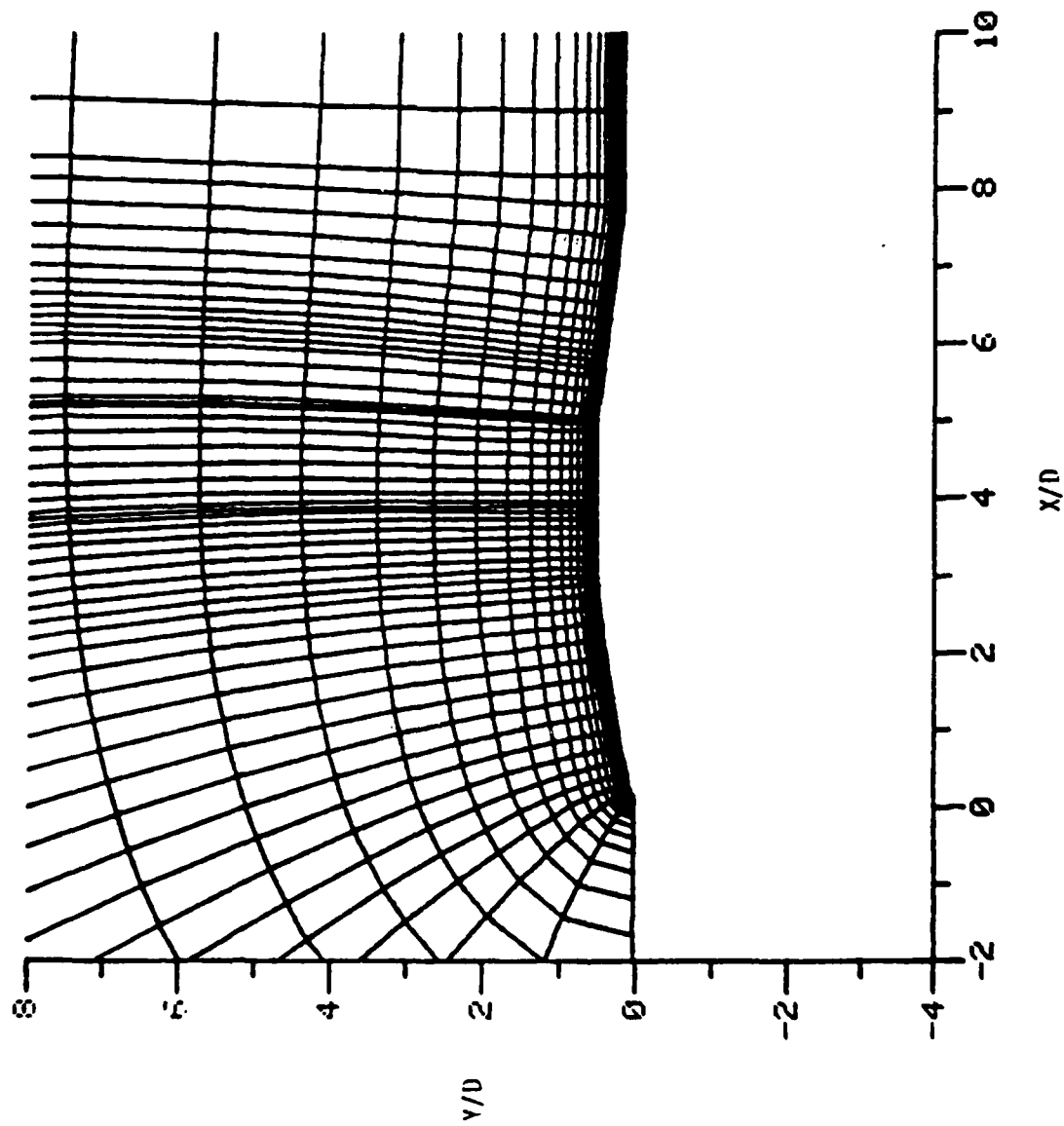


Figure 5b. Expanded View

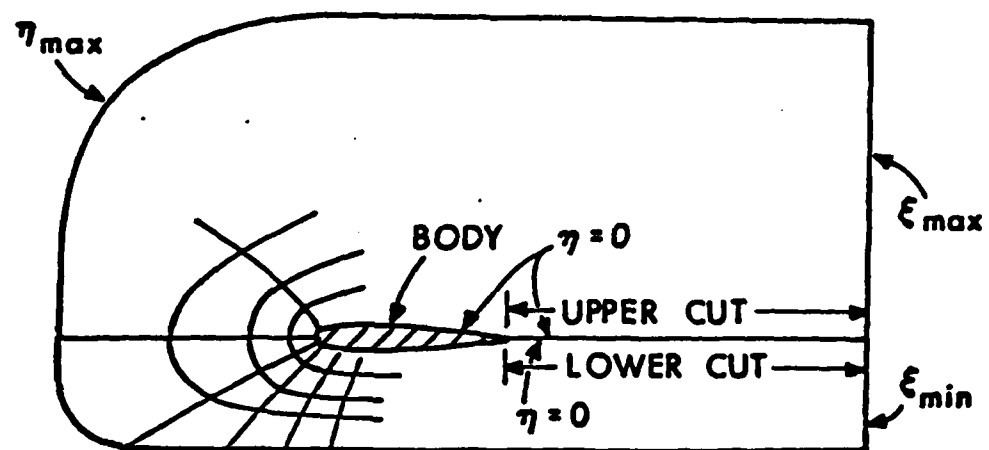
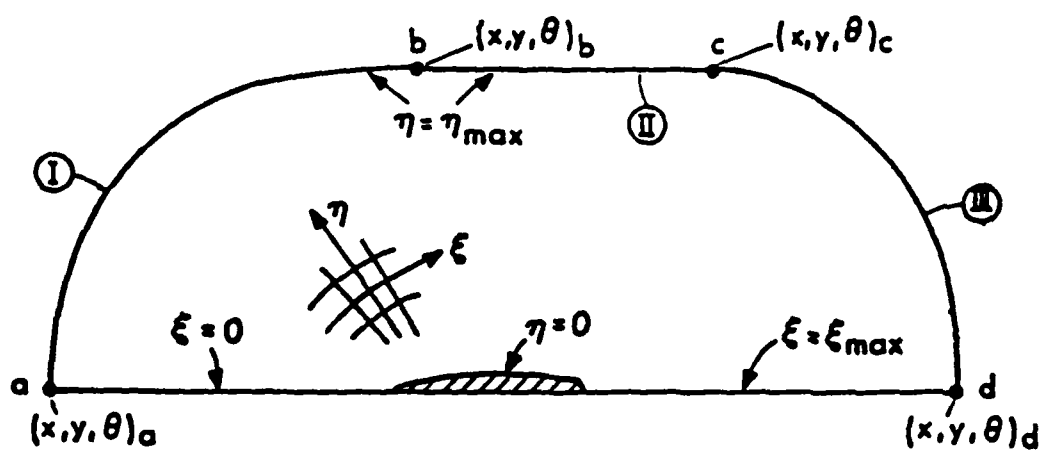
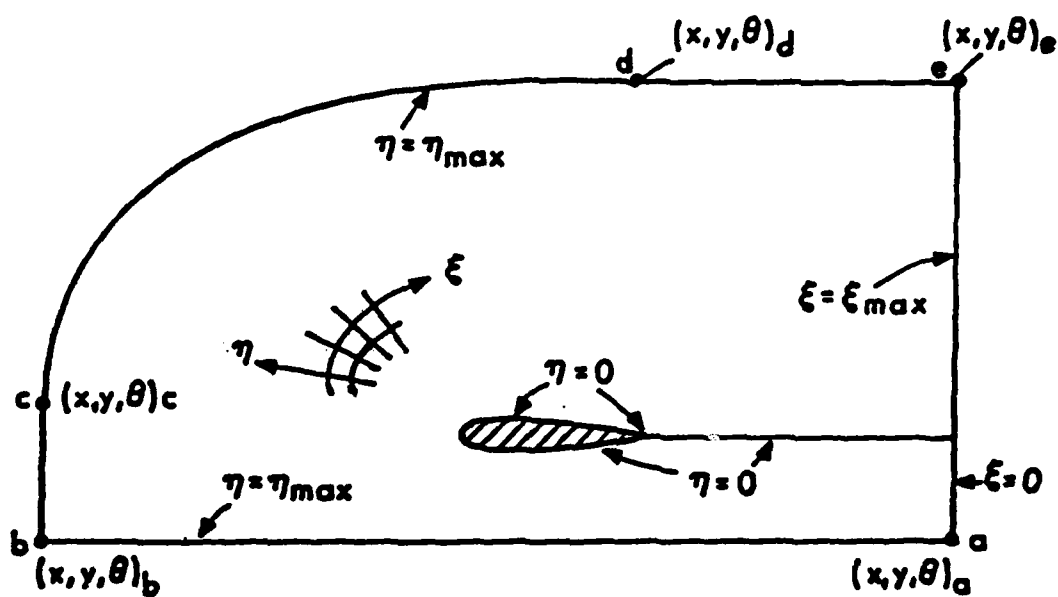


Figure 6. Tubular Projectile Grid or C-Grid



(a) STANDARD PROJECTILE GRID



(b) C-GRID FOR TUBULAR PROJECTILE

Figure 7. Outer Boundary Structure and Terminology for Two Classes of Grid

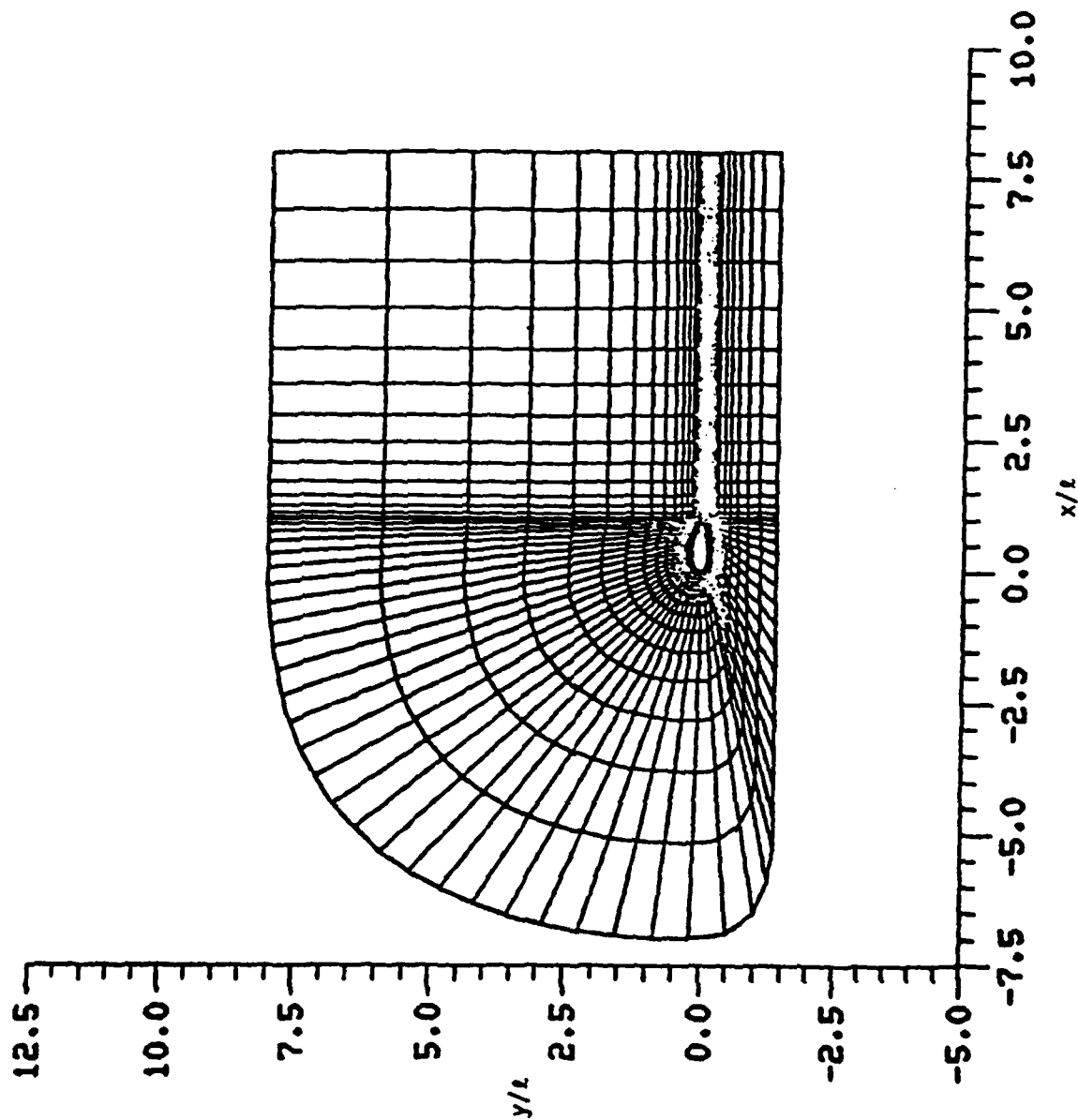


Figure 8a. Overview of Straight Ray Grid for Tubular Projectile

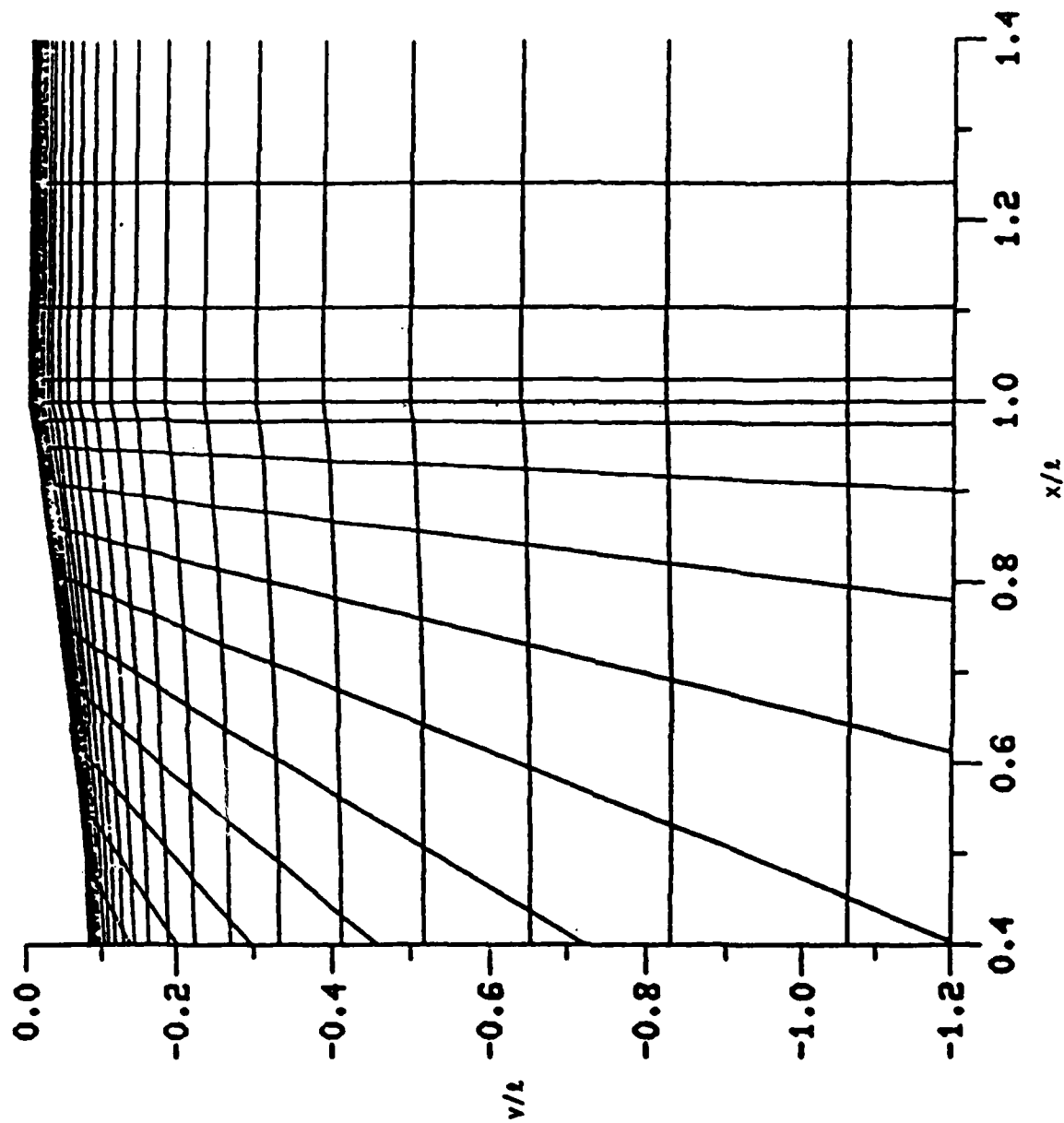


Figure 8b. Grid Detail near Lower Trailing Edge

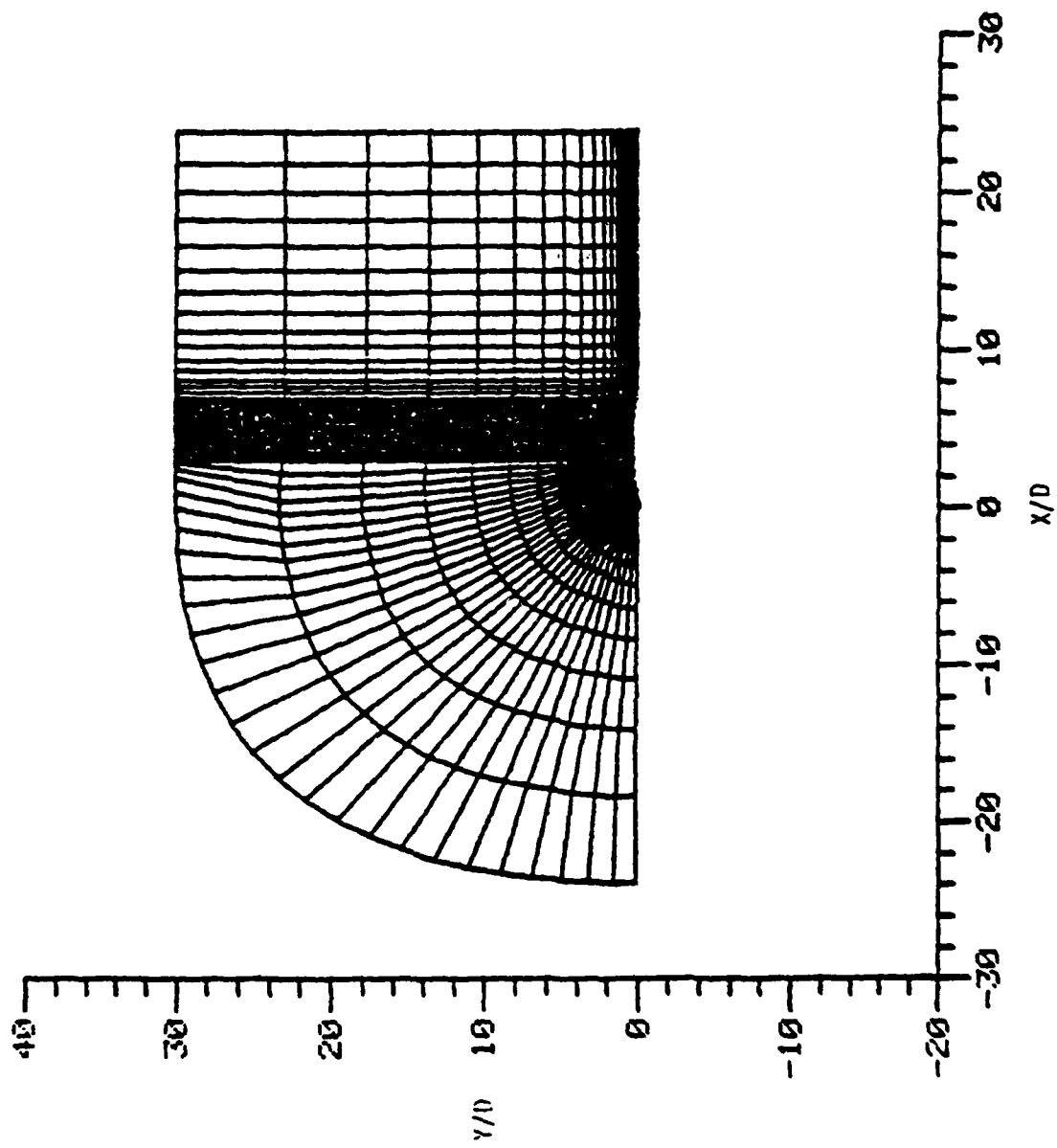


Figure 9a. Hybrid Elliptic and Straight Ray Grid for SOCBT With Sting

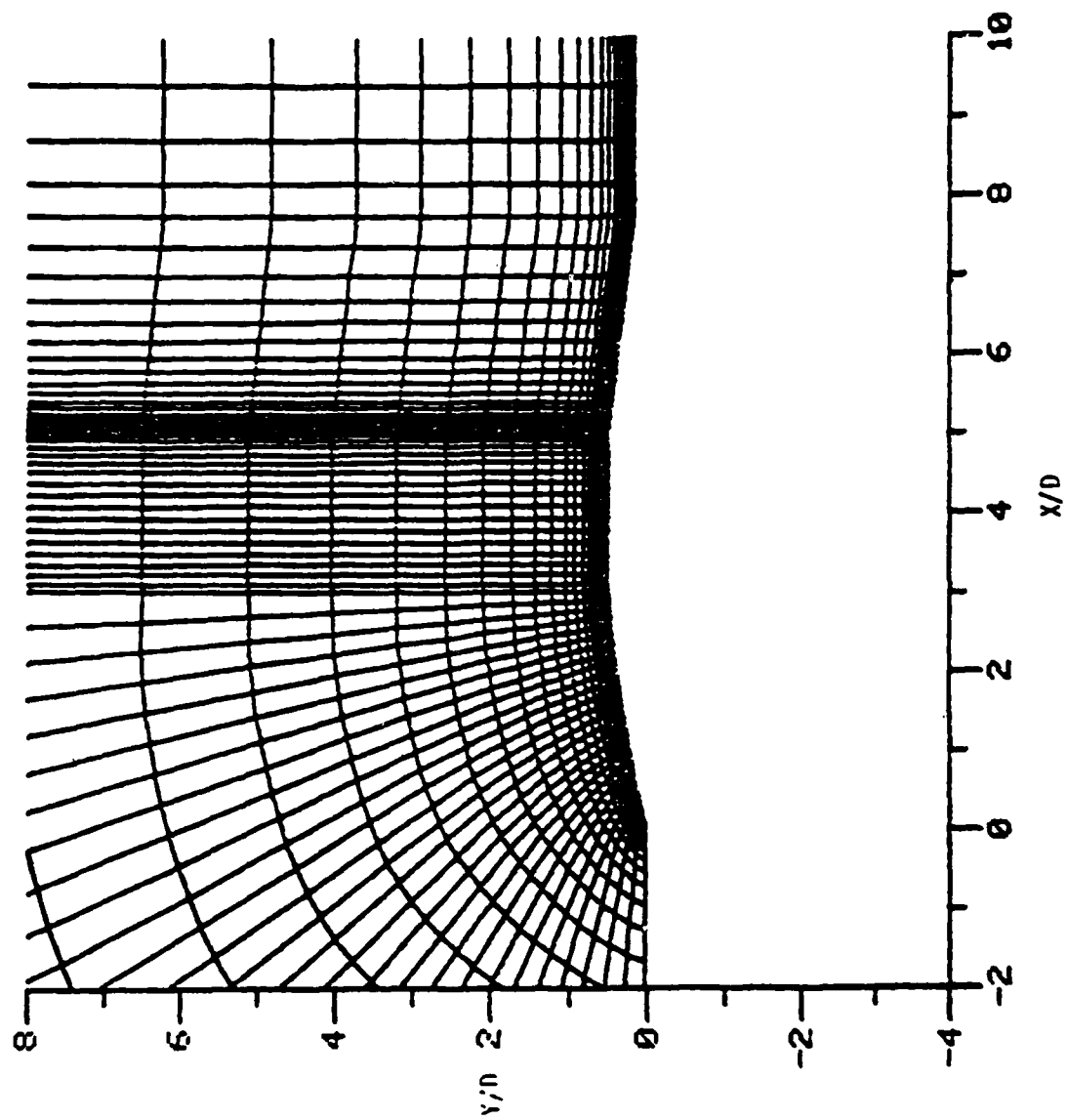


Figure 9b. Expanded View

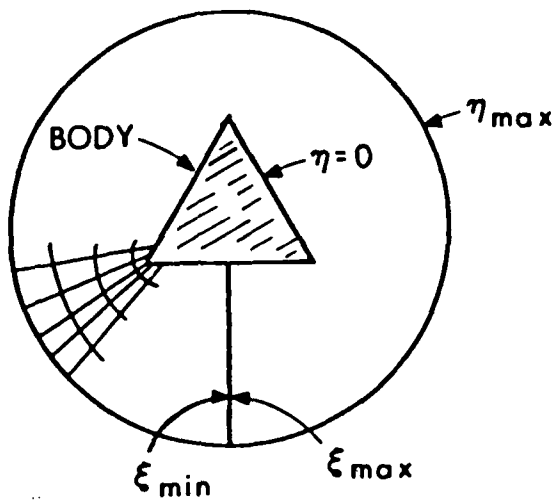


Figure 10a. Projectile Cross Section with Periodic B.C. (O-Grid)

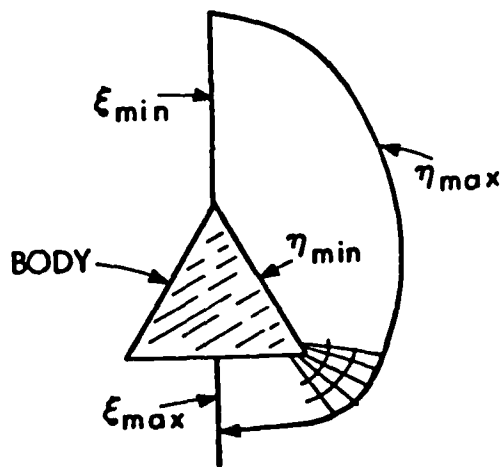


Figure 10b. Projectile Cross Section with Symmetry Plane (O-Grid)

IV. NUMERICAL ALGORITHMS

A majority of the research effort was devoted to numerical algorithm work so as to improve computational efficiency. Much of this work is described in two invited survey papers which are attached.

**Implicit Finite Difference Simulation
of Inviscid and Viscous Compressible Flow**

**presented at the Symposium on
Transonic, Shock and Multidimensional Flows
May 13-15, 1981, Madison, Wisconsin**

IMPLICIT FINITE DIFFERENCE SIMULATION

of

INVISCID AND VISCOUS COMPRESSIBLE FLOW

JOSEPH L. STEGER

I. INTRODUCTION.

It is not always convenient to use the simplified equations that extract the essential physics from the more complete set of inviscid and viscous fluid conservation-law-equations. Such a situation may occur if the usually inviscid outer flow is highly rotational and/or if the viscous layer is fully separated.

Numerical procedures for solving the system of conservation-law-equations of fluid flow are not as efficient as, say, the numerical procedures developed for the scalar nonlinear potential equation used in inviscid transonic flow analysis. Of course, the solution of a system of equations requires more work than the solution of a scalar equation. Equally or more significant, however, is that in dealing with a system of equations one often encounters characteristic speeds (i.e. eigenvalues) of disparate magnitude (stiffness) and of both positive and negative sign. These last conditions can make use of implicit differencing schemes desirable and put rather severe constraints on the choice of spatial differencing operators.

The purpose of this paper is to review the use of implicit finite difference schemes to solve the Euler and Navier-Stokes equations in primitive variables. In part one of this paper an approximate factorization (AF) implicit finite difference scheme for solving the Euler and Navier-Stokes equa-

tions is discussed. The equations are cast in generalized coordinates and partial differential equation grid generation techniques are used. In this approach the flux vectors of the equations are differenced as whole quantities and time-accurate or time-like iterative schemes are used to solve the equations for general boundary surfaces. In part two of this paper ways of splitting and reducing the governing equations are reviewed with an aim towards developing more accurate, efficient, and robust numerical algorithms. Again, implicit schemes are emphasized. Here, though, the methods are less developed.

II. IMPLICIT FINITE DIFFERENCE FLOW FIELD SIMULATION.

Over the last several years, a set [1-6] of versatile, somewhat robust computer codes has been developed for simulating steady or unsteady inviscid or viscous compressible flow. The computer programs make use of general coordinate transformations, numerical grid generation techniques, viscous modeling, and implicit finite difference algorithms to achieve a high degree of adaptiveness to flow conditions. In this section a review of this overall methodology is put forth. For brevity, the discussion here is restricted to two-dimensional compressible flow, although the basic procedures have also been applied to three-dimensional flow [2], incompressible flow [7], and supersonic flow solved by parabolic-like marching [8].

a) Transformed Thin-Layer Equations

As governing equations [1,9] we take the two-dimensional thin-layer Navier-Stokes equations subject to general coordinate transformation, but kept in conservation-law-form [10,11]

$$\partial_{\tau} \hat{Q} + \partial_{\xi} \hat{F} + \partial_{\eta} \hat{G} = \text{Re}^{-1} \partial_{\eta} \hat{S} \quad (2.1)$$

where

$$\xi = \xi(x, y, t), \quad \eta = \eta(x, y, t), \quad \tau = t$$

and the flux terms are defined as

$$\hat{Q} = J^{-1} \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix}, \quad \hat{F} = J^{-1} \begin{pmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ (e + p)U - \xi_t p \end{pmatrix}$$

$$\hat{G} = J^{-1} \begin{pmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ (e + p)V - \eta_t p \end{pmatrix}$$

$$\hat{S} = J^{-1} \begin{pmatrix} 0 \\ \eta_x \partial_\eta m_1 + \eta_y \partial_\eta m_2 \\ \eta_x \partial_\eta m_2 + \eta_y \partial_\eta m_3 \\ \eta_x \partial_\eta [um_1 + vm_2 + m_4] + \eta_y \partial_\eta [um_2 + vm_3 + m_5] \end{pmatrix}$$

where

$$m_1 = (2/3)\mu(2\eta_x u_\eta - \eta_y v_\eta)$$

$$m_2 = \mu(\eta_y u_\eta + \eta_x v_\eta)$$

$$m_3 = (2/3)\mu(2\eta_y v_\eta - \eta_x u_\eta)$$

$$m_4 = \kappa p_r^{-1} (\gamma - 1)^{-1} \eta_x \partial_\eta (c^2)$$

$$m_5 = \kappa p_r^{-1} (\gamma - 1)^{-1} \eta_y \partial_\eta (c^2)$$

Here ρ is density, p is pressure, u and v are Cartesian velocity components, and c is the sound speed. The total energy per unit volume, e , is defined by

$$e = (\gamma - 1)^{-1} p + 0.5\rho(u^2 + v^2) \quad (2.2)$$

and the (unscaled) contravariant velocities are defined as

$$U = \xi_t + \xi_x u + \xi_y v \quad (2.3a)$$

$$V = \eta_t + \eta_x u + \eta_y v \quad (2.3b)$$

The metrics ξ_t , ξ_x , etc., are determined once a mapping is defined. Usually, a numerical mapping is employed. The metrics are related to x_t , x_ξ , etc., by the relations

$$\begin{aligned} \xi_x &= J y_\eta, & \xi_y &= -J x_\eta, & \xi_t &= -x_t \xi_x - y_t \xi_y \\ \eta_x &= -J y_\xi, & \eta_y &= J x_\xi, & \eta_t &= -x_t \eta_x - y_t \eta_y \end{aligned} \quad (2.4)$$

$$J^{-1} = x_\xi y_\eta - x_\eta y_\xi$$

Here ξ varies around the body surface, and η varies away from the body surface, as indicated in Fig. 1-3. The symbol $\wedge$ denotes that the scalar or vector quantity is divided through by the Jacobian, J .

For practical viscous flow calculations, a turbulence model is needed. The algebraic two-layer eddy-viscosity model as proposed by Baldwin and Lomax [9] is used. The thin-layer approximation requires that $Re \gg 1$ and that the body coincide with an $\eta = \text{const}$ line.

The inviscid part of the governing equations is kept in conservation law (i.e. divergence) form so as to capture as accurately as possible the Rankine Hugoniot shock jump relations. Conservation-law-form is also useful in implicit formulations in that it can lead to cleaner local linearization formula. Conversely it can cause numerical inaccuracy unless metric transformation terms are properly dealt with.

b) Comments About the Transformed Equations

The transformed equations offer several significant advantages over the less complicated Cartesian form of the equations. Chief among these is that fact that the physical

boundary surfaces can coincide with transformed coordinate lines. This feature can be used to simplify the application of boundary conditions. Body conforming coordinates are also necessary to simplify the governing equations so as to permit the use of the thin-layer viscous model. Another significant aspect of coordinate transformations is that they can be used to cluster grid points to flow field action regions.

Generally we prefer to use a single transformation of coordinates to map the physical plane onto a uniform rectangular computational plane. Ideally the body boundary surfaces should lie on the boundaries of the computational domain. In this way one arrives at a well-ordered system of finite difference equations which is sparse, which can be efficiently solved for, and which is usually amenable to vectorized computer processing. These advantages are so significant that they are relinquished with great sorrow. The use of transforms and finite difference methods in no way requires such a simple topology, however. Although seldom used, it is possible to match or overlap more than one grid or coordinate system to treat complex geometries or to remove grid related stiffness of the equations, sketches 1 and 2 illustrate. In using such multiple grid systems, one will have to deal with a more complex program and take care not to introduce numerical instability at grid interfaces.

c) Metric Accuracy

Although the conservation-law-form of the equations is useful for capturing shocks and helps simplify the local linearization process, its use can lead to inaccuracy in certain difference formulations. In putting (2.1) into conservation-law-form, use is made of the exact relation

$$F[\partial_{\xi} y_{\eta} - \partial_{\eta} y_{\xi}] + G[-\partial_{\xi} x_{\eta} + \partial_{\eta} x_{\xi}] = 0 \quad (2.5)$$

In differencing the flow equations, (2.1), operators δ_{ξ} and δ_{η} are introduced to approximate ∂_{ξ} and ∂_{η} . If for example, the metrics $y_{\eta} = \xi_x/J$ etc. are exactly evaluated, then (2.5) cannot be equal to zero but is equal to a small error, the

truncation error. This error shows up in (2.1) as a source or sink, a "metric tare". For a highly stretched grid the metric tare in differencing (2.1) can be so appreciable as to invalidate the approximation and even numerical instability can result.

To avoid this problem steps must be taken as described in [1,2,5]. As first noted in [1], if $\xi_x/J = y_\eta$ etc. are centrally differenced with the same operators δ_ξ and δ_η used to difference the fluid flux terms, then the difference equations also exactly satisfy (2.5). Central differencing the metrics removes the source term difficulty, although flow field solution accuracy can still be poor if a very poor grid is used. This process can be extended to three-dimensions [2,5].

As an alternative to the above, good success has been obtained by making what we have termed "free stream subtraction" [2]. In this approach (2.1) is put into a perturbation-like form

$$\partial_\tau \hat{Q} + \partial_\xi (\hat{F} - \hat{F}_\infty) + \partial_\eta (\hat{G} - \hat{G}_\infty) = \text{Re}^{-1} \partial_\eta \hat{S} \quad (2.6)$$

That is, the "metric tare" is approximately subtracted off. This works on a well generated grid because the metric tare is usually only severe in the far field.

If one is willing to accept a "weak conservation-law-form", the equations could be modified as (here for inviscid flow only)

$$\partial_\tau \hat{Q} + \partial_\xi \hat{F} + \partial_\eta \hat{G} = F(\partial_\xi y_\eta - \partial_\eta y_\xi) + G(-\partial_\xi x_\eta + \partial_\eta x_\xi) \equiv \hat{H} \quad (2.7)$$

The right hand side source term $\hat{H}$ should not effect the shock strength or location, but it will contribute diagonal terms which by themselves can be weakly unstable.

d) Grid Generation

To take advantage of the transformed governing equations it is necessary to generate a smoothly-varying body-conforming grid. While this is a difficult task in general, a variety of algebraic [c.f. 12-14] and partial differential equation

[c.f. 14-22] schemes have been developed which at least handle the two-dimensional grid generation about a single body in a fairly automatic way. By automatic is meant that if the user carefully specifies grid points and clustering information along the mesh boundaries, the grid generation schemes will usually generate a smooth nonsingular interior grid. Of course the grid may not be as optimum as one would like, but the results are likely to be satisfactory. Moreover, progress is being made in treating more complex two and three-dimensional configurations.

The grid generation methods based on solving partial differential equations have always seemed especially appealing. This is because the numerical expertise one develops for solving the flow equations is directly applicable to the grid generation task. Figures 1a and 1b show an example of a grid using elliptic partial differential generating equations as taken from [23], while Figs. 2a and 2b show a grid obtained from hyperbolic partial differential generating equations as taken from [22]. In both cases the grid lines are orthogonal to the body and the grid spacing at the body is uniformly controlled. The grid generated with hyperbolic partial differential equations is ideal for many external flow configurations, it is essentially orthogonal throughout.

e) Difference Equations and Numerical Algorithm

An implicit, noniterative, time-accurate finite difference algorithm has been used to solve the transformed governing equations. Although viscous flows are ideally treated with an implicit scheme, the same numerical algorithm is used for inviscid flow calculations as well. By doing so one can base the time step size on accuracy considerations and not be overly concerned about highly clustered or distorted meshes. Computer programs that use explicit or semi-implicit (e.g. explicit in the streamwise direction, [c.f. 24]) schemes can be more efficient for a given problem, but are generally not as versatile.

The Beam-Warming delta form approximate factorization algorithm [25,26], with various adaptations, has been used to solve the thin-layer equations. It is remarked that similar numerical algorithms have been developed independently, and in aerodynamics applications the contributions of Briley and MacDonald [27,28] are notable. For either trapezoidal or Euler implicit temporal differencing the delta form differencing scheme for the thin-layer equation is given by:

$$\begin{aligned} & (I + h\delta_{\xi}\hat{A}^n - J^{-1}\epsilon_i h\nabla_{\xi}\Delta_{\xi}J)(I + h\delta_{\eta}\hat{B}^n - J^{-1}\epsilon_i h\nabla_{\eta}\Delta_{\eta}J \\ & - \text{Re}^{-1}h\bar{\delta}_{\eta}J^{-1}\hat{M}^n)(\hat{Q}^{n+1} - \hat{Q}^n) = -\Delta t(\delta_{\xi}\hat{F}^n + \delta_{\eta}\hat{G}^n - \text{Re}^{-1}\bar{\delta}_{\eta}\hat{S}^n) \\ & - \epsilon_e hJ^{-1}[(\nabla_{\xi}\Delta_{\xi})^2 + (\nabla_{\eta}\Delta_{\eta})^2]J\hat{Q}^n \end{aligned} \quad (2.8)$$

Here $h = \frac{\Delta t}{1+\alpha}$, $\alpha = 0$ or 1 for first or second order accuracy, and $\epsilon_i, \epsilon_e = O(1)$ with $\epsilon_i > 2\epsilon_e$ are added numerical dissipation terms. The operators $\delta_{\xi}, \delta_{\eta}$ are three point central differences e.g.

$$\delta_{\xi}F = \frac{F_{j+1} - F_{j-1}}{2\Delta\xi}$$

while ∇, Δ are conventional backward or forward operators, e.g.

$$\nabla_{\xi}Q = Q_j - Q_{j-1}.$$

Finally, $\bar{\delta}_{\eta}$ is the midpoint operator

$$\bar{\delta}_{\eta}S = \frac{S_{k+\frac{1}{2}} - S_{k-\frac{1}{2}}}{\Delta\eta}$$

where S itself contains midpoint differences so the viscous term uses three points. The coefficient matrices $\hat{A}, \hat{B}$, and $\hat{M}$

result from local linearization of the flux terms, with $\hat{A}$ and $\hat{B}$ the Jacobian matrices $[\partial \hat{F} / \partial \hat{Q}]$ and $[\partial \hat{G} / \partial \hat{Q}]$ while $\hat{M}$ contains derivative operators from linearizing $\hat{S}$.

The difference equations given by (2.8) are readily solved with an alternating direction-like sequence in which one inverts all block tridiagonals in ξ followed by all block tridiagonals in η . Boundary conditions can be imposed implicitly, approximate implicitly, or explicitly if the implicit stability range is not upset. These and additional details, including use of higher order space differencing, are described in [1-7,25,26].

f) Results

The numerical scheme described previously has been used in a variety of steady and unsteady state flow problems as reported in [1-7,23]. The simulation of aileron buzz represents a typical application of the code, so a few such calculations are presented below. These results are reproduced from [4].

In the buzz calculations a rigid aileron is allowed a one degree of freedom motion about its hinge line as described by the equation

$$I_a \ddot{\delta}_a = H(t) \quad (2.9)$$

Here I_a is the aileron mass moment of inertia about the hinge line, H is the aerodynamic forcing term, and δ_a is the aileron angle of deflection, see Fig. 3. Equation (2.9) is solved along with the thin-layer equations on a C-type grid in which the grid deforms to follow the aileron motion.

The results of a numerical simulation for a NACA 65-213 $a = 0.5$ airfoil section are indicated in Figures 4 to 8. Experimental data, albeit with some three-dimensional effect, is available on this airfoil from the tests of Erickson and Stephenson [29] in which they mounted the wing of the P-80 from the sidewall of the Ames sixteen foot tunnel.

According to the experiment, at $M_\infty = 0.82$ and $\alpha = -1$ deg, the aileron could be restrained at an angle near zero, and when freed, would buzz. In the numerical simulation the aileron was initially deflected to 4 deg, it would, on being released, buzz as indicated in Fig. 4. The computed frequency of 22.2 Hz is in good agreement with the experimental value of 21.2 Hz. However, in the numerical calculation, the aileron deflects ± 11.1 deg about the angle -1.1 deg, while the corresponding experimental values are ± 9.2 deg about -3 deg.

In the numerical calculations at a slightly higher Mach number, $M_\infty = 0.83$, the aileron does go into a buzz cycle when freed from a zero deflection position. An essentially steady-state solution was used as initial data. The build-up of aileron deflection angle as a function of time is indicated in Fig. 5. After four cycles, a quasi-steady-state is reached and the aileron oscillates at 22.7 Hz.

At a lower Mach number, $M_\infty = 0.79$, the aileron motion damps to a neutral value even though the flap was initially deflected 4 deg. Data from this calculation are displayed in Fig. 6.

Several frames from computer-generated film strips showing Mach contours are shown in Figs. 7a-7c at selected times for $\alpha = -1$ deg and $M_\infty = 0.82$. Contour levels of $M = 0.2, 0.4, 0.6, 0.98, 1.0$, and 1.02 were used in order to illustrate both the separated flow regions and sonic lines.

Finally, for the aileron held fixed at a higher Mach number, $M_\infty = 0.85$, we find that the viscous flow does not reach a steady state but buffets at a frequency of about 26.6 Hz. This is indicated by the unsteady hinge moment coefficient, C_H shown in Fig. 8. If the aileron is then released, it no longer oscillates in a simple sinusoidal motion. Viscous effects appear to be much more dominant and change the frequency and amplitude of the aileron motion. Similar type motion, but for a different airfoil, has been observed experimentally [30].

III. FLUX SPLIT SYSTEMS.

In a series of recent papers [31,26,32] splitting of the spatial flux terms of the conservative form of the Euler equations was proposed. The flux terms are usually split based on the positive and negative eigenvalues (characteristic speeds) of their appropriate Jacobian matrices. Nonconservative schemes based on the same principle have previously been proposed [33,34], and have recently been developed for aerodynamic applications [35-37]. Related older schemes have been identified and others are under extensive development [c.f. 38].

Splitting the flux vectors (conservative form) or coefficients matrices (nonconservative form) based on the sign of their eigenvalues allows the use of upwind (either backward or forward) spatial differencing schemes. Without use of such splitting only central spatial differencing can be used to approximate the Euler equation flux derivatives, except of course, for those spatial directions in which the coordinate velocity exceeds the sound speed. Upwind differencing schemes can offer some advantages over central differencing insofar that they are more dissipative, in some instances can follow the physics better, and can lead to new implicit approximate factorization schemes. It is this latter property which is the subject of this section.

The aim of the conservative form plus and minus flux vector splitting is to recast the inviscid portion of the equations into

$$\partial_t Q + \partial_x F^+ + \partial_x F^- + \partial_y G^+ + \partial_y G^- = \text{Viscous Part} \quad (3.1)$$

where Cartesian coordinates are used for illustration, i.e.

(2.1) with $\xi_x = 1 = \eta_y$ and $\xi_y = 0 = \eta_x$. The Jacobian matrices $\partial F^+ / \partial Q$ and $\partial G^+ / \partial Q$ are constructed to have positive real eigenvalues while $\partial F^- / \partial Q$ and $\partial G^- / \partial Q$ are to have negative real eigenvalues. The initial development of $F^\pm$ and $G^\pm$ relied on the fact that the Euler equations are homogeneous of degree one and proceeded as (illustrated for F)

$$\begin{aligned}
F &= AQ & A &= \partial F / \partial Q \\
&= SAS^{-1}Q \\
&= S(\Lambda^+ + \Lambda^-)S^{-1}Q & \text{with} & & 2\Lambda^\pm = (\Lambda \pm |\Lambda|) & (3.2) \\
&= S\Lambda^+S^{-1}Q + S\Lambda^-S^{-1}Q \\
&= F^+ + F^-
\end{aligned}$$

where S is a matrix of the eigenvectors of A while Λ is a diagonal matrix of its eigenvalues. This approach doesn't work exactly as desired because the crucial eigenvalues are those of the Jacobian matrices $\partial F^+ / \partial Q$ and $\partial F^- / \partial Q$, and $\partial F^\pm / \partial Q \neq S\Lambda^\pm S^{-1}$. Nevertheless the eigenvalues of $\partial F^\pm / \partial Q$ have the proper signs if not the proper magnitudes [39].

A general formula for the flux vectors $F^\pm$, $G^\pm$ is given by [32]:

$$\mathcal{F}_{II} = \frac{\rho}{2\gamma} \begin{bmatrix} 2(\gamma-1)\bar{\lambda}_1 + \bar{\lambda}_3 + \bar{\lambda}_4 \\ 2(\gamma-1)\bar{\lambda}_1 u + \bar{\lambda}_3(u+c\tilde{k}_1) + \bar{\lambda}_4(u-c\tilde{k}_1) \\ 2(\gamma-1)\bar{\lambda}_1 v + \bar{\lambda}_3(v+c\tilde{k}_2) + \bar{\lambda}_4(v-c\tilde{k}_2) \\ (\gamma-1)\bar{\lambda}_1(u^2+v^2) + \frac{\bar{\lambda}_3}{2}[(u+c\tilde{k}_1)^2 + (v+c\tilde{k}_2)^2] \\ + \frac{\bar{\lambda}_4}{2}[(u-c\tilde{k}_1)^2 + (v-c\tilde{k}_2)^2] + w_{II} \end{bmatrix} \quad (3.3)$$

where $\tilde{k}_1$ and $\tilde{k}_2 = 1$ or 0 for $F^\pm$ or 0 and 1 for $G^\pm$

$$w_{II} = \frac{(3-\gamma)(\bar{\lambda}_3 + \bar{\lambda}_4)c^2}{2(\gamma-1)}$$

and (in the present application)

$$\lambda_i^\pm = \frac{\lambda_i \pm |\lambda_i|}{2} = \bar{\lambda}_i \quad (3.4)$$

For example, the eigenvalues λ_i of F are u , u , $u+c$, $u-c$, and $F^\pm$ is defined from $\mathcal{F}_{II}$ using

$$\begin{aligned} 2\lambda_1^\pm &= u \pm |u| \\ 2\lambda_3^\pm &= u + c \pm |u+c| \\ 2\lambda_4^\pm &= u - c \pm |u-c| \end{aligned} \quad (3.5)$$

Similar relations hold for $G^\pm$ with v , v , $v+c$, $v-c$. In generalized coordinates $\hat{F}$ as given by (2.1) has eigenvalues [40,1] U , U , $U \pm c\sqrt{\xi_x^2 + \xi_y^2}$ and $\hat{F}^\pm$ is derived from (3.3) with

$$\begin{aligned} 2\lambda_1^\pm &= U \pm |U| & U &\equiv \xi_x u + \xi_y v + \xi_t \\ 2\lambda_3^\pm &= U + \tilde{c} \pm |U+\tilde{c}| & \tilde{c} &\equiv c\sqrt{\xi_x^2 + \xi_y^2} \\ 2\lambda_4^\pm &= U - \tilde{c} \pm |U-\tilde{c}| \end{aligned} \quad (3.6)$$

where $\tilde{k}_1 = \xi_x/(\xi_x^2 + \xi_y^2)^{1/2}$ and $\tilde{k}_2 = \xi_y/(\xi_x^2 + \xi_y^2)^{1/2}$.

A previously identified [32] difficulty with the above formulation is that $F^\pm$ and $G^\pm$ have discontinuous derivatives because $|\lambda|$ has a discontinuous derivative. As discussed in [32] it is necessary to smooth $|\lambda|$, and this is neatly accomplished [41] by replacing (3.4) with

$$2\lambda_i^\pm = \lambda_i \pm \sqrt{\lambda_i^2 + \epsilon} \quad (3.7)$$

where ϵ is small. As indicated in Fig. 9, this gives a smooth $\lambda_i^\pm$ variation which asymptotes to the old formulation. It also adds numerical dissipation whenever an eigenvalue changes sign. In numerical tests on a one-dimensional transonic nozzle the use of the new formulation (3.7) in place of (3.4) gives a smooth sonic line result that was not previously obtained, see Fig. 10.

Other split flux vectors have been proposed. For example, Bram van Leer [41] has suggested the form (here given in one-dimension)

$$u < c$$

$$F^+ = \begin{pmatrix} f_1^+ \\ f_2^+ \\ f_3^+ \end{pmatrix} = \begin{pmatrix} \rho c (M+1)^2 / 4 \\ \rho c^2 (M+1)^2 [(\gamma-1)M+2] / (4\gamma) \\ \gamma^2 [(f_2^+)^2 / f_1^+] / [2(\gamma^2-1)] \end{pmatrix} \quad \text{and } F^- = F - F^+ \quad (3.8)$$

$$u > c$$

$$F^+ = F \text{ and } F^- = 0$$

This splitting, devised from different arguments than (3.2), is naturally smooth at points where the eigenvalues change sign. In the above test problem use of the van Leer flux vector gives the pleasing result shown in Fig. 11.

Because upwind differencing schemes generate lower or upper triangular matrices, flux split implicit algorithms can be devised for the inviscid equations which are efficiently inverted. For example, a second order fully implicit differencing of (3.1) is obtained using three point upwind differencing in space and in time

$$(\delta_t^b Q + \delta_x^b F^+ + \delta_x^f F^- + \delta_y^b G^+ + \delta_y^f G^-)_{jk}^{n+1} = 0 \quad (3.9)$$

where

$$\delta_t^b Q^{n+1} = (3Q^{n+1} - 4Q^n + Q^{n-1}) / (2\Delta t)$$

$$\delta_x^f F_j = (-3Q_j + 4Q_{j+1} - Q_{j+2}) / (2\Delta x) \quad \text{etc.}$$

With use of local linearization to avoid iterative solution of the nonlinear terms, and with use of approximate factorization to simplify the inversion work, a delta form implicit differencing of (3.9) can be obtained as

$$(I + h\delta_{x+}^b A_+^n + h\delta_{y+}^b B_+^n)(I + h\delta_{x-}^f A_-^n + h\delta_{y-}^f B_-^n)(Q^{n+1} - Q^n) = \quad (3.10)$$

$$-h(\delta_{x+}^b F^+ + \delta_{x-}^f F^- + \delta_{y+}^b G^+ + \delta_{y-}^f G^-)^n + 1/3(Q^n - Q^{n-1})$$

where $h = (2\Delta t/3)$ and $A_+ = \frac{\partial F^+}{\partial Q}$, etc. Equation (3.10) can be put into its obvious algorithm form as

$$(I + h\delta_{x+}^b A_+^n + h\delta_{y+}^b B_+^n)\Delta Q^* = \text{RHS} \quad (3.11a)$$

$$(I + h\delta_{x-}^f A_-^n + h\delta_{y-}^f B_-^n)\Delta Q^n = \Delta Q^* \quad (3.11b)$$

$$Q^{n+1} = Q^n + \Delta Q^n \quad (3.11c)$$

where RHS represents the right hand side of (3.10). The first step of the algorithm (3.11a) requires a lower triangular inversion (i.e. solution) with 4x4 block elements. The second step, (3.11b), requires an upper triangular inversion. Both such solution processes are simple compared to the block tridiagonal inversions required with (2.8) when applied to only inviscid flow. The standard solution scheme (2.8) is still competitive with (3.10), however, because $A^\pm$ and $B^\pm$ are much more costly to form than A and B.

Various other implicit algorithms are possible with flux splitting [c.f. 32]. If the thin-layer viscous terms are included the following differencing has merit

$$\begin{aligned} & [I + h(\delta_{x+}^b A_+^n + \delta_{y+}^b B_+^n + \bar{\delta}_y M^n)][I + h\delta_{x-}^f A_-^n](Q^{n+1} - Q^n) \\ & = -h(\delta_{x+}^b F^+ + \delta_{x-}^f F^- + \delta_{y+}^b G^+ + \bar{\delta}_y S)^n + \alpha(Q^n - Q^{n-1})/3 \end{aligned} \quad (3.12)$$

where $h = \frac{(3 - \alpha)\Delta t}{3}$, $\alpha = 0$ or 1 for first or second order accuracy, and δ_y and $\bar{\delta}_y$ are the central difference operators defined previously. A solution algorithm for (3.12) entails block tridiagonal inversions carried on with a forward sweep in x, followed by a simple backsweep in the x-direction.

The schemes given by (3.10) and (3.12) have not yet been applied to as complex geometry situations as the Beam-Warming class of algorithms represented by (2.8). The scheme (3.10) has been used on stretched grids to compute inviscid transonic flow about a biconvex airfoil; however, thin airfoil boundary conditions were employed. A typical solution is shown in Fig. 12. This result was computed without the benefit of the transition smoothing, (3.7). A viscous supersonic wedge flow calculation using (3.12) is indicated by Fig. 13. This result is an old one that used an earlier flux splitting, namely (for F^+)

$$\begin{aligned} 2\lambda_1^+ &= u + |u| & 2\lambda_1^- &= u - |u| \\ 2\lambda_3^+ &= u + |u| + c & 2\lambda_3^- &= u - |u| \\ 2\lambda_4^+ &= u + |u| & 2\lambda_4^- &= u - |u| - c \end{aligned}$$

In this case, the exact geometry was fitted using shear transforms and a very fine grid was needed to resolve the viscous layer. A preliminary version of the turbulence model described in [9] was used in the calculation.

IV. REDUCED SYSTEMS.

Time-accurate or time-like iterative methods are frequently used to obtain steady state solutions. If only a steady state solution is sought, however, one can attempt to precondition and otherwise try to reduce the system of partial differential equations to obtain a more efficient solution. Not surprisingly, certain reductions of the Euler equations can begin to take on features of classical aerodynamic formulations. One such formulation [42], discarded several years ago in favor of the schemes discussed earlier, is being revived because of its excellent computational efficiency for steady rotational subsonic flow.

The nonconservative form of the Euler equations can be written in the matrix form

$$\tilde{A} \partial_x \tilde{Q} + \tilde{B} \partial_y \tilde{Q} = 0 \quad (4.1)$$

where

$$\tilde{Q} = \begin{pmatrix} \rho \\ u \\ v \\ p \end{pmatrix} \quad \tilde{A} = \begin{pmatrix} u & \rho & 0 & 0 \\ 0 & u & 0 & \rho^{-1} \\ 0 & 0 & u & 0 \\ 0 & \gamma p & 0 & u \end{pmatrix} \quad \tilde{B} = \begin{pmatrix} v & 0 & \rho & 0 \\ 0 & v & 0 & 0 \\ 0 & 0 & v & \rho^{-1} \\ 0 & 0 & \gamma p & v \end{pmatrix}$$

If the x-axis is aligned with the mean flow direction, one can perturb A and B about a reference state, o, with

$$u = u_o, v = 0, p = p_o, \rho = \rho_o. \quad (4.2)$$

Then (4.1) can be put into the perturbation form

$$\partial_x \tilde{Q} + \tilde{A}_o^{-1} \tilde{B}_o \partial_y \tilde{Q} = \epsilon \quad (4.3)$$

where

$$\epsilon = (\tilde{A}_o^{-1} \tilde{B}_o - \tilde{A}^{-1} \tilde{B}) \partial_y \tilde{Q} \quad (4.4a)$$

or

$$\epsilon = \tilde{A}_o^{-1} [(\tilde{A}_o - \tilde{A}) \partial_x \tilde{Q} + (\tilde{B}_o - \tilde{B}) \partial_y \tilde{Q}] \quad (4.4b)$$

The ϵ -formulation (4.4a) is obtained by multiplying (4.1) by $\tilde{A}^{-1}$ and placing $\tilde{A}^{-1} \tilde{B}$ in the perturbation form $\tilde{A}^{-1} \tilde{B} = \tilde{A}_o^{-1} \tilde{B}_o - (\tilde{A}_o^{-1} \tilde{B}_o - \tilde{A}^{-1} \tilde{B})$. To obtain the ϵ -formulation (4.4b), $\tilde{A}$ and $\tilde{B}$ are put in perturbation form and the equations are multiplied by $\tilde{A}_o^{-1}$. The (4.4a) formulation has been successfully used in numerical calculations (as described later) and requires fewer

derivative operators than the (4.4b) formulation. The (4.4b) formulation is not singular at a sonic or stagnation point, but it has not been attempted in numerical calculation.

The importance of (4.3) is the simplicity of the left hand side. As indicated in (4.2), v_0 is taken as zero so $\tilde{A}_0^{-1}\tilde{B}_0$ has the reduced form

$$\tilde{A}_0^{-1}\tilde{B}_0 = \begin{pmatrix} 0 & 0 & -\rho_0 u_0 \phi_0 & 0 \\ 0 & 0 & c_0^2 \phi_0 & 0 \\ 0 & 0 & 0 & (u_0 \rho_0)^{-1} \\ 0 & 0 & -\rho_0 u_0 c_0^2 \phi_0 & 0 \end{pmatrix} \quad (4.5)$$

where $\phi_0 = 1/(c_0^2 - u_0^2)$ and c_0 is the reference sound speed. The eigenvalues of $\tilde{A}_0^{-1}\tilde{B}_0$ are:

$$\sigma(\tilde{A}_0^{-1}\tilde{B}_0) = 0, 0, \pm \frac{c_0}{\sqrt{u_0^2 - c_0^2}}$$

For subsonic mean flow the nonzero eigenvalues are imaginary, for supersonic mean flow they are real. Thus the left hand side part reflects elliptic or hyperbolic behavior as u_0 is less than or greater than c_0 .

As the above matrix makes clear, only the third and fourth equations of the system (4.3) are strongly coupled. Written out we obtain

$$\partial_x \rho - \rho_0 u_0 \phi_0 \partial_y v = \epsilon_1 \quad (4.6a)$$

$$\partial_x u + c_0^2 \phi_0 \partial_y v = \epsilon_2 \quad (4.6b)$$

$$\partial_x v + (\rho_0 u_0)^{-1} \partial_y p = \epsilon_3 \quad (4.6c)$$

$$\partial_x p - \rho_0 u_0 c_0^2 \phi_0 \partial_y v = \epsilon_4 \quad (4.6d)$$

Thus if ϵ_i are known from some previous estimate, we can solve equations (4.6c) and (4.6d) for v and p . Once v is obtained, ρ and u are obtained from (4.6a) and (4.6b) via simple integration. Better estimates of ϵ_i can now be formed and the process is repeated until iterative solution of (4.1) is obtained.

The equations (4.6c) and (4.6d) form an elliptic system when $u_o < c_o$. This is readily apparent as the eigenvalues of

$$\begin{pmatrix} 0 & (u_o \rho_o)^{-1} \\ -\rho_o u_o c_o^2 \phi_o & 0 \end{pmatrix} \quad (4.7)$$

are imaginary as indicated previously. The equations, which if linearized by setting $\epsilon_i = 0$ could be transformed into Cauchy Riemann equations, can be differentiated to form a Poisson equation with either v or p as the dependent variable. For example

$$[1 - (u_o/c_o)^2] \partial_{xx} v + \partial_{yy} v = [(1 - (u_o/c_o)^2) (\partial_x \epsilon_3 - (\rho_o u_o)^{-1} \partial_y \epsilon_4)] \quad (4.8)$$

Once v is obtained, p is found by integrating (4.6d). If $u_o > c_o$, (4.8) represents a wave equation and the eigenvalues of (4.7) are real.

The (4.4a) formulation using (4.8) to update v has been successfully applied in two-dimensions [42] to compute rotational subsonic flow. Typical results for a lifting biconvex airfoil are shown in Fig. 14 for the incoming shear flow (jet or defect) profiles as depicted in Fig. 15. Computational times for a fully converged solution on a 59x90 grid averaged 18 seconds per case on a Control Data 7600 computer.

The algorithm using (4.8) is robust. For inexplicable reasons, elimination of v from (4.6c) and (4.6d) and solution of a Poisson equation for pressure, p , has always been a disappointment. The failure is believed to be keyed to the boundary condition treatment.

The above ideas readily extend to three-dimensions although no calculations have yet been undertaken. The three-dimensional perturbation form of the equations is given by:

$$\partial_x \bar{Q} + \bar{A}_0^{-1} \bar{B}_0 \partial_y \bar{Q} + \bar{A}_0^{-1} \bar{C}_0 \partial_z \bar{Q} = \epsilon \quad (4.9)$$

In this case, v , w , and p are strongly coupled through

$$\partial_x v + (\rho_0 u_0)^{-1} \partial_y p = \epsilon_3 \quad (4.10a)$$

$$\partial_x w + (\rho_0 u_0)^{-1} \partial_z p = \epsilon_4 \quad (4.10b)$$

$$\partial_x p - (\rho_0 u_0 c_0^2 \phi_0) (\partial_y v + \partial_z w) = \epsilon_5 \quad (4.10c)$$

Once v , w , p are obtained, ρ and u are found by integration of

$$\partial_x \rho = \epsilon_1 + (\rho_0 u_0 \phi_0) (\partial_y v + \partial_z w) \quad (4.10d)$$

$$\partial_x u = \epsilon_2 - (c_0^2 \phi_0) (\partial_y v + \partial_z w) \quad (4.10e)$$

Equations (4.10a) to (4.10c) can be differentiated so that v and w are eliminated to form a Poisson equation in pressure, that is

$$[1 - (u_0/c_0)^2] \partial_{xx} p + \partial_{yy} p + \partial_{zz} p = g \quad (4.11)$$

Alternately, as this was unsuccessful in two-dimensions, pressure can be eliminated and vector potential like equations can be formed for v and w . In particular

$$[1 - (u_0/c_0)^2] \partial_{xx} v + \partial_{yy} v + \partial_{yz} w = f_1 \quad (4.12a)$$

$$[1 - (u_0/c_0)^2] \partial_{xx} w + \partial_{zz} w + \partial_{yz} v = f_2 \quad (4.12b)$$

Here again, if (4.12a) and (4.12b) are solved for v and w , then p , ρ , and u can be found by simple integrations of (4.10c) to (4.10e).

Although not discussed previously, one can draw some interesting analogies between (4.10a) - (4.10c) to the incompressible irrotational relations

$$\partial_x u + \partial_y v + \partial_z w = 0 \quad (4.13a)$$

$$\partial_y w - \partial_z v = 0 \quad (4.13b)$$

$$\partial_z u - \partial_x w = 0 \quad (4.13c)$$

$$\partial_x v - \partial_y u = 0 \quad (4.13d)$$

The potential

$$\phi_x = u, \phi_y = v, \phi_z = w \quad (4.14)$$

satisfies (4.13b) to (4.13d) and from (4.13a) gives the Laplacian

$$\phi_{xx} + \phi_{yy} + \phi_{zz} = 0 \quad (4.15)$$

The particular vector potentials ψ , χ defined as [43]

$$u = \psi_y + \chi_z, v = -\psi_x, w = -\chi_x \quad (4.16)$$

satisfies (4.13a) and from (4.13c) and (4.13d) give

$$\psi_{xx} + \psi_{yy} + \chi_{yz} = 0 \quad (4.17a)$$

$$\chi_{xx} + \chi_{zz} + \psi_{yz} = 0 \quad (4.17b)$$

The analogies between (4.11) and (4.15) and between (4.12) and (4.17) are completely transparent and extend even to the boundary condition treatment. Curiously, (4.15) is elliptic, but (4.17) is not strictly elliptic according to the definition of [44]. Moreover, the system of equations comprised of (4.13a), (4.13c), and (4.13d) is not elliptic. This latter system presents no difficulty to numerical solution, however, and conventional numerical algorithms such as successive overrelaxation have been successfully applied to the solution of (4.17). It is conjectured, therefore, that one could devise rapid solution procedures for (4.12) and (4.10c) through (4.10e) since a similar approach worked in two-dimensions.

The perturbation schemes described here have only been used in numerical calculation of pure subsonic two-dimensional flow. Extensions to pure supersonic flow appear to be straight forward but such is not the case for transonic flow. The isentropic primitive variable approach of Martin [45] and the stream function sonic line treatment of Hafez [46] may offer guidance, however. Note there is no problem bringing the essentials of the perturbation formulation into conservative form. For example

$$\partial_x \bar{Q} + \bar{A}_O^{-1} \bar{B}_O \partial_y \bar{Q} = \bar{A}_O^{-1} [\partial_x (\bar{A}_O \bar{Q} - N_O^{-1} F) + \partial_y (\bar{B}_O \bar{Q} - N_O^{-1} G)] \quad (4.18)$$

where the matrix N relates Q to $\bar{Q}$ via

$$N = \frac{\partial Q}{\partial \bar{Q}}$$

and Q , F , G are the Cartesian form of the conservative fluxes of (2.1). The left hand side of (4.18) is identical to (4.3).

Rather than attempt to extend the method of this section into the realm of transonic flow, it may be better to use this procedure as the outer part of a chimera (i.e., multiple constituent) algorithm. For example, the semi-flux-split scheme (3.12) can be used to solve supersonic and subsonic flow, and it is very efficient for supersonic and high subsonic regions.

The scheme of this section is very efficient for subsonic flow. A combined chimera method for transonic flow simulation about an airfoil might entail use of the schemes (4.3) and (3.12). The outer subsonic flow would be solved with (4.3), the inner high subsonic and supersonic flow with (3.12), and both solution regions could be slightly overlapped [42,47] for iterative efficiency.

V. CONCLUDING REMARKS.

The successful simulation of the flow about really complex geometries will require further advances than those discussed here. Nevertheless, the combination of implicit finite difference procedures, generalized coordinates, and numerical grid generation techniques is proving to be effective. The overall methodology can be built on, and improvements in computational efficiency can be expected. In this overall development it would seem that the next crucial step is to develop efficient chimera schemes - that is, modularly coded numerical schemes that blend or overlay more than one grid system and more than one type of governing equation or numerical algorithm.

REFERENCES

1. Steger, J. L., Implicit finite-difference simulation of flow about arbitrary two-dimensional geometries, AIAA Journal 16 (1978), 679-686.
2. Pulliam, T. H. and J. L. Steger, Implicit finite-difference simulations of three-dimensional compressible flow, AIAA Journal 18 (1980), 159-167.
3. Nietubicz, C. J., T. H. Pulliam, and J. L. Steger, Numerical solution of the Azimuthal-invariant thin-layer Navier-Stokes equations, AIAA Paper No. 79-0010, 1979.
4. Steger, J. L. and H. E. Bailey, Calculation of transonic aileron buzz, AIAA Journal 18 (1980), 249.
5. Thomas, P. D. and C. K. Lombard, Geometric conservation law and its application to flow computations on moving grids, AIAA Journal 17 (1979), 1030-1037.

6. Pulliam, T. H. and D. S. Chaussee, A diagonal form of an implicit approximate-factorization algorithm, Journal of Computational Physics, 1980.
7. Steger, J. L. and P. Kutler, Implicit finite-difference procedures for the computation of vortex wakes, AIAA Journal 15 (1977), 581-590.
8. Schiff, L. B. and J. L. Steger, Numerical simulation of steady supersonic viscous flow, AIAA Journal 12 (1980), 1421-1430.
9. Baldwin, B. S. and H. Lomax, Thin layer approximation and algebraic model for separated turbulent flows, AIAA Paper No. 78-257, 1978.
10. Viviand, H., Conservative forms of gas dynamic equations, La Recherche Aerospaciale 1 (1974), 65-68.
11. Lapidus, A., A detached shock calculation by second-order finite differences, Journal of Computational Physics 2 (1967), 154-177.
12. Eiseman, P. R., A coordinate system for a viscous transonic cascade analysis, Journal of Computational Physics 26 (1978), 307-338.
13. Graves, Jr., R. A., Application of a numerical orthogonal coordinate generator to axisymmetric blunt bodies, NASA TM 80131, 1979.
14. McNally, W. D., Fortran program for generating a two-dimensional orthogonal mesh between two arbitrary boundaries, NASA TN D-6766, 1972.
15. Winslow, A. M., Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh, J. Comp. Phys. 2 (1967), 149-172.
16. Chu, W. H., Development of a general finite difference approximation for a general domain, J. Comp. Phys. 8 (1971), 392-408.
17. Godunov, S. K. and G. P. Prokopov, The use of moving meshes in gas-dynamical computations, USSR Comput. Math. Math. Phys., 12, 2 (1972), 182-195.

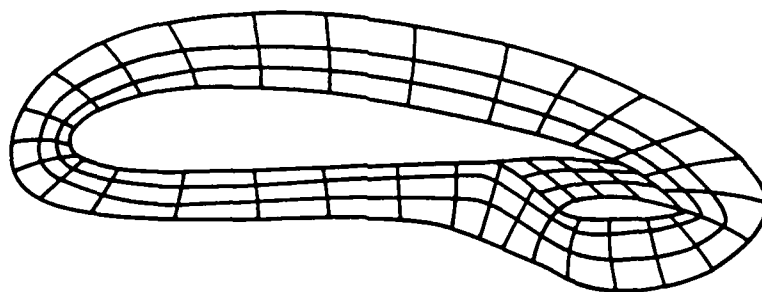
18. Thompson, J. F., F. C. Thames, and C. M. Mastin, Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies, *J. Comp. Phys.* 15 (1974), 299-319.
19. Steger, J. L. and R. L. Sorenson, Automatic mesh-point clustering near a boundary in grid generation with elliptic partial differential equations, *Journal of Computational Physics* 33 (1979), 405-410.
20. Sorenson, R. L., A computer program to generate two-dimensional grids about airfoils and other shapes by the use of Poisson's equation, NASA TM-81198, May 1980.
21. Starius, G., Constructing orthogonal curvilinear meshes by solving initial value problems, *Numerische Mathematik* 28 (1977), 25-48.
22. Steger, J. L. and D. S. Chaussee, Generation of body-fitted coordinates using hyperbolic partial differential equations, *SIAM J. Sci. Stat. Comput.* 1 (1980), 431-437.
23. Steger, J. L., T. H. Pulliam, and R. V. Chima, An implicit finite difference code for inviscid and viscous cascade flow, AIAA Paper No. 80-1427, 1980.
24. Shang, J., An implicit-explicit method for solving the Navier-Stokes equations, *AIAA Journal* 16 (1978), 496-502.
25. Beam, R. and R. F. Warming, An implicit finite-difference algorithm for hyperbolic systems in conservation-law-form, *Journal of Computational Physics* 22 (1976), 87-110.
26. Warming, R. F. and R. M. Beam, On the construction and application of implicit factored schemes for conservation laws, Symposium on Computational Fluid Dynamics, New York, Apr. 16-17, 1977; *SIAM-AMS Proceedings* 11 (1978), 85-129.
27. Briley, W. R. and H. McDonald, An implicit numerical method for the multi-dimensional compressible Navier-Stokes equations, United Aircraft Research Laboratories, Rept. M911363-6, 1973.
28. McDonald, H. and W. R. Briley, Computational fluid dynamics aspects of internal flows, AIAA Paper 79-1445, July 1979.

29. Erickson, A. L. and J. D. Stephenson, A suggested method of analyzing for transonic flutter of control surfaces based on available experimental evidence, NACA RM A7F30, Dec. 1947.
30. Lambourne, N. C., Some instabilities arising from the interactions between shock waves and boundary layers, AGARD Report 182, 1958.
31. Steger J. L., Coefficient matrices for implicit finite difference solution of the inviscid fluid conservation law equations, Computer Methods in Applied Mechanics and Engineering 13 (1978).
32. Steger, J. L. and R. F. Warming, Flux vector splitting of the inviscid gasdynamics equations with applications to finite difference methods, NASA TM-78605, 1979.
33. Anucina, N. N., Difference schemes for solving the Cauchy problem for symmetric hyperbolic systems, Soviet Mathematics 5 (1964), 60.
34. Gordon, P., The diagonal form of quasi-linear hyperbolic systems as a basis for difference equations. Sec. II-D of General Electric Company Document No. 69SD7056 (Final Report - Theoretical Investigation of Nuclear Burst Phenomena, Naval Ordnance Laboratory, Contract No. N 60921-7164), 1969.
35. Moretti, G., The λ -scheme, Computers and Fluids 7 (1979), 191.
36. Chakravarthy, S. R., D. A. Anderson, and M. D. Salas, The split coefficient matrix method for hyperbolic systems of gas-dynamic equations, AIAA Paper 80-0268, Pasadena, CA, 1980.
37. Daywitt, J., Improvements in techniques for computing supersonic blunt-body flows, AIAA Paper No. 81-0115, 1981.
38. van Leer, B., On the relation between the upwind-differencing schemes of Godunov, Engquist-Osher and Roe, ICASE Rep. 81-11, 1981.
39. Jespersen, D., Private communication, (1979).

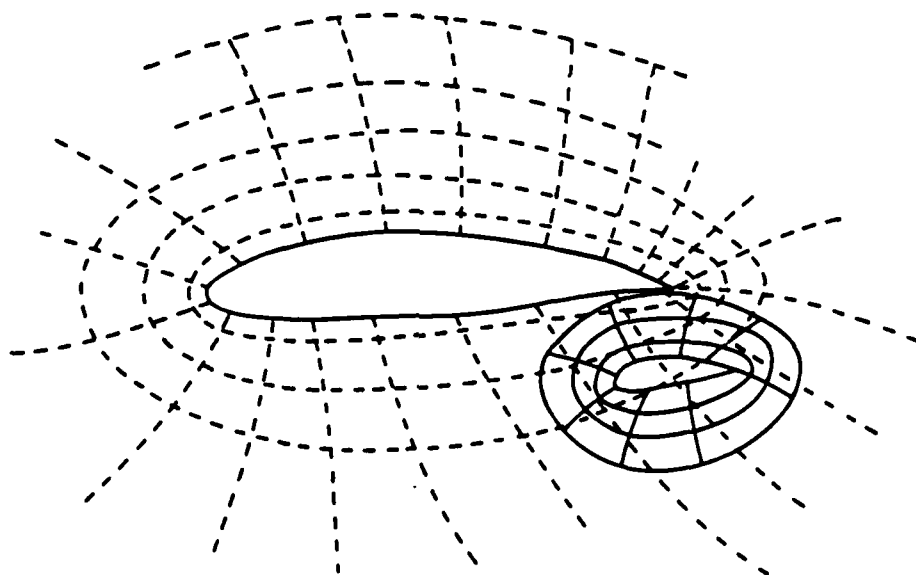
40. Warming R. F., R. Beam, and B. J. Hyett, Diagonalization and simultaneous symmetrization of the gas-dynamic matrices, *Mathematics in Computation* 29 (1975), 1037-1045.
41. Steger, J. L., A preliminary study of relaxation methods for the inviscid conservative gasdynamics equations using flux splitting, NASA CR 3415, 1981.
42. Steger, J. L. and H. Lomax, Calculation of inviscid shear flow using a relaxation method for the Euler equations, NASA SP-347, 1975, 811-836.
43. Moore, F. K., Three-dimensional compressible laminar boundary layer flow, NACA TN 2279, 1951.
44. Agmon, D. A. and L. Nirenberg, Estimates near the boundary for solutions of elliptic partial differential equations satisfying general boundary conditions, *Comm. Pure Appl. Math.* 17 (1964), 35-92.
45. Martin, E. D., Progress in application of direct elliptic solvers to transonic flow computations, *Aerodynamic Analyses Requiring Advanced Computers, Part II*, NASA SP-347, 1975, 839-870.
46. Hafez, M. M., Numerical solution of transonic full stream function equations in conservative form, Flow Research Note No. 178, Flow Research Company, Kent, Washington.
47. Atta, E., Component-adaptive grid interfacing, AIAA Paper No. 81-0382, 1981.

The author was partially supported by Army Research Office Contract DAAG29-81-K-0013.

Department of Aeronautics and
Astronautics
Stanford University
Stanford, CA 94305



Sketch 1. Showing patched grid system with all body surfaces mapped to grid outer boundaries. This approach requires interpolation along boundaries.



Sketch 2. Showing overset grid system. In this approach various interior points must be turned-off and interpolation is required at grid interfaces, however, grids are easy to generate and have minimum distortion.

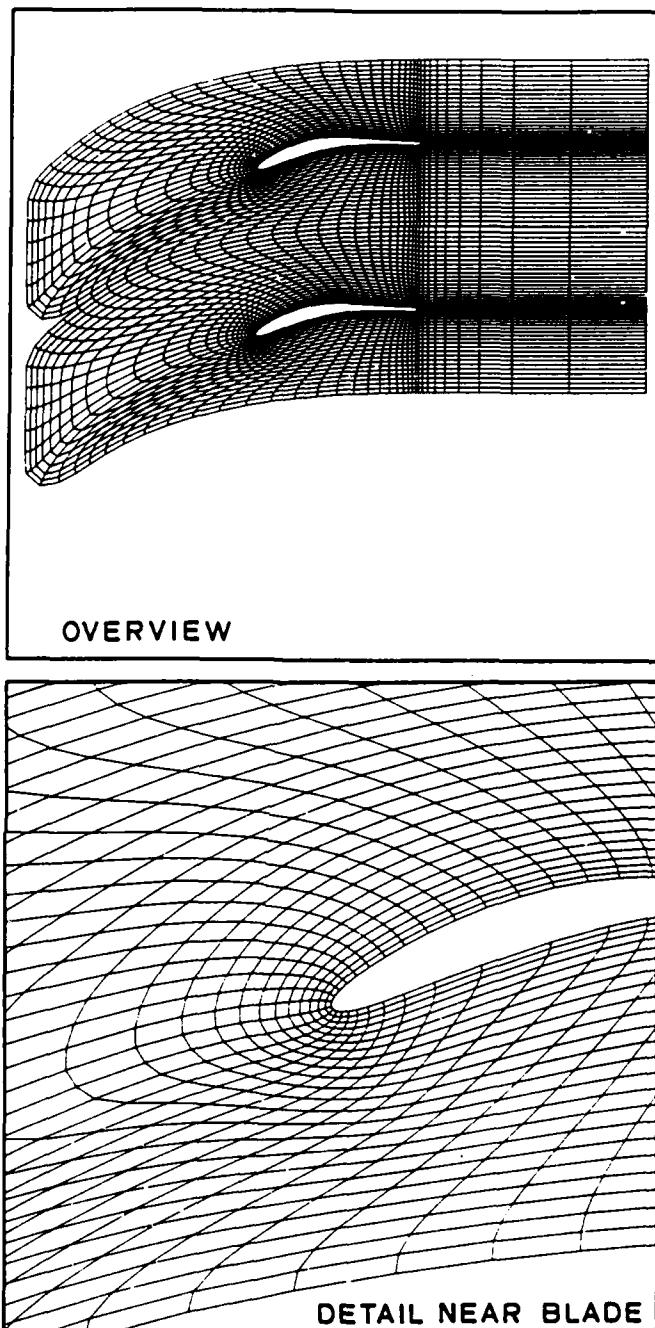


Fig. 1. Cascade grid generated using elliptic partial differential equations.

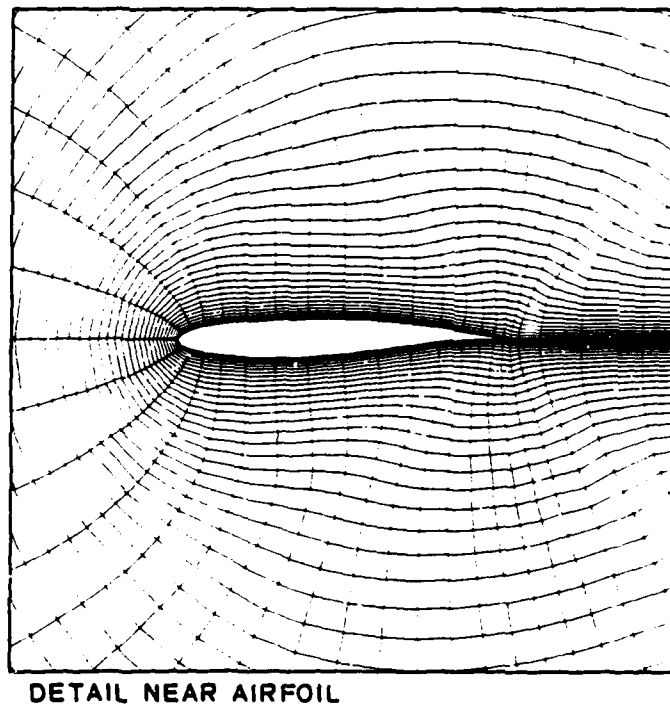
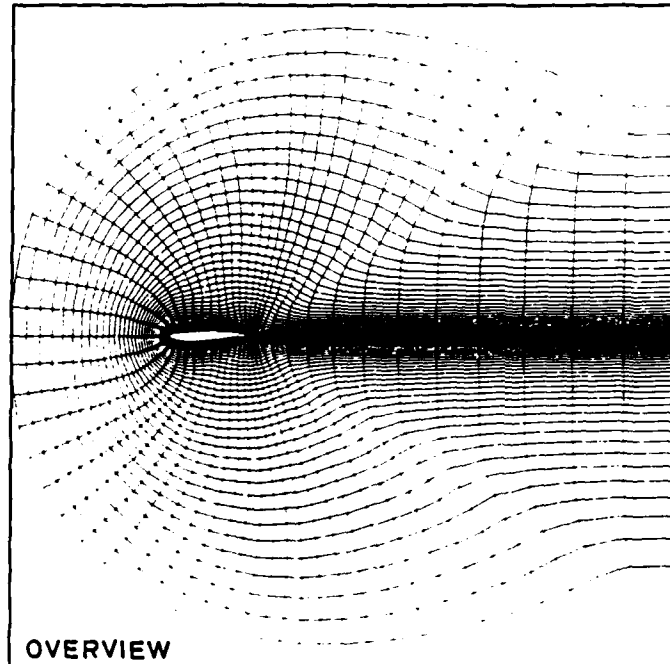


Fig. 2 Grid generated about airfoil using hyperbolic partial differential equations.

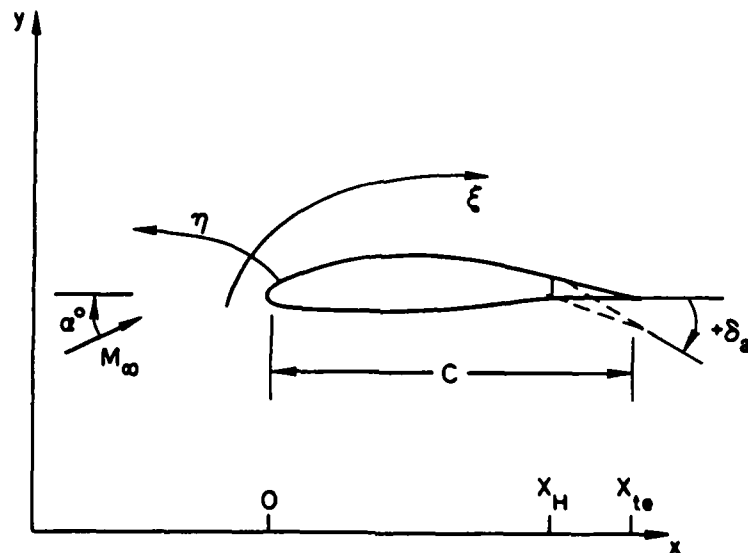


Fig. 3. Schematic defining geometric quantities.

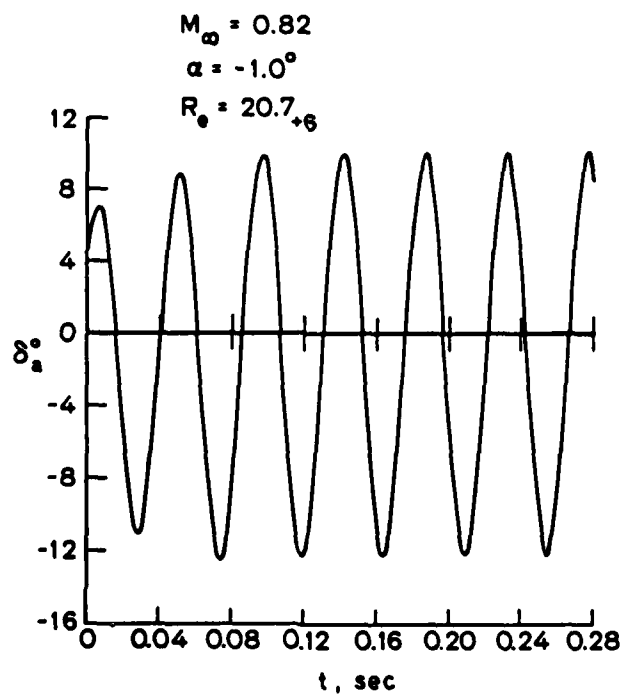


Fig. 4. Computed variation of aileron angle with time for NACA 65 213 airfoil.

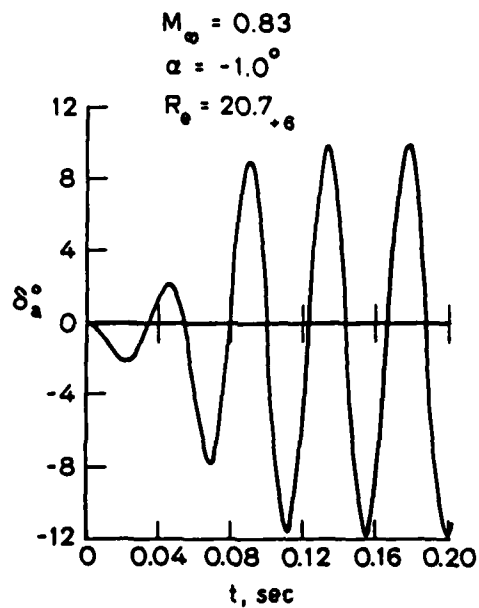


Fig. 5. Computed variation of aileron angle with time; aileron initially set at 0 deg.

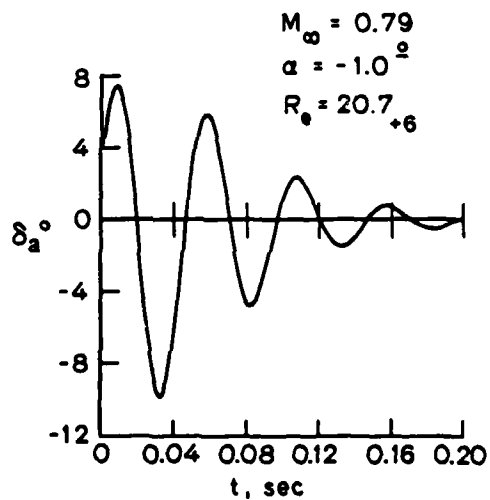


Fig. 6. Decay of aileron buzz with time.

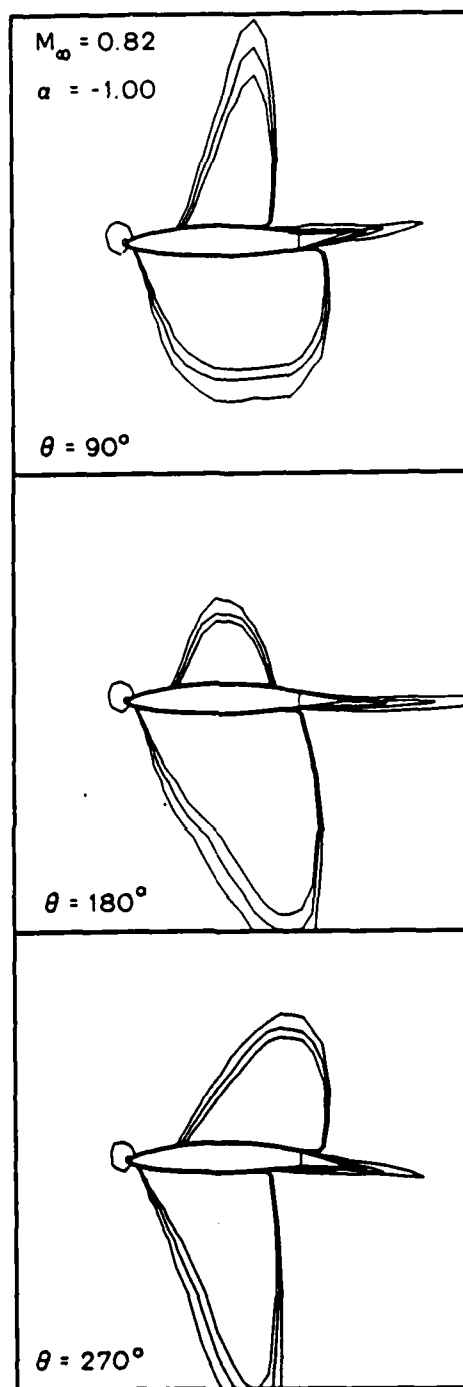


Fig. 7. Mach contour lines during an aileron buzz cycle, $M_\infty = 0.82$, $\alpha = -1$ deg, $R_e = 20.7 \times 10^6$. Here $\theta = (t/t_p) * 360$ deg where t_p is the time to complete one cycle.

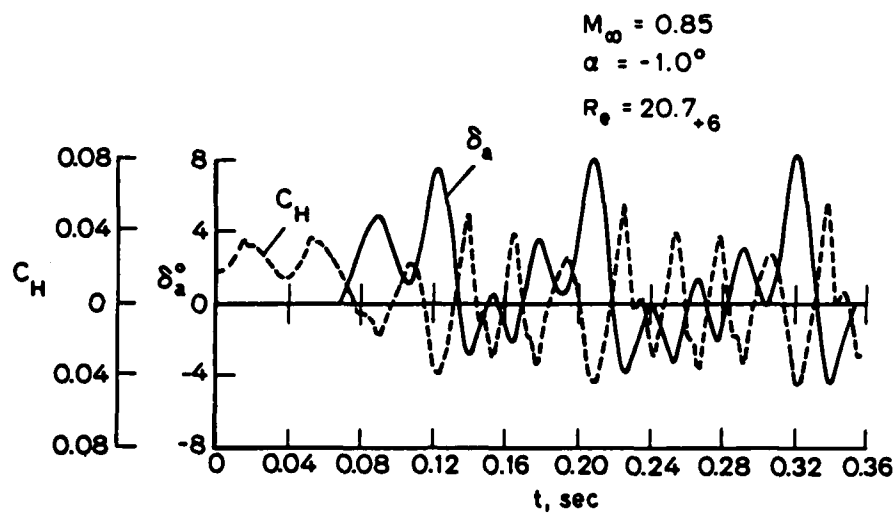


Fig. 8. Growth of aileron deflection with time.

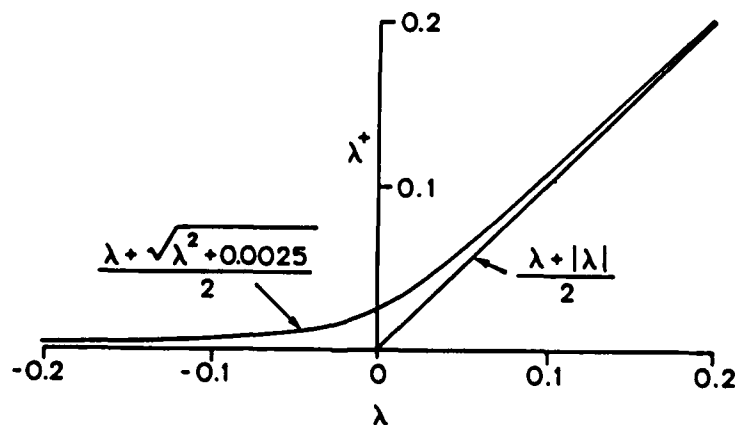


Fig. 9. Comparison of continuous derivative λ^+ formulation to discontinuous derivative λ^+ formulation.

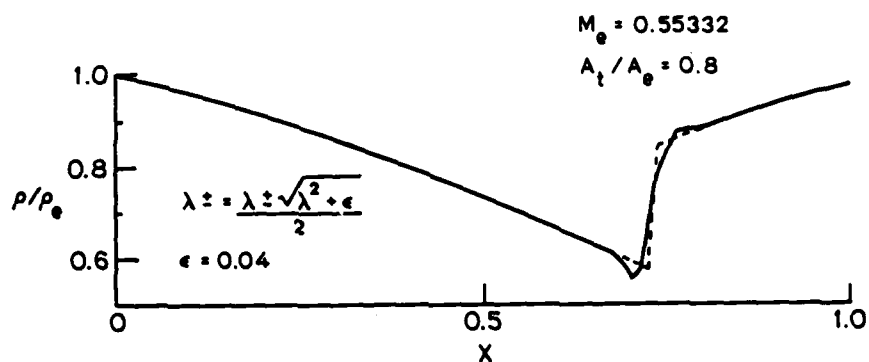
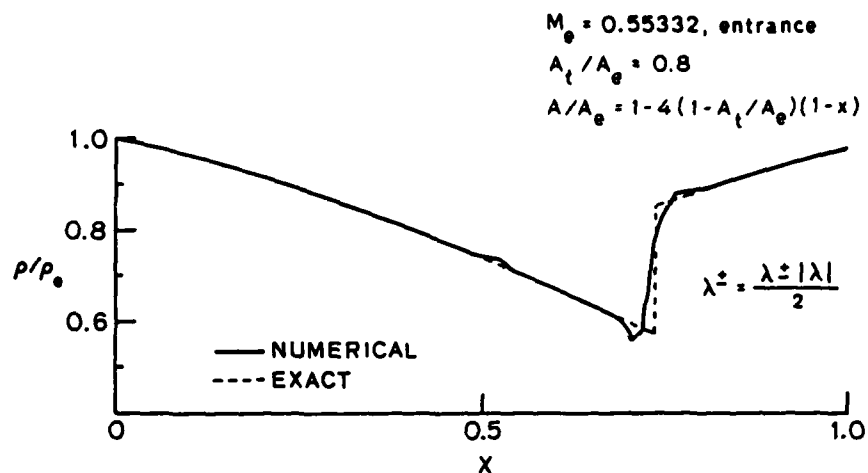


Fig. 10. Computational comparison of λ^+ formulations.

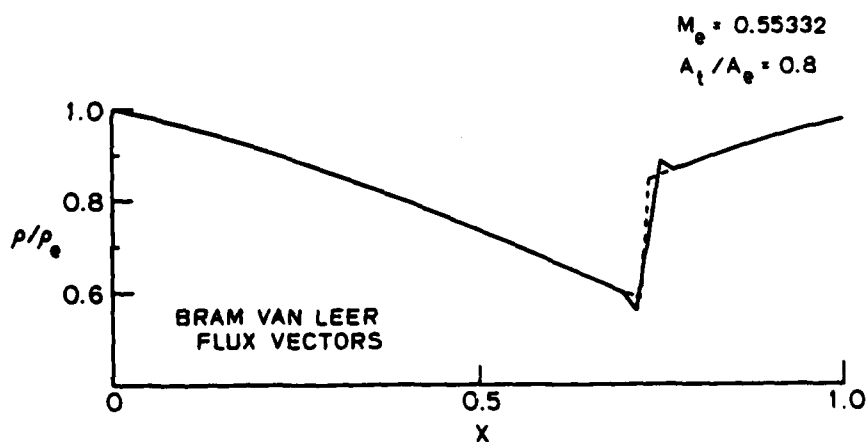


Fig. 11. Solution to one dimensional nozzle using continuous derivative flux splitting - Bram van Leer flux vectors.

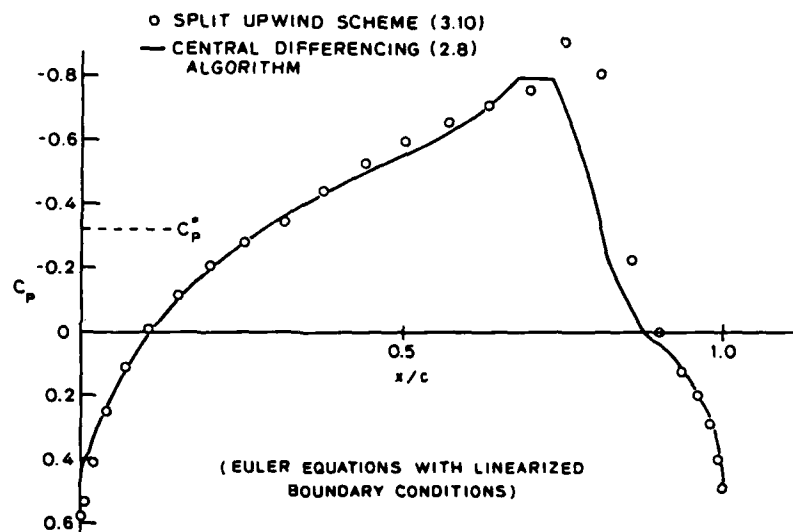


Fig. 12. Steady state solution for 11.4% thick parabolic arc airfoil, $M_\infty=0.84$.

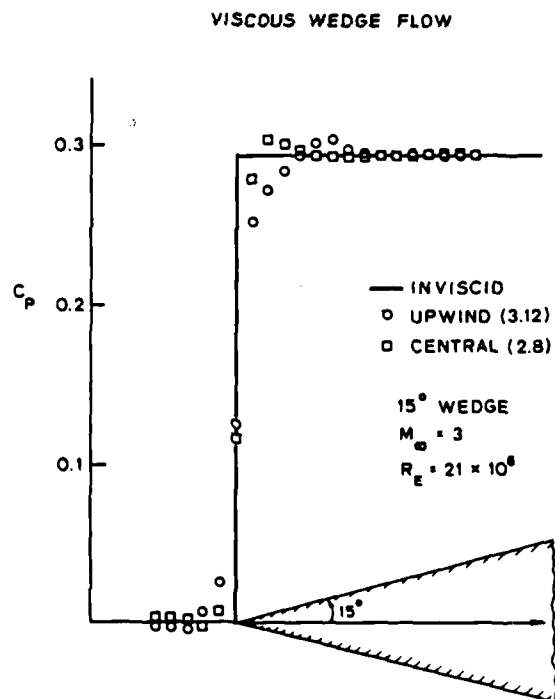
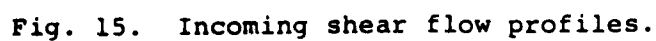
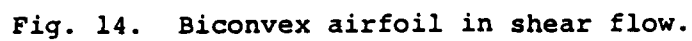


Fig. 13. Steady state viscous wedge flow solutions.



**On Application of Body Conforming Curvilinear Grids
for Finite Difference Solution of External Flow
in Numerical Grid Generation**

from

Numerical Grid Generation

J. F. Thompson, Editor

North-Holland, 1982

ON APPLICATION OF BODY CONFORMING CURVILINEAR GRIDS
FOR FINITE DIFFERENCE SOLUTION OF EXTERNAL FLOW

Joseph L. Steger

Department of Aeronautics and Astronautics

Stanford University, Stanford, CA 94305

I. INTRODUCTION

Finite difference practitioners frequently make use of arbitrary coordinate transforms and introduce body conforming curvilinear grid systems. The coordinate transforms may either be built in globally in mappings from physical space to computational space, or they may be built in locally in the finite volume sense. The advantages of using body conforming curvilinear grids in finite difference flow field simulation include the following: Body conforming grids simplify the application of boundary conditions insofar that grid lines will coincide with the body boundary. Curvilinear grids may be clustered to flow field action regions to improve solution accuracy. Body conforming grids may allow simplification of the governing equations. Such grids can also help maintain a well-ordered system of algebraic equations suitable for vector-computer processing or approximate-factorization-implicit techniques.

The task of generating suitable body conforming curvilinear grids is not an easy one. The grids should be generated in an automatic manner requiring minimum user input. Yet the user will wish to maintain considerable control of where points will be distributed along the boundary surface and how they are clustered in the interior field. Moreover the grid must be tailored in some degree to the numerical algorithm because some numerical algorithms are more sensitive than others to grid smoothness, skewness, and stretching.

Although use of body conforming curvilinear grids can offer the advantages cited previously their careless application can lead to difficulties. This is particularly true when the governing equations are differenced in conservative (i.e., divergence) form and transform metric terms are brought inside the difference operators. Then as noted above, some numerical algorithms are far more mesh-sensitive than others, and numerical accuracy

BEGIN HERE

and computational efficiency can be affected by how rapidly a grid changes or how far away it is from orthogonality.

The subject of this paper is not the generation of body conforming curvilinear grids; rather it is the use of such grids in finite difference applications. In Section II of this paper the difficulties of solving the transformed equations in conservative form are discussed. In Section III various experiences are cited to suggest that considerable computational efficiency can yet be gleaned by further improvements of the grid. Concluding remarks follow in Section IV.

II. CONSERVATIVE DIFFERENCING OF TRANSFORMED EQUATIONS

a) Background

In aerodynamics applications we frequently try to difference the governing equations in conservative or divergence form. Conservative form differencing is preferred because it best maintains the correct weak solution of the governing equations. Thus if a shock wave is captured by simply solving the difference equations (as opposed to fitting a shock wave discontinuity into the difference equations), then the speed, location and jump of the shock can only be correct if the partial differential equations are in conservative form. The difference equations must also satisfy the divergence relation, at least in the vicinity of the shock. The conservative form of the equations may also be desirable for its numerical properties. For example, nonlinear equations in conservative form can be more cleanly linearized about a previous state than those in nonconservative form. This can be advantageous in implicit marching procedures in order to avoid iterative solutions of nonlinear equations with each marching step.

Let the conservation-law-form of the equations be represented in Cartesian coordinates

as

$$\partial_t Q + \partial_x F + \partial_y G = 0 \quad (1)$$

LINES TO
BOTTOM

where for simplicity only two dimensions are considered. This strict divergence form of the equations can be maintained for new independent variables

AD-A137 935

COMPUTATION OF VISCOUS FLOW ABOUT ADVANCED PROJECTILES

2/2

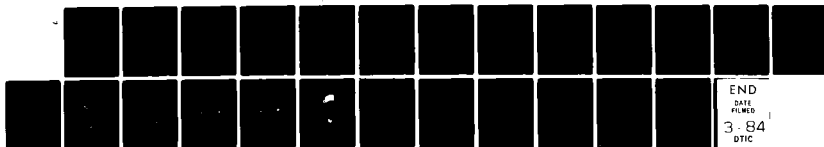
(U) STANFORD UNIV CA DEPT OF AERONAUTICS AND

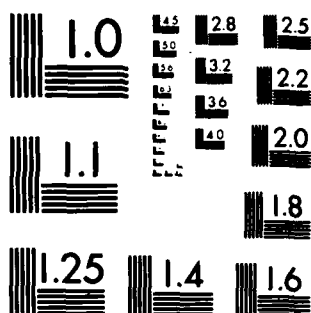
ASTRONAUTICS J L STEGER 09 SEP 83 SUDAA-CFD-83-2

UNCLASSIFIED ARO-18000.3-EG DAAG29-81-K-0013

F/G 20/4

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963 A

$$\begin{aligned}\xi &= \xi(x, y, t) \\ \eta &= \eta(x, y, t) \\ \tau &= t\end{aligned}\tag{2}$$

as (c.f. [1])

$$\partial_\tau \dot{Q} + \partial_\xi \dot{F} + \partial_\eta \dot{G} = 0\tag{3}$$

where

$$\dot{Q} = J^{-1}Q\tag{4a}$$

$$\dot{F} = J^{-1}(\xi_\tau Q + \xi_x F + \xi_y G)\tag{4b}$$

$$\dot{G} = J^{-1}(\eta_\tau Q + \eta_x F + \eta_y G)\tag{4c}$$

and where J is the transform Jacobian

$$J = \xi_x \eta_y - \xi_y \eta_x\tag{5}$$

For a thermally perfect gas Q , F , and G may represent the Cartesian inviscid and viscous flux quantities for conservation of mass, momentum and energy. For example, for inviscid flow

$$Q = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix}, \quad F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(e + p) \end{pmatrix}, \quad G = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(e + p) \end{pmatrix}\tag{6}$$

where ρ is fluid density, u , v and w are Cartesian velocities components, p is pressure and e is given by

$$e = (\gamma - 1)^{-1} p + \frac{1}{2} \rho (u^2 + v^2)\tag{7}$$

Alternately in the case of compressible potential flow

$$Q = \rho, \quad F = \rho \phi_x, \quad G = \rho \phi_y\tag{8}$$

and $\rho = \rho(\phi)$ is determined by the Bernoulli relation.

Although the transformed governing equations (3) are more complex than (1), apparent simplicity is returned to the inviscid flow equations with introduction of the unscaled

contravariant velocities

$$U = \xi_t + \xi_x u + \xi_y v = \xi_x(u - \dot{x}) + \xi_y(v - \dot{y}) \quad (9a)$$

$$V = \eta_t + \eta_x u + \eta_y v = \eta_x(u - \dot{x}) + \eta_y(v - \dot{y}) \quad (9b)$$

For example, the transformed conservation of mass relation is given by

$$\partial_r(\rho/J) + \partial_\epsilon(\rho U/J) + \partial_\eta(\rho V/J) = 0 \quad (10)$$

and does not appear too much more complex than the Cartesian form. Moreover, if $\eta = 0$ coincides with the body boundary surface then flow boundary conditions such as tangency and no slip are especially elegant and are expressed as

$$V = 0 \quad (\text{tangency}) \quad (11)$$

$$U, V = 0 \quad (\text{no slip}) \quad (12)$$

To motivate further discussion it is noted that the transformed equations (3) can be derived by first performing chain rule expansions on the terms of (1). For example

$$F_x = \xi_x F_\xi + \eta_x F_\eta + \tau_x F_\tau \quad (13)$$

where $\tau_x = 0$. The equations are then scaled by J^{-1} and metrics are brought back inside the operators using differentiation by parts. For example,

$$\frac{\xi_x}{J} \partial_\epsilon F = \partial_\epsilon \left(\frac{\xi_x F}{J} \right) - F \partial_\epsilon \left(\frac{\xi_x}{J} \right)$$

and so on. Terms are then collected to give equation (3) as well as combinations of metric terms that will be found to be zero. That is

$$\begin{aligned} \partial_r \dot{Q} + \partial_\epsilon \dot{F} + \partial_\eta \dot{G} = & Q \left[\partial_\epsilon \left(\frac{\xi_t}{J} \right) + \partial_\eta \left(\frac{\eta_t}{J} \right) + \partial_\tau \left(\frac{1}{J} \right) \right] + F \left[\partial_\epsilon \left(\frac{\xi_x}{J} \right) + \partial_\eta \left(\frac{\eta_x}{J} \right) \right] \\ & + G \left[\partial_\epsilon \left(\frac{\xi_y}{J} \right) + \partial_\eta \left(\frac{\eta_y}{J} \right) \right] \end{aligned} \quad (14)$$

All such right-hand-side combinations of metric terms are found to be zero because of the relations

$$\begin{aligned} \xi_x/J &= \eta_y & \eta_x/J &= -\eta_z \\ \xi_y/J &= -\xi_z & \eta_y/J &= \xi_z \\ \xi_z &= -\xi_x \xi_y - \eta_z \xi_y & \eta_z &= -\xi_x \eta_y - \eta_z \eta_y \end{aligned} \quad (15)$$

for example,

$$F\left[\partial_\xi\left(\frac{\xi_x}{J}\right) + \partial_\eta\left(\frac{\eta_x}{J}\right)\right] = F(\eta_z \eta_y - \eta_z \eta_y) = 0 \quad (16)$$

and so on.

The point of all this is that the metric quantities have been worked inside the differential operators. This is possible because combinations of metric terms such as

$$\partial_\xi\left(\frac{\xi_x}{J}\right) + \partial_\eta\left(\frac{\eta_x}{J}\right) = 0 \quad (17)$$

are found. Note also that unlike equation (1), equation (3) has been scaled by J^{-1} .

b) Metric Differencing

The fact that the transformed governing equations now have the metrics brought inside the difference operations can lead to numerical errors. This is because the metric variation is now being differenced along with the flow field quantity. In typical external aerodynamics applications, the flow quantities far from the body are essentially constant or uniform. Difference terms should therefore be zero, and this will be true if equation (1) is differenced on a uniform mesh. For a nonuniform mesh the transformed equations will not yield zero in regions of constant flow, however, unless proper differences of the metric identities are zero, that is from (14)

$$\delta_\xi(\xi_x/J) + \delta_\eta(\eta_x/J) = 0 \quad (18a)$$

$$\delta_\xi(\xi_y/J) + \delta_\eta(\eta_y/J) = 0 \quad (18b)$$

$$\delta_\xi(\xi_z/J) + \delta_\eta(\eta_z/J) + \delta_r(1/J) = 0 \quad (18c)$$

where δ_ξ , δ_η and δ_r are the difference operators used in the solution algorithm for (3). If, for example, the metrics ξ_x/J , etc., could be exactly evaluated (as they can be in, say,

a cylindrical coordinate), then equations (18) will not be exactly zero but will be zero to within the order of accuracy of the operators δ_ξ , etc. For a rapid variation of the metrics and for large grid spacing—a phenomenon frequently occurring in aerodynamic applications in the far field—this error can be very appreciable. It can, in fact, add a error source to the equations that can overwhelm the solution accuracy. However, if the metrics themselves are differenced such that equations (18) are exactly zero, then this error is controlled. For example, in two dimensional steady flow relations (18a) and (18b) become using (15)

$$\delta_\xi(y_\eta) - \delta_\eta(y_\xi) = 0 \quad (19a)$$

$$-\delta_\xi(x_\eta) + \delta_\eta(x_\xi) = 0 \quad (19b)$$

If y_η and y_ξ are differenced as $\delta_\eta y$ and $\delta_\xi y$ where δ_η and δ_ξ are the same difference operators used in the solution algorithm for (3), then the metric identities (19) exactly difference to zero. The importance of satisfying these relations was pointed out in [2] in which three point central spatial differences were used in the solution algorithm [3] as well as for the metric quantities.

In three dimensions it becomes more difficult to exactly difference the metric identity relations. For steady or simple unsteady grid motion, Pulliam and Steger [4] introduced an averaging process for the steady terms that works for any difference operator that can be differenced in parts as

$$\delta(uv) = (\mu v)(\delta u) + (\mu u)(\delta v) \quad (20)$$

where μ is an averaging operator. An example of (20) is given by

$$\nabla uv = (\mu v)(\nabla u) + (\mu u)\nabla v \quad (21)$$

where $\nabla u = u_j - u_{j-1}$, $\mu u = \frac{u_j + u_{j-1}}{2}$, etc. In extended work Thomas and Lombard [5] correctly treated the unsteady metric variations and cleverly simplified the calculation of the spacial metric terms. They also coined the term "metric conservation law" to describe the fact that the metric relations (18) must be differenced to be zero.

As one introduces higher order central or one-sided spatial difference operators, or uses predictor-corrector schemes, it becomes more and more difficult to correctly satisfy

the metric relations (18). This ultimately lead [4] to an approximate cancellation method that relies on solving the differenced equations in a simple perturbation form:

$$\delta_t(\dot{Q} - \dot{Q}_\infty) + \delta_\xi(\dot{F} - \dot{F}_\infty) + \delta_\eta(\dot{G} - \dot{G}_\infty) = 0 \quad (22)$$

In the far field $\dot{F} \rightarrow \dot{F}_\infty$, etc., and any consistent difference scheme is satisfied regardless of how rapidly the metrics vary. When F appreciably varies from F_∞ , etc., it is assumed that the grid is sufficiently smooth and refined so that the metric error is no greater than the error of differencing the flux terms. In external flow applications this process has worked quite well, including successful use with the potential flow equation (cf. [6]).

A direct means of cancelling the metric errors is the straightforward one of subtracting the error, for example, for a stationary grid

$$\delta_t \dot{Q} + \delta_\xi \dot{F} + \delta_\eta \dot{G} = F(\delta_\xi y_\eta - \delta_\eta y_\xi) + G(-\delta_\xi x_\eta + \delta_\eta x_\xi) \quad (23)$$

This puts the equation into a weak conservation law form that in principle does not degrade the shock capturing properties of the scheme. It could, however, contribute to a mild source-term weak instability that would be alleviated somewhat by spatial averaging of the right-hand side F and G flux terms. These right-hand-side flux terms can also be approximately evaluated at ∞ in somewhat duplication of (22) above.

The whole problem of differencing the metrics has been avoided in [7-10]. In this approach the Cartesian equations are expanded by chain rule and then simply left that way. That is

$$\partial_r Q + \xi_t \partial_\xi Q + \eta_t \partial_\eta Q + \xi_x \partial_\xi F + \eta_x \partial_\eta F + \xi_y \partial_\xi G + \eta_y \partial_\eta G = 0 \quad (24)$$

and is called the quasi-linear form by Shamorth and Gibeling [10] or chain-rule conservation form by Hindman [9]. Although the Jacobian is never divided through, this form is somewhat similar to the weak conservation law form (23) just discussed, particularly so with averaging of F and G . It should also properly capture the correct jump relations. The chain-rule conservation form may well be a good compromise to differencing the transformed equations in conservative form. For certain algorithms (e.g., Beam-Warming [3]) it appears to require more work than using the strong conservation law form with free stream correction.

c) Perturbation Form Digression

The above idea of subtracting out the free stream metric variation, equation (22), discussed previously is a special case of perturbing the solution about a known function which in some sense is also a nearby or approximate solution. Let $Q = Q_0 + Q'$ where Q_0 is the nearby or approximate solution and let Q' be the perturbation. The terms of equation (3) can be rewritten as,

$$\begin{aligned}\partial_r \dot{Q} &= \partial_r \dot{Q}_0 + \partial_r \dot{Q}' \\ \partial_\epsilon \dot{F}(Q) &= \partial_\epsilon \dot{F}(Q_0) + \partial_\epsilon (\dot{F}(Q) - \dot{F}(Q_0)) \\ &\text{etc.}\end{aligned}$$

Then assuming functions of Q_0 are sufficiently simple to be very accurately (or exactly differentiated) with operators $\bar{\delta}$, the differencing of equation (3) can be represented as

$$\begin{aligned}\delta_r \dot{Q}' + \delta_\epsilon [\dot{F}(Q) - \dot{F}(Q_0)] + \delta_\eta [\dot{G}(Q) - \dot{G}(Q_0)] \\ = -[\bar{\delta}_r \dot{Q}_0 + \bar{\delta}_\epsilon \dot{F}(Q_0) + \bar{\delta}_\eta \dot{G}(Q_0)]\end{aligned}\quad (25)$$

where δ_r , δ_ϵ , and δ_η represent the algorithm difference operators and $\bar{\delta}_r$, $\bar{\delta}_\epsilon$, $\bar{\delta}_\eta$ represent the very accurate differencing. In the case $Q_0 = Q_\infty$ the right-hand side is analytically zero and equation (25) is identical to equation (22).

Such a perturbation form of the differenced equation has been proposed in internal spin-up problems to remove the axisymmetric variation from the dependent Cartesian velocity variable [11,12]. It might also be used in problems in which certain fine details might be otherwise lost in a coarse grid. For example, a nonuniform incoming flow profile can be represented in Q_0 that would otherwise be lost in a far field coarse grid. Assuming in this case that Q_0 satisfies the Euler equations, the right-hand side of (25) is identically zero. Calculations using this particular technique for incoming inviscid shear flows have been tested by Buning and Steger [13]. Although not yet tried, Q_0 might be chosen as an approximate solution, in which case the right hand side of (25) would not be zero. In regions in which $Q \rightarrow Q_0$, one could hope to use a much coarser grid without losing solution accuracy.

III. CALCULATIONS ON CURVILINEAR GRIDS

Finite difference and related finite volume calculations using body fitted curvilinear grid systems have been carried out for some time. Solution variables have included velocity potential, stream function, and the primitive variables. Computed results include incompressible and compressible flow around airfoils, projectiles, cascades, inlets, wings, wing-body combinations, etc. In some cases the body deforms with time (e.g., airfoil with moving aileron) and occasionally solutions for multiple non-connected bodies appear (e.g., airfoil with detached flap). No attempt will be made to review this work—the interested reader will find much of the material in the AIAA Journal, the Proceedings of the International Conference on Numerical Methods in Fluid Mechanics, Computers and Fluids, and the Journal of Computational Physics. What is apparent from this literature is that while we are becoming more adept at solving the flow about complex configurations, considerable computational efficiencies are yet to be obtained.

In order to illustrate points to be discussed later, the results of a finite difference simulation, due to Nicolet *et al.* [14] for flow about an X-24C configuration is reproduced in Figs. 1 to 3. A head-on view of the X-24C is indicated in Fig. 1, while Fig. 2 shows typical views of the grid fit between the body surface and an analytically fit outer bow shock. The grid in this case is generated with a hyperbolic partial differential equation grid generation scheme [15]. The overall three dimensional grid is formed by generating two dimensional grids at each station along the body as the solution progresses by marching the steady parabolized Navier-Stokes (PNS) equations. The hyperbolic grid generation scheme is fast enough to be used within the flow field marching scheme. Moreover, each two dimensional grid is itself generated using the same kind of numerical algorithm used for solving the PNS equations—a sort of conservation of numerical algorithm knowledge. Computed surface pressure and heat transfer at a station just prior to the beginning of the wing are compared to experimental data in Fig. 3.

In carrying out the preceding calculation or any similar calculation on a generalized grid it is found that the solution accuracy depends on the grid. This is not surprising because unless one has a very fine mesh throughout the field, an accurate solution will require that grid points be clustered to the flow field action regions—the change of gradient

regions. The grid in Fig. 2, for example, is exceedingly fine near the body surface in order to resolve viscous gradients along the wall. (The less than perfect agreement with the experimental heating rate shown in Fig. 3 is apparently not due to inadequate resolution in this direction). The outer grid line also coincides with the bow shock, and points are clustered along the body, for example to resolve the cross flow expansion around the chime (i.e., lower right corner in Fig. 2) when the vehicle is at angle of attack (here at 6°). Otherwise no other attempt was made in this calculation to adapt the grid to computed flow field gradients.

Besides proper grid clustering, the smoothness of the grid, the skewness of the grid, and sometimes the aspect ratio of the grid can affect the accuracy of a numerical solution or the efficiency with which it is obtained. The grids shown in Fig. 2 are nearly orthogonal close to the body surface and they have a smooth, gradual variation. These grids would be judged felicitously. However, the quality of a grid seems to be hard to quantify because various numerical algorithms appear to behave differently to the properties of grid smoothness, skewness, and stretching. Numerical algorithms that use a very compact stencil of points to evaluate fluxes and metrics, for example, generally seem to be less sensitive to grid spacings that change rapidly or even discontinuously. Thus computers using the MacCormack finite volume method for Navier-Stokes equations sometimes change the grid spacing by a factor of 2 or 4 in a given region. Such a change is not allowed, for example, when using high order central spacial differencing operators.

Some numerical algorithms appear very sensitive to mesh cell aspect ratio, i.e., the ratio of Δx to Δy or $(x_i^2 + y_i^2)^{1/2}$ to $(x_{i+1}^2 + y_{i+1}^2)^{1/2}$. Thus Jameson [16] in developing a multigrid relaxation algorithm for the transonic potential equation abandoned SLOR iterative methods. In its place he used an alternating-direction-implicit scheme as the multigrid iterative solver because it is less sensitive to cell aspect ratio. The very efficient approximate-factorization-implicit relaxation scheme of Holst [17] appears to degrade if uniform fine grid spacing is used along the body, prompting Holst to generate his grids with this constraint in mind. For his approximate-factorization scheme the grid shown in Fig. 4 is much preferred to that shown in Fig. 5. A user of a standard alternating-direction-implicit relaxation scheme, however, may very well opt instead for the grid of Fig. 5 over

that of Fig. 4 simply because of its finer grid spacing near the body and presumably greater accuracy.

Avoiding certain undesirable grid properties such as skewness can lead to more complex computer programs and perhaps other difficulties. The cascade C-grid shown in Fig. 6 for example is too highly skewed. While the implicit Beam-Warming algorithm for the Euler or Navier-Stokes equations functions on such a grid, it runs far from optimum. An alternative grid to that shown in Fig. 6 might use an overlapped or patched grid system. For example, the overlapped grid system shown schematically in Fig. 7 is suggested because each grid is easy to generate and has minimum distortion. However, such a grid system requires extensive modification of existing numerical algorithms and computer programs. This is because certain grid points will have to be flagged off, and grid interfaces will have to be joined without degrading numerical stability. Nevertheless, overlapped or patched grid systems will ultimately be needed as body boundary configurations become more complex, for example, in computing flow about a wing with engine nacelles.

Finally, it should be remarked that the effect of a poorly spaced grid will sometimes not be observed until the data is displayed or utilized in a different way. The unpublished result due to Seidel [18] that is shown in Fig. 8 is an example. The plots of generalized pitching moment versus reduced frequency show an essentially exact solution (dashed line) and a low frequency transonic small disturbance finite difference solution with the nonlinear terms removed. The flow is about a flat plate subjected to an angle of attack pulse. The small oscillations shown in the finite difference result are a significant error in a flutter calculation. They were traced back to a discontinuous change of grid stretching more than a chord away from the airfoil, and were eliminated by using a smoothly stretched grid throughout.

IV. CONCLUDING REMARKS

Finite difference methods coupled with body conforming curvilinear grid systems are being used to solve a variety of complex flow fields. Current numerical algorithms and grids are tuned to flow field applications that can be computed in reasonable times on

present day machines. These numerical algorithms rely on sparse-equation time-accurate or iterative-solution methods that function best on well ordered grids.

Curvilinear body conforming grids have made modern finite difference schemes into practical engineering tools. They simplify the application of boundary conditions and allow flow field gradients to be resolved in an orderly manner. However, one must be careful in differencing transformed equations, especially when the equations are in conservative form and transform metrics are brought inside the difference operators. Finite difference algorithms are also sensitive to grid smoothness, skewness and stretching with some algorithms being much more adversely affected than others.

As finite difference methods are extended to more complex geometries, it becomes obvious that more than one grid system will have to be used. Exactly how multiple grids should best be joined, patched, or overset together remains a research topic, but a number of approaches will likely give satisfactory results. The simultaneous development of multiple grid systems and finite difference schemes suitable for multiple grids is underway and will be a major pacing item in computational fluid dynamics.

Acknowledgements: This work was partially supported by Army Research Office Contract DAAG29-81-K-0013 and Air Force Flight Dynamics Contract F33615-81-K-3020.

REFERENCES

1. Viviand, H., Conservative forms of gas dynamic equations, *La Recherche Aerospatiale* 1 (1974), 65-68.
2. Steger, J.L., Implicit finite-difference simulation of flow about arbitrary two-dimensional geometries, *AIAA Journal* 16 (1978), 679-686.
3. Beam, R., and R.F. Warming, An implicit finite-difference algorithm for hyperbolic systems in conservation-law-form, *Journal of Computational Physics* 22 (1976), 87-110.
4. Pulliam, T.H. and J.L. Steger, Implicit finite-difference simulations of three-dimensional compressible flow, *AIAA Journal* 18 (1980), 159-167.
5. Thomas, P.D. and C.K. Lombard, Geometric conservation law and its application to flow computations on moving grids, *AIAA Journal* 17 (1979), 1030-1037.
6. Steger, J.L., Implicit finite difference simulation of inviscid and viscous compressible flow, Symposium on Transonic, Shock and Multidimensional Flows, Madison, Wisconsin, May 13-15, 1981.
7. Shan, J.S. and Hankey, W.L., Numerical Solution of the Navier-Stokes Equations for a Three-Dimensional Corner, *AIAA J.* 15 (1977), 1575-1582.
8. Lerat, A. and Slides, J., Numerical Calculation of Unsteady Transonic Flows. Paper presented at AGARD Meeting of Unsteady Airloads in Separated and Transonic Flow, Lisbon, April 1977.
9. Hindman, R.G., Geometrically induced errors and their relationship to the form of the governing equations and the treatment of generalized mappings, Proceedings of the AIAA 5th Computational Fluid Dynamics Conference, Palo Alto, CA. 1981.
10. Shamroth, S.J. and Gibeling, H.J., Navier-Stokes Solution of the Turbulent Flowfield About an Isolated Airfoil, *AIAA J.* 18, December 1980.
11. Steger, J.L., A computer program for the internal flow of spin-stabilized liquid-filled shells. Flow Simulations Contract Report 79-03, submitted to ARRADCOM, Oct. 1979.
12. Chakravarthy, S.R., Numerical simulation of laminar incompressible flow within liquid filled shells, Rockwell International Science Center Report SC5248.2FRD submitted to ARRADCOM, Oct. 1981.

13. Buning, P.G., and J.L. Steger, Simulation of the two-dimensional Euler equations with generalized coordinate transformation using flux vector splitting. AIAA Paper No. 82-0971 to be presented June, 1982.

14. Nicolet, W.E., S. Shanks, G. Srinivasan, and J.L. Steger, Flowfield predictions about lifting entry vehicles. AIAA Paper 82-0026, 1982.

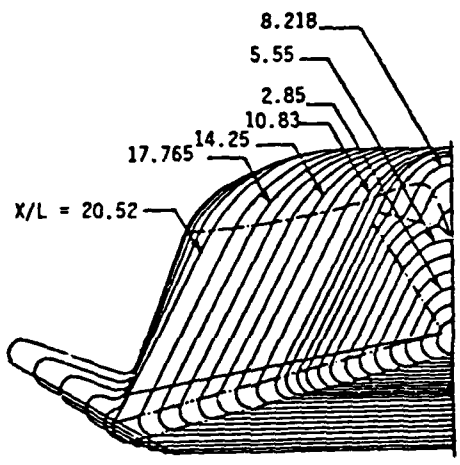
15. Steger, J.L., and D.S. Chaussee, Generation of body-fitted coordinates using hyperbolic partial differential equations, SIAM J. Sci. Stat. Comput. 1 (1980), 431-437.

16. Jameson, A., Acceleration of transonic potential flow calculations on arbitrary meshes by the multiple grid method. AIAA Paper No. 79-1458. Proceedings of AIAA Computational Fluid Dynamics Conference, Williamsburg, VA. July 1979.

17. Holst, T.L., and D. Brown, Transonic airfoil calculations using solution-adaptive grids. AIAA Paper No. 81-1010-CP. 1981.

18. Seidel, David A., Private Communication, NASA Langley Research Center, Hampton, VA, 1982.

BEGIN HERE



LINES T
807702

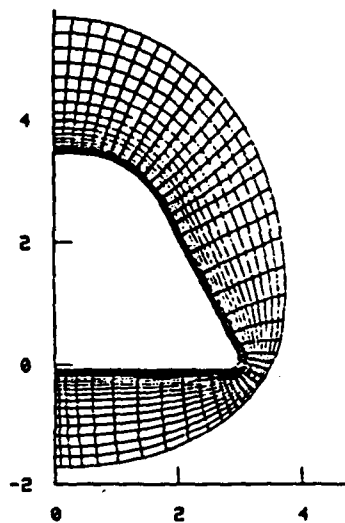
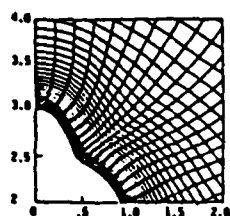
Fig. 1 X-24C configuration.

4
3
2
1
0

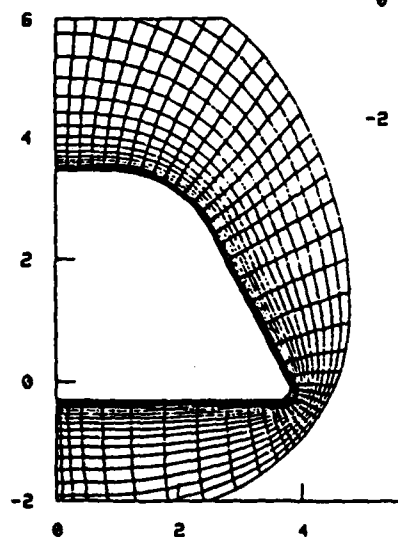
BEGIN HERE

COMPUTED X-21C GRIDS

$X/L = 7.3$



$X/L = 13.3$



$X/L = 17$

LINES TO
BOTTOM

Fig. 2 Typical grids for X-24C configuration.

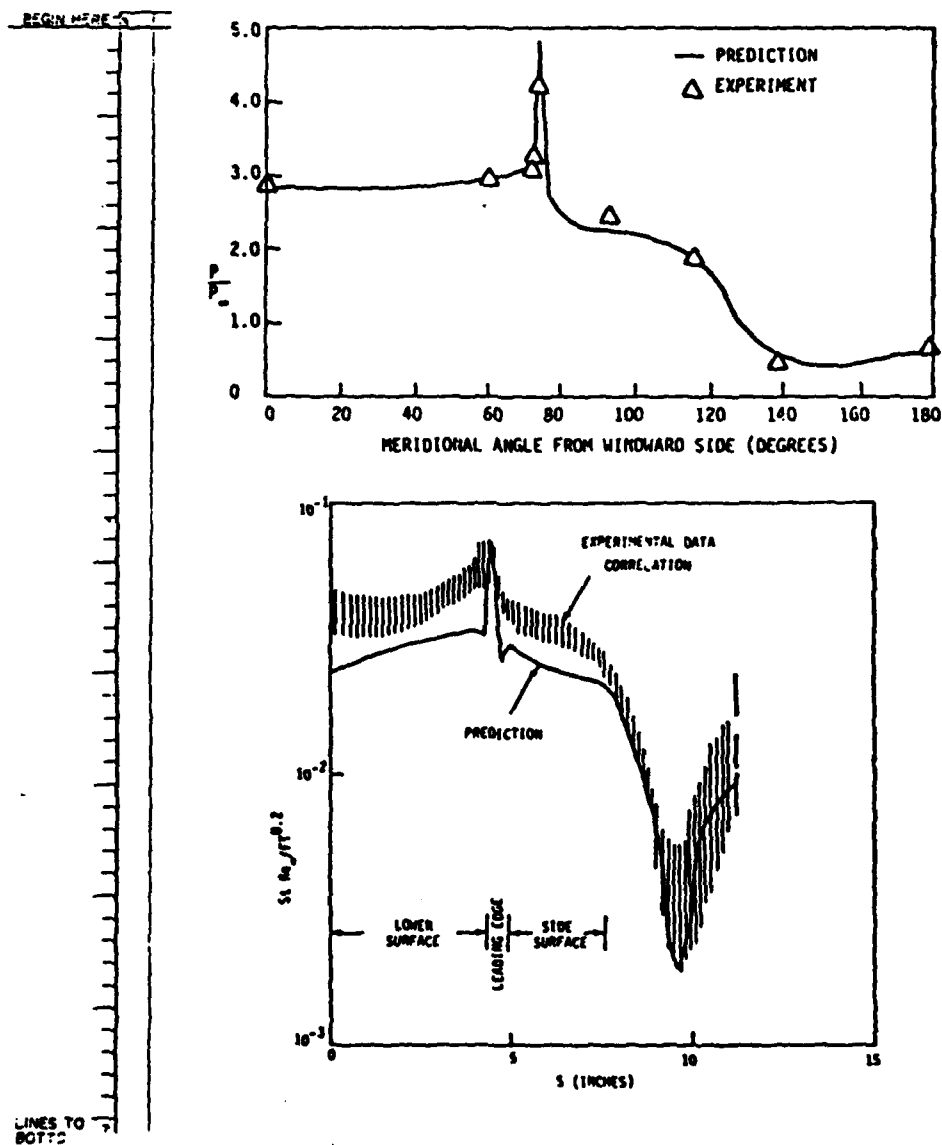
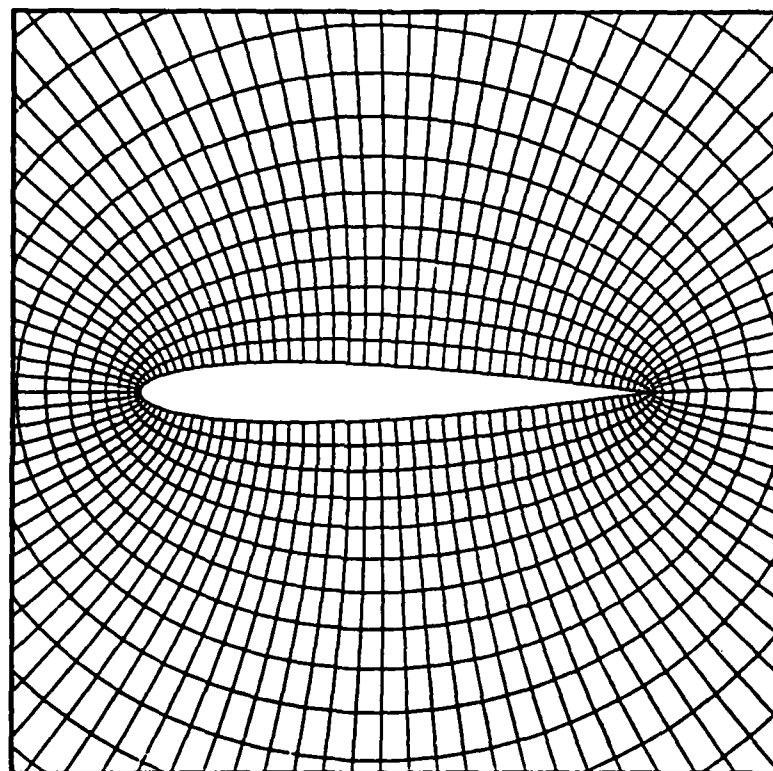


Fig. 3 Solution results obtained using parabolized Navier-Stokes equations for X-24C model at $M_\infty = 5.05$, $\alpha = 6^\circ$, $Re = 5 \times 10^6/ft$ at $x = 20.52$ inches.

BEGIN HERE



XMN--0.25

XMX- 1.25

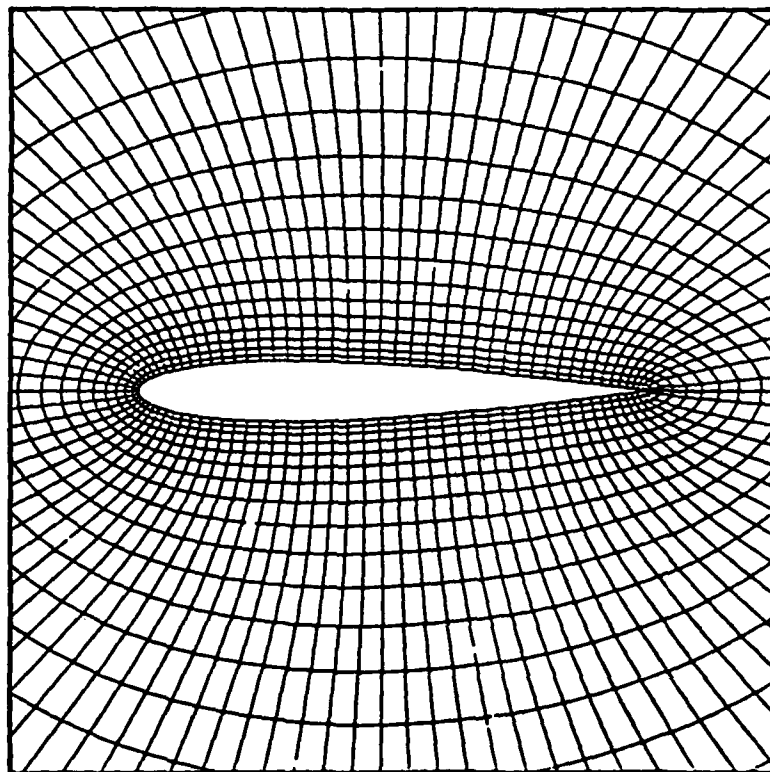
YMN--0.75

YMX- 0.75

LINES TO
BOTTOM

Fig. 4 Low aspect ratio grid used for inviscid flow calculations.

BEGIN HERE



$X_{MN} = -0.25$

$X_{MX} = 1.25$

$Y_{MN} = -0.75$

$Y_{MX} = 0.75$

LINES TO
BOTTOM

Fig. 5 Inviscid flow grid with uniform mesh spacing adjacent to the body.

0
1
2
3
4

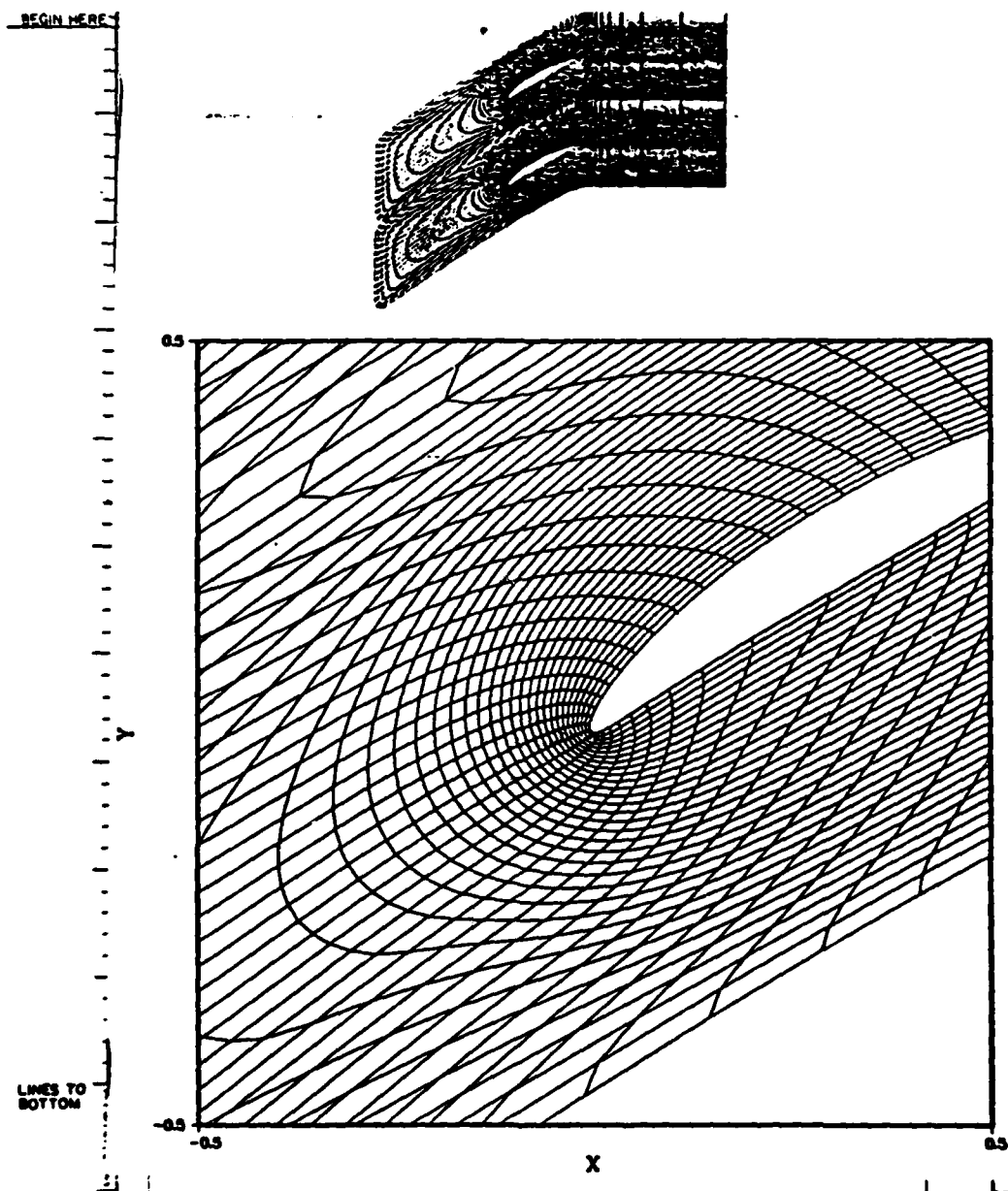
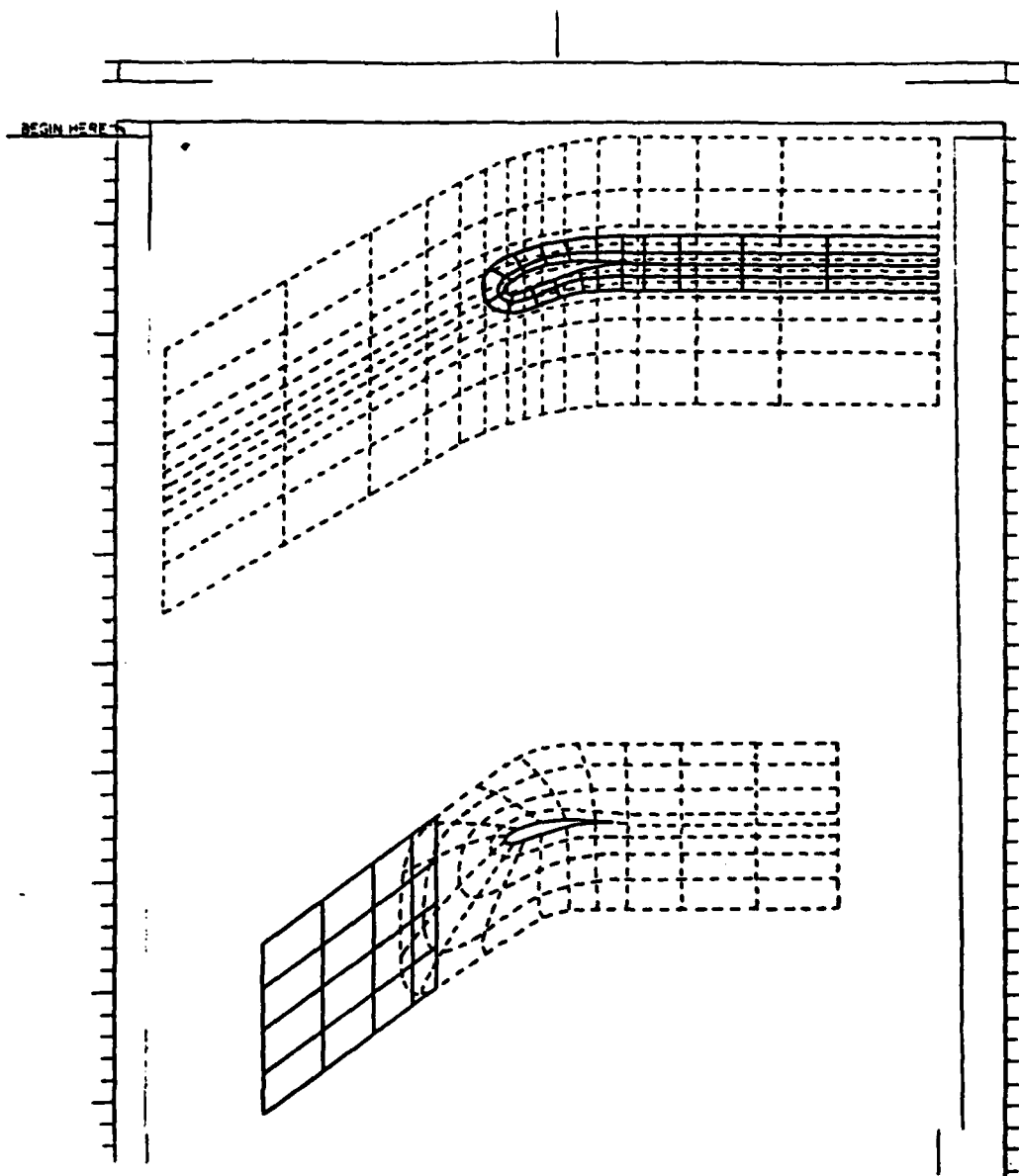


Fig. 6 Highly skewed C-grid for cascade.



LINES TO
BOTTOM

Fig. 7 Schematic indicating how overlapped grids could be used for cascade flow applications to remove grid distortion.

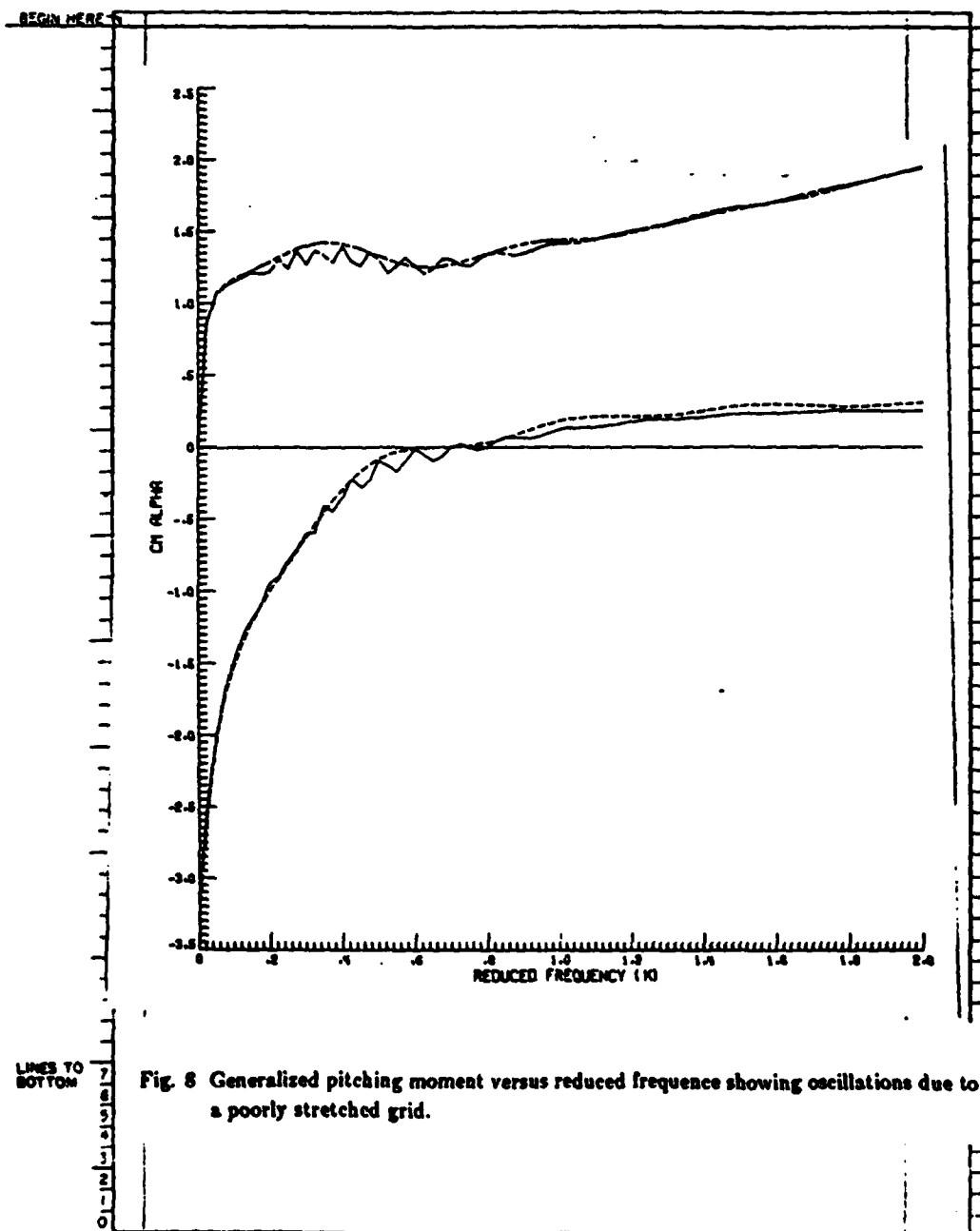


Fig. 8 Generalized pitching moment versus reduced frequency showing oscillations due to a poorly stretched grid.

V. FUTURE WORK

Two new flow simulation codes for projectiles are now underway at BRL. One of these, which is being implemented by Nietubicz, extends our segmented projectile code for the calculation of ringed or tubular projectiles. This advanced projectile flies a more level trajectory than conventional rounds. Figures 1 and 2 show a typical configuration and a preliminary result obtained by Nietubicz.

The other new projectile code is still in the planning stage. A three dimensional transonic projectile code with base will use the same segmentation process used with the two dimensional code. In order to minimize the number of grid points needed to resolve a three dimensional field, we plan to use a spectral method in the circumferential direction. Although our solution method is implicit, the spectral method only has to be implemented explicitly. (This has been shown by K. C. Reddy and has been independently verified by my student, Mr. T. Barth.) This code should be operational by December 1984. My involvement will continue as an employee of NASA Ames.

NOMINAL PHYSICAL CHARACTERISTICS

PROJECTILE BODY WEIGHT	2.722 Kg	(6.00 lbs)
FUEL WEIGHT	.295 Kg	(0.65 lbs)
PROJECTILE FLIGHT WEIGHT	3.017 Kg	(6.65 lbs)
SABOT WEIGHT	2.325 Kg	(5.13 lbs)
TOTAL THROW WEIGHT	5.342 Kg	(11.78 lbs)
CENTER OF GRAVITY, FROM BASE	133.35 mm	(5.25 in)
AXIAL MOMENT OF INERTIA	.00233 Kg m ²	(.00194 Slug ft ²)
TRANSVERSE MOMENT OF INERTIA	.02284 Kg m ²	(.01685 Slug ft ²)

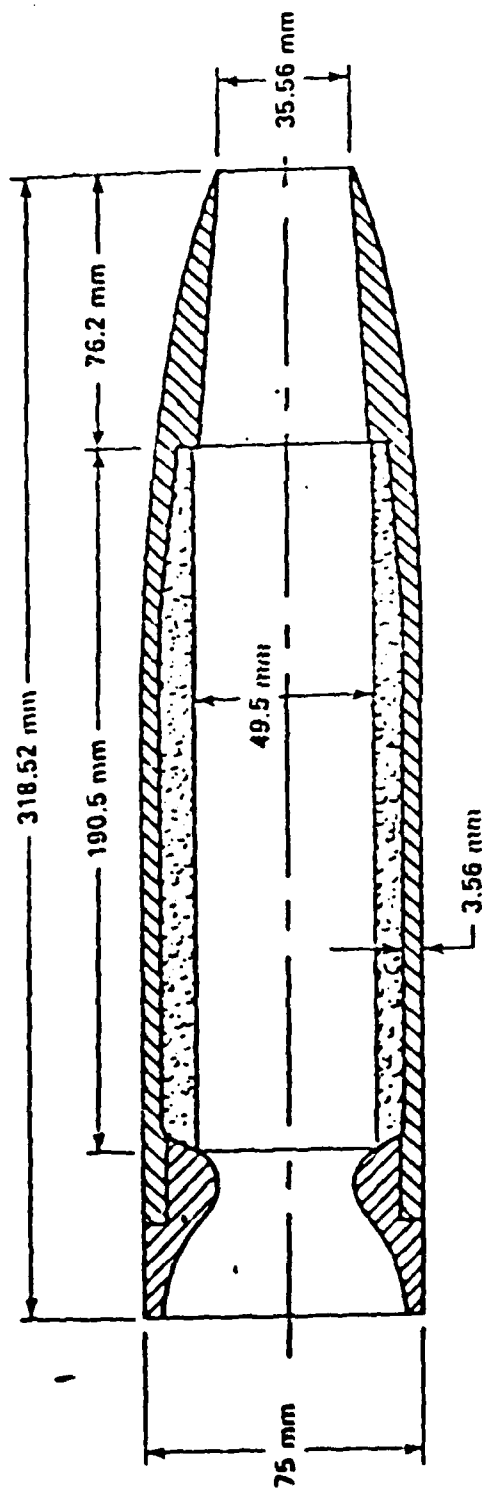


FIGURE 1. 75MM SFRJ TUBULAR PROJECTILE CONFIGURATIONAL DETAILS

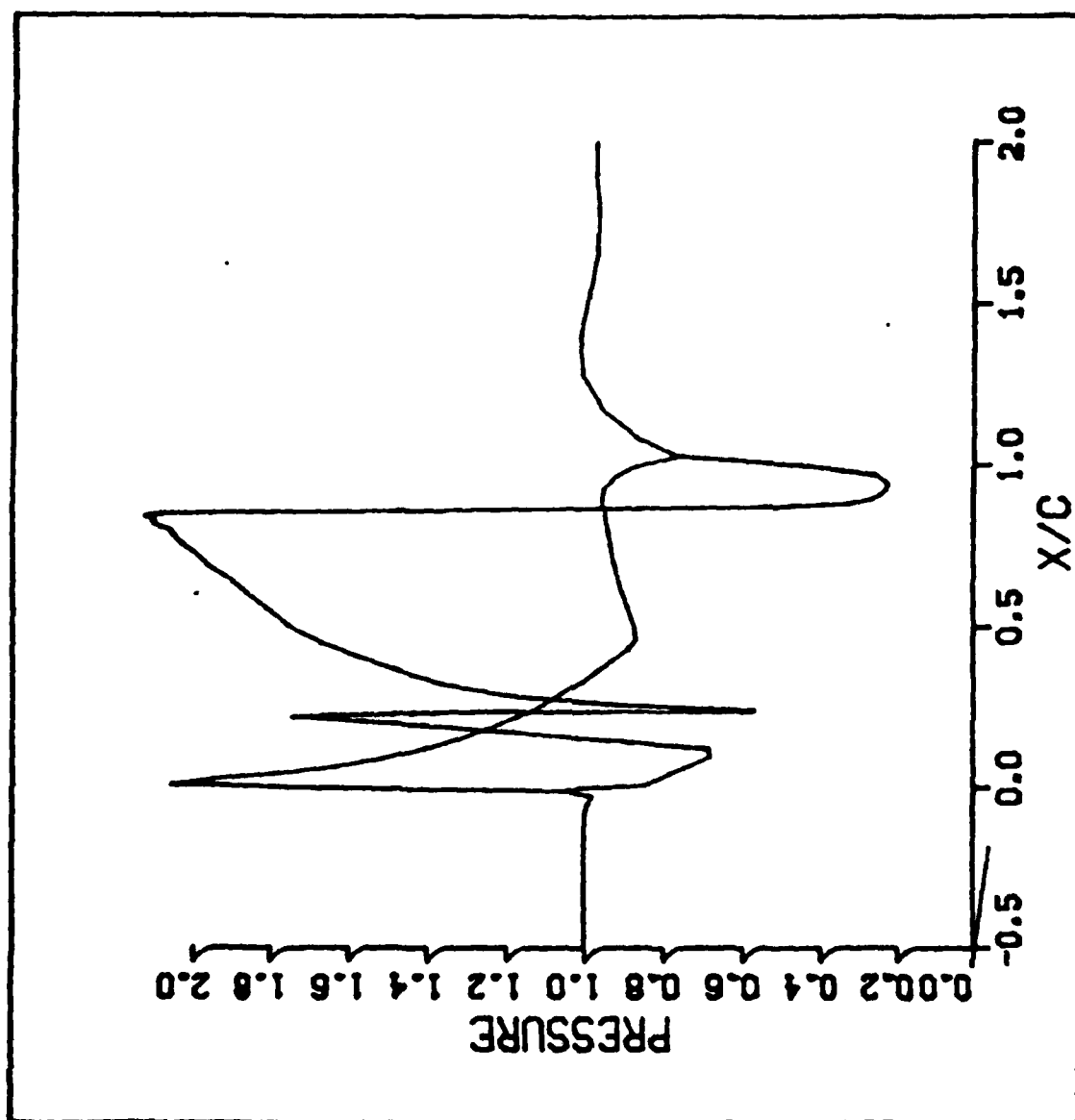


Figure 2.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AA CFD 83-2	2. GOVT ACCESSION NO. AD A137935	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTATION OF VISCOUS FLOW ABOUT ADVANCED PROJECTILES		5. TYPE OF REPORT & PERIOD COVERED Final 10/15/80 to 09/30/83
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Joseph L. Steger		8. CONTRACT OR GRANT NUMBER(s) DAA 629-81-K-0013
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Aeronautics and Astronautics Stanford University Stanford, CA 94305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office Post Office Box 12211 Research Triangle Park, NC 27709		12. REPORT DATE 09/09/83
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) None
		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 THE VIEW, OPINIONS, AND/OR FINDINGS CONTAINED IN THIS REPORT ARE THOSE OF THE AUTHOR(S) AND SHOULD NOT BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION, POLICY, OR DE- CISION, UNLESS SO DESIGNATED BY OTHER DOCUMENTATION.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Transonic Flow, Projectile Flow Simulation, Finite Differences		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The research program has been aimed at developing computational procedures for simulation of flow about projectiles. This final report includes summaries on numerical algorithm developments, methods of grid generation, and descriptions of flow field solution codes for projectiles in the transonic range. <i>47</i>		