NASA-CR-4165 19880018655

NASA Contractor Report 4165

An Improved Version of NCOREL

A Computer Program for 3-D Nonlinear Supersonic Potential Flow Computations

Michael J. Siclari

CONTRACT NAS1-16758
JULY 1988

NASA Contractor Report 4165

An Improved Version of NCOREL

A Computer Program for 3-D Nonlinear Supersonic Potential Flow Computations

Michael J. Siclari
Grumman Corporate Research Center
Bethpage, New York

Prepared for
Langley Research Center
under Contract NAS1-16758

National Aeronautics
and Space Administration

Scientific and Technical
Information Division

1988

CONTENTS

CONTENTS (continued)

ILLUSTRATIONS

ILLUSTRATIONS (continued)

ILLUSTRATIONS (continued)

ILLUSTRATIONS (continued)

ILLUSTRATIONS (continued)

1. INTRODUCTION

The numerical solution of supersonic flow problems using the full potential equation is an attractive alternative to solving either Euler equations or the linearized potential flow equations. The full potential equation retains most of the nonlinear features of the flow field that the linearized potential (e.g., the more popular linear panel methods) inherently neglects, while having the simplicity of a single variable irrotational solution. In the absence of separated flow and strong shocks (e.g., Mach number normal to shocks greater than 1.4) the full potential and Euler equations should agree very closely at low to moderate supersonic Mach numbers.

A computer code called NCOREL (for Nonconical Relaxation) has evolved into a relatively sophisticated tool for solving supersonic full potential flows over complex geometries. The method first solves for the conical flow at the apex and then marches downstream in a spherical coordinate system. Implicit relaxation techniques are used to numerically solve the full potential equation at each subsequent crossflow plane. Reference 10 reports on an earlier pilot code that reliably computed flows about isolated wings and bodies. Since that time, many additional options and modifications have been incorporated into the NCOREL to increase its usefulness, applicability, and reliability.

One of the significant improvements is to the numerics and grid generation to allow the reliable treatment of wing-body-inlet configurations. Another significant modification is the inclusion of entropy corrections for higher Mach number flows. The following is a summary of the major new features included in NCOREL:

- o Improved numerics for the computation of wing-body configurations
- o Improved grid generation
  - better wing-body mapping capability
  - cluster options
  - capability of treating bodies that are highly axially cambered
- o Flow-through-inlet simulation

- Entropy corrections with bow shock fit option
  - modified potential method
  - Rankine-Hugoniot conditions applied to bow shock

- Accelerated relaxation scheme
  - approximate relaxation method (AF1Z)

- New geometry package that accepts Harris Wave Drag input data

- Wake model.

2. COORDINATE SYSTEMS

The finite-difference supersonic marching techniques generally begin with a Cartesian coordinate frame (x, y, z) with the centerline of the geometry coincident with the z-axis and marching downstream in z. This procedure has the drawback of requiring the axial Mach number to be greater than unity throughout the flowfield. This condition can be easily violated along the leading edge of the wing at low supersonic Mach numbers and practical sweep angles. This usually limits the applicability of Cartesian marching methods to a condition where $M_\infty > 1.8 \sim 2.0$. This can be alleviated with the utilization of a spherical coordinate frame ψ, ω, r where

$$\begin{aligned} x &= r \cos \omega \sin \psi \\ y &= r \sin \omega \sin \psi \\ z &= r \cos \psi \end{aligned} \tag{1}$$

with the centerline of the body along the r-axis at $\psi = 0$. The numerical solution will be obtained by marching on spherical surfaces in the r direction. Hence, the leading edge of a constant sweep angle or delta wing will be coincident with the marching direction.

To solve the exact potential flow equation with exact boundary conditions on relatively general wings, a transformed domain on each spherical surface must be generated with the mapped body surface a coordinate line and with adequate mesh spacing in regions of steep flow gradients. In Ref 1, a series of mappings were developed for conical bodies which meet these requirements. A sketch of the series of transformations is shown in Fig. 1. First, a preliminary mapping, called a stereographic projection, (Ref 2) is used to open the sphere r = constant to a planar surface by

$$p + iq = \tan \frac{\psi}{2} e^{i\omega}. \tag{2}$$

The body cross section is then transformed to a near circle by a conformal mapping which maps (p,q) to (ρ,θ) coordinates. For example, a Joukowski mapping has the form,

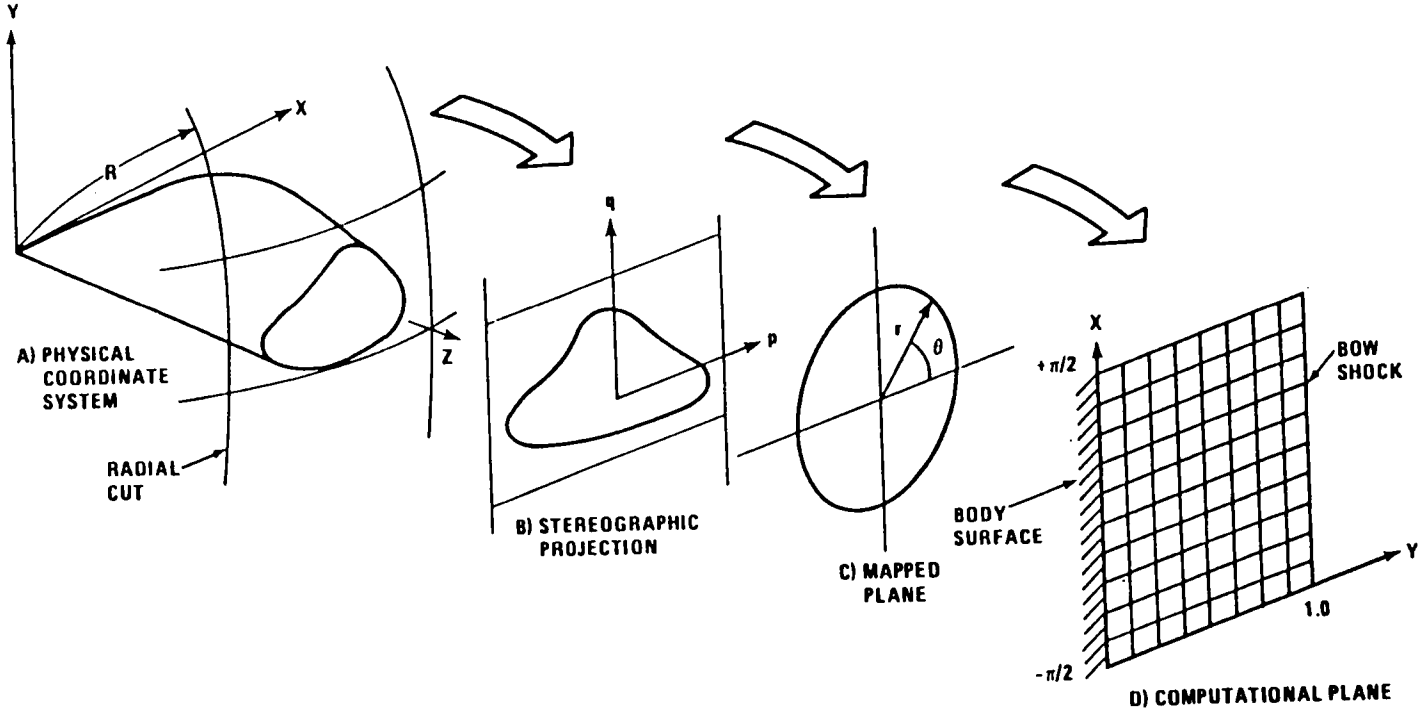


Fig. 1 NCOREL Coordinate Transformations

$$\frac{S-S_o}{S+\bar{S}_o} = \left(\frac{\Gamma-S_o/2}{\Gamma+\bar{S}_o/2}\right)^2 \tag{3}$$

where

$$\begin{aligned} S &= p + iq \\ \Gamma &= \rho e^{i\theta} \end{aligned} \tag{4}$$

and $S_o = S_o(r)$ is the location of the singularity in the p, q plane. In general, the conformal transformation (3) can be replaced by any conformal mapping or series of mappings.

The mapped domain, consisting of the curvilinear coordinates ρ, θ and $R = r$, forms the basis of the computation. The nonorthogonality of the coordinate system enters from the dependency of the singularity location on r, i.e., $S_o = S_o(r)$. Thus, derivatives with respect to R (with ρ and θ constant) generally differ from derivatives with respect to r (at constant ψ and ω). However, as will be noted in the next section, the coordinates ρ, θ are orthogonal with respect to each other at R = constant. It is convenient to introduce here the two metric lengths for the orthogonal coordinates

$$\begin{aligned} h_\rho &= \left[\left(\frac{\partial x}{\partial \rho}\right)^2 + \left(\frac{\partial y}{\partial \rho}\right)^2 + \left(\frac{\partial z}{\partial \rho}\right)^2\right]^{1/2} \\ &= RH(\rho,\theta,R) \end{aligned} \tag{5}$$

$$\begin{aligned} h_\theta &= \left[\left(\frac{\partial x}{\partial \theta}\right)^2 + \left(\frac{\partial y}{\partial \theta}\right)^2 + \left(\frac{\partial z}{\partial \theta}\right)^2\right]^{1/2} \\ &= \rho RH(\rho,\theta,R) \end{aligned}$$

where

$$H = \frac{\sin\psi}{(p^2+q^2)^{1/2}}\, h \tag{6}$$

with h being the metric of the conformal mapping at constant r

$$h = \left|\frac{\partial S}{\partial r}\right| = \left|\frac{(S-S_o)(S+\bar{S}_o)}{(r-S_o/2)(r+\bar{S}_o/2)}\right|. \tag{7}$$

The derivatives of the transformations in the r direction, denoted h_1 and h_2, are given by:

$$h_1(\rho,\theta,R) \equiv \frac{\partial \rho}{\partial r} = \frac{\partial \xi}{\partial r}\cos\theta + \frac{\partial \eta}{\partial r}\sin\theta$$

$$h_2(\rho,\theta,R) \equiv \frac{1}{\rho}\frac{\partial \theta}{\partial r} = \frac{\partial \xi}{\partial r}\sin\theta - \frac{\partial \eta}{\partial r}\cos\theta \tag{8}$$

where

$$\xi = \rho\cos\theta$$

$$\eta = \rho\sin\theta$$

$$\xi_r + i\eta_r = r_r .$$

The numerical computation is finally performed in a general nonorthogonal sheared coordinate system defined by

$$X = \theta$$

$$Y = \frac{\rho - b(\theta,R)}{c(\theta,R)-b(\theta,R)} \tag{9}$$

$$Z = R$$

where $b(\theta,R)$ is the mapped body surface coordinate (near-circle). $c(\theta,R)$ is a somewhat-arbitrary predefined smooth surface located outside the bow wave if the bow shock capture method (BSC) is used and $c(\theta,R)$ is coincident with the bow shock if the bow shock fit (BSF) method is employed.

3. GOVERNING EQUATIONS AND BOUNDARY CONDITIONS

3.1 GOVERNING EQUATIONS

For inviscid flow, the governing steady flow equations in terms of the velocity vector $\bar{Q}$ and the speed of sound, a, may be written as

$$a^2 \nabla \cdot \bar{Q} - \frac{1}{2} \bar{Q} \cdot \nabla(\bar{Q} \cdot \bar{Q}) = 0 \tag{10}$$

and

$$a^2 + \frac{\gamma-1}{2} \bar{Q} \cdot \bar{Q} = a_o^2 \tag{11}$$

where a_o is the stagnation speed of sound and γ is the ratio of specific heats. A perturbation potential ϕ is introduced such that

$$\bar{Q} = \bar{q} + \bar{q}_\infty = \nabla\phi + \bar{q}_\infty \tag{12}$$

where the freestream velocity vector, at angle of attack, α, may be written in the original Cartesian coordinates as

$$\bar{q}_\infty = \sin \alpha \, \hat{j} + \cos \alpha \, \hat{k}. \tag{13}$$

In order that the potential ϕ reduce to the conical flow potential F as defined in Ref 1, let

$$\phi = RF(\rho,\theta,R) \tag{14}$$

where, for conical flows, F will only depend upon ρ and θ.

The transformation of the governing equations (10) and (11) in the nonorthogonal coordinate system ρ, θ, R through equations (1) to (8) is quite tedious. A convenient device for handling the algebra develops from the fact that the ρ and θ coordinates are mutually orthogonal along surfaces with R = constant. This derivation follows a technique used by Moretti (Ref 3) for a nonorthogonal mapped Cartesian coordinate system. The procedure basically consists of transforming the vector operations in Eq (10) to spherical

coordinates using standard orthogonality rules. Then, the crossflow terms (ψ,ω) are further transformed to the (ρ,θ) coordinates using the appropriate two-dimensional orthogonality relationships. The remaining r derivatives are then transformed separately. The procedure is simplified by considering the mixed velocity vector

$$\bar{Q} = V\hat{i}_\rho + U\hat{i}_\theta + W\hat{i}_r \tag{15}$$

Following this prescription, we have

$$\nabla\phi = \nabla_c\phi + \frac{\partial\phi}{\partial r}\,\hat{i}_r = v\hat{i}_\rho + u\hat{i}_\theta + w\hat{i}_r \tag{16}$$

where

$$\nabla_c\phi = \frac{1}{RH}\frac{\partial\phi}{\partial\rho}\,\hat{i}_\rho + \frac{1}{\rho RH}\frac{\partial\phi}{\partial\theta}\,\hat{i}_\theta. \tag{17}$$

The notation ∇_c refers to the "crossflow" gradient, which can be shown to follow the transformation properties of a two-dimensional orthogonal coordinate system.

Introducing the definition, Eq (14), into Eq (16), (17), and (12), the velocity components are obtained as

$$\begin{aligned} V &= v + v_\infty = \frac{1}{H}\frac{\partial F}{\partial\rho} + v_\infty(\rho,\theta,R) \\ U &= u + u_\infty = \frac{1}{\rho H}\frac{\partial F}{\partial\theta} + u_\infty(\rho,\theta,R) \\ W &= w + w_\infty = F + R\frac{\partial F}{\partial r} + w_\infty(\rho,\theta,R) \end{aligned} \tag{18}$$

where

$$v_\infty = \frac{1}{H}\left[\sin\alpha\left(\psi_\rho \cos\psi \sin\omega - \frac{1}{\rho}\psi_\theta \cos\omega\right) - \psi_\rho \cos\alpha \sin\psi\right]$$

$$u_\infty = \frac{1}{H}\left[\sin\alpha\left(\psi_\rho \cos\omega + \frac{1}{\rho}\psi_\theta \cos\psi \sin\omega\right) - \frac{1}{\rho}\psi_\theta \cos\alpha \sin\psi\right] \tag{19}$$

$$w_\infty = \sin\alpha \sin\psi + \cos\alpha \cos\psi$$

and ψ, ω are related to ρ, θ, R through Eq (1) - (7).

Considering the first term in the governing Eq (10), introducing Eq (12) and noting that $\bar{q}_\infty$ is a constant vector, then

$$\nabla \cdot \bar{Q} = \nabla \cdot \bar{q} = \nabla_c \cdot \bar{q} + \frac{1}{r^2}\frac{\partial}{\partial r}(r^2 w). \tag{20}$$

Similarly, the second term of Eq (10) becomes

$$\bar{Q} \cdot \nabla(\bar{Q} \cdot \bar{Q}) = \bar{Q} \cdot [\nabla(\bar{q} \cdot \bar{q}) + 2\nabla(\bar{q}_\infty \cdot \bar{q})] \tag{21}$$

where

$$\bar{Q} \cdot \nabla(\bar{q} \cdot \bar{q}) = \bar{Q} \cdot \nabla_c(\bar{q} \cdot \bar{q}) + W\frac{\partial}{\partial r}(\bar{q} \cdot \bar{q}) \tag{22}$$

and utilizing irrotationality conditions

$$\bar{Q} \cdot \nabla(\bar{q}_\infty \cdot \bar{q}) = \bar{Q} \cdot (\bar{q}_\infty \cdot \nabla)\bar{q} = \bar{Q} \cdot \left[\bar{q}_\infty \cdot \nabla_c + w_\infty \frac{\partial}{\partial r}\right]\bar{q}. \tag{23}$$

The last term of Eq (23) can be developed through direct transformation of Eq (1) - (8) as

$$\frac{\partial \bar{q}}{\partial r} = \left[\frac{\partial v}{\partial r} - \frac{u}{\rho}\frac{\partial}{\partial \theta}\left(\frac{\partial \rho}{\partial r}\right)\right]\hat{i}_\rho + \left[\frac{\partial u}{\partial r} + \frac{v}{\rho}\frac{\partial}{\partial \theta}\left(\frac{\partial \rho}{\partial r}\right)\right]\hat{i}_\theta + \frac{\partial w}{\partial r}\hat{i}_r. \tag{24}$$

The only terms remaining to be transformed involve the r derivatives. Noting that

$$\frac{\partial}{\partial r} = \frac{\partial}{\partial R} + \frac{\partial \rho}{\partial r}\frac{\partial}{\partial \rho} + \frac{\partial \theta}{\partial R}\frac{\partial}{\partial \theta} \equiv \frac{\partial}{\partial R} + h_1\frac{\partial}{\partial \rho} + \frac{1}{\rho}h_2\frac{\partial}{\partial \theta} \tag{25}$$

where $h_1 = h_1(\rho,\theta,R)$ and $h_2 = h_2(\rho,\theta,R)$ are defined in Eq (8). Substituting the expressions (14-25) into the governing Eq (10) and (11), and performing a number of algebraic manipulations, yields the following governing equations:

$$(a^2-V^2)\frac{\partial^2 F}{\partial \rho^2} - 2UV\left(\frac{1}{\rho}\frac{\partial^2 F}{\partial \rho \partial \theta} - \frac{1}{\rho^2}\frac{\partial F}{\partial \theta}\right) + (a^2-U^2)\left(\frac{1}{\rho^2}\frac{\partial^2 F}{\partial \theta^2} + \frac{1}{\rho}\frac{\partial F}{\partial \rho}\right)$$

$$+ (V^2-U^2)\left(v\frac{\partial H}{\partial \rho} - \frac{u}{\rho}\frac{\partial H}{\partial \theta}\right) + 2UV\left(u\frac{\partial H}{\partial \rho} + \frac{v}{\rho}\frac{\partial H}{\partial \theta}\right) + H^2 w(2a^2-V^2-U^2)$$

$$= RH[(W^2-a^2)RHh_1+2WV]\left[h_1\frac{\partial^2 F}{\partial \rho^2} + \frac{h_2}{\rho}\frac{\partial^2 F}{\partial \rho \partial \theta} + \frac{\partial^2 F}{\partial \rho \partial R} + H\left(v\frac{\partial h_1}{\partial \rho} + u\frac{\partial h_2}{\partial \rho} - \frac{u}{\rho}h_2\right)\right]$$

$$+ \frac{1}{\rho}RH[(W^2-a^2)RHh_2+2WU]\left[h_1\frac{\partial^2 F}{\partial \rho \partial \theta} + \frac{h_2}{\rho}\frac{\partial^2 F}{\partial \theta^2} + \frac{\partial^2 F}{\partial \theta \partial R} + H\left(v\frac{\partial h_1}{\partial \theta} + u\frac{\partial h_2}{\partial \theta}\right)\right]$$

$$+ RH^2(W^2-a^2)\left[R\left(h_1\frac{\partial^2 F}{\partial \rho \partial R} + \frac{h_2}{\rho}\frac{\partial^2 F}{\partial \theta \partial R} + 2\frac{\partial^2 F}{\partial R^2}\right) + 2\frac{\partial F}{\partial R} + 2H(vh_1+uh_2)+RH\left(v\frac{\partial h_1}{\partial R} + u\frac{\partial h_2}{\partial R}\right)\right] \tag{26}$$

and

$$a^2 + \frac{\gamma-1}{2}(V^2 + U^2 + W^2) = a_o^2. \tag{27}$$

The above equations, although appearing somewhat formidable, are actually in a significantly simpler form than the general nonorthogonal second-order potential equation which contains 27 first and second derivative transformation terms. For conical flows, all R derivatives vanish and hence the right-hand side of Eq (26) goes to zero, leaving the identical equation developed in Ref 1. This feature enables an initial starting solution for the marching procedure to be developed from a conical flow. The mixed elliptic-hyperbolic nature of the crossflow (U,V) plane is apparent in the left-hand side of Eq (26) and the similarity to the two-dimensional transonic full potential equation can be noted. The equations will be hyperbolic in the R-direction and solutions may be obtained by marching, provided W, the velocity in the r-direction, is greater than the speed of sound, as seen from the coefficient of the F_{RR} term. W is further defined to be:

$$W = w + w_\infty$$

where

$$w = F+R(F_R + h_1 F_\rho + \frac{h_2}{\rho} F_\theta).$$

The governing equation in the final computational domain is obtained through the shearing transformation, Eq (9), as

$$(A_1-B_1)F_{XX}+(A_2-B_2)F_{XY}+(A_3-B_3)F_{YY}+(A_4-B_4)F_Y+A_5F_X+A_6w+(A_7-B_7)$$

$$= B_8F_{ZZ}+B_9F_{XZ}+B_{10}F_{YZ}+B_{11}F_Z \qquad (28)$$

where the coefficients A_i are identical to the conical terms in Ref 1.

$$A_1 = (a^2-U^2)\frac{1}{\rho^2}$$

$$A_2 = -2UV\frac{1}{\rho}Y_\rho+2(a^2-U^2)\frac{1}{\rho^2}Y_\theta$$

$$A_3 = (a^2-V^2)Y_\rho^2-2UV\frac{1}{\rho}Y_\rho Y_\theta+(a^2-U^2)\frac{1}{\rho^2}Y_\theta^2$$

$$A_4 = -2UV\frac{1}{\rho}(Y_{\rho\theta}-\frac{1}{\rho}Y_\theta)+(a^2-U^2)\frac{1}{\rho}(\frac{1}{\rho}Y_{\theta\theta}+Y_\rho)$$

$$A_5 = 2UV\frac{1}{\rho^2}$$

$$A_6 = H^2(2a^2-V^2-U^2)$$

$$A_7 = (V^2-U^2)[vY_\rho H_Y-u\frac{1}{\rho}(H_X+Y_\theta H_Y)]+2UV[uY_\rho H_Y+v\frac{1}{\rho}(H_X+Y_\theta H_Y)]$$

The nonconical terms, B_i are

$$B_1 = \frac{C_2h_2}{\rho^2}$$

$$B_2 = \frac{1}{\rho}(C_1h_2Y_\rho+2C_2h_2Y_\theta+C_2Y_R+C_3Rh_2Y_R+C_2h_1Y_\rho)$$

$$B_3 = C_1 h_1 Y_\rho^2 + \frac{Y_\theta}{\rho}\left(\frac{C_2 h_2 Y_\theta}{\rho} + C_1 h_2 Y_\rho + C_2 h_1 Y_\rho + C_2 Y_R\right) + C_1 Y_\rho Y_R + C_3 R Y_R\left(\frac{h_2 Y_\theta}{\rho} + h_1 Y_\rho + Y_R\right)$$

$$B_4 = C_1\left(h_1 Y_{\rho\rho} + \frac{h_2}{\rho} Y_{\rho\theta} + Y_{\rho R}\right) + \frac{C_2}{\rho}\left(\frac{h_2}{\rho} Y_{\theta\theta} + Y_{\theta R} + h_1 Y_{\rho\theta}\right) + C_3 R\left(h_1 Y_{\rho R} + \frac{h_2}{\rho} Y_{\theta R} + Y_{RR} + \frac{2Y_R}{R}\right)$$

$$B_7 = C_1 H\left(v h_{1\rho} + u h_{2\rho} - \frac{u h_2}{\rho}\right) + \frac{C_2 H}{\rho}\left(v h_{1\theta} + u h_{2\theta}\right) + C_3 H\left[2(u h_2 + v h_1) + R(u h_{2R} + v h_{1R})\right]$$

$$B_8 = C_3 R$$

$$B_9 = \frac{C_2}{\rho} + \frac{C_3 R h_2}{\rho}$$

$$B_{10} = \frac{C_2 Y_\theta}{\rho} + \frac{C_3 R h_2 Y_\theta}{\rho} + C_1 Y_\rho + C_3 R\left(h_1 Y_\rho + 2Y_R\right)$$

$$B_{11} = 2C_3$$

where

$$C_1 \equiv RH\left[(W^2 - a^2) RH h_1 + 2WV\right]$$

$$C_2 \equiv RH\left[(W^2 - a^2) RH h_2 + 2WU\right]$$

$$C_3 \equiv RH^2 (W^2 - a^2).$$

The marching method and nonconical equations were first presented in Ref 4.

The final mapped coordinate derivatives, $h_1 = \rho_r$ and $h_2 = \rho\theta_r$, can be derived by considering an intermediate coordinate system (ξ, η, ζ) defined as

$$\xi = \rho \cos\theta$$

$$\eta = \rho \sin\theta$$

$$\zeta = R.$$

The coordinate derivatives can now be transformed:

$$\rho_r = \rho_\xi \xi_r + \rho_\eta \eta_r$$
$$\theta_r = \theta_\xi \xi_r + \theta_\eta \eta_r. \tag{29}$$

The derivatives ξ_r and η_r must be determined from the conformal mapping, (Eq (3)), or

$$\frac{S-S_o(r)}{S-\bar{S}_o(r)} = \left(\frac{\Gamma-S_o(r)/2}{\Gamma-\bar{S}_o(r)/2}\right)^2$$

where

$$\Gamma = \xi + i\eta$$

and

$$\Gamma_r = \xi_r + i\eta_r.$$

The conformal transformation can be rewritten:

$$\Gamma^2 - \Gamma S + \frac{1}{4}[S(S_o-\bar{S}_o)+S_o\bar{S}_o] = 0.$$

Differentiating Eq (3) with respect to r yields:

$$\Gamma_r = \frac{1}{4}\,\frac{[(S-S_o)\bar{S}_{or}-(S_o+\bar{S}_o)\bar{S}_{or}]}{(2\Gamma-S)}\,.$$

Now,

$$\xi_r = \mathrm{Re}(\Gamma_r)$$

$$\eta_r = \mathrm{Im}(\Gamma_r)$$

and from Eq (29):

$$h_1 = \rho_r = \cos\theta\ \mathrm{Re}(\Gamma_r) + \sin\theta\ \mathrm{Im}(\Gamma_r)$$

and (30)

$$h_2 = \rho\theta_r = \cos\theta\ \mathrm{Im}(\Gamma_r) - \sin\theta\ \mathrm{Re}(\Gamma_r).$$

Thus, given the singularity derivatives S_{o_r}, or $\bar{S}_{o_r}$ (which are necessarily determined numerically) and the conformal transformation, the coordinate derivatives h_1 and h_2 can be determined directly from Eq (30).

3.2 SURFACE BOUNDARY CONDITION

The surface boundary condition requires the vanishing of the normal component of velocity or flow tangency condition. The body surface in the mapped (ρ, θ, R) space is defined as:

$$\rho = b(\theta,R).$$

Let (31)

$$G(\rho,\theta,R) \equiv \rho - b(\theta,R) = 0.$$

The body surface unit normal is defined as:

$$\bar{I}_N = \frac{\nabla G}{|\nabla G|}. \tag{32}$$

Flow tangency then requires that

$$\bar{q} \cdot \bar{I}_N = 0.$$

This condition, using Eq (31) and (32), becomes

$$bV - Ub_\theta + WbRH\left(h_1 - \frac{h_2 b_\theta}{b} - b_R\right) = 0 \tag{33}$$

where

$$U = \frac{1}{\rho H}\left[F_X - \frac{b_\theta F_Y}{(c-b)}\right] + u_\infty$$

$$V = \frac{1}{H}\left(\frac{F_Y}{c-b}\right) + v_\infty$$

$$W = F+R\left[F_Z - \frac{b_R F_Y}{(c-b)}\right] + \frac{h_1 R F_y}{(c-b)} + \frac{h_2 R}{b}\left[F_X - \frac{b_\theta F_Y}{(c-b)}\right] + w_\infty .$$

The body boundary condition is implemented by defining a dummy row of coordinates below the body surface. The value of the potential at this grid line is then determined by solving Eq (33) for the derivative F_Y, or

$$F_Y = (c-b)\left[\frac{b_\theta F_X + bH(b_\theta u_\infty - bv_\infty) - RH^2\lambda^2(bF+bRF_Z+h_2RF_X+bw_\infty)}{(b^2+b_\theta^2 + R^2H^2\lambda^2)}\right] \qquad (34)$$

where

$$\lambda \equiv bh_1 - h_2 b_\theta - bb_R .$$

If j=2 defines the body surface coordinate line, then the potential at j = 1 is determined by:

$$F_{i,1} = F_{i,3} - 2\Delta Y\ Fy .$$

In the far field, outside the bow wave, the perturbation potential and its first derivatives must vanish

$$F, \frac{\partial F}{\partial X}, \frac{\partial F}{\partial Y}, \frac{\partial F}{\partial Z} \rightarrow 0 \quad \text{outside the bow wave.} \qquad (35)$$

This Page Intentionally Left Blank

4. NUMERICAL FORMULATION

4.1 BASIC SLOR SCHEME

The numerical solution of the governing three-dimensional full-potential Eq (28), subject to boundary conditions (33) and (35), is developed through a hyperbolic marching procedure in the Z (mapped spherical radius) direction. First-order backward differences are used to approximate all Z-derivatives of the potential F, on the right-hand side of Eq (28).

In the case of conical flow, $\partial/\partial r = 0$ (i.e., $\partial/\partial R = 0$, $h_1 = 0$, $h_2 = 0$), Eq (28) becomes independent of the spherical radius R(Z), and the terms involving the second derivatives of F are of the identical form as the two-dimensional transonic full-potential equation. The resulting conical equation is of mixed type, being elliptic when the crossflow is:

$$Q_c = (U^2+V^2)^{1/2} < a$$

and hyperbolic when $Q_c > a$. Since Q_c must necessarily be zero on the body at the symmetry plane, it is to be expected that, if the crossflow becomes supersonic, it will terminate in an embedded shock wave.

In order to utilize Jameson's rotated difference scheme (Ref 5), the local streamline direction, s, and the direction normal to the streamline n in the crossflow plane (R = constant) are defined as:

$$Q_c \frac{\partial}{\partial s} = V \frac{\partial}{\partial \rho} + \frac{U}{\rho} \frac{\partial}{\partial \theta} = V_1 \frac{\partial}{\partial Y} + U_1 \frac{\partial}{\partial X} \tag{36a}$$

$$Q_c \frac{\partial}{\partial n} = - U \frac{\partial}{\partial \rho} + \frac{V}{\rho} \frac{\partial}{\partial \theta} = - U_2 \frac{\partial}{\partial Y} + V_2 \frac{\partial}{\partial X} \tag{36b}$$

where Q_c is the magnitude of the crossflow velocity, and

$$V_1 = VY_\rho + (U/\rho)Y_\theta, \quad U_1 = U/\rho \tag{37a}$$

$$U_2 = UY_\rho - (V/\rho)\, Y_\theta, \quad V_2 = V/\rho. \tag{37b}$$

The second derivatives in the s and n directions can be approximated by their principal terms as:

$$Q_c^2 \frac{\partial^2}{\partial s^2} = V_1^2 \frac{\partial^2}{\partial Y^2} + 2U_1V_1 \frac{\partial^2}{\partial X \partial Y} + U_1^2 \frac{\partial^2}{\partial X^2} \tag{38a}$$

$$Q_c^2 \frac{\partial^2}{\partial n^2} = U_2^2 \frac{\partial^2}{\partial Y^2} - 2U_2V_2 \frac{\partial^2}{\partial X \partial Y} + V_2^2 \frac{\partial^2}{\partial X^2}. \tag{38b}$$

Hence, the leading conical terms of Eq (28) can be written in terms of local crossflow streamline directions as

$$A_1 \frac{\partial^2 F}{\partial X^2} + A_2 \frac{\partial^2 F}{\partial X \partial Y} + A_3 \frac{\partial^2 F}{\partial Y^2} = (a^2 - Q_c^2) \frac{\partial^2 F}{\partial s^2} + a^2 \frac{\partial^2 F}{\partial n^2}. \tag{39}$$

Equation (39) will be utilized in the numerical solution of Eq (28).

It can be seen from Eq (39) that in the case of conical flow ($B_i = 0$) in Eq (28), Eq (39) provides an explicit and exact insight into the elliptic and hyperbolic nature of the governing equation. Equation (39) also provides a means for correctly differencing the governing equation to obtain the correct domain of dependence and stability.

In formulating difference equations to approximate the governing Eq (28), the mixed-flow line relaxation procedure which has been successful for transonic flows are utilized. These procedures stem from the work by Murman and Cole (Ref 6) for the transonic small-disturbance equation. In particular, the formulation postulated by Jameson (Ref 5) for the transonic full-potential equation is adopted.

The computation is started by first solving the conical part of Eq (28) at R = 0, the apex of the wing. The numerical formulation is broken into two regions, subsonic and supersonic crossflow.

The windward finite difference forms of the F_{ss} derivatives lead to artificial time derivatives of an equivalent time-dependent equation. The F_{ss} derivative will create a term proportional to:

$$(a^2-Q_c^2)\left(\frac{U}{W_c}\frac{\Delta t}{\Delta X}+\frac{V}{Q_c}\frac{\Delta t}{\Delta Y}\right)F_{st}$$

in the equivalent time-dependent equation.

In order to maintain stability, diagonal dominance, and uniform convergence when the crossflow Mach number is close to unity, an upwind approximation to F_{st} is further added at both subsonic and supersonic points (Ref 7). This term has the form

$$\sim \varepsilon \Delta t F_{st}$$

where ε is a constant and the temporal cross derivative is approximated by

$$F_{st} = \frac{V_1}{Q_c}F_{Yt} \pm \frac{U_1}{Q_c}F_{Xt} = \frac{V_1}{Q_c \Delta Y}\left(F^+_{i,j+1}-F^+_{i,j}-F_{i,j+1}+F_{i,j}\right)$$

$$\pm \frac{U_1}{Q_c \Delta X}\left(F^+_{i,j}-F^+_{i-1,j}-F_{i,j}+F_{i-1,j}\right) \tag{40}$$

As mentioned previously, the surface boundary condition, Eq (33), and the use of a dummy grid line below the body surface (j = 1) allows the use of the same finite difference expressions at the body boundary j = 2 as the interior points.

Similarly, at the symmetry planes i = 2 and i = ic, we introduce dummy columns i = 1 and i = ic + 1 and symmetry conditions mandate that

$$F_{i,j} = F_{3,j}$$

$$F_{ic+1,j} = F_{ic-1,j}$$

At the outer boundary j = jc, $\rho = c(\theta)$, the value of the potential is set as

$$F_{i,jc} = 0$$

A fully-implicit, coupled set of finite-difference equations is obtained for the perturbation potential F at all values of the crossflow plane (i,j) at the present Z station. This particular formulation of the marching procedure has the advantage of retaining the successful conical-flow relaxation procedure of Ref 1 with the second Z-derivatives included implicitly. Furthermore, since the crossflow difference equations are fully implicit, the radial step size ΔZ will not be subject to any stability limitations except those dictated by geometric accuracy requirements.

The crossflow solution is determined using type-dependent line relaxation techniques (SLOR) similar to those developed by Jameson (Ref 5) for the transonic full-potential equation. The crossflow relaxation process was shown to very accurately capture the bow wave and embedded shocks for conical flows (Ref 1).

In the present three-dimensional applications, the crossflow plane Z = constant will not be normal to the bow and imbedded shocks, and the rotated difference scheme will not be as effective in capturing shock waves as they become more oblique to the grid. This will result in oblique shocks being spread over sightly more mesh points than for normal shocks. This can be eliminated at the bow shock by fitting the bow shock as a boundary. Fitting the bow shock makes the computation conservative except in regions downstream of embedded shocks. A subsequent section (Section 10) deals with improvements to the 3-D numerical scheme, in order to more reliably capture embedded oblique shocks.

Two complete crossflow solutions for the potential function F are needed to march due to the second derivative terms in Z. If the geometry begins from a conical body, the starting solutions are determined from the governing equations (27) and (28), with all the Z (or R) derivatives set to zero. The right-hand side of Eq (28) then vanishes and the conical solution is found, as described in Ref 1. For nonconical initial geometries, a very small conical "nose" is assumed to exist and conical initial conditions are again utilized. This procedure should be more accurate than the more approximate, and often time-consuming methods used for the initialization of Euler's equations solutions.

The basic numerical procedure outlined here might appear to be quite inefficient, because the marching solution requires the complete convergence

of an implicit relaxation method at each Z station, a step usually requiring many iterations for conical flows. However, it has been found for the applications considered here, that each crossflow station does not vary substantially from the previous one and, except for the initial conical solution, the remaining crossflow solutions usually converge in 20-40 iterations.

The finite-difference analog of Eq (28) along with the boundary conditions, Eq (34), (45), and (46), are written in tridiagonal form for values of the correction $F^{+}_{i,j} - F_{i,j}$, along a coordinate line. These expressions can be solved for by Gaussian elimination, using successive line over relaxation (SLOR) along rows (j = constant) or columns (i = constant).

Employing an SLOR technique along i = constant lines (from body to bow shock or outer boundary) was found to be more efficient (see Ref 7 or 8) than the earlier methods using ring relaxation (Ref 1). References 9 and 10 employ the SLOR method to solve the governing equations. More recently, approximate factorization methods have also been found to be even more efficient (Ref 11). In the AF methods, which are discussed in the next section, both ring and column relaxations are used simultaneously to insure the fastest propagation or communication of disturbances.

4.2 APPROXIMATE FACTORIZATION SCHEMES

Numerical techniques are now presented for accelerating convergence in comparison with the standard SLOR methods. The primary candidate for this is the alternating-direction-implicit (ADI) (Ref 12) or, as it is more commonly referred to in its application to nonlinear transonic flows, approximate factorization (AF) schemes. These AF schemes have been applied successfully to a variety of transonic flow problems. Initially, AF schemes were applied to the Transonic Small Disturbance (TSD) equation by Ballhaus, et al (Ref 13). Holst (Ref 14) successfully applied an AF2 type scheme to the conservative full potential equation for transonic flows. The nonconservative full potential equation was treated successfully by Baker(Ref 15 and 16) for 2D transonic flows and should be applicable to the nonconservative full potential supersonic/transonic crossflow problem of the present study. Two basic AF algorithms are considered, ADI or AF1 and the AF2 scheme which splits one of the second derivatives into two first-order derivative operators. The latter scheme has reportedly been the most stable in supersonic flow regions

for the transonic flow problem.

4.2.1 BASIC FORMULATION - CONICAL FLOWS

The nonconservative form of the 3D full potential equation is written in a spherical coordinate system (ω, ψ, r). The governing equation is then transformed via a conformal stereographic projection to (p, q, t) coordinates and further by a crossflow conformal Jowkowski mapping to (ρ, θ, R) coordinates. In terms of a reduced potential F, where $Q = \nabla\phi + q_\infty$ and $\phi = RF(\rho,\theta,R)$, the full potential equation can be written as

$$\frac{(a^2-U^2)}{\rho^2} F_{\theta\theta} - \frac{2UV}{\rho} F_{\rho\theta} + (a^2-V^2)F_{\rho\rho} + \ldots$$

$$= RH \left[(W^2-a^2)RHh_1 + 2WV\right]\left[h_1 F_{\rho\rho} + \frac{h_2}{\rho} F_{\rho\theta} + F_{\rho R}\right.$$

$$+ \frac{1}{\rho}\left[(W^2-a^2)RHh_2 + 2WU\right]\left[h_1 F_{\rho\theta} + \frac{h_2}{\rho} F_{\theta\theta} + F_{\theta R}\right]$$

$$+ RH(W^2-a^2)\left[h_1 F_{\rho R} + \frac{h_2}{\rho} F_{\theta R} + F_{RR}\right] + \ldots. \qquad (41)$$

where U, V, and W are the velocities in the θ, ρ, and R directions, respectively, and H is the combined metric of the two mappings. In general, the conformal mapping in the crossflow plane leads to nonorthogonal coordinate derivatives if the mapping singularity is a function of r. This mapping dependence on r leads to mesh derivatives defined as $h_1 = \rho_r$ and $h_2 = \rho\theta_r$. The radial marching direction r, remains unchanged due to the transformations, or $r = t = R$.

Unlike transonic flow, the supersonic flow problem is contained within a finite crossflow mesh bounded by a bow shock. The bow shock may be captured within a prescribed outer boundary (Ref 1 or 4) or the bow shock can be fitted as the outer boundary (Ref 7 or 8) using the isentropic shock jump conditions.

A shearing transformation is applied to Eq (41), between the body $\rho = B(\theta, R)$ and outer boundary or bow shock $\rho = C(\theta, R)$, where

$$X = \theta$$

$$Y = \frac{\rho - B}{C - B} \tag{42}$$

$$Z = R$$

which yields a final rectangular computational mesh.

Equation (41) can be represented as the sum of a conical plus a nonconical operator, in the form

$$L(\phi_{i,j}) = L_C(\phi_{i,j}) + R \cdot L_{NC}(\phi_{i,j}) \cdot \tag{43}$$

The nonconical coefficients on the RHS of Eq (41) all have an R dependence and vanish identically at R = 0 for the quasi-two-dimensional conical flow problem.

For conical flows, after applying the shearing transformation, Eq (41) can be written as

$$L_C(\phi_{i,j}) = A_1 F_{XX} + A_2 F_{XY} + A_3 F_{YY} + \cdots \tag{44}$$

where

$$A_1 = \frac{(a^2-U^2)}{\rho^2}$$

$$A_2 = \frac{-2UV}{\rho} Y_\rho + 2\frac{(a^2-U^2)}{\rho^2} Y_\theta \tag{45}$$

$$A_3 = (a^2-V^2)Y_R^2 - \frac{2UV}{\rho} Y_\rho Y_\theta + \frac{(a^2-U^2)}{\rho^2} Y_\theta^2 \cdot$$

Equation (44) closely resembles the nonconservative form of the 2-D transonic flow equation. The difference is that the type dependency of the conical part of Eq (41) or Eq (44) is linked to the nature of the crossflow velocity defined by $Q_c^2 = U^2 + V^2$. An upwind bias in the difference equations must occur when the crossflow velocity is supersonic or $Q_c^2 > a^2$. The crossflow velocity component V is always negative and toward the body

surface. The U component of velocity can be positive or negative depending upon the geometry and angle of attack. Equation (44) has heretofore been solved successfully using transonic SLOR techniques and the rotated difference scheme of Jameson (Ref 5).

The principal part of Eq (44) can be rewritten in a rotated difference format as, for $Q_c^2 > a^2$

$$A_1 F_{XX} + A_2 F_{XY} + A_3 F_{YY} = (a^2 - Q_c^2)\, F_{ss} + a^2 F_{nn} \tag{46}$$

where

$$F_{ss} = \frac{V_1^2}{Q_c^2} F_{YY} + \frac{2U_1 V_1}{Q_c^2} F_{XY} + \frac{U_1^2}{Q_c^2} F_{XX}$$

$$F_{nn} = \frac{U_2^2}{Q_c^2} F_{YY} - \frac{2U_2 V_2}{Q_c^2} F_{XY} + \frac{V_2^2}{Q_c^2} F_{XX}$$

and

$$U_1 = U/\rho \quad , \quad V_1 = VY_\rho + \frac{UY_\theta}{\rho}$$

$$V_2 = V/\rho \quad , \quad U_2 = UY_\rho - \frac{VY_\theta}{\rho} \, .$$

An upwind bias is applied to the finite difference representation of the F_{ss} terms, and central differences are used for the F_{nn} terms in supersonic crossflow regions.

The following sections will present an adaptation of the two basic AF algorithms, ADI or AF1 and AF2, to the present supersonic flow problem.

4.2.2 AF1 Factorization

An ADI or AF1 type factorization can be applied to the principal terms of Eq (44), for subsonic crossflow $Q_c^2 < a^2$, in the form

$$\left(\alpha - A_1 \frac{\vec{\delta}_X \vec{\delta}_X}{\Delta X^2}\right)\left(\alpha - A_3 \frac{\vec{\delta}_Y \vec{\delta}_Y}{\Delta Y^2}\right) \Delta_{i,j}^{n+1} = \alpha\omega L_c(\phi_{i,j}^{n}) \tag{47}$$

where $\Delta_{i,j}$ is the correction to the reduced potential $F_{i,j}$ or $\Delta_{i,j} = F_{i,j}^{n+1} - F_{i,j}^{n}$ at the $^{n+1}$ interaction. ω is a relaxation parameter and $L_c(\phi_{i,j})$ is the residual of Eq (44) at the nth iteration. α is an acceleration parameter which is varied in a cyclic fashion during the iteration process. The two first-order difference operators result in a second-order central difference. The first-order difference operators are defined as

$$\begin{aligned} \vec{\delta}_X &= (\ )_{i+1,j} - (\ )_{i,j} \\ \overleftarrow{\delta}_X &= (\ )_{i-1,j} - (\ )_{i,j} \\ \vec{\delta}_Y &= (\ )_{i,j+1} - (\ )_{i,j} \\ \overleftarrow{\delta}_Y &= (\ )_{i,j} - (\ )_{i,j-1}. \end{aligned} \tag{48}$$

The basic premise behind an approximate factorization scheme can be revealed if the LHS of Eq (47) is expanded and terms not resembling those of Eq (44) are neglected, or

$$\left(A_1 \frac{\vec{\delta}_X \overleftarrow{\delta}_X}{\Delta X^2} + A_3 \frac{\vec{\delta}_Y \overleftarrow{\delta}_Y}{\Delta Y^2} \right) \Delta_{i,j}^{n+1} = - \omega L_c(\phi_{i,j}^{n}) \ . \tag{49}$$

If $\omega = 1$, Eq (49) would be equivalent to solving Eq (44) with the cross derivative evaluated using old values of the potential. Equation (49) is typically solved in a two-step format by defining an intermediate variable, $G_{i,j}$, where

$$\begin{aligned} \left(\alpha - A_1 \frac{\vec{\delta}_X \overleftarrow{\delta}_X}{\Delta X^2} \right) \ G_{i,j}^{n+1} &= \alpha\omega L_c(\phi_{i,j}^{n}) \\ \left(\alpha - A_3 \frac{\vec{\delta}_Y \overleftarrow{\delta}_Y}{\Delta Y^2} \right) \ \Delta_{i,j}^{n+1} &= G_{i,j}^{n+1} \ . \end{aligned} \tag{50}$$

Equation (50) represents two tridiagonal systems of equations involving differences only in the computational X or Y direction. Equation (50) must be modified in regions of supersonic crossflow to include the proper upwind bias. The following form of the supersonic crossflow factorization is essentially identical to the AF1 scheme of Baker (Ref 15 and 16) equation. Hence, for $Q_c^2 > a^2$,

$$\left(\alpha - A_{1c} \frac{\vec{\delta}_X \overleftarrow{\delta}_X}{\Delta X^2} - K_1 A_{1u} \frac{\overleftarrow{\delta}_X \overleftarrow{\delta}_X}{\Delta X^2} - K_2 A_{1u} \frac{\vec{\delta}_X \vec{\delta}_X}{\Delta X^2} \right) G_{i,j}^{n+1} = \alpha\omega L(\phi_{i,j}^{n}) \tag{51}$$

$$\left(\alpha - A_{3c} \frac{\vec{\delta}_Y \overleftarrow{\delta}_Y}{\Delta Y^2} - A_{3u} \frac{\vec{\delta}_Y \vec{\delta}_Y}{\Delta Y^2} \right) \Delta_{i,j}^{n+1} = G_{i,j}^{n+1} .$$

The central (subscript c) and upwind coefficients (subscript u) are given by the rotated difference scheme Eq (46) as

$$A_{1c} = \frac{V_2^2}{Q_c^2} \quad , \quad A_{3c} = \frac{U_2^2}{Q_c^2}$$

$$A_{1u} = \frac{(a^2 - Q_c^2)}{Q_c^2} U_1^2 \quad , \quad A_{3u} = \frac{(a^2 - Q_c^2)}{Q_c^2} V_1^2 . \tag{52}$$

The first factor allows for the U component of velocity to be positive or negative, where if

$$U > 0, \quad K_1 = 1, \quad K_2 = 0$$

$$U < 0, \quad K_1 = 0, \quad K_2 = 1. \tag{52a}$$

As mentioned earlier, the V velocity is always negative for supersonic crossflow and, hence, only forward upwind differences occur in the second factor.

In general, the first factor involves a pentadiagonal matrix and the second factor a quadradiagonal matrix. As suggested by Baker (Ref 15), these differences can be replaced by

$$\overleftarrow{\delta}_X\overleftarrow{\delta}_Y = (\ )^{n+1}_{i,j} - (\ )^{n}_{i,j} - (\ )^{n}_{i-1,j} + (\ )^{n}_{i-2,j}$$

$$\overrightarrow{\delta}_X\overrightarrow{\delta}_X = (\ )^{n+1}_{i,j} - (\ )^{n+1}_{i+1,j} - (\ )^{n}_{i+1,j} + (\ )^{n}_{i+2,j} \tag{53}$$

$$\overrightarrow{\delta}_Y\overrightarrow{\delta}_Y = (\ )^{n+1}_{i,j} - (\ )^{n+1}_{i,j+1} - (\ )^{n}_{i,j+1} + (\ )^{n}_{i,j+2} \ .$$

This reduces the set of Eq (51) to the following tridiagonal form, for $Q_c^2 > a^2$,

$$\left[\alpha - A_{1c}\frac{\overrightarrow{\delta}_X\overleftarrow{\delta}_X}{\Delta X^2} - I_s\frac{A_{1u}}{\Delta X^2}\left(K_1\overleftarrow{\delta}_X + K_2\overrightarrow{\delta}_X\right)\right] G^{n+1}_{i,j} = \alpha\omega L_c(\phi^n_{i,j}) \tag{54}$$

$$\left(\alpha - A_{3c}\frac{\overrightarrow{\delta}_Y\overleftarrow{\delta}_Y}{\Delta Y^2} + A_{3u}\frac{\overrightarrow{\delta}_Y}{\Delta Y^2}\right)\ \Delta^{n+1}_{i,j} = G^{n+1}_{i,j}$$

where, for $U > 0$, $I_s = 1$ and $U < 0$, $I_s = -1$.

4.2.3 AF2 Factorization

The AF2 algorithm has been found by several investigators (Ref 13 and 14) to generally be the more stable AF factorization for transonic flows. In the AF2 factorization, one of the second derivatives is split between the two factors. Following a similar AF2 factorization used by Baker (Ref 15), for subsonic crossflow, $Q_c^2 < a^2$, the AF2 algorithm becomes

$$\left(\frac{-\alpha}{H_m}\frac{\overrightarrow{\delta}_Y}{\Delta Y} - A_1\frac{\overrightarrow{\delta}_X\overleftarrow{\delta}_X}{\Delta X^2}\right)\left(\alpha + A_3H_m\frac{\overleftarrow{\delta}_Y}{\Delta Y}\right)\ \Delta^{n+1}_{i,j} = \alpha\omega L_c(\phi^n_{i,j}) \ . \tag{55}$$

This form of factorization is thought to be more stable since the Y operator in the first factor yields a ϕ_{Yt} term, unlike the ϕ_t term of the AF1 scheme (Ref 16). The term H_m in the two factors accounts for the transformation derivatives of the mapped and sheared mesh. As illustrated by Catherall (Ref 17), the proper splitting of these transformation derivatives can yield optimum convergence, and neglecting these derivatives can

considerably degrade convergence performance. The coefficient A_3 in Eq (44) contains the shearing transformation derivatives Y_ρ and Y_θ/ρ. The mapped plane (ρ,θ) velocities, V and U respectively, contain the metric $H(\rho,\theta)$ of the two mappings. Hence, a suitable form for the factor H_m might be

$$H_m = \frac{\Delta Y}{\Delta X}\frac{Y_\rho}{H} \quad \text{or} \quad \frac{\Delta Y}{\Delta X}\frac{1}{H}\left(Y_\rho - \frac{Y_\theta}{\rho}\right) . \tag{56}$$

The first-order forward difference operator on Y is placed in the first factor since this term does not switch for supersonic crossflow conditions. For supersonic crossflow, the AF2 factorization is modified to include an upwind bias and a tridiagonal form as

$$\left[\frac{-\alpha}{H_m}\frac{\overleftarrow{\delta}_Y}{\Delta Y} - A_{1c}\frac{\overleftarrow{\delta}_X\delta_X}{\Delta X^2} - \frac{I_s A_{1u}}{\Delta X^2}\left(K_1\delta_X + K_2 d\delta_X\right)\right] G^{n+1}_{i,j} = \alpha\omega L_c(\phi^n_{i,j}) \tag{57}$$

$$\left(\alpha + A_{3c}H_m\frac{\overleftarrow{\delta}_Y}{\Delta Y} + A_{3u}H_m\frac{\overleftarrow{\delta}_Y}{\Delta Y}\right)\Delta^{n+1}_{i,j} = G^{n+1}_{i,j} .$$

Since the forward Y-operator is included in the first factor, the sweep is constrained to be toward the body surface or decreasing J.

An alternate factorization to the AF2 scheme described above would be to split the X-derivative. Unlike transonic flow, the difficulty in splitting the X-derivative in the supersonic problem is that the U-velocity in the supersonic crossflow region can be either positive or negative. This occurs primarily at the captured bow shock. In transonic flow, the X- or U-velocity direction is much like the Y-direction of the present problem in that a negative supersonic U-velocity is unlikely to occur. Hence, there is no first order X-operator that, in general, does not switch. One could propose a scheme where the factorization is set up for $U < 0$, and if $U > 0$, the upwind coefficient is either neglected or a shift operator is imposed. This scheme was found to be unstable or would not work for this problem.

Other AF2 factorizations were considered, including the AF3 factorization of Baker (Ref 15) where both coefficients are brought into the first factor. This would seem to be a candidate for a faster scheme since the differential operators would not act on the coefficients and lead to spurious terms. Baker

(Ref 15), in fact, has reported his AF3 scheme to be considerably faster than either the AF1 or AF2 schemes. Unfortunately, this scheme could not be applied successfully to Eq (57).

4.2.4 Boundary Conditions

Figure 2 illustrates the conformally mapped and sheared computational crossflow plane domains. Symmetry conditions are imposed at $\theta = \pm \pi/2$ or I = 2 and IC for the symmetric half-plane problem. Hence, periodic end conditions apply on Y = constant lines. On X = constant lines, J = 2 corresponds to the body surface and J = JMAX corresponds either to the outer boundary (BSC) or the bow shock (BSF). In both the bow shock capture or fit methods, the outer boundary has the same condition that the correction to the potential vanish, or

$$\Delta_{i,jmax} = 0 \ . \tag{58}$$

A dummy row or Y = constant line at J = 1 is used to implement the body boundary condition of flow tangency. This condition relates the values of the correction at J = 1 to those at J = 3, or for conical flow yields:

$$\Delta_{i,1} = \Delta_{i,3} \ . \tag{59}$$

In this way, central derivatives can be used for F_Y and the body surface coordinate line can be treated like any other coordinate line.

The order of the factors in both AF schemes were chosen so that the first sweep is carried out on Y = constant lines. This was chosen over the reverse factorization so that periodic end conditions could be imposed on the intermediate variable G, or

$$\begin{aligned} G_{1,j} &= G_{3,j} \\ G_{IC+1,j} &= G_{IC-1,j} \end{aligned} \tag{60}$$

which most certainly is a reasonable assumption. If the factors in the AF schemes are reversed, then somewhat arbitrary boundary conditions must be

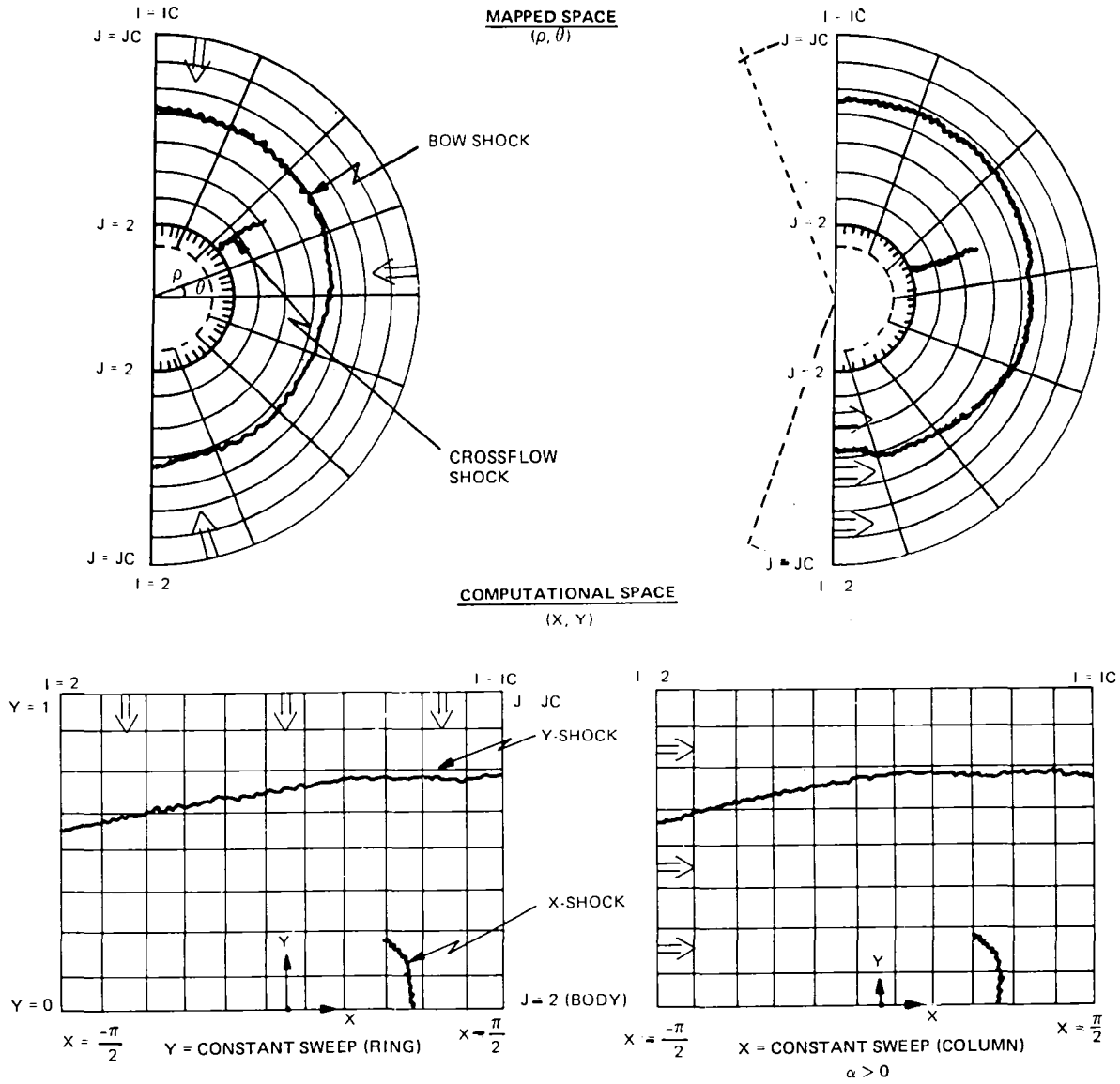


Fig. 2 Sweep and Mesh Definitions

imposed on the intermediate variable G. The end conditions (58) and (59) then apply to the second sweep on the X = constant lines.

4.2.5 Temporal Damping

It has been indicated that the basic AF1 scheme may be unstable in supersonic regions. To allow for this possibility, the AF1 scheme has been generalized to include an explicit temporal damping F_{st} (e.g., ϕ_{st}) term. Jameson's generalized AF scheme (Ref 18) includes similar terms in both factors. In Reference 14, it was indicated that this term may also be required in the AF2 scheme. It was found that adding this term explicitly to the first factor was sufficient to maintain stability for flows with captured shocks. This stability or temporal damping term has the form (Ref 8):

$$-\varepsilon\ \delta_{st} = -\ \varepsilon\ \frac{V_1}{Q_c}\ \bar{\delta}_Y + \frac{U_1}{Q_c}\ (K_1\bar{\delta}_X + K_2\bar{\delta}_X)$$

where

$$\frac{K}{|1-M_c|} \leq \varepsilon \leq \varepsilon_{max}\ K \qquad (61)$$

and

$$\varepsilon_{max} = 10.$$

In general, the addition of the temporal damping slows convergence. Hence, the form of the factor ε was chosen to maximize the damping in the vicinity of sonic lines or across shocks so as not to cause an overall degradation in the convergence rate. The constant K is chosen to be as small as possible for the optimum convergence.

In general, there are no restrictions on the sweep due to the velocity directions in an AF1 scheme. The first sweep can be from the outer boundary to the body surface or the reverse. In order to properly include temporal damping in the AF1 scheme, the first sweep must be in the direction of the supersonic crossflow. This requires that the Y = constant sweep must be towards the body or decreasing J since $V < 0$.

In the supersonic problem (unlike the transonic case), it was observed that the AF1 scheme required little or no temporal damping, except for the most difficult cases, whereas the AF2 scheme required considerably higher values of ε for the strong Y-shock solutions.

4.2.6 Acceleration Parameters

For the AF1 scheme, the maximum and minimum values of the acceleration parameter α were taken as

$$\alpha_{max} = A_{max}\left(\frac{1}{\Delta X^2}, \frac{1}{\Delta Y^2}\right)$$
$$\alpha_{min} = A_{min}\left(\frac{1}{\Delta X}, \frac{1}{\Delta Y}\right). \tag{62}$$

The coefficients A_{min} and A_{max} were taken anywhere from 0.5 to 4.0 for all the cases computed. The convergence rate could be affected by as much as a factor of two by a judicious choice of these parameters.

In the AF2 scheme, these parameters were chosen as

$$\alpha_{max} = A_{max}\left(\frac{1}{\Delta X}, \frac{1}{\Delta Y}\right)$$
$$\alpha_{min} = A_{min}\left(\frac{1}{\Delta X}, \frac{1}{\Delta Y}\right) \tag{63}$$

where A_{min} varied between 0.5 and 6 and A_{max} between 3 and 6. Typically, unity for A_{min} and three for A_{max} was sufficient for most cases. The theoretical value of α_{min} = 1 or 2 could never be achieved, possibly due to the effect of the various coordinate transformations.

For both AF schemes, the cyclic variation of α took the form

$$\alpha_I = \alpha_{max}\left(\frac{\alpha_{min}}{\alpha_{max}}\right)^{\frac{I-1}{IMAX-1}} \quad \text{for } I = 1, IMAX \tag{64}$$

where IMAX = 3 for the AF1 scheme and IMAX = 6 for the AF2 scheme. The minimum number of cycles seemed to be the best choice for the AF1 scheme. The convergence rate of the AF2 scheme was affected insignificantly for IMAX between 3 and 6. Further increase in IMAX reduced the convergence rate. In all of the cases computed, ω_{AF1} = 1.50 or 1.70 and ω_{AF2} = 1.33. Departures from Eq (64) did not seem to affect significantly the convergence rates.

4.2.7 Conical Results

Two techniques are available for the computation of supersonic conical flows: the bow shock capture (BSC) and bow shock fit (BSF) methods. In the BSC method, an outer mesh boundary $\rho = C(\theta)$ is prescribed, and the bow shock is captured within this boundary. The BSC method is a more stringent test of the AF schemes in that two shocks may be present in the flow, the bow shock (Y-shock) and an embedded crossflow shock (X-shock) as illustrated in Fig. 2. The bow shock is the most critical in that its position and strength largely determine the internal flow. The bow shock also extends around the entire field encompassing more points than the embedded crossflow shock. The BSF method fits the bow shock as the outer boundary and, hence, eliminates the bow shock from the internal flow calculation; if an embedded supersonic crossflow region is not present, the internal flow problem becomes elliptic. The conical convergence rate of the BSF method is largely determined by the implicit shock fitting procedure where the shape of the boundary is updated and usually underrelaxed until the isentropic shock jump condition is fulfilled at each shock mesh point. The conical BSF method also requires more computational time per iteration because the crossflow mesh and metrics must be recomputed for each iteration after the bow shock shape has been updated.

For conical flow, the BSC method is used primarily to evaluate the AF schemes. Figure 3 shows the effect of reversing the order of the factors in the AF1 scheme for a thin elliptic cone at $M_\infty = 2.0$, $\alpha = 0°$. The AF1XY scheme represents the order of the factors indicated in Eq (7) and (14) where periodic end conditions are used for the intermediate variable G. The factors were then reversed (AF1YX) with the first sweep occurring on X = constant lines. Two different boundary conditions were used: setting $G_{i,1} = 0$ as the end condition in the tridiagonal on the dummy row below the body; and using $G_{i,1} = G_{i,3}$ as the periodic end condition, as is the case for the condition on the correction $\Delta_{i,1}$.

All three cases were run with the same α variation. As shown in Fig. 3, the results are quite sensitive to the boundary condition on the intermediate variable, and making the intermediate variable mimic the correction seems to be the best choice. Even with this boundary condition, the YX factorization does not give identical results to the XY factorization and seemed to be

somewhat more sensitive to the α variation. Hence, the AF1XY scheme was used for all the computations.

Figure 4 shows a comparison of the convergence rate of the maximum residual for the AF1 and SLOR schemes for a thin subsonic leading edge elliptic cone (θ_c = 20°, δ_c = 2°) on a (48 x 38) crossflow mesh at M_∞ = 2 and angles of attack α = 0° and 10°. The SLOR scheme found to be optimum for this problem in Ref 8 is one that sweeps around the body on X = constant lines. The "column" SLOR scheme was found to be two to five times faster than the alternate SLOR scheme which sweeps toward the body on Y = constant lines. It is interesting to note that after one or two orders of magnitude reduction in the maximum residual, the SLOR scheme for the supersonic freestream problem does not exhibit the typical slowdown in convergence rate that occurs in transonic flow problems. A break in the SLOR curve occurs after one order of magnitude, but remains linear for further reductions. It is also interesting to note that the SLOR convergence rates of the α = 0° and α = 10° flows are not dramatically different, both taking about 350 to 400 iterations, considering that the 10° case is a multishocked flow. At α = 10°, a strong crossflow shock develops but is evidently overshadowed by the convergence of the captured bow shock. The AF1 scheme at α = 0° converges very quickly. Essentially, these flows are converged when the maximum residual reaches 10^{-2}. For α = 0°, this occurs at about 10 iterations or when the log (RESMAX) = -2. The AF1 scheme is an order of magnitude faster than the SLOR scheme for α = 0°. As the angle of attack increases, the AF1 scheme slows down by a factor of 3 while the SLOR remains about the same. Overall, the AF1 scheme is at least three times faster iteration wise than the SLOR scheme. The relaxation factor was 1.5 for these cases in both schemes, and three cycles were used in the AF1 scheme. A larger number of cycles did not seem to enhance convergence. Figure 5 shows the surface pressure distributions for the elliptic cone computed in Fig. 4 at α = 0°, 5°, and 10° and M_∞ = 2.0 on a finer mesh. Both α = 5° and α = 10° have crossflow shocks on the leeward surface. Figure 6 shows the computed crossflow streamline pattern at α = 10°. As mentioned earlier, the V component of velocity is negative. The crossflow streamlines emanate from the bow shock and travel toward the body surface coalescing at the leeward and windward vortical singularities. One streamline stagnates on the body and wets the body surface. Also shown is the

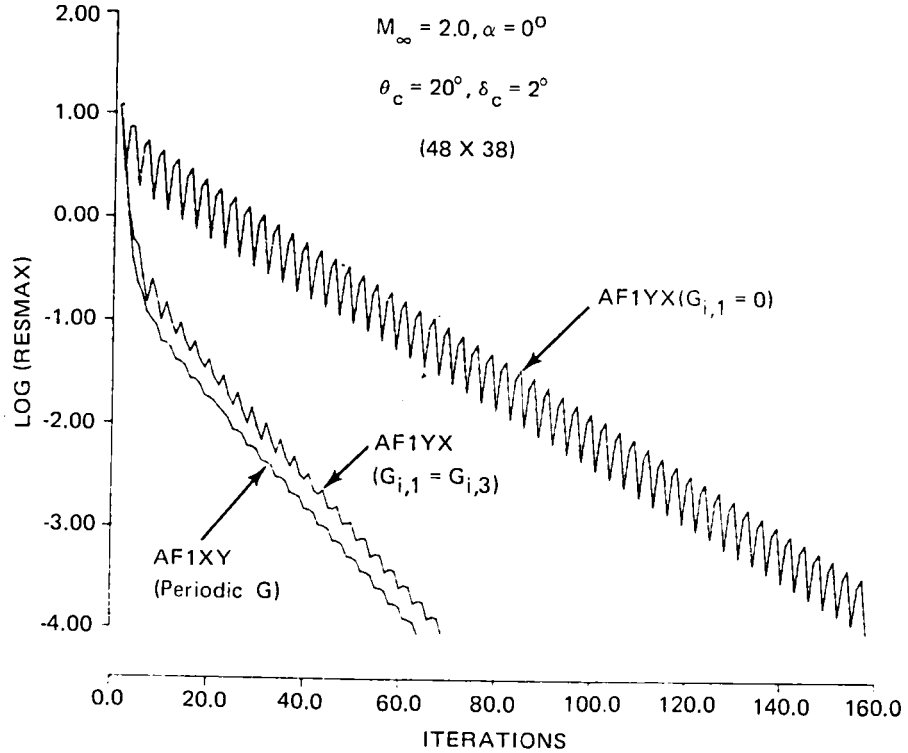


Fig. 3 Effect of Surface Boundary Condition and Factor Sequence on the AF1 Scheme (BSC) Convergence Rate

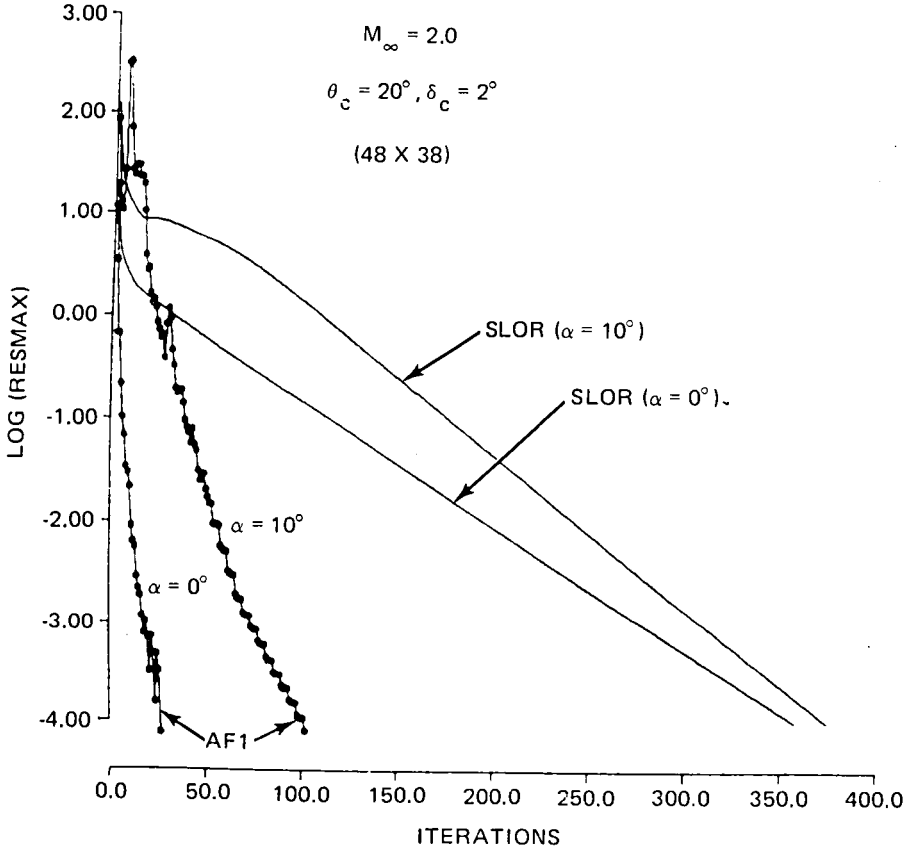


Fig. 4 Comparison of AF1 Convergence Rate with SLOR (BSC)

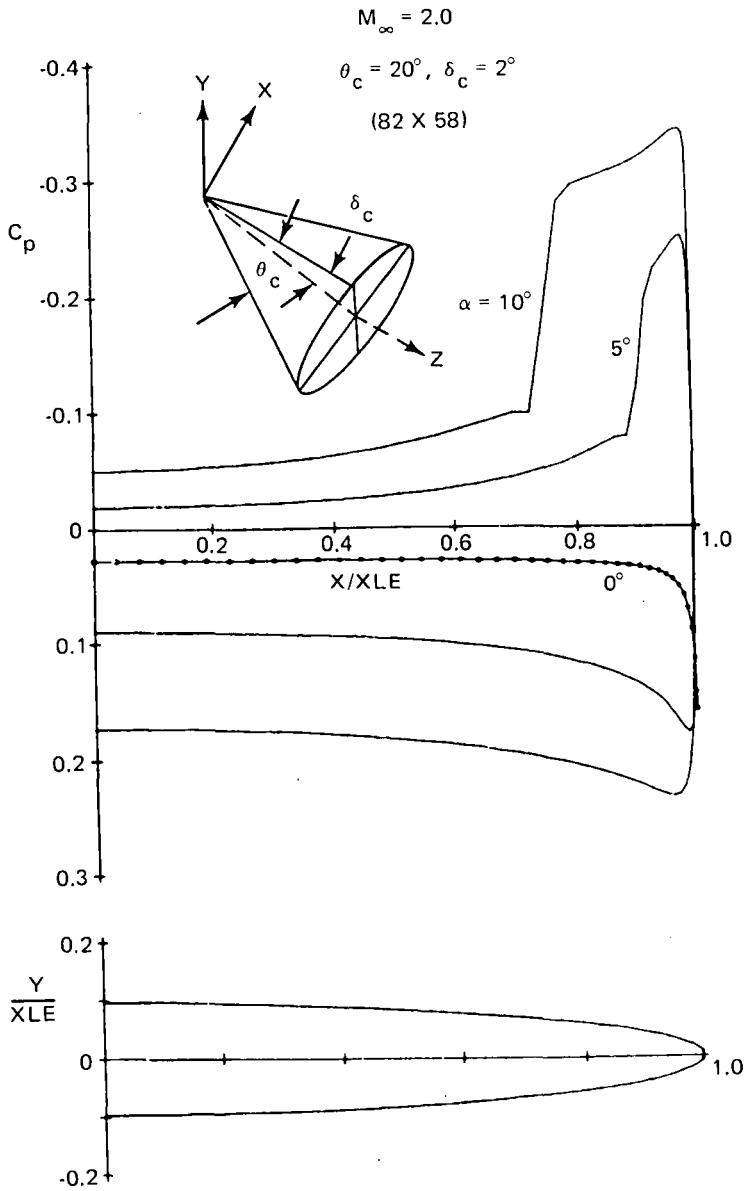


Fig. 5 Surface Pressure Distributions for Subsonic Leading Edge Elliptic Cone (BSC)

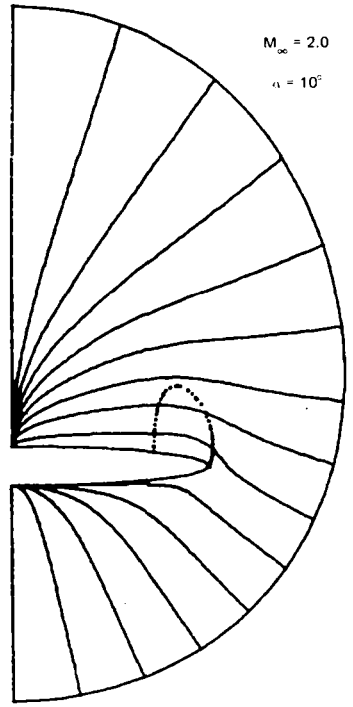


Fig. 6 Crossflow Streamline Pattern for Subsonic Elliptic Cone (BSF)

extent of the embedded supersonic crossflow region, which terminates at the crossflow shock.

Before the AF2 scheme was implemented, the effect of splitting the transformation derivatives between the two factors was studied. Figure 7 indicates the effect of using different forms for the term H_m in Eq (56). If the transformation derivatives are neglected in the factorization, the case could only be run when the α variation was increased significantly. Increasing the minimum value of α generally degrades the convergence rate. The best convergence was achieved when both the metric and the shearing transformation derivatives were included in the term H_m. The two curves in Fig. 7 with H_m other than unity were obtained with an alpha variation that diverged when $H_m = 1$. Hence, the AF2 scheme seems to be sensitive to the coordinate transformations, and the convergence rate can be affected significantly. The form of H_m is not considered to be optimum, and further analytical and numerical studies should be conducted to study its effect on the convergence rate. A nonoptimum H_m may also affect the minimum values of α that can be used.

Figure 8 shows a comparison of the convergence rates of the AF1 and AF2 schemes for the elliptic cone of Fig. 4 at $\alpha = 10°$. A comparable convergence rate was eventually achieved with the AF2 scheme. Further investigation of the AF2 scheme led to more problems (see Ref 11) and more sensitivity (i.e., not user friendly) to the choice of acceleration parameters. Hence, the AF2 scheme does not seem to have any advantages over the AF1 scheme and was not further considered for implementation in NCOREL.

4.2.8 Mesh Refinement

In Ref 14, it was observed that for transonic flow, mesh refinement enhanced convergence for SLOR but did not significantly accelerate convergence with the AF schemes. This is apparently not the situation with supersonic flows, where the bow shock formation is a critical factor. This is demonstrated in Fig. 9 for the subsonic elliptic cone at $\alpha = 5°$. The mesh refinement in the AF1 scheme yields a factor-of-three enhancement in convergence rate over the same computation on a single fine mesh. Mesh refinement also enhances the convergence rate of the SLOR scheme. The mesh refined AF1 indicates a factor of six to seven over the mesh refined SLOR result, which is similar to the reduction gained for a single mesh solution.

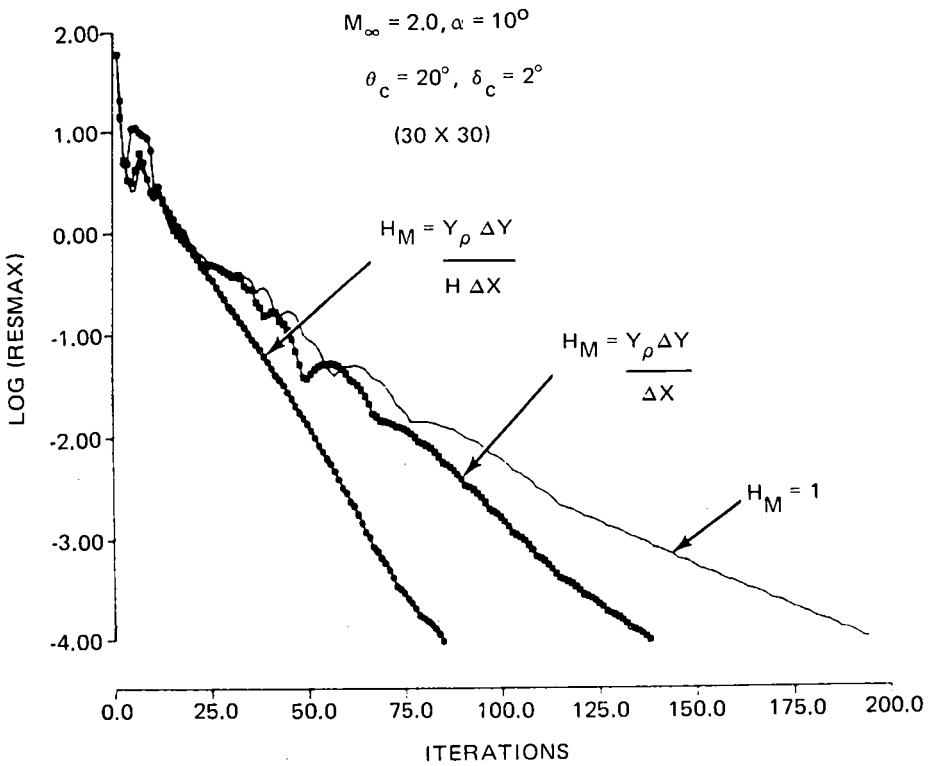


Fig. 7 The Effect of Transformation Derivatives on the AF2 Scheme (BSC)

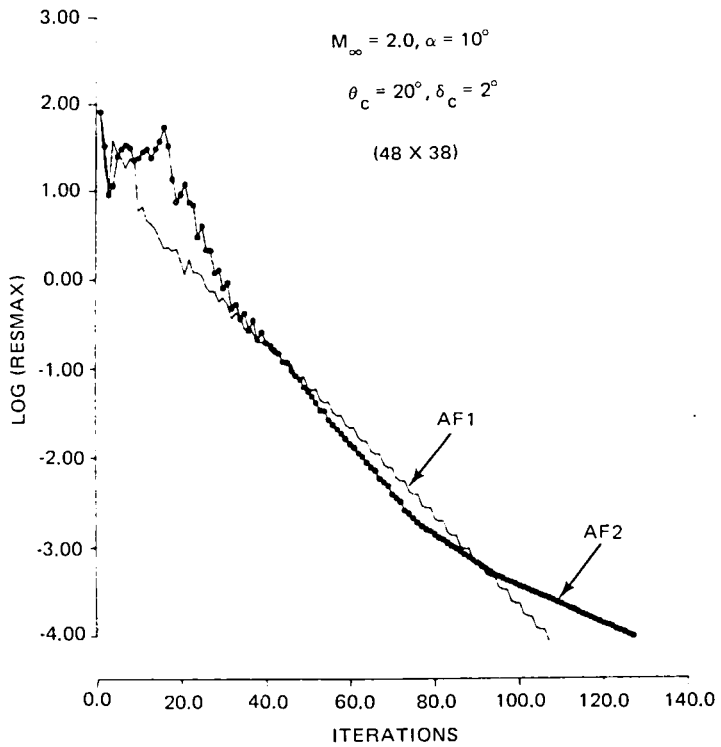


Fig. 8 Comparison of AF1 and AF2 for a Multi-shock Flow (BSC)

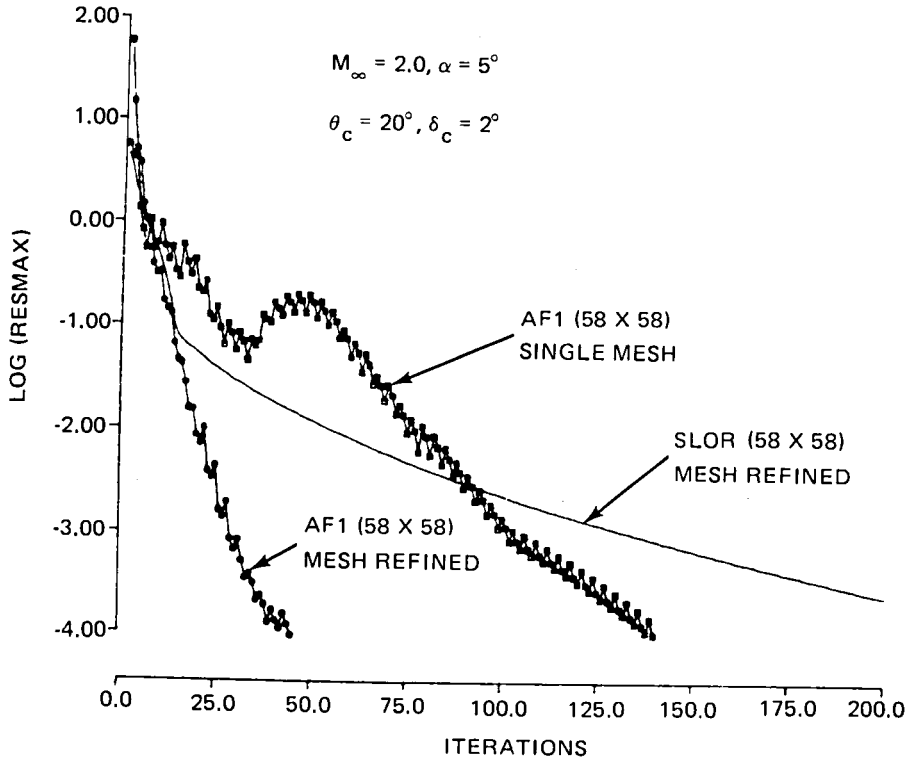


Fig. 9 Effect of Mesh Refinement on the AF1 Scheme

4.2.9 Bow Shock Fit Scheme

The AF1 scheme was also adapted to the conical bow shock fit method. In the BSF method, the capture of the bow shock is eliminated from the internal flow field. For any given outer boundary within or coincident with the true bow shock, the internal flow field will quickly converge. Hence, the convergence rate is largely governed by the iteration scheme for determining the bow shock position that satisfies the isentropic shock jump conditions. Typically, the updated bow shock boundary must be underrelaxed at each iteration so as not to disturb the convergence of the internal flow field computation. The bow shock boundary also has to be underrelaxed so as not to overshoot the correct bow shock position, which would cause the internal flow to capture a portion or all of the bow shock, thus leading to divergence. The major advantage in applying the AF schemes to the BSF method is that the correct bow shock information will propagate faster to and around the boundary and allow greater values of the shock relaxation parameter (ω_S), leading to an enhanced overall convergence rate. Figures 10 and 11 illustrate the AF1 convergence rate of the subsonic elliptic cone in comparison to SLOR at $\alpha = 5°$ and 10°, respectively. In Fig. 10 and 11, the maximum residual convergence is shown. In the SLOR curves, the shock relaxation parameter was 0.75 and 0.50, respectively, for $\alpha = 5°$ and $\alpha = 10°$. Higher values cause divergence of the flow field. It was observed that with the AF1 scheme the shock relaxation parameter did not require under-relaxation for these cases and could be taken to be equal to unity. Hence, in the AF1 scheme, both the bow shock and internal flow field converge faster than the SLOR. As a result, a similar enhancement in convergence rate is obtained for the BSF method as compared to the earlier results indicated for the BSC method.

4.2.10 Nonconical or Three-dimensional Flows (AF1Z Scheme)

The AF1 and AF2 schemes both worked well for quasi-two-dimensional conical flow yielding convergence rates two to ten times faster than SLOR. The AF2 scheme was somewhat more sensitive because of the split Y-derivative and the need to include the coordinate transformation derivatives in the factorization. Hence, for the present study, only the AF1 scheme was considered for the nonconical or three-dimensional flow problem. When the shearing transformation is applied to Eq (41), the principal terms of the 3-D full potential equation can be rewritten as

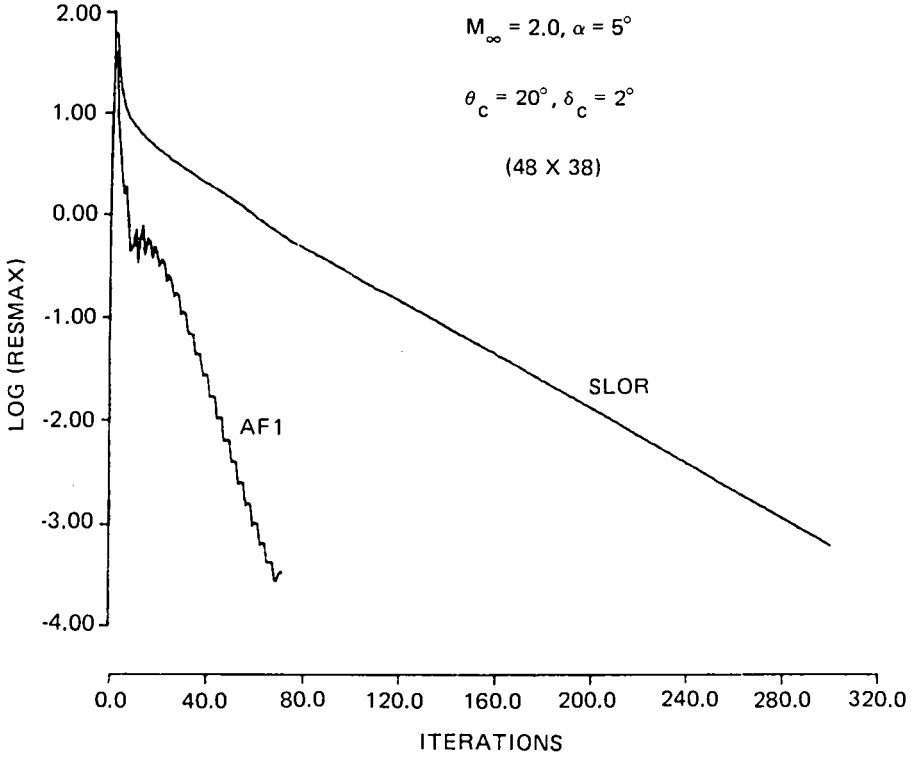


Fig. 10 Comparison of AF1 and SLOR Convergence Rates for the Bow Shock Fit Method at M_∞ = 2.0, α = 5°

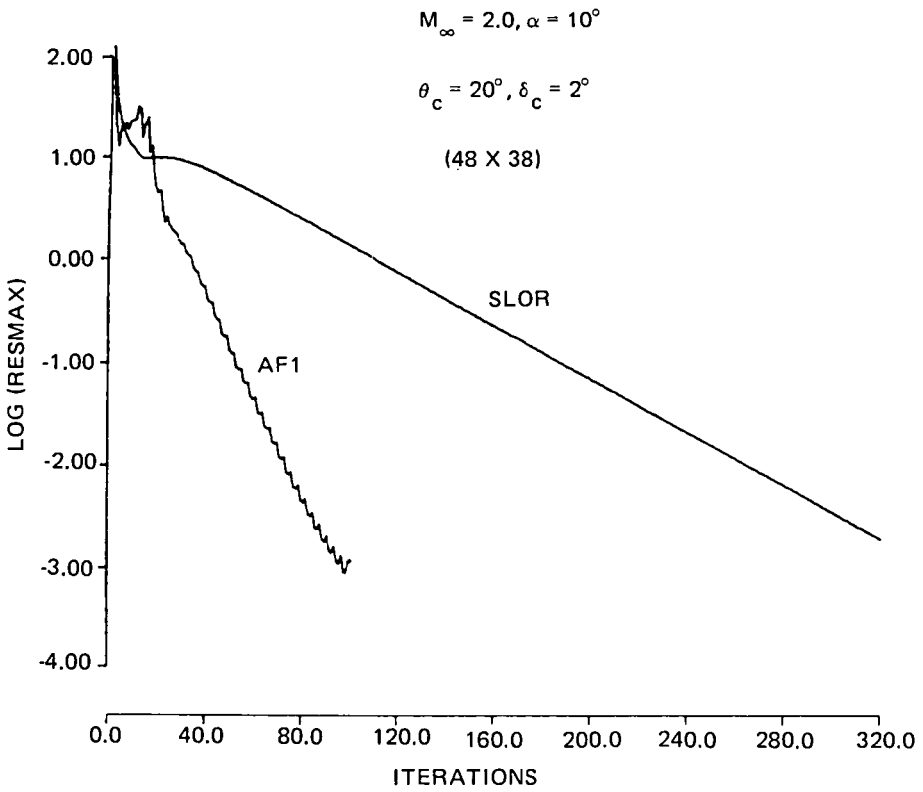


Fig. 11 Comparison of AF1 and SLOR Convergence Rates for the Bow Shock Fit Method at M_∞ = 2.0, α = 10°

$$L(\phi_{i,j}) = (A_1 + B_1)F_{XX} + (A_2 + B_2)F_{XY}$$
$$+ (A_3 + B_3)\, F_{YY} + B_8 F_{ZZ} + B_9 F_{XZ} \qquad (65)$$
$$+ B_{10} F_{YZ} + B_{11} F_Z + \cdots$$

where the B_i coefficients represent the additional nonconical R or 3-D terms.

The geometry is assumed to be conical at the apex or R = 0 of the configuration. Marching solutions are then obtained on spherical R or Z = constant surfaces. The terms F_Z and F_{ZZ} always have upwind differences, whereas the F_{XZ} and F_{YZ} terms are smoothly switched from central subsonic differences to supersonic upwind formulae. A first-order accurate F_{ZZ} difference requires information at two previous planes. Initially, the AF1 scheme was applied to the crossflow plane XY terms of Eq (65) with the Z derivatives treated as forcing terms evaluated with old values of the potential. This scheme turned out to be slower and resulted in divergence in many cases when compared to the optimum SLOR, which includes all the principal Z terms in the tridiagonal matrix. Hence, a factorization was sought that would maintain the XY crossflow convergence rate of the AF scheme and still retain the SLOR efficiency for the Z terms. The following AF1Z factorization was proposed for subsonic crossflow, $Q_c^2 < a^2$:

$$\left[\alpha - (A_1 + B_1)\frac{\overset{+}{\delta}_X \overset{+}{\delta}_X}{\Delta X^2} \quad \frac{-B_8}{\Delta Z^2} \quad \frac{-B_9 \delta_{Xc}}{2\Delta X \Delta Z} \quad \frac{-B_{11}}{\Delta Z}\right] G_{i,j}^{n+1} = \alpha\omega L(\phi_{i,j}^n) \qquad (66)$$

$$\left[\alpha - (A_3 + B_3)\frac{\overset{+}{\delta}_Y \overset{+}{\delta}_Y}{\Delta Y^2} - B_{10}\frac{\delta_{Yc}}{2\Delta Y \Delta Z}\right] \Delta_{i,j}^{n+1} = G_{i,j}^{n+1}$$

where δ_{Xc} and δ_{Yc} represent second-order, central, first-derivative operators, or

$$\delta_{Xc} = (\ )_{i+1,j} - (\ )_{i-1,j}$$
$$\delta_{Yc} = (\ )_{i,j+1} - (\ )_{i,j-1} \cdot \qquad (67)$$

The Z terms (e.g., F_{ZZ}) do not factor because of the hyperbolic nature of the

problem and hence are added explicity to the two factors.

The F_{ZZ} difference contains the unknown value of the potential at the current station and the two known values of the potential at the two previous stations. Except for the nonconical coefficients of the F_{XX} and F_{YY} terms, the Z terms are added explicitly to each factor. Including the Z terms in the factorization was also necessary to maintain diagonal dominance and convergence. In many cases, neglecting the Z terms in the AF scheme not only slowed convergence but actually resulted in divergence. For supersonic crossflow, the factorization was modified to include the upwind Z terms or, for $Q_c^2 > a^2$,

$$\left[\alpha - A_{1c}\frac{\vec{\delta}_X\overleftarrow{\delta}_X}{\Delta X^2} - I_s\frac{(A_{1u}+B_1)}{\Delta X^2}(K_1\vec{\delta}_X - K_2\overleftarrow{\delta}_X) - \frac{B_8}{\Delta Z^2} - \frac{B_9}{\Delta X\Delta Z} - \frac{B_{11}}{\Delta Z}\right] G^{n+1}_{i,j} = \alpha\omega L(\phi^n_{i,j})$$

$$\left[\alpha - A_{3c}\frac{\vec{\delta}_Y\overleftarrow{\delta}_Y}{\Delta Y^2} + (A_{3u}+B_3)\frac{\vec{\delta}_Y}{\Delta Y^2} - B_{10}\frac{\vec{\delta}_Y}{\Delta Y\Delta Z}\right] \Delta^{n+1}_{i,j} = G^{n+1}_{i,j} \,. \tag{68}$$

It was found that the off-diagonal terms of the F_{XZ} derivative could be included in the subsonic crossflow region, but not in the supersonic region leading to the following for the δ_{XZ} operator for $U > 0$, or

$$\vec{\delta}_{XZ} = (\,)^{n+1}_{i,j,k} - (\,)^{n}_{i-1,j,k} - (\,)_{i,j,k-1} + (\,)_{i-1,j,k-1}$$

$$\overleftarrow{\delta}_{XZ} = (\,)^{n+1}_{i+1,j,k} - (\,)^{n}_{i,j,k} - (\,)_{i+1,j,k-1} + (\,)_{i,j,k} \tag{69}$$

where the subscript K refers to the present R station and K-1 refers to the known values of the potential at the previous station. Hence, care must be taken to preserve the proper balance of new and old values of the potential.

Because of the split nature of the governing equation and R dependence, the AF1Z scheme was found to converge most reliably and optimally if the acceleration parameter α was scaled with R, or

$$\alpha_{min} \sim \alpha_{min} \left(\frac{1}{\Delta X} , \frac{1}{\Delta Y} \right) (1 + R)$$

$$\alpha_{max} \sim \alpha_{max} \left(\frac{1}{\Delta X^2} , \frac{1}{\Delta Y^2} \right) (1 + R) \ . \tag{70}$$

Hence, the acceleration parameter variation reduces to the conical value at R = 0 and increases linearly with R as the marching solutions are obtained. Since the nonconical coefficients are scaled with R, the acceleration parameters must also be scaled. This scaling will cause the retention of approximately the same convergence rate for a body of length unity versus an arbitrary dimensional length.

4.2.11 Three-dimensional Applications of the AF1Z Scheme

Although it is impossible to test the AF1Z scheme for all possible situations, test case computations were carried out on a variety of arbitrary wings and bodies. The following test cases were computed with the bow shock fit option.

Figure 12 illustrates the convergence history of the first case of a highly- swept delta wing at $M_\infty = 1.80$, $\alpha = 0°$. The wing geometry consists of a parabolic centerline thickness distribution with elliptic spanwise cross sections. As is the case for most three-dimensional wings, the geometry commences near the apex with a thick (e.g., 3:1 major-to-minor axis ratio) cross section which becomes thin and eventually approaches a flat plate cross section at the trailing edge. Figure 12 shows the number of iterations required per marching step to reduce the maximum residual to 10-3 (i.e., a minimum of four to five orders of magnitude) for the SLOR and AF1Z schemes. Step 0 refers to the conical or R = 0 solution required to start the computation. Fitting the bow shock as the outer boundary reduces the internal flow field computation to an elliptic problem if an embedded supersonic crossflow region does not form. For this case, the entire flow field was elliptic. Mesh refinement is used only at the conical station, and the marching proceeds on the fine mesh using the previous station solution as an initial guess. The iterations at step 0 reflect only the fine grid convergence. The AF1Z scheme shows large gains for the initial steps. At the initial steps, the geometry changes most rapidly. The AF1Z gains taper off as the cross section becomes quite thin. The SLOR scheme has some difficulty

computing the latter stations in comparison with the AF1Z scheme. Step 30 corresponds to the centerline trailing edge. The calculation is taken beyond the trailing edge in order to compute the entire wing. The wake is assumed to be a flat plate, which in this case is an exact assumption. The total iterations for the run are also shown in Fig. 12. The SLOR computation took 2010 fine grid iterations, and the AF1Z scheme required only 672 iterations, producing an overall factor-of-three reduction in iterations. The actual computation time, which includes geometry and mesh generation, was reduced by a factor of two.

The next case, shown in Fig. 13, is for a 67° sweep angle arrow wing at $M_\infty = 1.75$, $\alpha = 5°$. The geometry consists of a symmetrical, NACA 4% thick, four-digit airfoil imposed chordwise on the wing. The wake is approximated as a flat plate. In this calculation, the flow field very quickly becomes supercritical at R = 3 or STEP 4. The same trends apply for this case except that the SLOR scheme does not have any rise in iterations near the trailing edge of the wing. The interesting aspect of the AF1Z scheme is that the number of nonconical iterations required per step is relatively constant and independent of the geometry variation. Almost a factor-of-three reduction in iterations is again achieved by the AF1Z scheme corresponding to a factor of two in running time. Hence, the appearance of supercritical crossflow and a crossflow shock does not seem to deteriorate the AF1Z scheme significantly. In addition, aft of the trailing edge, the crossflow shock merges with the trailing edge shock.

Another case, shown in Fig. 14, is for a realistic, supersonic maneuver, demonstration wing designed with the aid of NCOREL and tested by NASA Langley. Details of this wing can be found in Ref 19. This wing has a variable sweep leading and trailing edge. The leading edge planform angle varies from 25° to 33°. The wing also has significant twist and camber. Figure 14 shows the convergence histories for the two schemes at $M_\infty = 1.62$ and $\alpha = 14°$. The wake is approximated by a flat plate extension of the wing spanwise camberline. For this wing, supercritical flow also appears at R = 3 or STEP 4. Similar trends apply for this wing except that the AF1Z scheme gains are reduced aft of the trailing edge of the wing. The overall reduction in iterations is similar, being almost a factor of three and corresponding to a factor-of-two reduction in running time.

All of the previous cases were run on a relatively fine grid (58 x 58). Figure 15 shows the iteration ratio of the SLOR to AF1Z scheme for several grids, corresponding to the DEMO wing computation of Fig. 14. The AF1Z scheme performs almost as well on the cruder meshes as on the finest mesh, retaining between a factor-of-two-to-three reduction in number of iterations. Figure 15 does indicate that the performance gain will increase on the finer grids.

The AF1Z scheme performs well for wing computations. The next set of cases will test the scheme for body computations. Figure 16 shows the convergence history for an axisymmetric circular arc cylinder body at M_∞ = 1.60, α = 10° on a 58 x 58 mesh. For this case, the crossflow is subsonic along the entire length of the body and, hence, shockless. The AF1Z scheme performs nicely by reducing the number of iterations by a factor of four.

Figures 17 to 19 show another set of computations for a more difficult body shape: an elliptic cross section (3:1 axis ratio) with a Haacke-Adams area distribution. Figure 17 shows the computations at M_∞ = 1.60, α = 5° on a 58 x 58 mesh. Better than a factor of four reduction in number of iterations is achieved for this condition. Figure 18 shows a higher Mach number case (M_∞ = 2, α = 5°) for the same body. For this condition, just under a factor-of-three reduction was achieved. Both schemes seem to have difficulty in the supercritical crossflow region that commences at around 20 in the marching steps. Figure 19 shows an even more extreme and difficult condition to compute, M_∞ = 2, α = 10°. For this condition, the crossflow is supersonic at the conical start, with the crossflow shock becoming increasingly stronger toward the aft end of the body. The AF1Z scheme performs well initially, but both schemes begin to have difficulty towards the aft end of the body. Not quite a factor-of-two reduction in iterations is achieved.

In summary, the AF1Z scheme seems to be reasonably reliable and can be made to run faster than the SLOR scheme over a wide range of configurations. Even for difficult bow shock fit computations, the AF1Z will run faster, although the gains achieved are not as great as at the lower Mach number conditions. The AF1Z scheme was found to be more sensitive and difficult to achieve convergence for supersonic blunt leading edge wings. It must be mentioned that the effect of the AF parameters was not intensively explored for the previous computations and, hence, a more thorough study of these parameters might yield greater reductions.

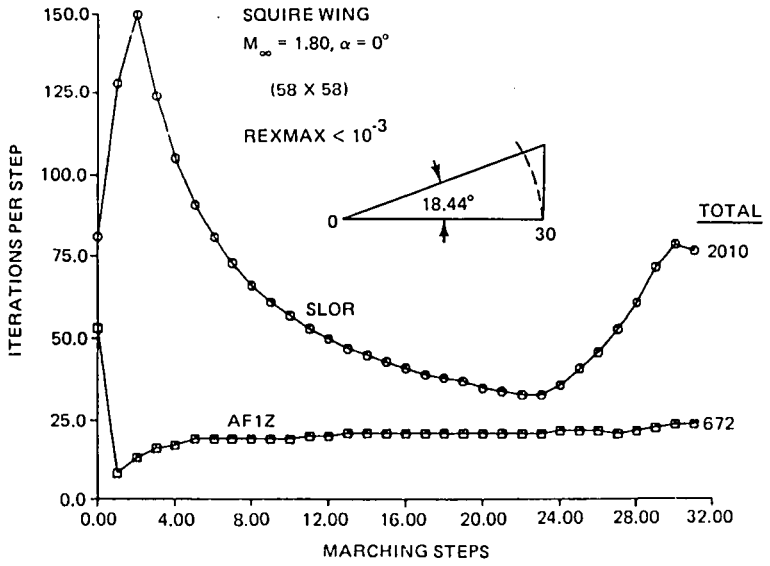


Fig. 12 Squire Wing Computation, AF1Z vs SLOR (BSF)

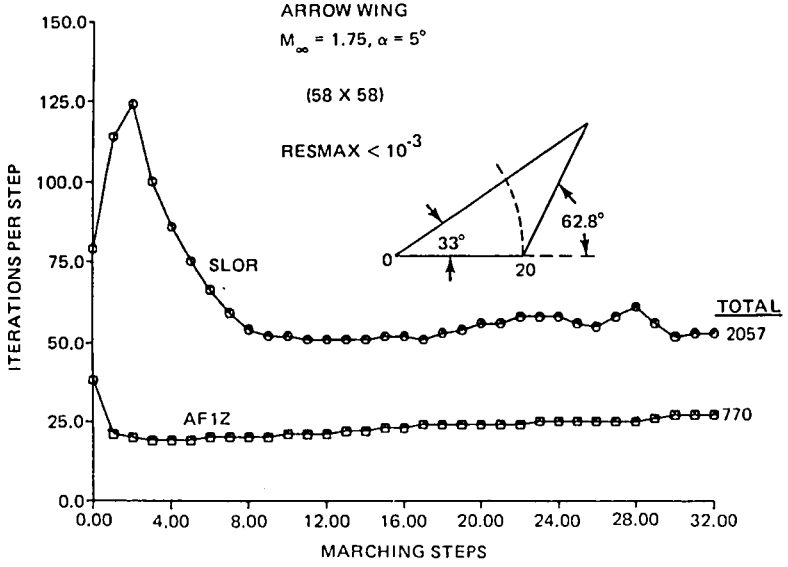


Fig. 13 Arrow Wing Computation, AF1Z vs SLOR (BSF)

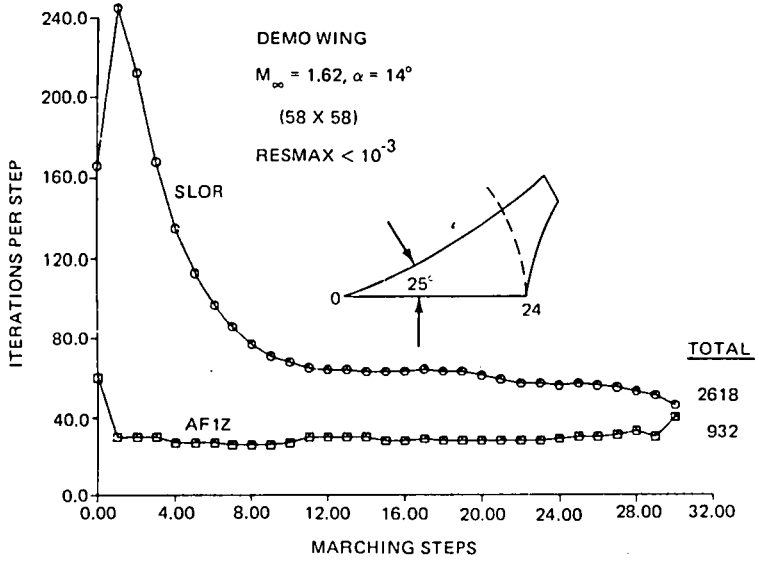


Fig. 14 Demonstration Wing Computation, AF1Z vs SLOR (BSF)

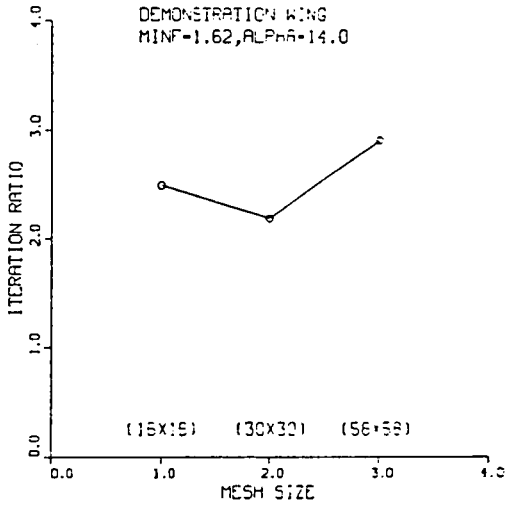


Fig. 15 SLOR vs AF1Z Convergence Ratio for Varying Mesh Sizes

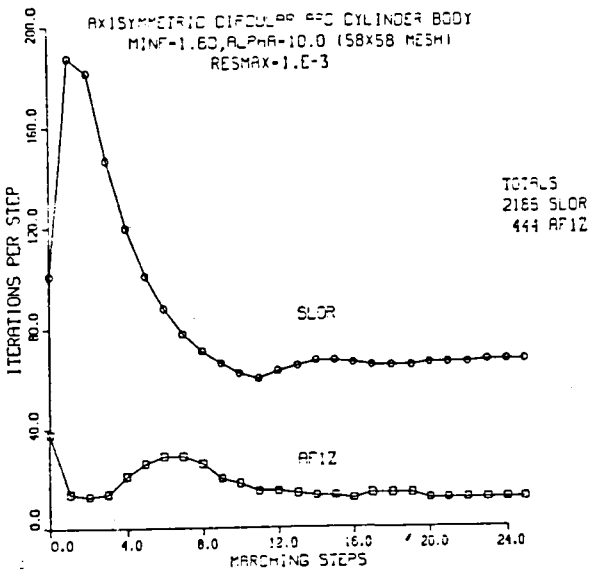


Fig. 16 Axisymmetric Circular Arc-cylinder Body Computation at $M_\infty = 1.60$, $\alpha = 1^\circ$, AF1Z vs SLOR

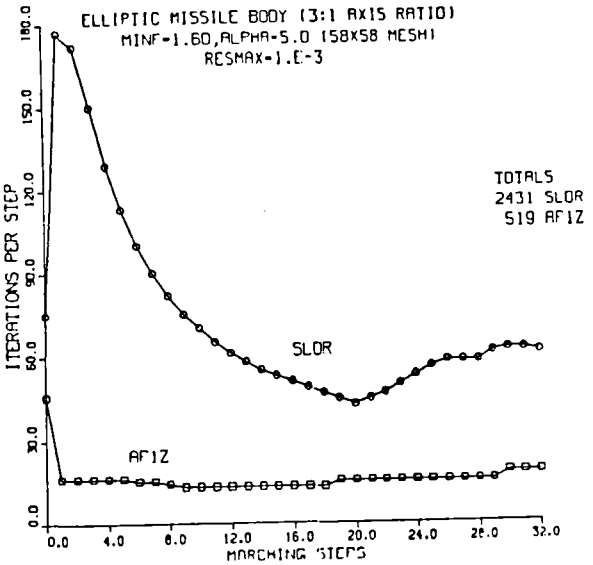


Fig. 17 Elliptic Missile Body Computation at $M_\infty = 1.60$, $\alpha = 5^\circ$, AF1Z vs SLOR

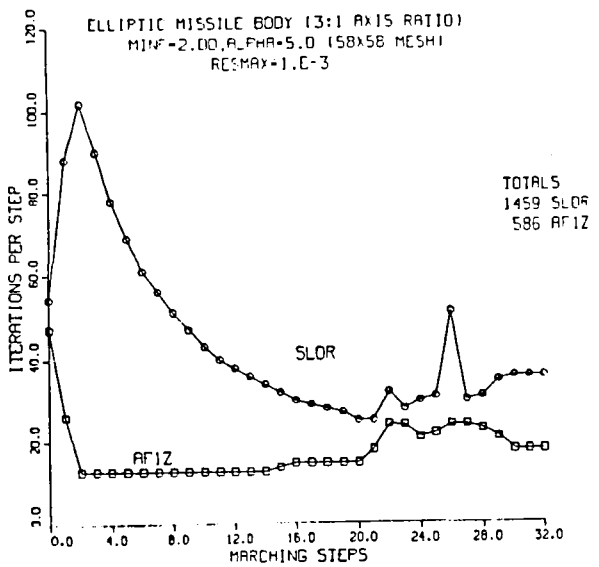


Fig. 18 Elliptic Missile Body Computation at $M_\infty = 2.0$, $\alpha = 5^\circ$, AF1Z vs SLOR

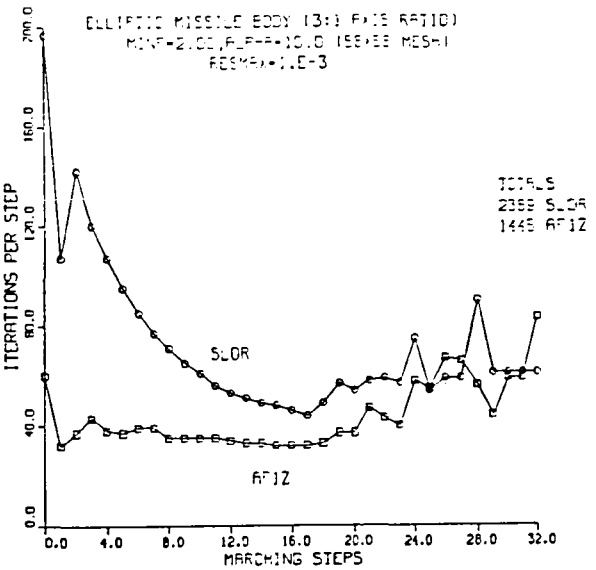


Fig. 19 Elliptic Missile Body Computation at $M_\infty = 2.0$, $\alpha = 10^\circ$, AF1Z vs SLOR

4.2.12 Parameter Selection

Although it is impossible to test the AF1Z scheme for all possible situations, test case computations were carried out on a variety of arbitrary wings and bodies. The α variation that was found to work best in a variety of cases was

$$\alpha_{max} = \frac{2}{\Delta Y^2}(1 + R) = \frac{A_{max}}{\Delta Y^2}(1 + R)$$
$$\alpha_{min} = \frac{1}{\Delta X}(1 + R) = \frac{A_{min}}{\Delta x}(1 + R) \qquad (71)$$

where the cycles between α_{max} and α_{min} are two or three for crude-to-medium meshes (16x16 → 48x48) and four for finer meshes (58x58 and above). The exception to this rule is for more difficult bow shock fit cases at high Mach number or incidence. For these cases, the alpha range of Eq (71) should be raised. For example, in Eq (71), $A_{min} = 1$, $A_{max} = 2$; these values might be raised to $A_{min} = 2$, $A_{max} = 4$ for the crude mesh to get the bow shock fit computation started. In addition, the number of cycles ITMAX might be raised to 6 on the crude mesh. After the crude mesh is successfully computed, the default values can probably be used on the finer and nonconical meshes.

The AF1Z scheme can be run with a relaxation factor (ω) for the residual up to 1.7. Since the AF1Z scheme converges faster than the SLOR scheme, the bow shock relaxation (ω_S) factors can also be increased both at the conical start and at the nonconical stations. Typically, $\omega_S = 3.0$ was used at the nonconical stations. The temporal artificial viscosity or damping terms controlled by the parameter EST are also included in the AF1Z scheme. For most cases, with the exception of difficult cases with strong embedded shocks, the AF1Z sheme can be run with the EST parameter set to zero. For cases that do require damping, the value of EST should be set at very small values (e.g., $-0.001 \geq EST \geq -0.10$). For most cases -0.01 or -0.001 is sufficient. Nonzero values of EST in the AF1Z scheme degrade the convergence considerably and should be avoided.

This Page Intentionally Left Blank

5. WAKE FLOWS

5.1 GENERAL CONDITIONS

In general, wake flows are characterized by a discontinuity or slip in velocity with a continuity in pressure. In potential flows, a jump in potential is prescribed across the wake which accounts for the circulation or lift about the wing or airfoil. In two-dimensional potential flows the velocity is continuous, but the potential has a constant jump across the wake streamline. Matching pressures at the trailing edge in the wake cut also imposes a Kutta condition and causes the wake streamline to leave the trailing edge in a direction along the trailing edge bisector. For the Euler equations, the situation can be different because of entropy losses across shock waves. If shock waves of different strengths exist on the upper and lower surfaces of an airfoil, a slip line with a jump in velocity will arise in the wake, due to the differences in total pressure and entropy along the upper and lower surface streamlines.

On the other hand, in three-dimensional flows, contact discontinuities arise in the wake for both Euler and potential flow formulations. In addition, in three dimensions, depending on the trailing edge geometry and flow conditions (e.g., cusped or finite angle), the trailing edge streamline will leave tangentially to one of the surfaces or at the mean angle of the trailing edge (Ref 20). In potential flows, the slip in velocity is due to a discontinuity in the direction of the total velocity vector at the wake surface. The magnitude of the velocity vector must match on the wake to impose the Kutta condition and continuity of pressure along the wake surface. The discontinuous direction of the wake surface total velocity vector causes at least two of the three components of velocity to be discontinuous or to have a slip. For the most part, in three-dimensional transonic flows, the wake is treated in a similar fashion to its two-dimensional counterpart. The jump in potential for a particular airfoil section is assumed constant to downstream infinity. This approximation matches the longitudinal or axial velocity. A continuous velocity is also imposed through the wake surface. The spanwise velocity on the wake surface is not matched and is just proportional to the variation in circulation or lift and, hence, to the variation in the spanwise potential jump for each

airfoil section. Hence, the magnitude of the total velocity vector and resultant pressure is only approximately matched on the wake surface. This is a good approximation for wings that are not highly swept or tapered. A similar wake treatment is used by Shanker, et al (Ref 21) for three-dimensional supersonic flows.

Up to the present time, no attempt had been made to model the wake in NCOREL, and a flat plate treatment had been utilized. The successful application of NCOREL to three-dimensional wings and bodies has suggested that a wake treatment might further extend its applicability to wing-body and more complex configurations.

In the present formulation (see Ref 22), the wake is also modeled approximately as a planar cut in three dimensions. However, the Kutta condition at the trailing edge and the continuity of pressure on the wake surface are imposed exactly. Therefore, as distinct from other treatments, the behavior for the jump in potential is not prescribed a priori, but rather computed at every point on the wake surface.

5.2 WAKE MODEL BOUNDARY CONDITIONS

The wake is modeled as a planar surface cut of infinitesimal thickness. The wing cross section and wake cut (see Fig. 20) are then mapped to a near circle (ρ, θ, R) using conformal mappings and further sheared to a computational domain (X, Y, Z).

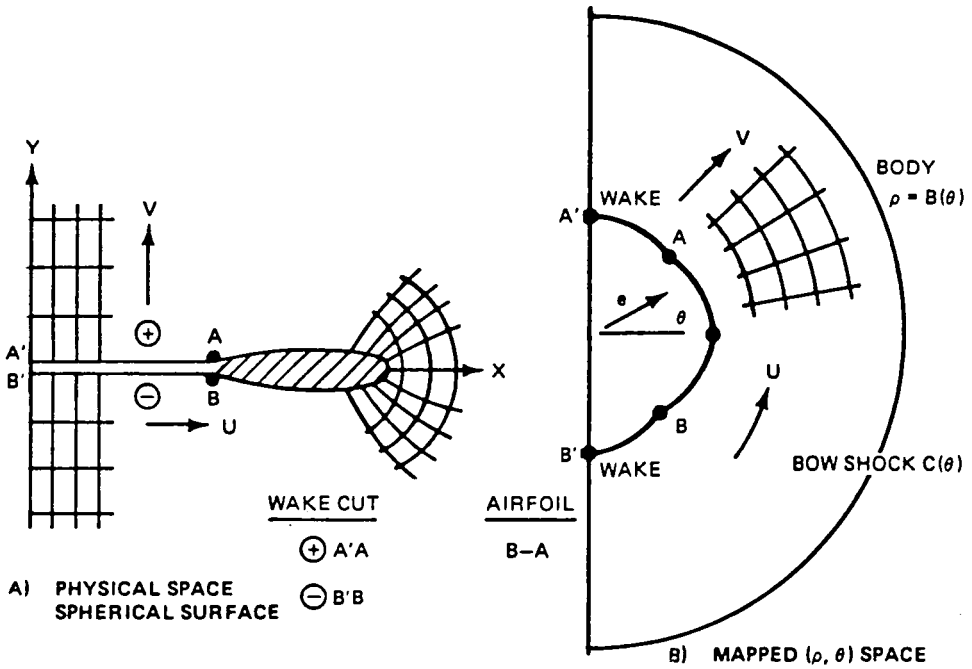
Flow tangency is obtained by using a dummy row of grid points around the body whose values are obtained by imposing the vanishing of the normal velocity through the F_Y derivative. For positive angle of attack, the flow on the lower surface of the wake cut is computed with the boundary condition that the normal velocity component through the wake cut is continuous.

For V_N to be continuous, the condition that

$$V_{Nw}^{-} = V_{Nw}^{+} \tag{72}$$

must be imposed on the wake cut.

The wake condition (72) is imposed by computing the potential derivatives in a one-sided fashion on the upper wake cut, and is then used as the boundary condition on the lower wake cut. The computational method proceeds by



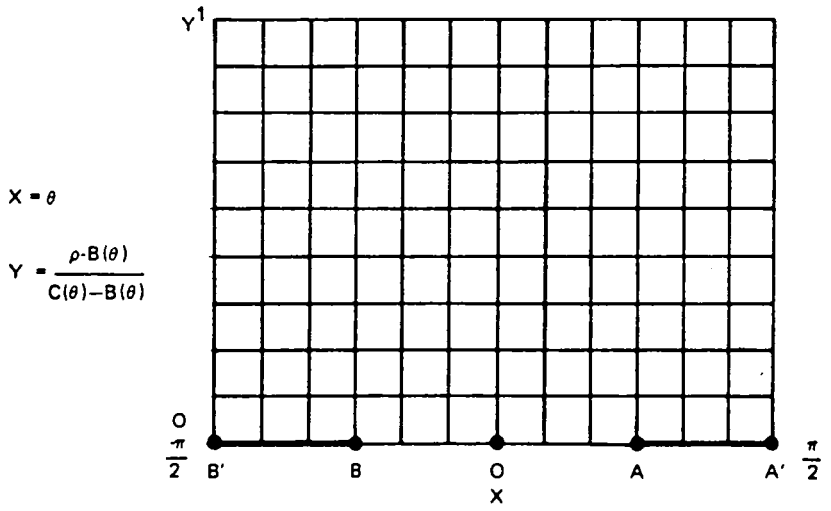


Fig. 20 Conical Wake Problem

computing on lines X = constant commencing at the lower symmetry plane and sweeping around to the upper symmetry plane. On the lower wake cut, the governing full potential equation is satisfied subject to condition (72). On the wing surface, flow tangency is satisfied. Both conditions use a Neumann boundary condition given by the dummy row of potentials. On the upper wake cut, the full potential equation is not satisfied. Instead, a jump in potential is assumed to exist at every point on the upper wake cut as

$$F_W^+ = F_W^- + \Delta F(\theta_W) \quad . \tag{73}$$

The upper wake cut X = constant lines are then solved as a Dirichlet problem once a jump in potential has been imposed.

In summary, the X = constant lines emanating from the lower wake cut are solved by using a Neumann-type boundary condition obtained from condition (72), where V_{Nw}^+ is derived from a one-sided difference in the upper wake plane. Hence, communication across the wake exists without explicitly differencing across the wake cut, thus eliminating the necessity for interpolation. An equation for the jump in potential is now needed that matches the pressures all along the wake cut. For potential flows, it is a sufficient condition to match the total speed along the wake cut for the pressures to match. Equality of total speed can be expressed by the equation

$$\bar{U}(U_{W+}-U_{W-}) + \bar{V}(V_{W+}-V_{W-}) + \bar{W}(W_{W+}-W_{W-}) = 0 \tag{74}$$

where

$$\bar{U} = U_{W+} + U_{W-}$$

$$\bar{V} = V_{W+} + V_{W-}$$

$$\bar{W} = W_{W+} + W_{W-}$$

In practice, Eq (74) can be implemented recursively by solving for ΔF in terms of the U, V and average velocities on the wake cut. The new value of the jump in potential is computed based on old values of the velocity and is

underrelaxed until the full potential equation, flow tangency, and conditions (72) and (74) are all satisfied.

5.3 GENERAL THREE-DIMENSIONAL WAKES

The jump in potential at each point on the wake surface emanating from a three-dimensional wing has as its origin an upstream trailing edge point. Hence, for a lifting wing, in the crossflow plane, a variable jump in potential exists along the wake surface proportional to the circulation about an upstream streamline section.

A good initial guess for the jump in potential in the crossflow plane can be generated from the jump in potential across the wake and airfoil section of the previous plane. Some underrelaxation must be used in updating the potential jump, but the overall rate of convergence is not affected significantly, except when a strong shock interaction occurs at the trailing edge.

In general, both subsonic and supersonic trailing edges can occur in the 3-D supersonic wake flow problem. Subsonic trailing edges have local Mach numbers in a plane normal to the trailing edge that are less than unity. For this situation, the wake streamline leaves the trailing edge smoothly. When the local normal trailing edge Mach number is supersonic, the airfoil surface streamline will either expand or pass through a shock in satisfying the pressure condition. Depending upon the value of the local normal Mach number at the upper and lower surface trailing edge points, several flow situations can exist. For supersonic trailing edges, these are

- Shock - shock
- Shock - expansion.

A shock or expansion can also occur on one surface in combination with smooth subsonic behavior on the other surface. Figure 21 sketches some of the basic flow situations and their character in the spherical crossflow plane. The trailing edge shock is really a three-dimensional surface which takes a shape similar to that shown in Fig. 21a in the crossflow plane at zero incidence. A crossflow shock can also coexist on the surface of the cross section that will interact with the trailing edge shock(s) and/or expansion. One such complicated interaction is sketched in Fig. 21d from observations of computed crossflow plane isobar patterns. The crossflow shock and trailing edge shock intersect, forming two resultant shocks. The crossflow shock after

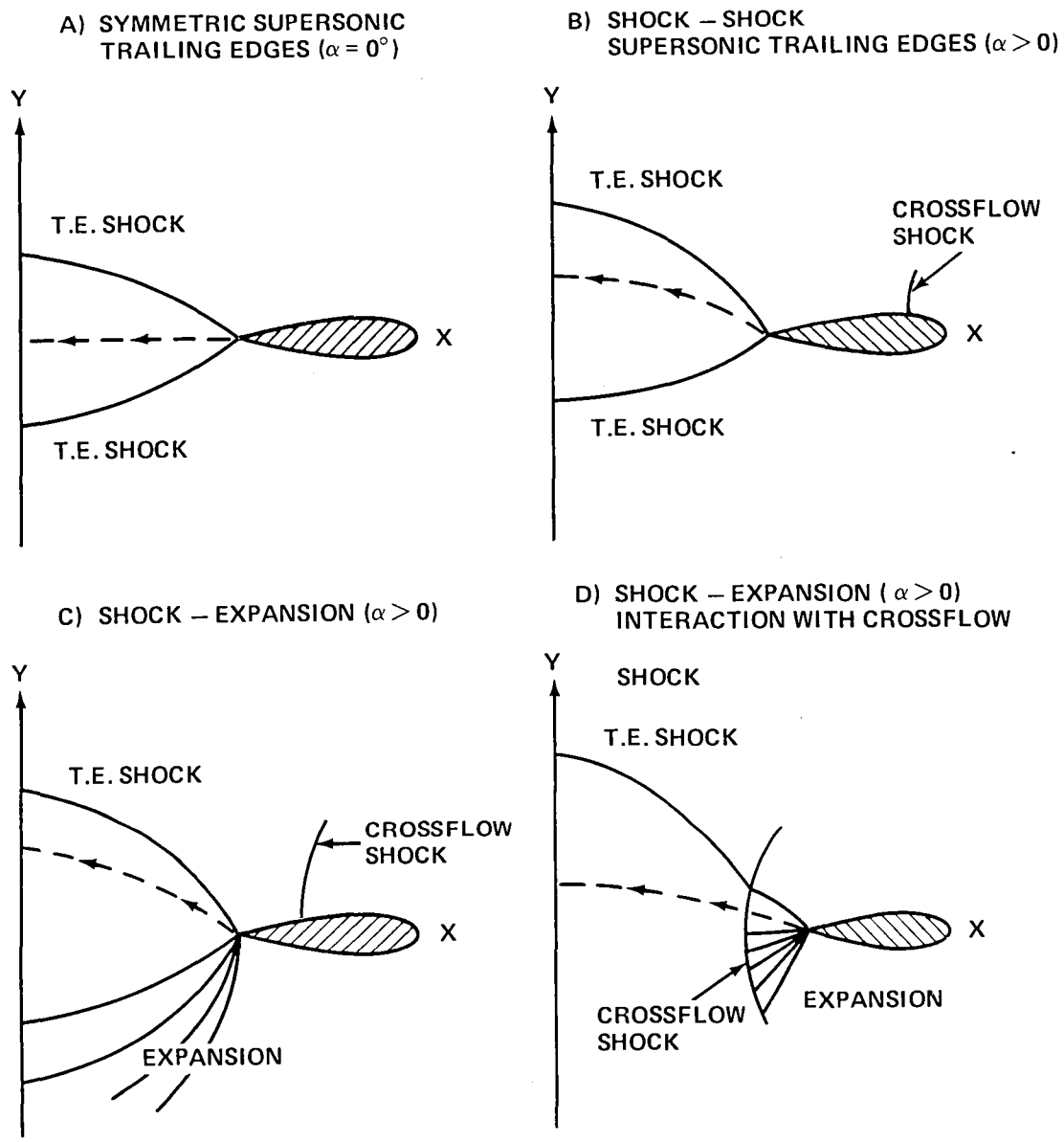


Fig. 21 Sketch of Trailing Edge Shock Interactions (3-D) in the Crossflow Plane

intersection with the trailing edge shock exists in the flow on the wake cut (i.e., passes through the trailing edge shock) and is attenuated below the wake cut by the lower surface expansion.

5.4 SYMMETRIC CROSS-SECTIONAL GEOMETRIES

Symmetric geometries were treated initially. The meshes generated for these geometries by NCOREL yield corresponding grid points in the upper and lower surfaces of the wake cut. Hence, no interpolation is required at corresponding upper and lower wake points for the potential or the speed. Figure 22 shows a selected sample of the crossflow plane surface pressure solutions for a symmetric arrow wing with 70° leading edge and 45° trailing edge sweep angles at $M_\infty = 1.75$, $\alpha = 5°$ commencing with the spherical surface that cuts through the centerline trailing edge. Some of the computed wake pressure distributions are compared with the pressure distributions obtained by assuming a flat impermeable plate for the wake. A crossflow shock exists on the upper surface of the airfoil. The upper surface pressures indicate a trailing edge shock. The lower surface pressures indicate a smooth flow behavior or a very slight expansion. The flat plate solution indicates shocks on both surfaces of the airfoil. Further down the wing, the trailing edge shock becomes somewhat stronger, and the crossflow shock approaches the trailing edge of the wing. Finally, both shocks merge just as the crossflow shock intersects the trailing edge and form a single strong shock at the trailing edge. At the last station, the crossflow shock has passed through the trailing edge shock and exists on the wake cut. The trailing edge solution indicates a shock-expansion behavior at this point. The trailing edge shock, crossflow shock, and expansion interaction are similar to that sketched in Fig. 21d.

Figure 23 indicates a similar behavior at $M_\infty = 1.75$, $\alpha = 10°$ for the same arrow wing. The two shocks merge, causing a very strong shock at the trailing edge, and eventually the crossflow shock passes through the trailing edge shock and sits on the wake cut with a shock expansion at the trailing edge. Naturally, this interaction has the built-in approximation of modeling the wake cut as a planar surface. The effect of fitting the wake exactly for these complicated interactions is not known at this point.

The important aspect of modeling the wake properly is to be able to predict the effect of the wake of a lifting surface on a fuselage or other

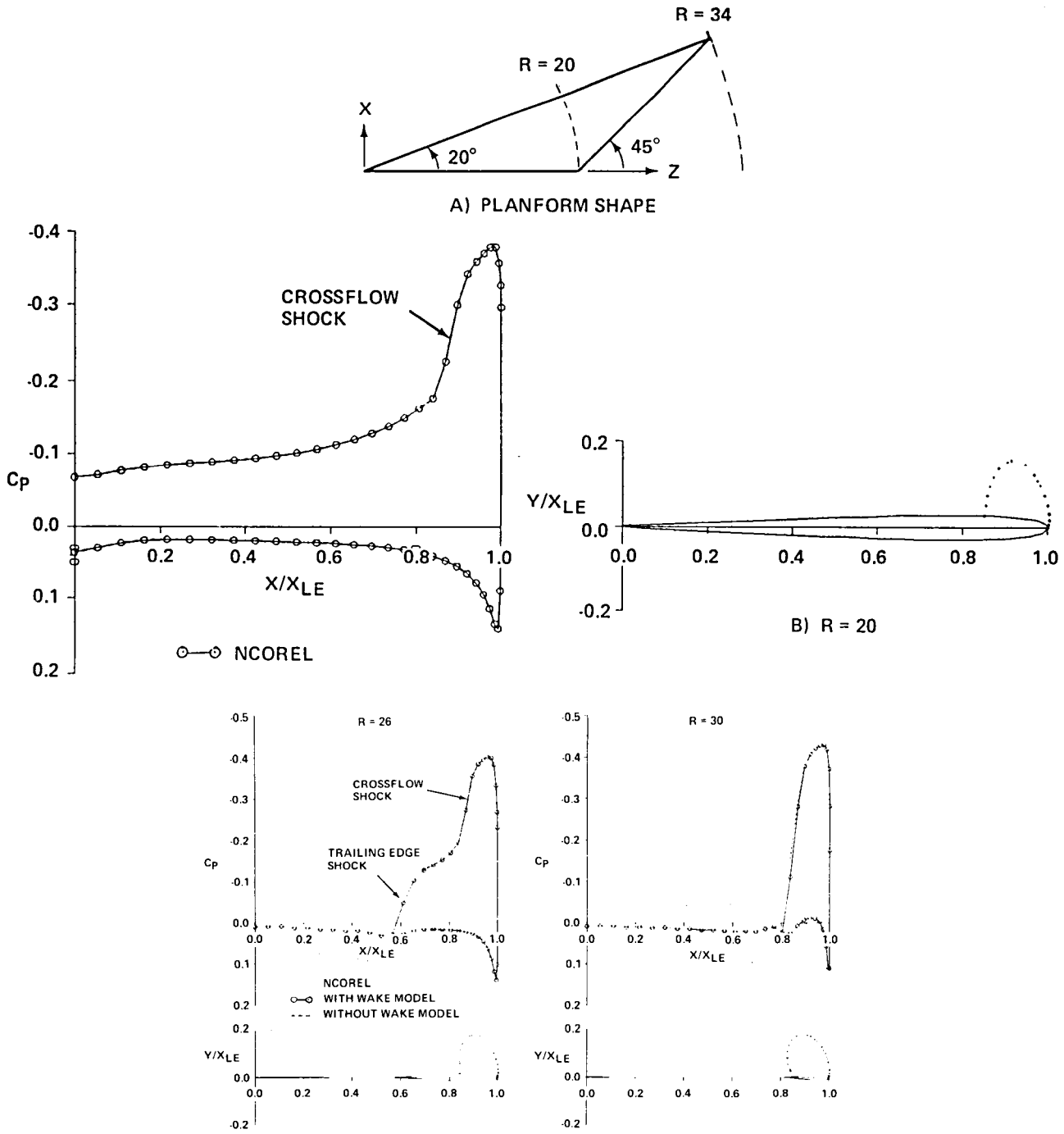


Fig. 22 Sample Arrow Wing Computation at $M_\infty = 1.75$, $\alpha = 5^\circ$ (Sheet 1 of 2)

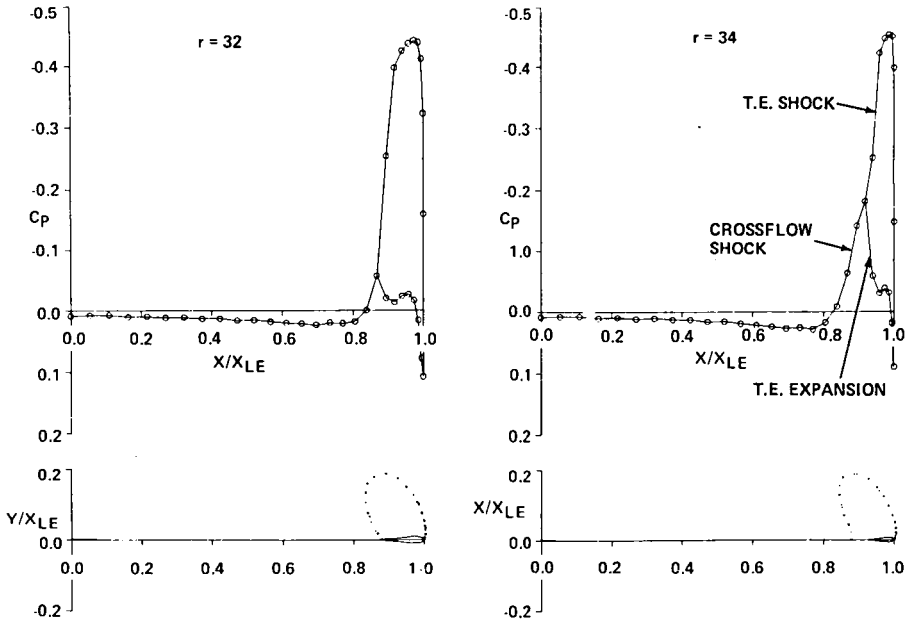


Fig. 22 Sample Arrow Wing Computation at $M_\infty = 1.75$, $\alpha = 5^\circ$ (Sheet 2 of 2)

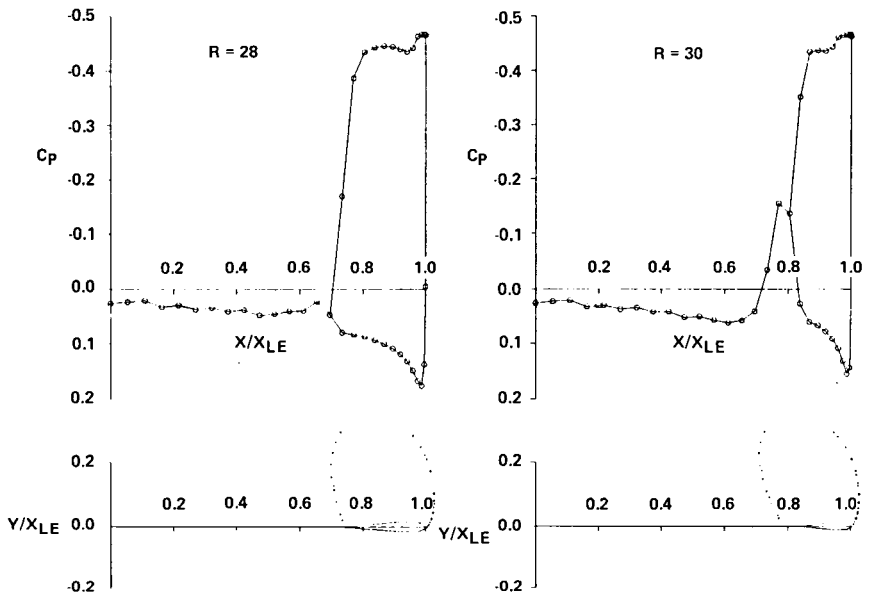
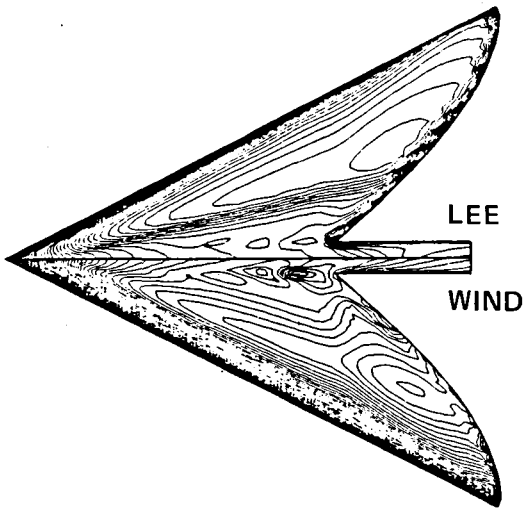
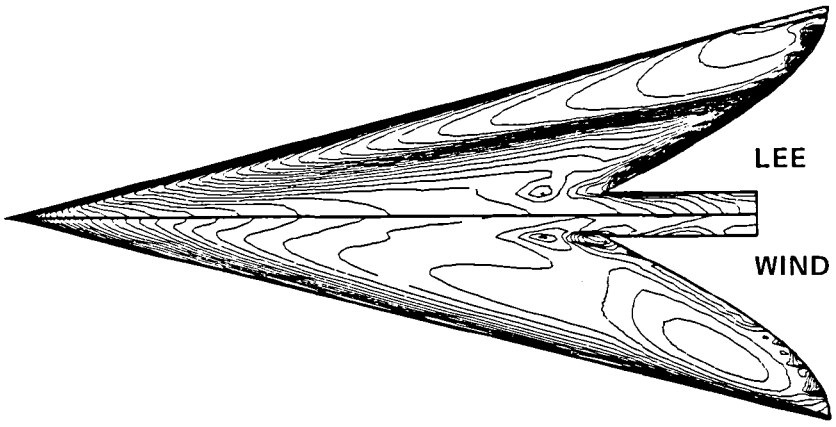


Fig. 23 Sample Arrow Wing Computation at $M_\infty = 1.75$, $\alpha = 10^\circ$

lifting surfaces. To test the wake model, a series of computations were carried out on a set of arrow wings built and tested by NASA Langley (see Ref 23). Figure 24 shows a sample of the isobar patterns computed, along with the planform shape and centerbody for two of the four models tested at Mach numbers of 2.36, 2.96, and 4.63. Model 1 has a leading edge sweep of 63.4°, and Model 2 was more highly swept with a leading edge sweep of 76°. The models were instrumented for pressure, and detailed pressure measurements are available on both the wing and centerbody at flow spanwise sections. One of the spanwise stations was downstream of the centerline trailing edge and serves as a test for the accuracy of the present wake model. One of these wings was computed previously using NCOREL without a wake model, with the result that the body pressures could not be predicted (see Ref 9). The planform isobar patterns clearly show the crossflow shock on the leeward surface. The crossflow shock intersects the trailing edge wherein the complicated interaction of Fig. 21d takes place. Figure 25 shows a comparison of measured and computed pressures for Model 1 at two different conditions, $M_\infty = 2.36$, $\alpha = 6°$ and $M_\infty = 2.96$, $\alpha = 10°$. Excellent agreement is achieved at the lower Mach number on both the wing and centerbody. At the higher Mach number and incidence, slightly lower pressures are computed in the vicinity of the upper surface trailing edge. Excellent agreement is achieved for the lower wing surface. The body pressures are in good agreement except near the upper surface symmetry plane. Unfortunately, the resolution of the body is quite poor on the mesh that is currently used because the relative size of the body with respect to the wing is quite small for Model 1. Comparisons for the highly swept Model 2 at $\alpha = 3°$ and $6°$ are shown in Fig. 26 and 27 for $M_\infty = 2.36$ and 2.96, respectively. Better resolution of the body is obtained for this model because of its relatively large size in comparison to the wing. Good to excellent agreement is achieved for body pressures. Good agreement is achieved for the lower surface of the wing. Poor correlation is achieved on the upper surface of the wing at the higher incidence. The higher pressure supercritical plateau shown by the measured data at $\alpha = 6°$ is usually indicative of leading edge flow separation and vortex formation and, hence, correlation with computed pressures would not be expected.



MODEL 1 AT M_∞ = 2.96, α = 6°



MODEL 2 AT M_∞ = 2.96, α = 6°

Fig. 24 Isobar Planform Pattern for NASA Arrow Wings

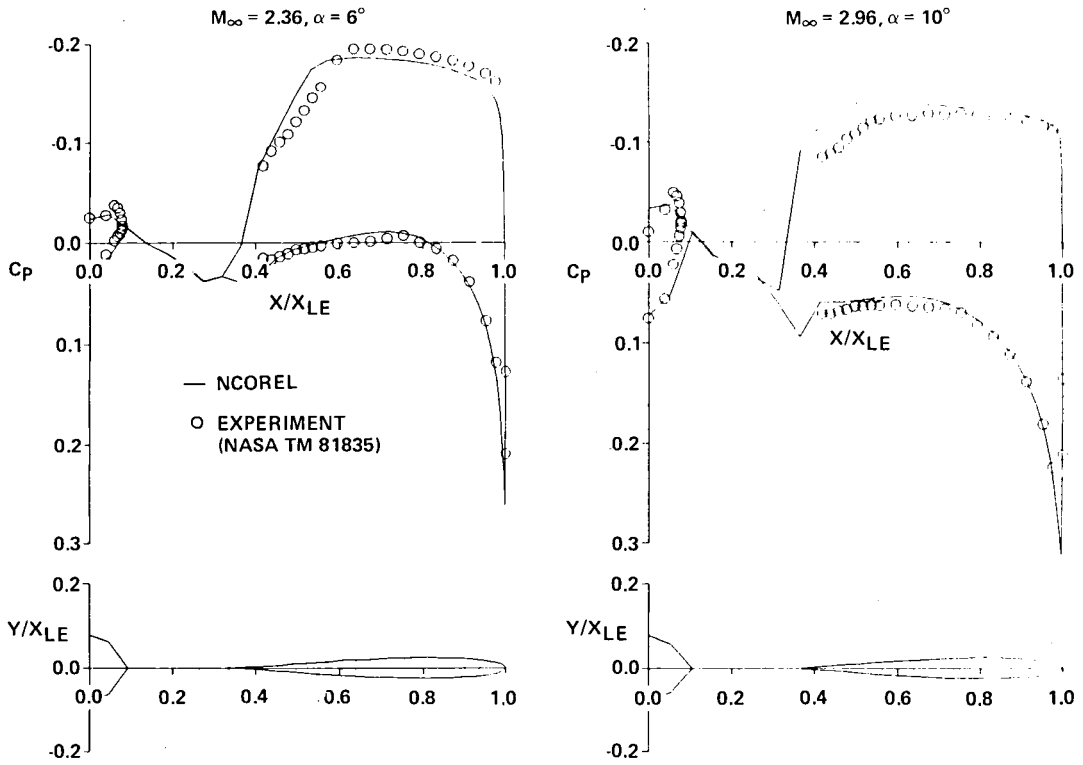


Fig. 25 Model 1 Comparison with Experiment at Spanwise Wake Station

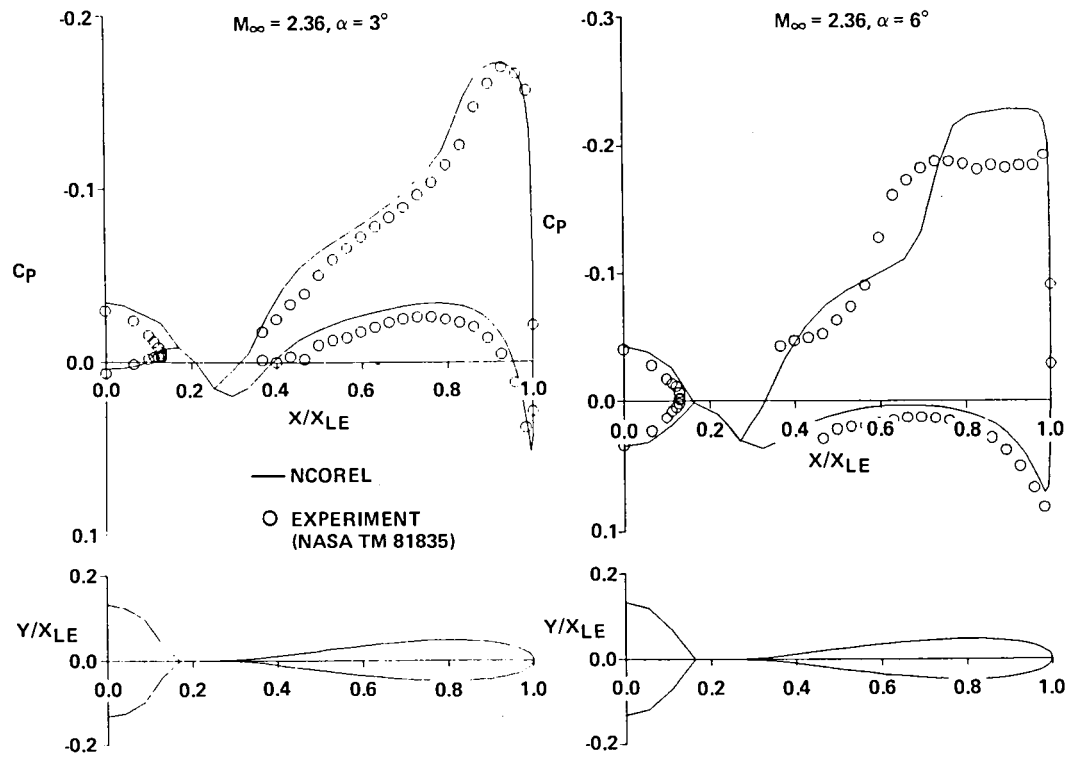


Fig. 26 Model 2 Comparison with Experiment at Spanwise Wake Station

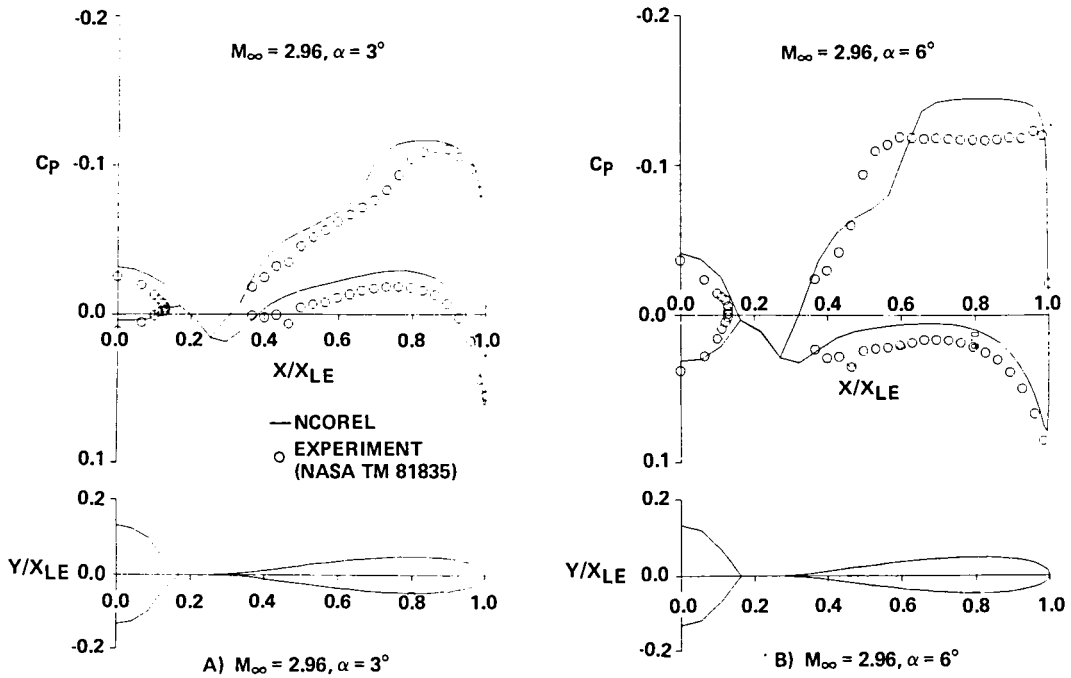


Fig. 27 Model 2 Comparison with Experiment at Spanwise Wake Station

5.5 ARBITRARY CROSS-SECTIONAL GEOMETRIES

In general, the wing cross-sectional geometry is not symmetric and can have camber or twist associated with it. If the leading edge is drooped, the placement of the singularity of the conformal wing mapping will generate a grid that will cause a translation of the grid points in the wake cut. Hence, a lower wake cut grid point will not have a corresponding upper wake point. This internal grid generation complicates the implementation of the wake computation. Interpolation of the potential and speed from the lower wake mesh locations into the upper wake must be carried out in order to match the pressures and transverse velocity in the wake cut.

Figure 28 shows a selected sample of the crossflow plane solutions for a cambered arrow wing at $M_\infty = 1.75$, $\alpha = 5°$. This is the same arrow wing as Fig. 22, but with leading edge chordwise droop implemented with a parabolic chord-wise camber. Figure 22a shows an early station in the wake. The translation of the upper and lower wake cut points is evident. This becomes increasingly apparent farther downstream in the wake, where twice as many points appear on the wake cut. The interpolation scheme seems to work well for general geometries and has also been implemented for wing-body geometries.

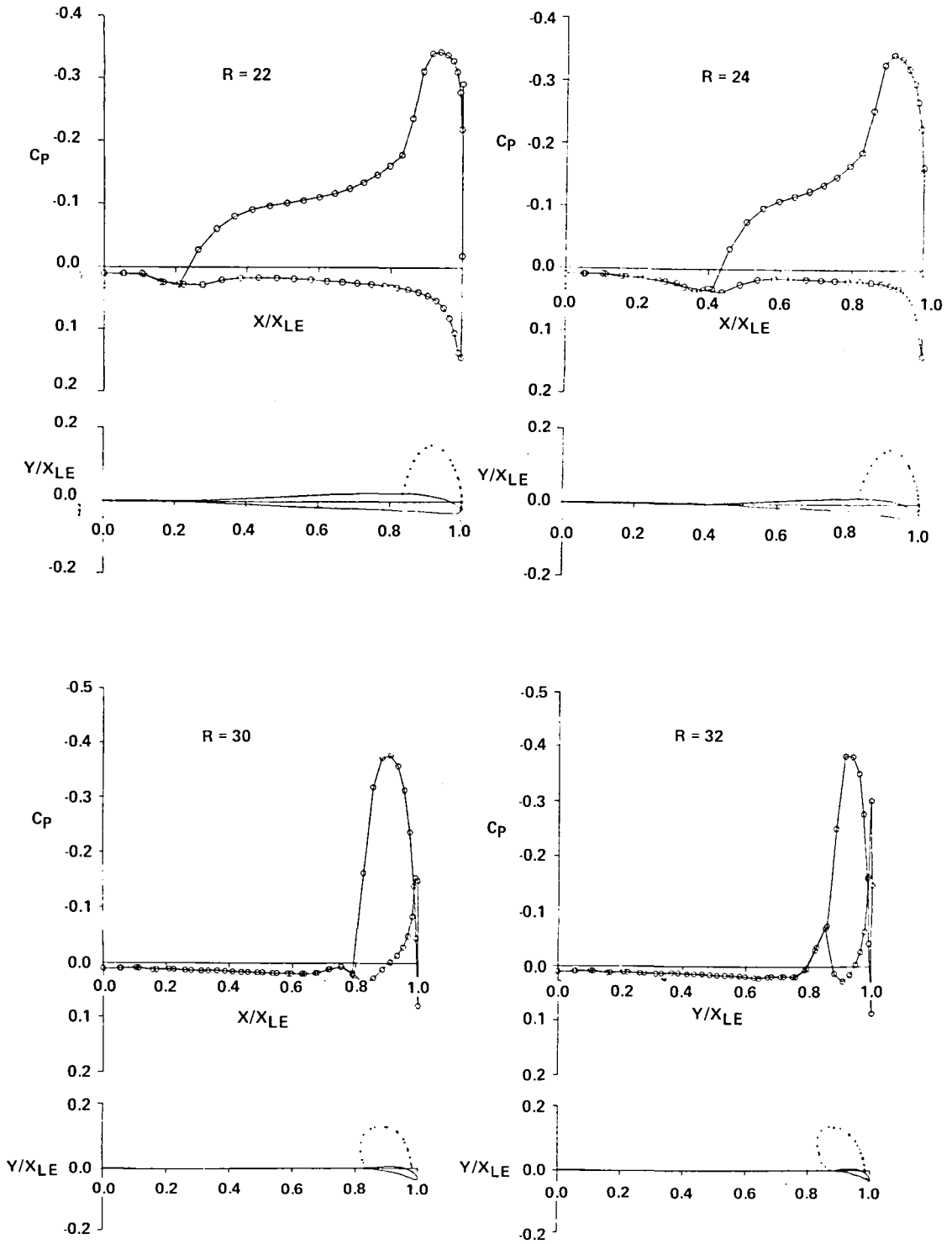


Fig. 28 Cambered Arrow Wing ($\Omega_{LE} = 70°$, $\Omega_{TE} = 45°$)
$M_\infty = 1.75$, $\alpha = 5°$ ($C_r = 20$)

This Page Intentionally Left Blank

6. ENTROPY CORRECTIONS - MODIFIED POTENTIAL METHOD

6.1 BASIC FORMULATION

An option exists in the NCOREL program that allows the user to fit the bow shock as an external boundary. The shock jump conditions used are the isentropic jump conditions obtained from the conservation of mass normal to the shock given by

$$\bar{\rho}_H q_{NH} = \bar{\rho}_L q_{NL} \tag{75}$$

where q_N refers to the normal velocity to the shock surface, $\bar{\rho}$ is the normalized density, and L and H refer to the low- and high-pressure sides of the discontinuous shock surface. The density is defined by the isentropic relation,

$$\bar{\rho}_{H_i} = \bar{\rho}_L \left[\frac{1 + (\frac{\gamma-1}{2})M_L^2}{1 + (\frac{\gamma-1}{2})M_N^2}\right]^{\frac{1}{\gamma-1}} \tag{76}$$

where M refers to the Mach number.

The accuracy of the above conditions tends to diminish as the normal Mach number to the shock exceeds 1.4. Hence, for NCOREL computations where the normal Mach number is in the range of 2.0, large inaccuracies in the computed pressures are to be expected.

In an attempt to eliminate some of this error to the isentropic formulation, the Rankine-Hugoniot (R-H) conditions were installed as an option in NCOREL (see Ref 24). The entropy (i.e., $\Delta S/R$) is assumed zero in the isentropic formulation. Instead, the entropy can be computed via the R-H conditions as a function of the incoming normal Mach number as,

$$\frac{\Delta S}{R} = \frac{1}{\gamma-1} \log \left[\frac{2\gamma M_{NL}^2 - (\gamma-1)}{\gamma+1}\right] - \frac{\gamma}{\gamma-1} \log \left[\frac{(\gamma+1)M_{NL}^2}{2+(\gamma-1)M_{NL}^2}\right]. \tag{77}$$

The shock jump conditions can be made to satisfy the R-H conditions by modifying the density as

$$\bar{\rho}_H = \bar{\rho}_{H_i} e^{-\Delta S/R} \tag{78}$$

where $\bar{\rho}_{H_i}$ is computed from Eq (76). The introduction of the nonisentropic shock conditions poses no difficulties and requires very little additional computation, and is implemented iteratively during the overall implicit relaxation procedure. The R-H conditions described by Eq (75) to Eq (78) will alter the position and shape of the bow shock when entropy production becomes significant. This, in turn, alters the entire velocity field computed within the shock layer. Figure 29 shows an example of this application to a 30° cone at zero angle of attack. The dashed line with x-symbols indicates the surface cone pressure as a function of M_∞ or $M_{N\infty}$, as computed by the default isentropic jump conditions in NCOREL. The straight line indicates the exact Euler solution. The departure from the Euler solution becomes dramatic as the Mach number normal to the shock begins to exceed about 1.5. If the R-H shock conditions are implemented in NCOREL and the pressure is computed from isentropic relations, the dashed curve with square symbols results. A pure implementation of the R-H conditions to the bow shock, neglecting the effect of entropy on pressure, yields less accuracy than the full isentropic jump conditions. Therefore, the pressure must be corrected by the entropy produced by the bow shock. This can be accomplished by the following relations:

$$\frac{p}{p_\infty} = \left(\frac{p_i}{p_\infty}\right) e^{-\Delta S/R} \tag{79}$$

where

$$\frac{p_i}{p_\infty} = \left[1 - \frac{\gamma-1}{2} M_\infty^2 (q^2/q_\infty^2 - 1)\right]^{\frac{\gamma}{\gamma-1}} \tag{80}$$

and p_i is the isentropic pressure computed as a function of the total velocity. The pressure field is modified, then, by both the altered shock position and the velocity field, and also by the bow shock entropy. The entire pressure field for the cone at zero angle of attack is corrected by the bow shock entropy fed down to the surface on radial coordinate lines. When the R-H conditions are satisfied and the pressure is corrected using Eq (79) and (80), the results for the surface pressure are the circles lying on the Euler curve in Fig. 29. Hence, Eq (75) to Eq (80) allow duplication of the

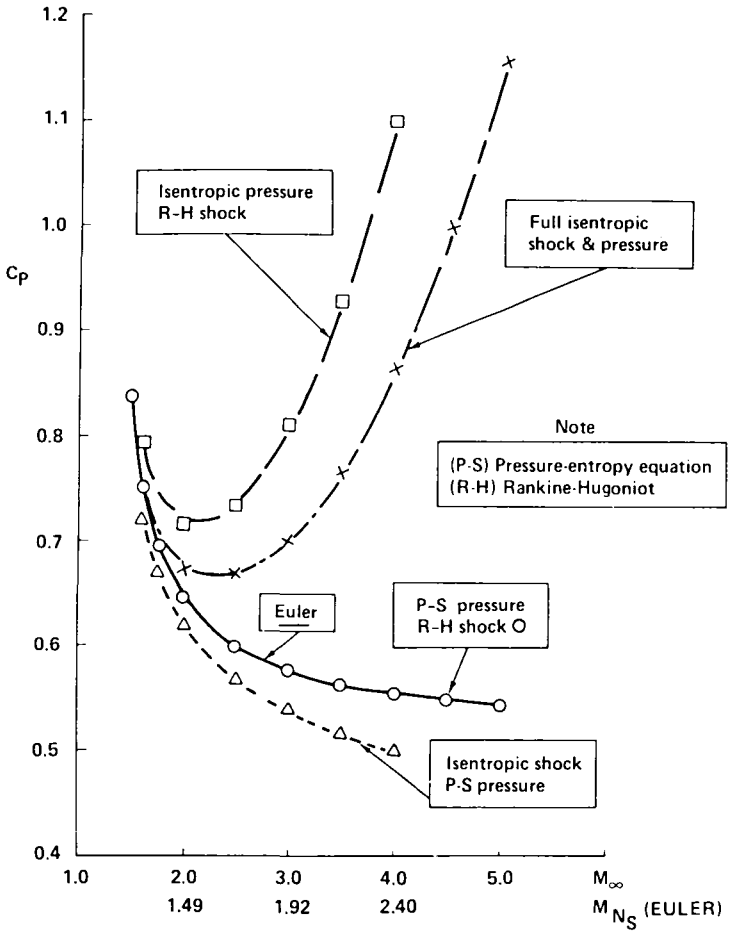


Fig. 29 Effects of Entropy Corrections on a 30° Circular Cone at $\alpha = 0°$

exact Euler results for any Mach number or deflection angle for a cone at zero angle of attack. This occurs because the shock layer for a cone at zero angle of attack is irrotational and, if the entropy, jump is taken into account, the Euler solution can be exactly reproduced by the full potential equation using the procedure outlined above.

For general shapes at angle of attack, the identical procedure is applied. The modified potential method inherently neglects rotational flow effects due to the circumferential variation of entropy. Some discrepancy should therefore be expected in comparison with Euler solutions when rotational effects dominate. For general shapes, the pressure field is corrected by the bow shock entropy, which is allowed or assumed to propagate down to the surface via coordinate lines.

6.2 CONICAL FLOWS

To demonstrate the modified potential method installed in NCOREL, several comparisons with Euler solutions and experiments for conical solutions will be shown. The first case is illustrated in Fig. 30 for a 30° cone at $M_\infty = 3$ and $\alpha = 15°$. Both the isentropic full potential and the modified potential solutions are shown for surface pressures and bow shock shape in comparison to the Euler solution. The maximum bow shock entropy and variation is fairly large for this case $\left(0.65 \leq \frac{\Delta S_s}{R} \leq .054\right)$, as indicated by the displacement of the modified potential R-H shock from the isentropic potential shock. Excellent agreement in surface pressures compared to the Euler curve is obtained by the modified potential method. Excellent agreement in bow shock position is achieved on the windward side, but some discrepancy is apparent on the leeward side. This discrepancy is most likely the result of neglecting rotational effects in the shock layer which are largest in the leeward plane.

Figure 31 shows a very extreme example of the application of the modified potential method for a 10° cone at $M_\infty = 10$, $\alpha = 8°$. The isentropic and modified potential results are compared to an Euler solution. Excellent agreement in surface pressures is obtained by the modified potential method. Considering the large entropy production $(1.5 \leq \Delta S/R \leq 10^{-1})$, very good agreement in the windward bow shock location is obtained with some discrepancy in the leeward shock shape. The discrepancy in the leeward shock

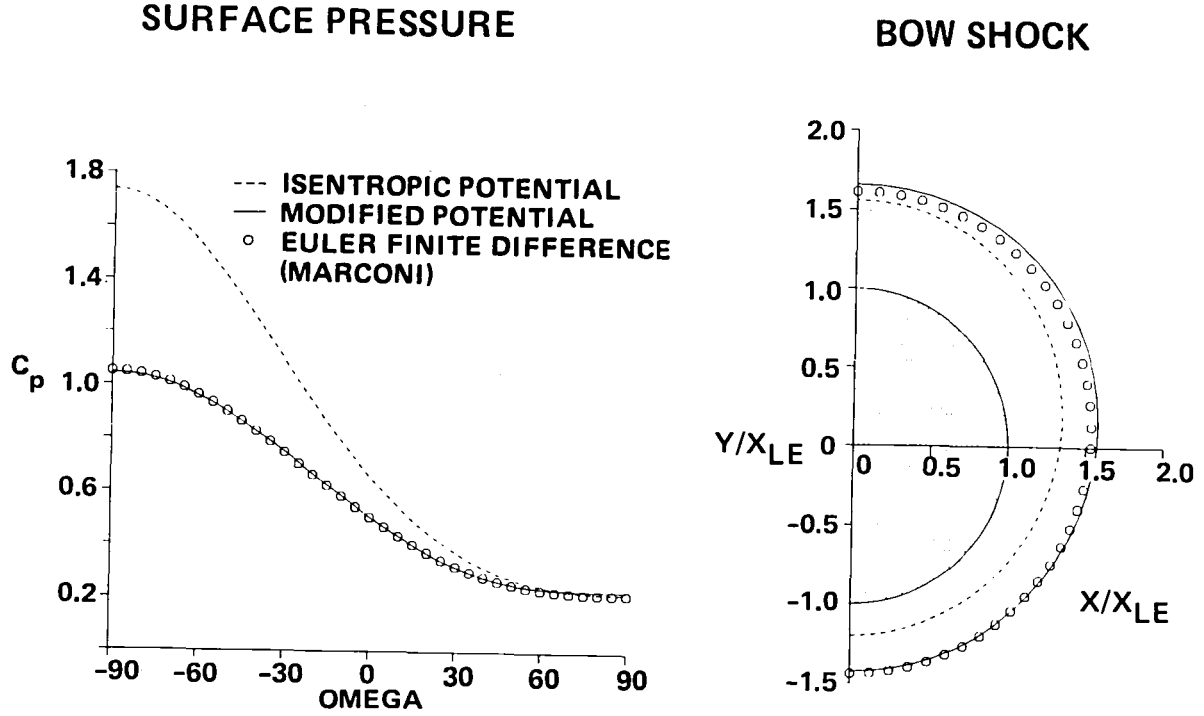


Fig. 30 Modified Potential Solution for a 30° Circular Cone at $M_\infty = 3$, $\alpha = 15°$, $\theta = 30°$

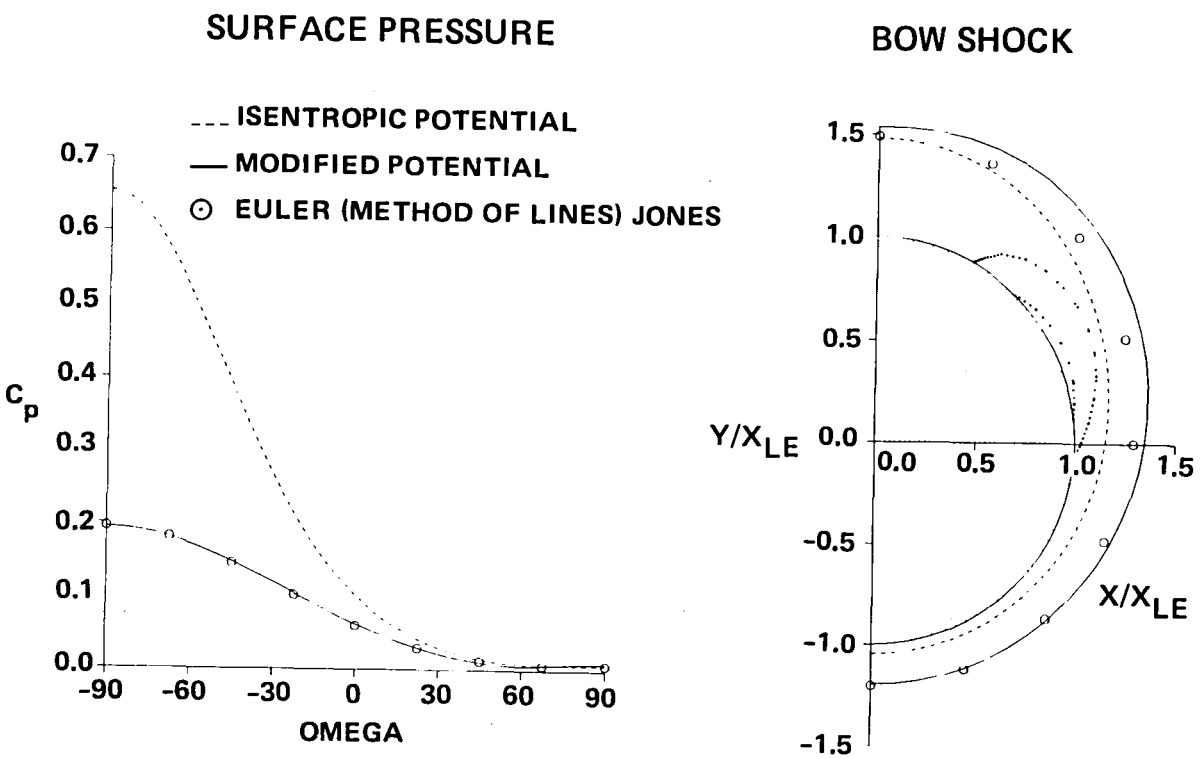


Fig. 31 Modified Potential Solution for a 10° Circular Cone at $M_\infty = 10$, $\alpha = 8°$, $\theta = 10°$

shape does not lead to large discrepancies in pressure because the bow shock is relatively weak in this region.

In general, the isentropic or standard potential formulation will overpredict the windward pressures. The largest discrepancy in the modified potential bow shock occurs in the lateral leeward region of the shock. Evidently, rotational effects are largest in this region.

It has been shown that the modified potential method yields a remarkable improvement in accuracy for cones at angle of attack even for flows with very large entropy production and variation. The method will now be demonstrated for more general conical shapes, in particular, ellipses. Since simple Euler solutions are not available for ellipses at high angle of attack, the potential solutions are compared to the experimental data of Ref 25 and 26. Figure 32 shows a comparison of the potential solutions to experiment for an elliptic cone at $M_\infty = 6.0$, $\alpha = 20°$ and a major-to-minor axis ratio of 1.79. Excellent agreement with experimental surface pressure data is achieved with the modified potential method. It should be noted that the major entropy effect occurs in the windward region. Figure 33 shows another elliptic cone at $M_\infty = 4.5$, $\alpha = 20°$ with a major-to-minor axis ratio of 3.0. Good agreement in surface pressure is achieved on the windward side, but only fair agreement is achieved on the leeward surface of the cone. For this case, the potential solutions indicate crossflow shocks on the leeward side of the cone. The leeward side may be dominated not only by rotational effects, but also by viscous separated flow effects. Figure 34 shows the normal and axial force coefficients and pitching moment for a 20° circular cone at $\infty = 4.5$ for a range of incidence up to 20°. The predictions by both the isentropic potential and modified potential methods are shown in comparison to the experimental data of Ref. 26. The marked improvement in accuracy by the modified potential method is readily apparent. Figure 35 shows a similar comparison for 20° elliptic cone with a 3:1 axis ratio at $M_\infty = 4.5$.

6.3 GENERAL 3-D APPLICATIONS

The modified potential method outlined previously for conical flows is directly applicable to three-dimensional flows. In three-dimensional flow, the R-H conditions are satisfied at each spherical marching plane. The entropy of the bow shock at each station is then used to further correct the pressures resulting from the modified potential velocity field. Figure 36

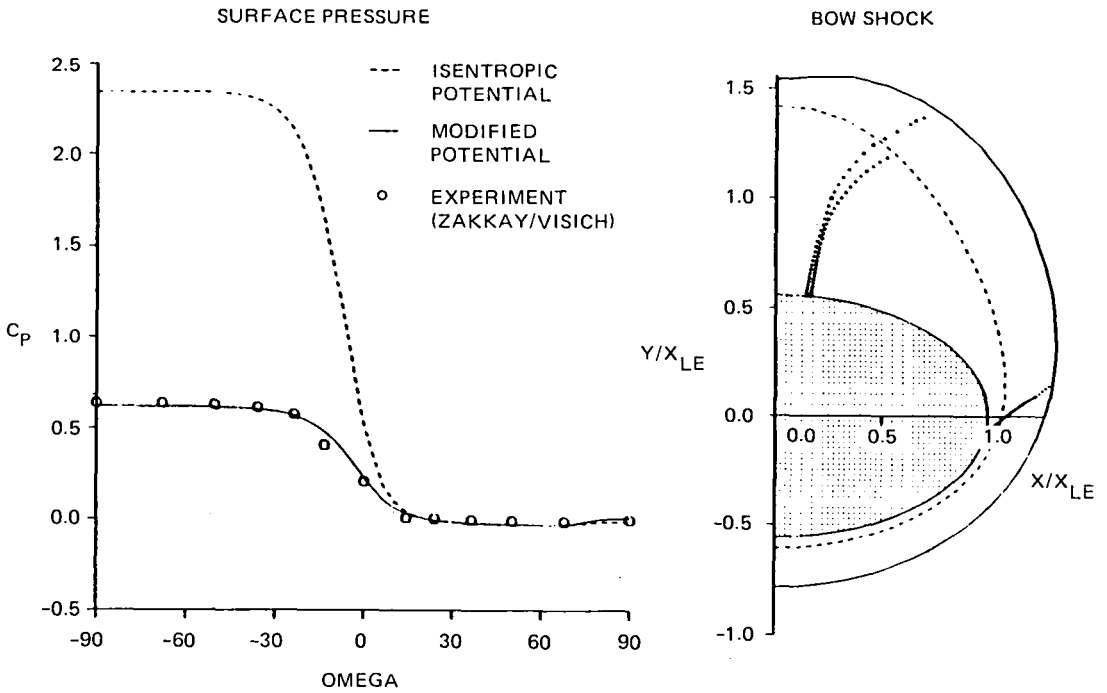


Fig. 32 Modified Potential Solution for an Elliptic Cone at $M_\infty = 6.0$, $\alpha = 20°$, $\theta = 22°$, a/b = 1.79

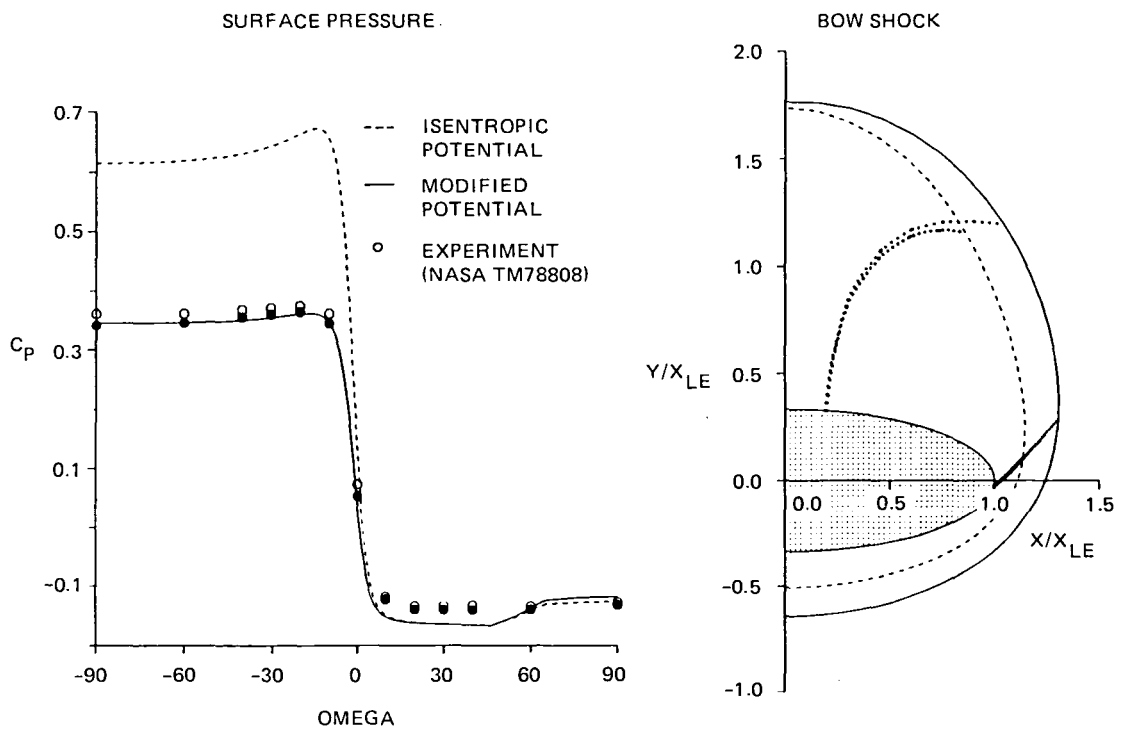


Fig. 33 Modified Potential Solution for an Elliptic Cone at $M_\infty = 4.5$, $\alpha = 20°$, $\theta = 20°$, a/b = 3

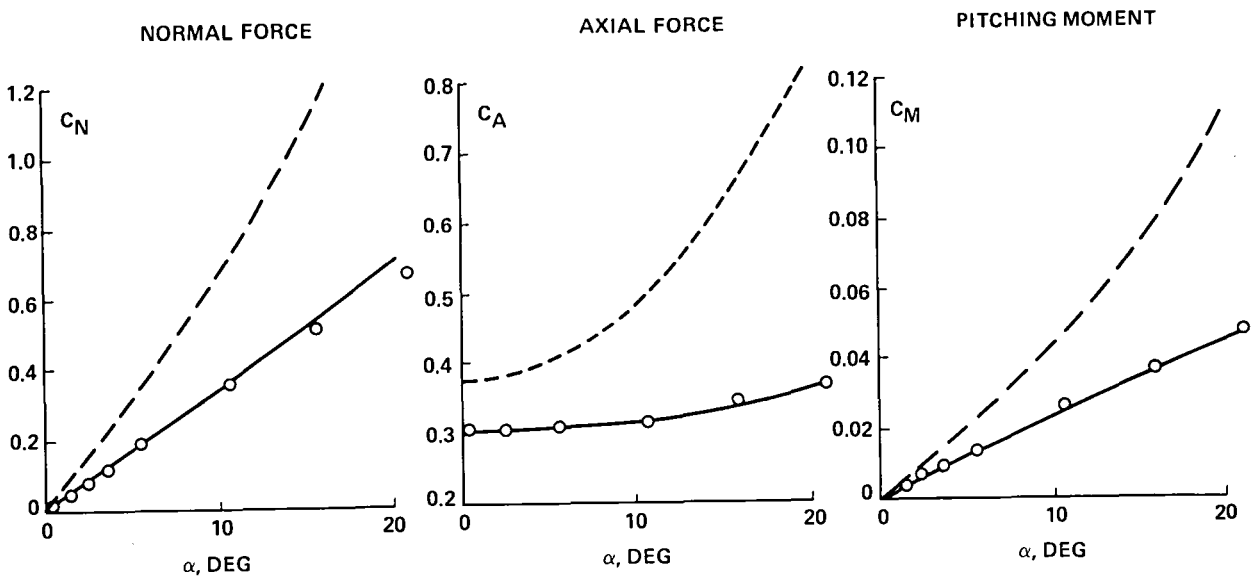


Fig. 34 Comparison of Force and Moment Prediction with Experiment for a Circular Cone ($M_\infty = 4.5$, $\theta = 20°$)

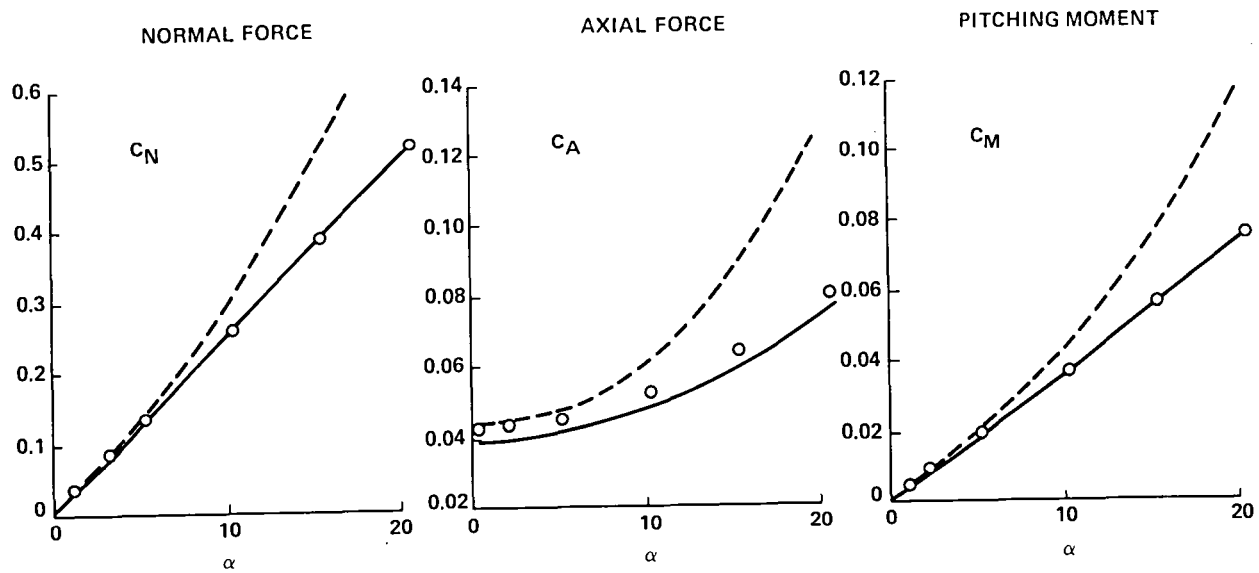


Fig. 35 Comparison of Force and Moment Prediction with Experiment for an Elliptic Cone ($M_\infty = 4.5$, $\theta = 20°$, a/b = 3)

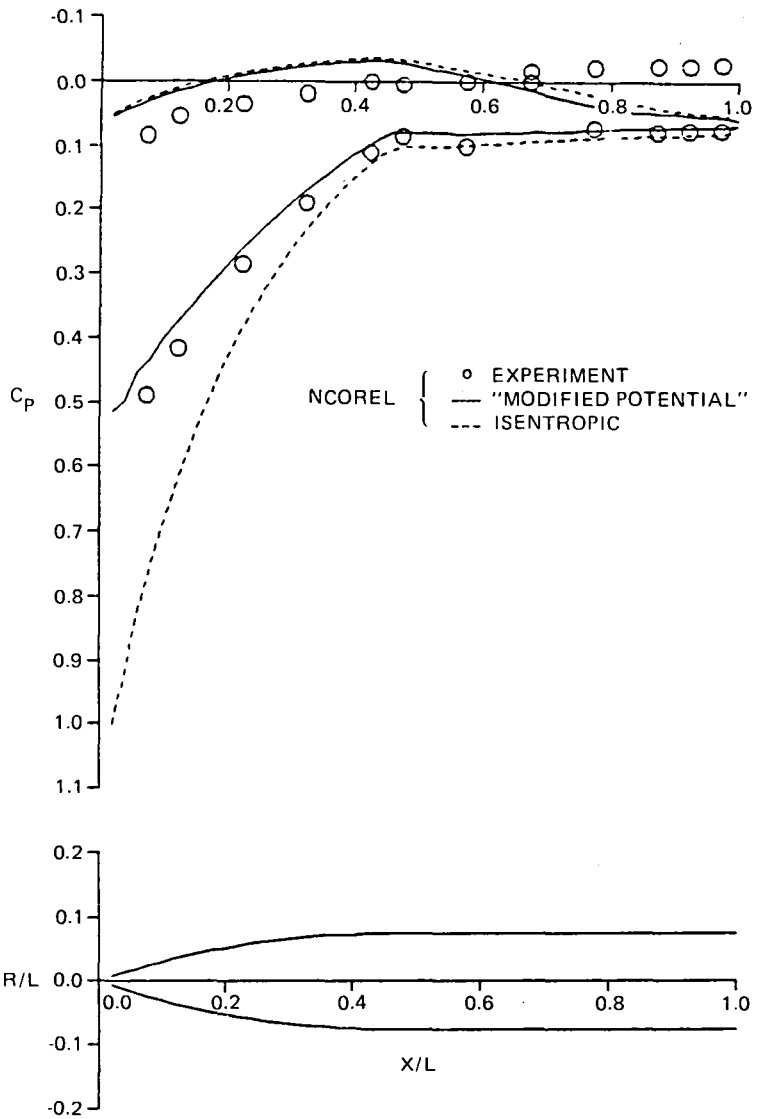


Fig. 36 Example of Modified Potential & Standard Isentropic Potential Solutions in NCOREL for a Circular-Arc-Cyclinder Body of Revolution at $M_\infty = 4.63$, $\alpha = 12°$

shows a sample computation for a circular-arc-cylinder body of revolution at $M_\infty = 4.63$, $\alpha = 12°$. The modified potential method is shown in comparison to experimental axial pressure data of Ref 27 and the standard isentropic potential solution. The modified potential method markedly improves the correlation with experimental pressures on the windward side of the body. Both modified and standard isentropic potential methods yield similar correlation with pressure data in the leeward plane. Viscous effects typically dominate the lee side of bodies at high incidence and, hence, good correlation is not expected.

Typical for bodies is the fact that the entropy production is most dominant near the apex, where the bow shock has the greatest strength. The effect of entropy diminishes rapidly for expanding body shapes. At high angle of attack, the entropy correction is dominant on the windward side.

The modified potential method is designed primarily to account for entropy effects generated by the primary bow shock. Secondary embedded shocks are not accounted for in the present method and, hence, the pressure rise across strong embedded shocks may be overpredicted by NCOREL.

7. SECOND-ORDER ACCURATE COMPUTATIONS

7.1 BOW SHOCK FITTING

Currently, NCOREL fits the bow shock using a simple, explicit first-order marching technique that computes the shock slopes in the mapped space at the present station and uses these slopes to extrapolate the shock shape to the next station. The internal mesh is governed by the shock shape because of the shearing transformation. If the extrapolated shock shape is held constant, the internal flow computation iterates on a fixed mesh. This procedure is the most cost efficient, because the mesh, free-stream velocities, and metrics need only be computed once. This procedure works well as long as the shock curvature is not large; this is usually the situation for moderate Mach numbers. At higher Mach numbers (e.g., $M_\infty > 4$), instabilities in the code have been encountered primarily near the apex of the configuration, where the bow shock has the largest curvature.

The instabilities arise from the incompatibility of the explicit bow shock computation with the implicit internal flow solution technique. Large marching step sizes are usually used and, if large bow shock curvature exists, the bow shock computation may require smaller marching step sizes to retain accuracy. One solution to this problem would be to use small step sizes near the apex of the configuration and increase the step size as the computation proceeds. The drawback to this procedure is that it does not work well or at all for bodies that are locally blunt near the apex. For this class of bodies, NCOREL automatically computes a sharp conical nose cap and, for locally blunt noses, the solution will eventually become subsonic due to the large deflection angles as the step size is reduced. This may cause the computation to diverge. An alternate solution is to use an implicit second-order accurate method to compute the bow shock in regions of large curvature compatible with the internal flow solution. This requires an iteration of the bow shock position simultaneously with the internal flow field in a similar fashion to the conical computation. This procedure is computationally costly because the nonconical mesh is no longer fixed, but must be updated at each iteration of the flow field, and new metrics and free stream velocities must be recomputed at each mesh point. This procedure has been implemented in NCOREL as an option and has been found to eliminate the bow shock

instabilities at higher Mach numbers without having to resort to using inordinately small step sizes. In addition, the second order bow shock method can be switched off downstream of the apex.

Figure 37 shows a sample computation for a circular-arc-cylinder body of revolution at $M_\infty = 5.40$, $\alpha = 0°$. Both first- and second-order accurate bow shock fit results are shown. These NCOREL solutions were obtained with the entropy option switched on. Severe oscillations in the first order solution occur near the apex. These oscillations are essentially eliminated with the use of the second-order method.

7.2 SUPERSONIC CROSSFLOW COMPUTATIONS

Large gradients typically occur near the leading edges of spanwise wing cross sections, which ultimately lead to the generation of supersonic crossflow velocities terminating in a crossflow shock. The original finite difference formulation in NCOREL computed supersonic crossflow points with first-order accuracy. A second-order accurate treatment of these points has now been included in NCOREL as an option. Implementing the second-order accurate supersonic crossflow option tends to reduce the crossflow Mach numbers around the leading edges of thin-wing sections or wing sections with small leading edge nose radii. This also reduces the appearance of low-pressure peaks or suction spikes near the leading edge.

Figure 38 shows an example of this option on a thin elliptic cone at $M_\infty = 2.0$, $\alpha = 10°$ and a relatively crude 30x30 grid in comparison to a fine grid computation.

The first- or second-order supersonic crossflow options affect the second (X,Y) derivatives of the potential at supersonic crossflow points. The implementation of the second-order supersonic option also requires more computational time, since each supersonic point now requires an additional upwind point in its second-derivative computation.

For example,

$$\Delta X^2 F_{XX_{i,j}} = F_{i,j} - 2F_{i-1,i} + F_{i-2} \qquad \text{(First Order)}$$

$$\Delta X^2 F_{XX_{i,j}} = 2F_{i,j} - 5F_{i-1,j} + 4F_{i-2,j} - F_{i-3,j} \qquad \text{(Second Order)} \tag{81}$$

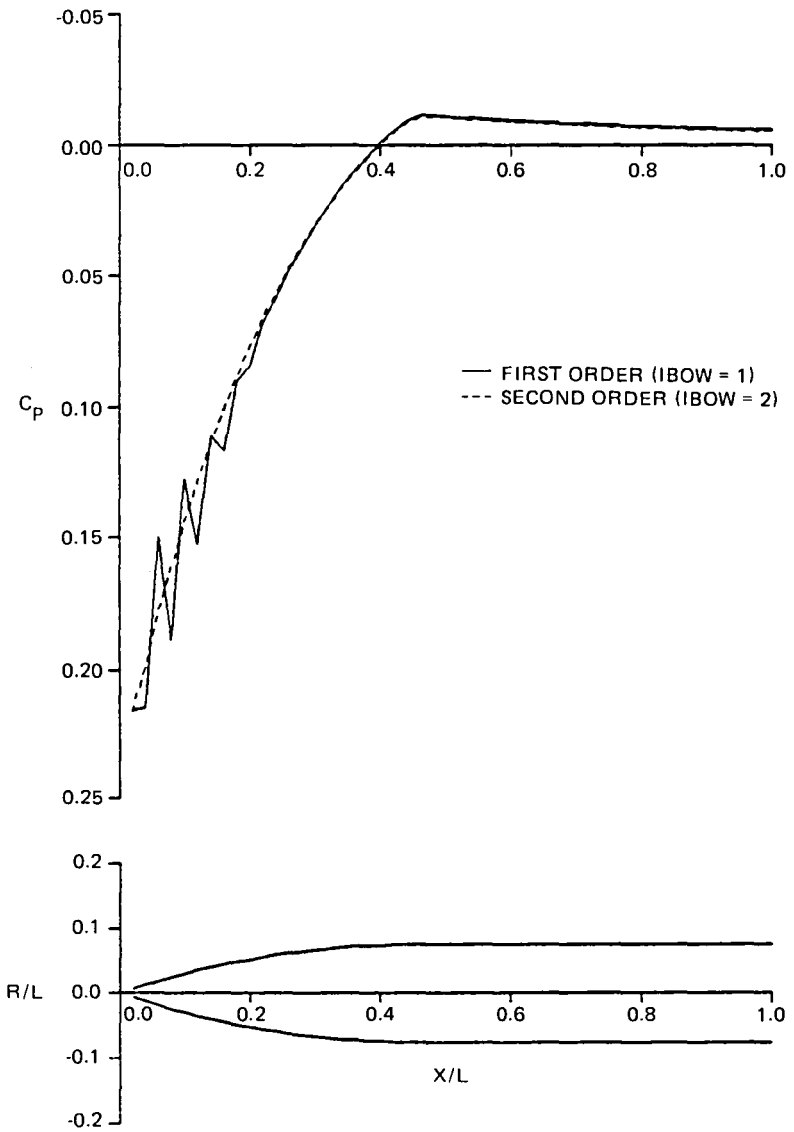


Fig. 37 Example of First & Second Order Bow Shock Fit Methods in NCOREL at $M_\infty = 5.40$, $\alpha = 0^\circ$

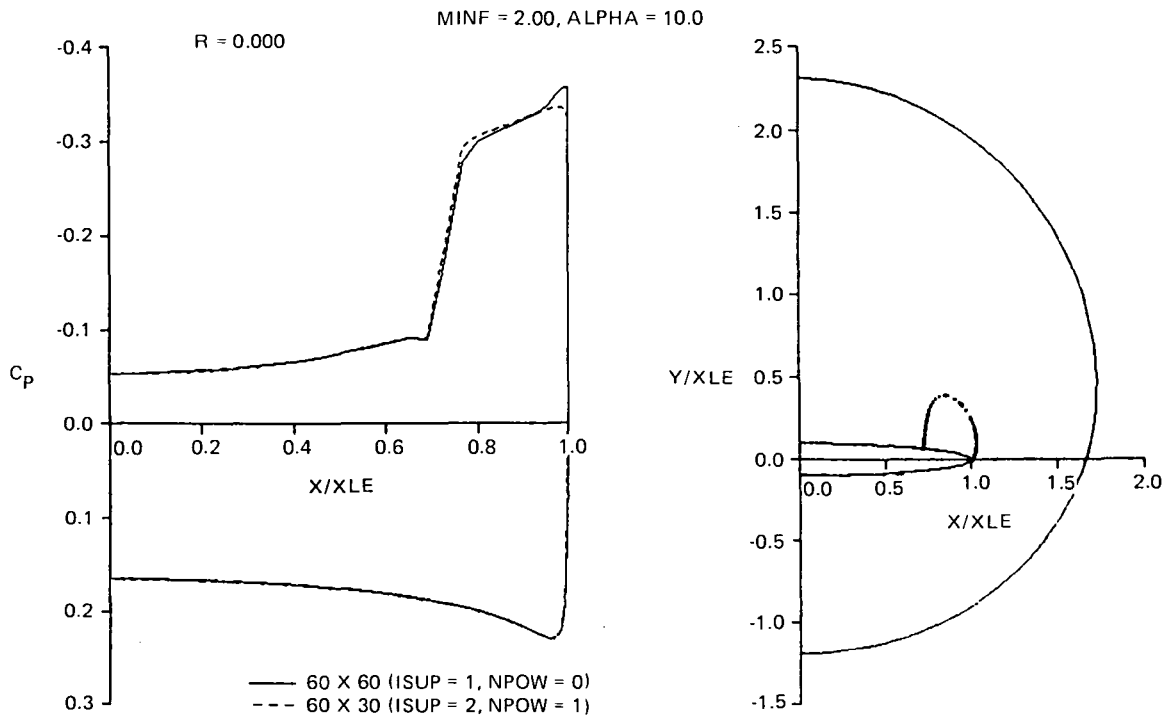


Fig. 38 Example of Second Order Supersonic Crossflow & Radial Grid Stretching Options

Similar treatments are included for the F_{XY}, F_{YY}, F_{XZ}, and F_{YZ} terms at supersonic crossflow points. The Z derivatives remain first-order accurate marching derivatives.

This Page Intentionally Left Blank

8. INLET SIMULATION

There are two possible methods by which an inlet can be simulated. The first method is simply to swallow or capture all of the mass included within the inlet face with little or no disturbance to the external flow. The second method is to prescribe a normal velocity boundary condition on the node points that lie on the inlet face. After some preliminary investigation, the inlet capture method was adopted for several reasons, including ease of implementation. In the velocity boundary condition method, there is no way of knowing the appropriate magnitude of the prescribed velocity boundary condition. The mass flow at the inlet face would have to be determined a priori. This is an unreliable procedure because, if the wrong velocity condition is applied, subsonic conditions may be generated and the code would fail. Imposing freestream velocity may sound logical, but the local flow may be dramatically different from freestream and this condition might either accelerate or decelerate the flow in the vicinity of the inlet. Hence, in terms of user friendliness, the inlet mass capture method is the more desirable technique.

To develop the inlet capability in NCOREL, a sample analytical 3-D inlet fuselage geometry subroutine was created. Figure 39 shows the computational grid on a crude mesh for three views; the geometry is highly 3-D. It was built by starting with a circular arc cylinder, circular cross-section body, and then adding on a canopy and an inlet. The inlet is a side-mounted inlet described in cross section by a super ellipse which takes on a rectangular appearance. Figure 40 shows an isometric view of the computational mesh.

Four additional subroutines were added to NCOREL to include the inlet capture method as an option. Only a couple of changes are required to implement the method in NCOREL with these new routines. An r-station is specified as RINLET where the inlet is to be started. At $r \geq$ RINLET, an inlet switch (INLETT) is tripped from zero to unity. The inlet switch also trips the geometry cross section to add on the inlet geometry. Once the new inlet cross section has been generated, it is mapped and sheared and a new cross-sectional inlet on mesh is generated. A routine is called that computes the pressures on the old mesh. The potential and pressures are then interpolated onto the new inlet mesh. The new cross section and pressures are output. Two pressure distributions are then obtained at r = RINLET: one corresponding to

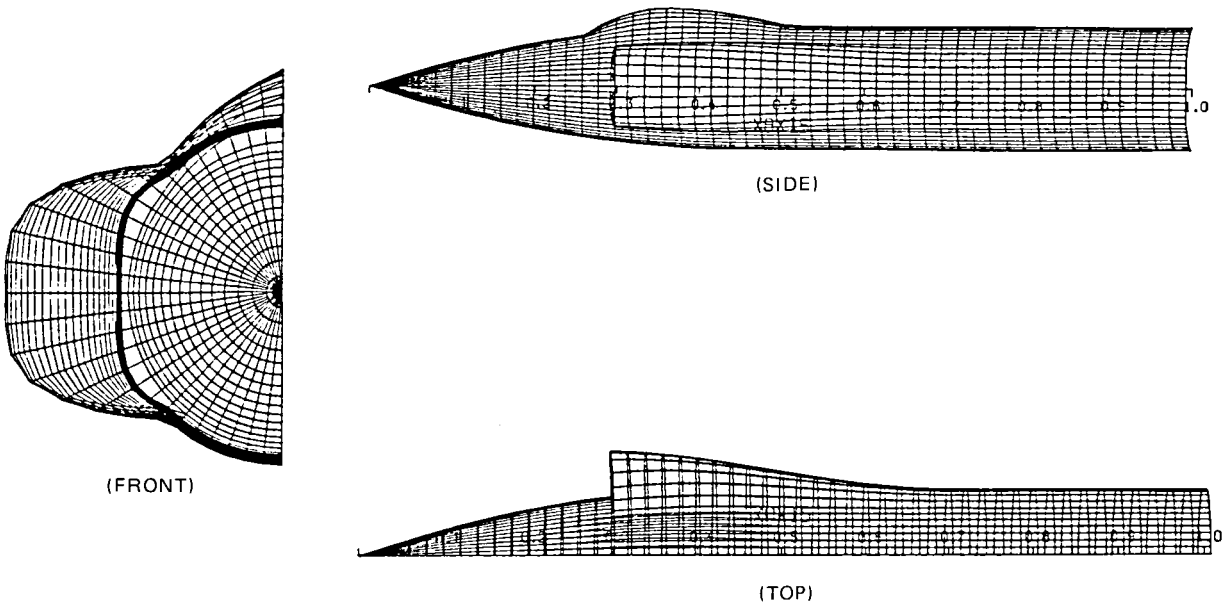


Fig. 39 Three Views of Crude Computational Mesh for Fuselage Inlet Configuration

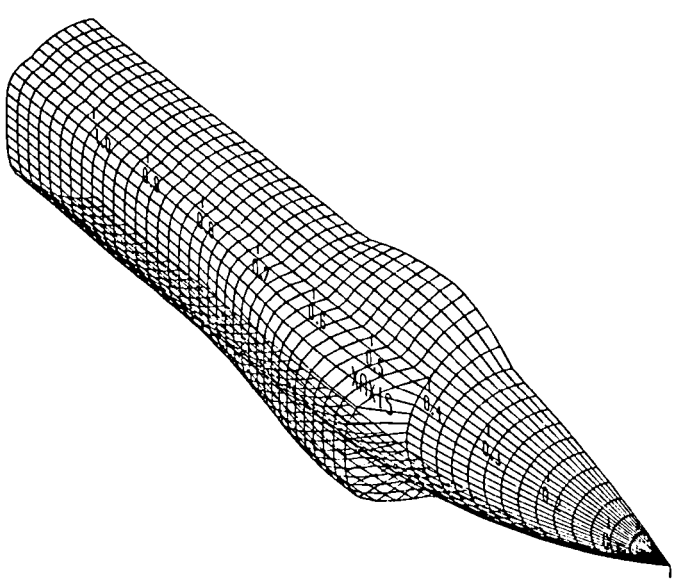


Fig. 40 Isometric View of Crude Computational Mesh for Fuselage Inlet Configuration

inlet off and the other to the interpolated inlet on cross section. Metrics and freestream velocities are then recomputed for the new inlet mesh for the computation to proceed.

Figure 41 shows the computed axial surface pressures in the upper and lower symmetry planes for $M_\infty = 1.70$, $\alpha = 5°$ for the fuselage with a side-mounted inlet. A very fine mesh was used in order to uncover problems (i.e., 100x60x60). A very large canopy shock is generated which shocks the flow to almost sonic conditions on the upper surface. This is followed by an equally large expansion around the canopy. A recompression shock also occurs just downstream of the canopy.

Figure 42 shows the axial surface pressure distribution on the side of the fuselage. The pressure jumps discontinuously at the station where the cross section changes character. A higher pressure occurs in the flow at the inlet side, which is reasonable, since the flow at the surface is expanding. A large expansion takes place as the inlet computation starts followed by a recompression.

The degree of complexity in computing a realistic isolated fuselage configuration is quite evident from these figures. A variety of shocks can be present in the flow field, including bow shock, canopy shock, canopy-fuselage recompression shocks, and external inlet shocks. For this particular configuration an external inlet shock was not readily apparent.

A complete wing-body-inlet configuration will be treated in Section 10.

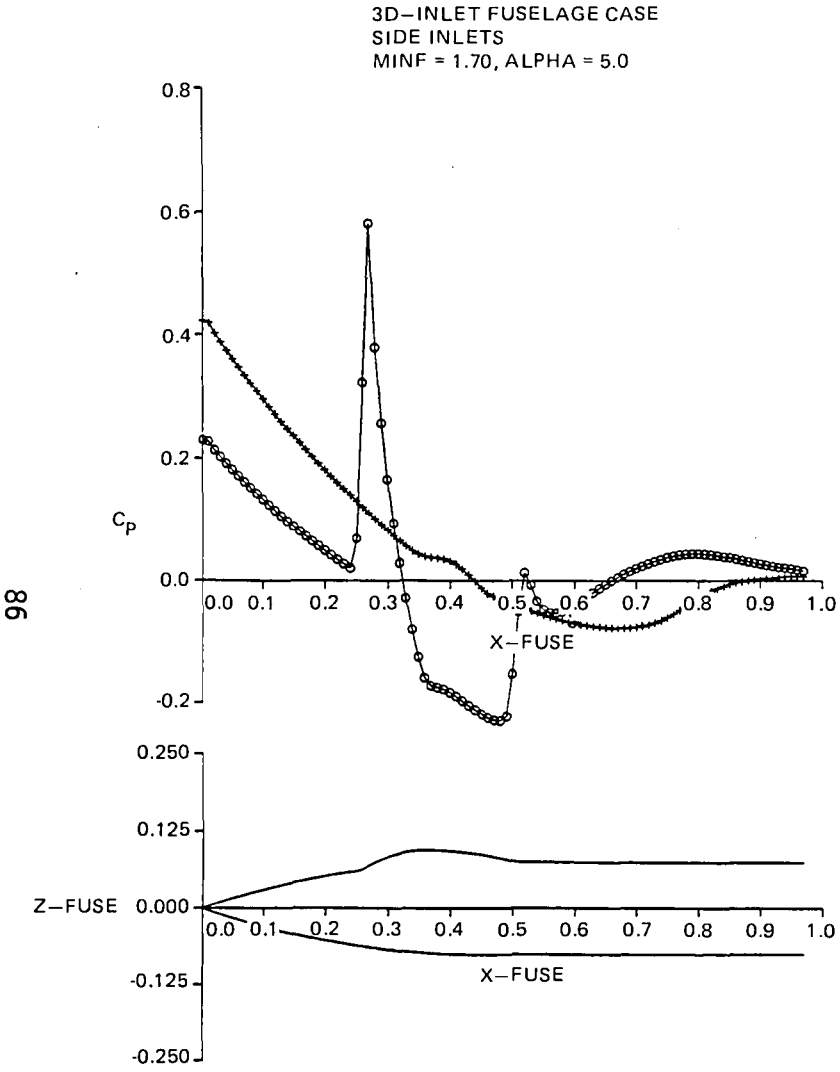


Fig. 41 Axial Surface Pressure Distributions in Plane of Symmetry for Inlet Fuselage

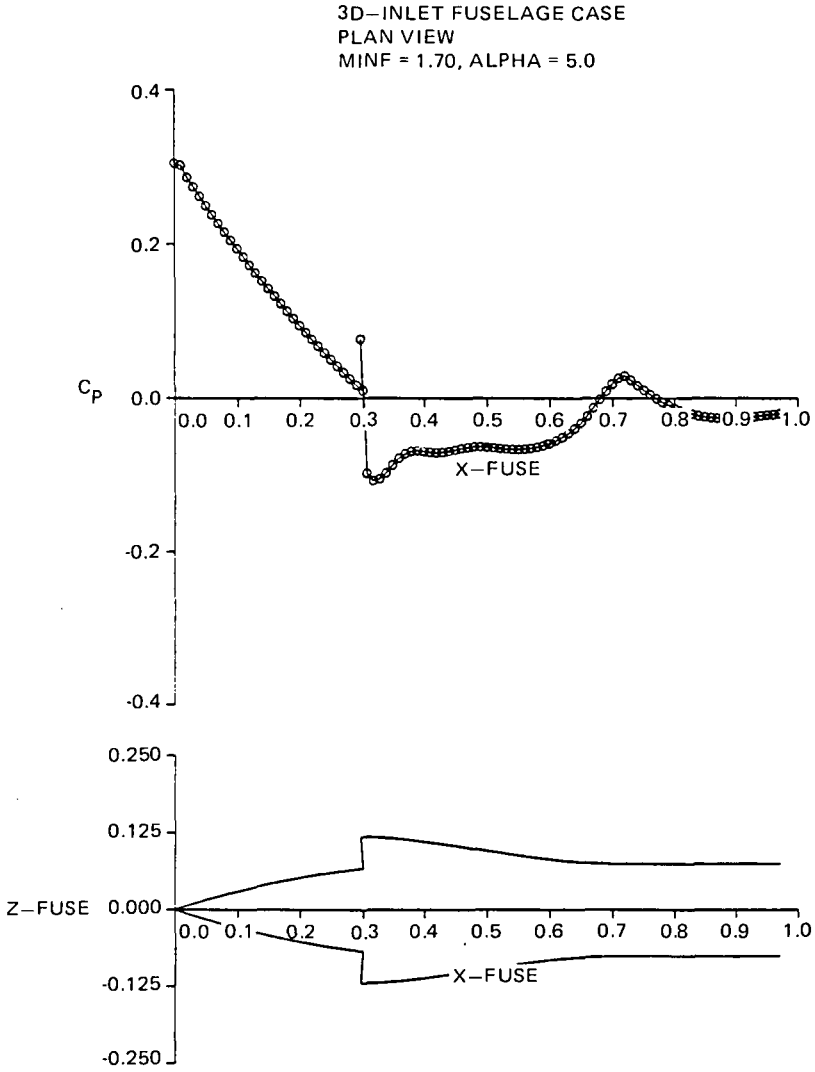


Fig. 42 Axial Surface Pressure Distribution on Side of Inlet Fuselage Configuration

9. GRID GENERATION AND MAPPINGS

9.1 MODIFIED SHEARING TRANSFORMATION

An alternate shearing transformation has been implemented in NCOREL that produces a crossflow radial stretching in the mesh generation. The shearing transformation is applied as a final stage in the grid generation process. The shearing transformation operates on the conformally mapped grid. This option is switched on by a parameter NPOW when it is set equal to unity. A zero value of NPOW causes the code to generate the standard mesh. This new transformation is applied to the Y shearing in the form of a hyperbolic stretching. The transformation has the form:

$$Y = \frac{(\rho-B)}{(c-B)} \frac{1}{\left(\frac{\rho}{c}\right)^{NPOW}}. \tag{82}$$

Equation (82) reverts to the standard shearing for NPOW = 0. The simple modification to Eq (82) for hyperbolic stretching has a much larger overall impact in the code. All of the Y-shearing derivatives had to be modified in NCOREL to reflect the new transformation. These derivatives are more easily represented if we let

$$\lambda = \left(\frac{c}{\rho}\right)^{NPOW} \quad \text{and} \quad Y = \frac{(\rho-B)}{(c-B)} \lambda.$$

The new form of the derivatives become somewhat more complicated. For example:

$$Y_\theta = \frac{1}{(c-B)}\left[(\rho-B)\lambda_\theta - B_\theta\lambda - Y(C_\theta-B_\theta)\right].$$

Similar modifications occur for the following set of derivatives:

$$Y_\theta,\ Y_{\theta\theta},\ Y_\rho,\ Y_{\rho\theta}$$
$$Y_R,\ Y_{RR},\ Y_{\rho R},\ Y_{\theta R}.$$

One additional complication arises in the implementation of the new shearing transformation. In the standard shearing $Y_{\rho\rho} = 0$, and in the new stretching $Y_{\rho\rho} \neq 0$. This necessitates some additional terms in the governing equation in the form of the coefficients of the F_y term. These new terms are

$$[(a^2-v^2) + C_1h_1]Y_{\rho\rho}F_y$$

where $C_1 = R(C_3h_1-2WVh)$ and $C_3 = - RH^2 (W^2 - a^2)$.

As examples of the new mesh generation, Fig. 43 and 44 show both the standard grid and the new stretched grids (30x30) for 20° circular and elliptic cones at $M_\infty = 2$, $\alpha = 10°$. The new grid allows for better resolution in the leeward shoulder region where the flow gradients are largest at high incidence. It should be noted that increasing the resolution near the body removes grid points near the bow shock. The user should be careful in retaining enough grid points for accurate leeside bow shock computations at high incidence.

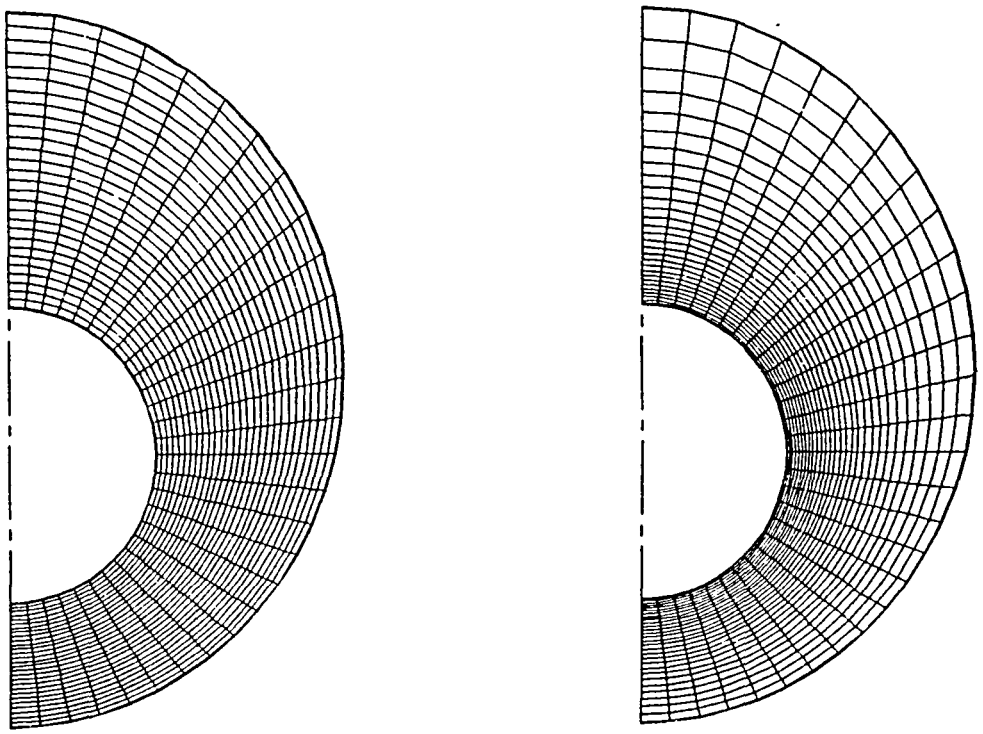
9.2 NEW CONFORMAL MAPPINGS

A more general procedure is under development for mapping arbitrary cross sections in NCOREL. The procedure was originally developed by Moretti (Ref 3) and applied by Marconi (Ref 28). It consists of using a series of Karman-Trefftz Conformal transformations. The mapping has the general form:

$$\frac{Z - Z_o}{Z + \overline{Z}_o} = \left(\frac{W - Z_o}{W + \overline{Z}_o}\right)^{\delta}. \qquad (83)$$

Z_o is the location of the singularity in the physical space $Z(x,y)$. W is the mapped space $W(\xi,\eta)$ or $W(\rho,\theta)$; δ is the power of the mapping. If $\delta = 2$, the transformation (83) reduces to the original wing mapping in NCOREL. If $\delta = 1/2$, the mapping has the ability to open corners. $\pi\delta$ can be interpreted as the external angle of the mapping. To map a slit or a thin wing, the external angle is 2π. To open a 90° external corner, the angle is $\pi/2$.

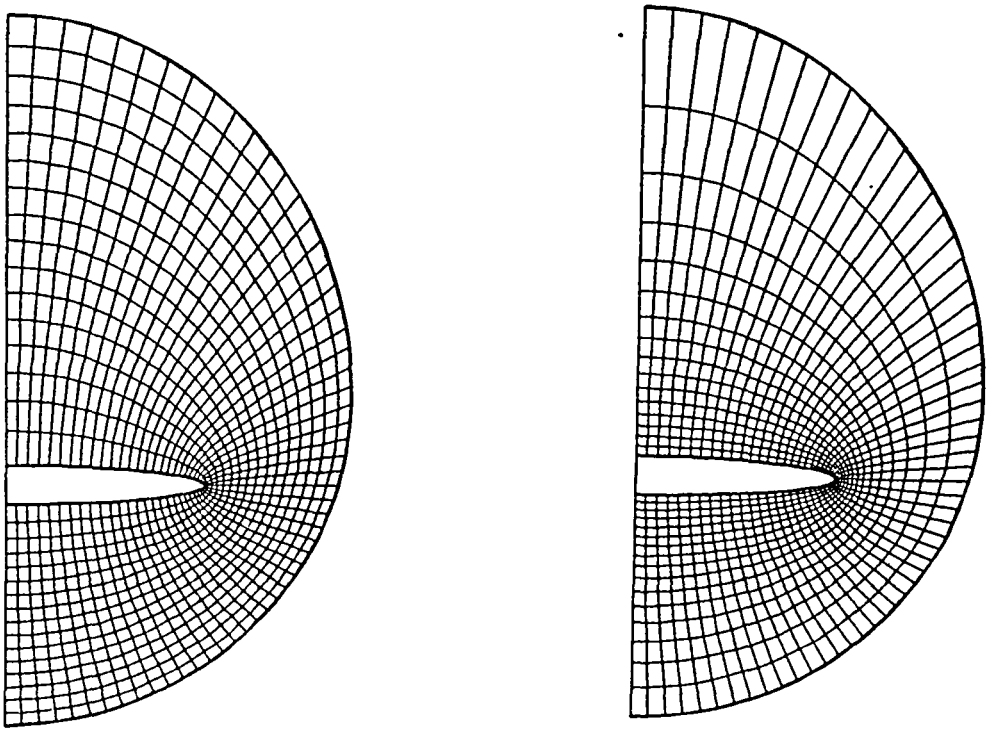
The mappings are applied in a progressive fashion -- first mapping the largest angles and then the corners. Many intermediate mapped spaces can be involved depending on the number of times the transformation (83) is



(a) STANDARD GRID
NPOW = 0

(b) STRETCHED GRID
NPOW = 1

Fig. 43 Examples of Standard & Stretched Grids on a 20° Circular Cone at $M_\infty = 2.0$, $\alpha = 10°$



(a) STANDARD GRID
NPOW = 0

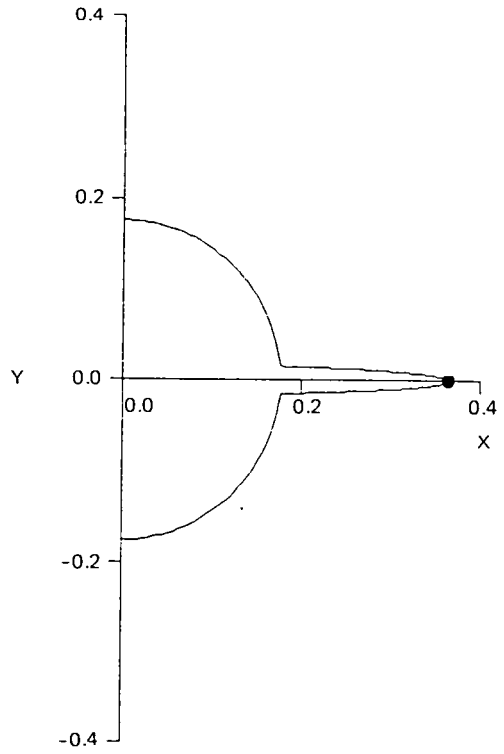
(b) STRETCHED GRID
NPOW = 1

Fig. 44 Examples of Standard & Stretched Grids on a Thin Elliptic Cone at $M_\infty = 2.0$, $\alpha = 10°$

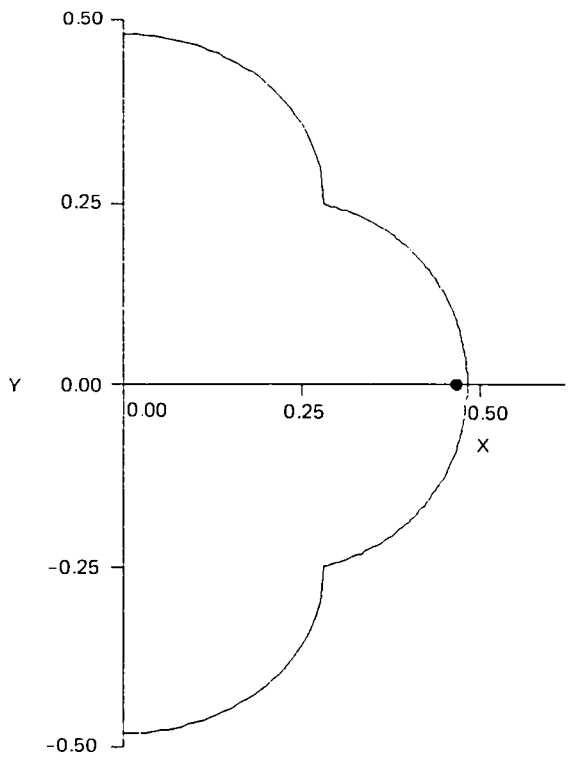
applied. For example, if three mappings are used, the three locations of each singularity are specified in the physical space. The first mapping is applied and the two remaining singularity locations are mapped by the first transformation. The second mapping is then applied and the third singularity location is then mapped by the second transformation. The third transformation is then applied. In applying the transformations, a shift in the body cross section will typically occur. Thus, after the last transformation of the form (83), a translation or shift takes place to center the mapped cross section. The last transformation that is applied is done automatically in the grid generation code. This transformation checks to see if the mapped cross section is vertically elliptic in nature. If so, a last transformation of the form

$$Z = W + \frac{\left(\frac{Z_o}{Z}\right)^2}{W} \tag{84}$$

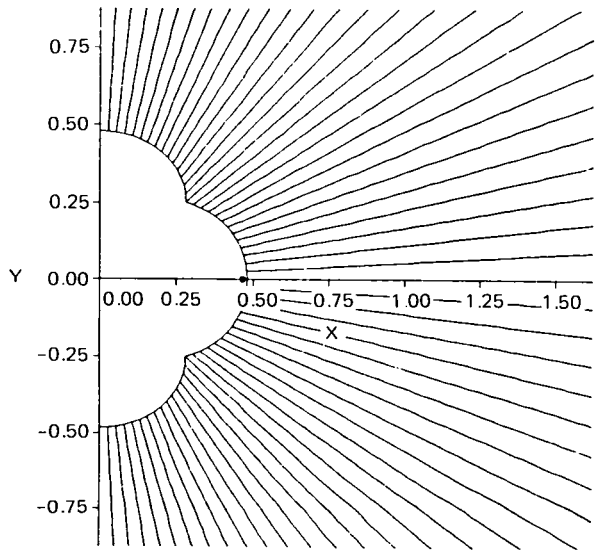
is applied. This causes the final mapped body to have the same radius in both the horizontal and the vertical axes. This last transformation is necessary for vertically elliptic bodies and wing-body cross sections. For a wing-body cross section (see Fig. 45), the wing is first mapped using transformation (83). The presence of the body will generate a mapped cross section that is vertically elliptic. Transformation (84) is then applied which results in a final mapped body as shown in Fig. 45b. The horizontal mapped radius becomes equal to the vertical radius of the body. Using transformation (84) improves the orthogonality of the grid around the body. Figure 45c shows a blowup of the mapped space body and radial grid lines at equal theta spacing. In this example, the wing-body juncture was not mapped. Hence, the grid lines are nearly orthogonal to the surface at the wing leading edge and centerline of the body. In the vicinity of the wing-body juncture, the lines are not nearly orthogonal. Figure 45d shows the final mesh generated by this procedure in the physical space. The grid appears to be partitioned in nature, the dividing line being the wing-body juncture. Near the body, the grid is body-like, and near the wing the grid is like the single mapped wing grid. This is a relatively good grid, being only slightly deficient in the wing-body juncture. The nonorthogonality, unless it is extreme, usually does no harm to the computation. The largest drawback to this grid is the abrupt change in



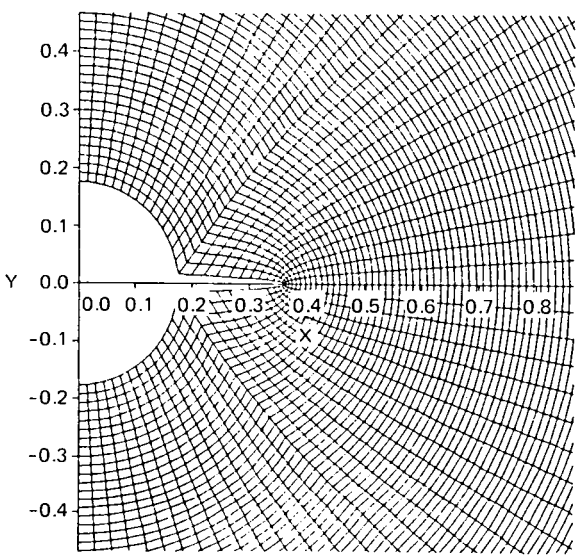
(a) PHYSICAL SPACE



(b) FINAL MAPPED SPACE BODY



(c) MAPPED SPACE θ = CONSTANT COORDINATE LINES



(d) COMPUTATIONAL MESH IN PHYSICAL SPACE

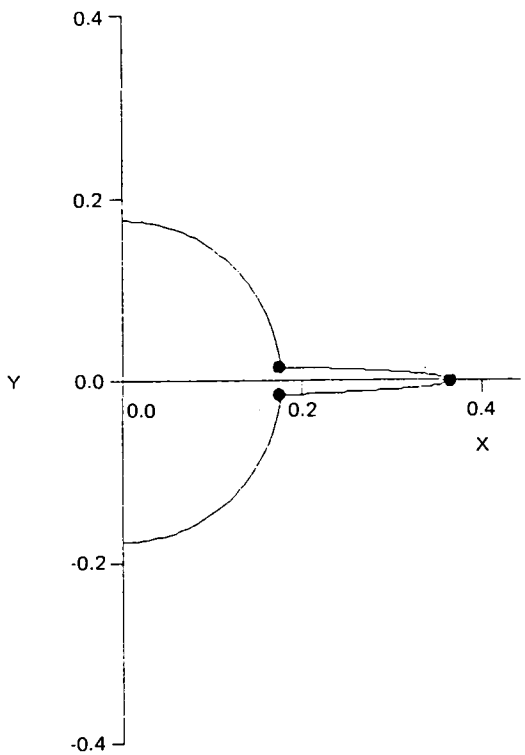
Fig. 45 Basic Wing-Body Mapping

shape of the circumferential grid lines around the body at the wing-bodyjuncture line. This abrupt change in slope can cause spurious oscillations in the solution.

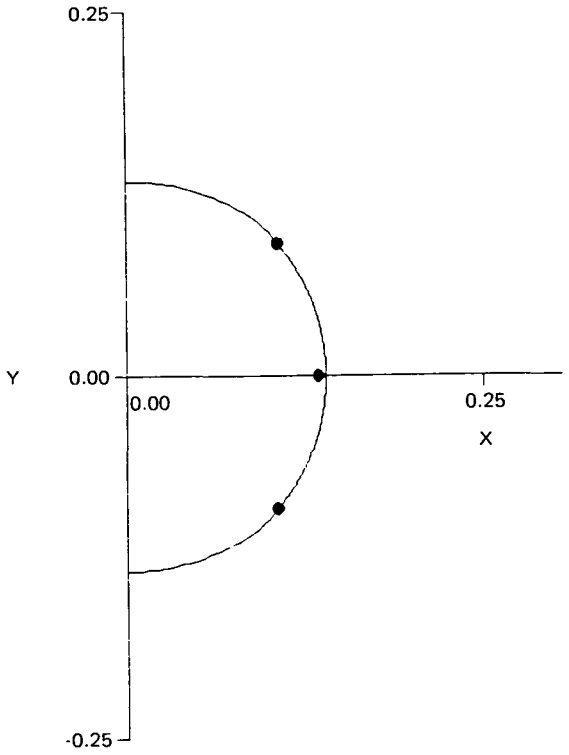
The orthogonality and curvature of the grid can be improved if the wing-body juncture points are also mapped using a form of Eq (83). Figure 46 shows an example of this grid generation for the same wing-body geometry. Figure 46a shows the locations of the three singularities or control points for the mappings. Two additional singularities are located just inside the wing-body juncture intersections. A power of 0.75 was used in Eq (83) for these two transformations that are applied after the wing mapping. Hence, a total of four mappings are applied to the body. Figure 46b shows the body shape in the final mapped space; the body is nearly circular in nature. A perfect orthogonal grid will be generated if the body is mapped to a perfect circle. Figure 46c shows the radial grid lines in relation to the body shape. In comparison to Fig. 45c, a much more orthogonal grid has been generated by mapping the wing-body juncture points. Figure 46d shows the resulting grid in the physical space. The discontinuous change in slope of the grid lines around the body has been almost eliminated, and the grid appears to be much more orthogonal.

A deficiency of the corner mappings can be seen in Fig. 46d. The corner mapping tends to eliminate grid lines in both the radial and circumferential directions in the vicinity of the juncture. Consequently, we have achieved a more orthogonal grid but have less resolution in the wing-body juncture. It also must be noted that the more mappings that are used, the greater the computational time. Whether or not it will be necessary to use corner mappings will depend upon the quality of the solution obtained on grids similar to Fig. 45.

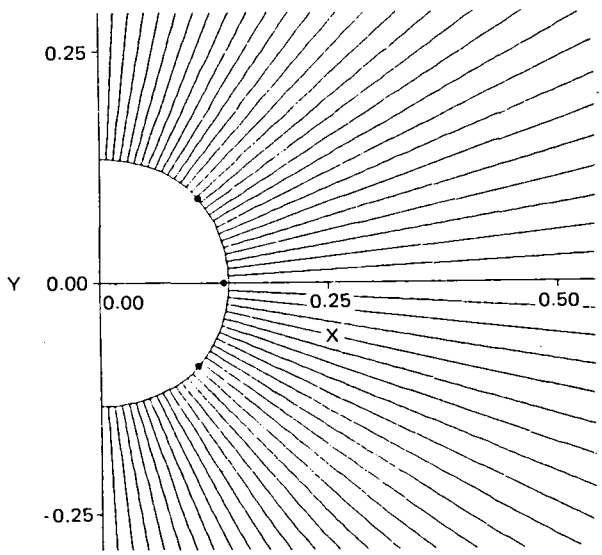
The deficiencies of the corner mappings were investigated to find a remedy, if needed, to the loss of resolution created by these mappings. The loss of radial resolution can be easily remedied by using the stretching already incorporated as an option in NCOREL that will cluster points near the body surface. No circumferential clustering is currently available for specified points around the body. A clustering function in the mapped θ coordinate was developed that is capable of clustering points at two specified theta locations in the mapped space.



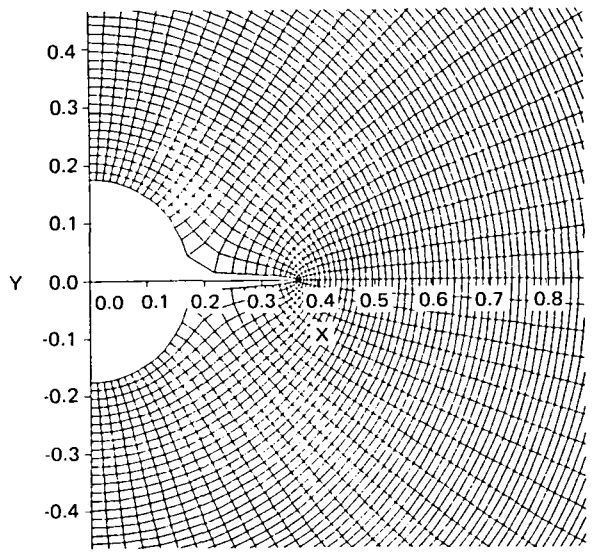
(a) GEOMETRY & SINGULARITY LOCATIONS IN PHYSICAL SPACE



(b) FINAL MAPPED SPACE BODY



(c) MAPPED SPACE θ = CONSTANT COORDINATE LINES



(d) COMPUTATIONAL MESH IN PHYSICAL SPACE

Fig. 46 Wing-Body Juncture Mappings

The general form of the function is as follows:

$$\theta = \pi \left[\frac{F(X) - F(-\frac{\pi}{2})}{F(\frac{\pi}{2}) - F(-\frac{\pi}{2})}\right] - \frac{\pi}{2} \tag{85}$$

where

$$F(X) = X + \beta \tanh\left[\alpha(X-X_1)\right] + \beta \tanh\left[\alpha(X-X_2)\right]. \tag{86}$$

In Eq (85) and (86), θ is the mapped space polar angle and X is the final computational coordinate. Introducing this function increases somewhat the complexity of the code. Prior to this transformation, $X = \theta$, and hence $X_\theta = 1$ and $X_{\theta\theta} = 0$; with the new transformation, $X \neq \theta$ and $X_{\theta\theta} \neq 0$. Therefore, these new transformation derivatives and terms must be introduced into NCOREL. In addition, B_θ, $B_{\theta\theta}$, $B_{\rho\theta}$, C_θ, $C_{\theta\theta}$, $C_{\rho\theta}$ ... etc cannot be computed as before because $B_\theta \neq B_X$. Instead, $B_\theta = B_X X_\theta$ and $B_{\theta\theta} = B_X X_{\theta\theta} + B_{XX} X_\theta^2$, where B_X and B_{XX} are now the numerical derivatives computed by the program.

In Eq (86), α and β are constants that control the degree of clustering. The clustering is achieved by a superposition of linear with hyperbolic tangent functions. X_1 and X_2 are the specified locations for the clustering. The superposition allows for a discrete clustering without disturbing the overall mesh distribution. Thus, the clustering can be made very discrete or smeared depending upon the choice of constants. Care must be taken in applying Eq (85) and (86) to eliminate the possibility of multiple values. To eliminate this possibility, θ_X must be constrained to be positive or zero. Hence, a parameter ε controls the slope of the function. ε must be positive or zero. A discrete clustering can be achieved if ε is near zero, and a more continuous clustering or a smeared clustering is achieved if ε is about 0.25 to 0.50. Since X_1 and X_2 appear in Eq (86), there is no guarantee that the resulting cluster will occur at the specified locations θ_1 and θ_2. To insure this, an iterative scheme (Newton method in two variables) is used along with the constant β in Eq (86). The constant β is iterated upon to insure the proper cluster locations. The constant α also controls the discreteness although the sensitivity to α is not great. Values of α around 1.0 or 1.5 are usually used, and ε is used to control the discreteness.

Figure 47 shows the same wing-body cross-section geometry as illustrated in Fig. 45 and 46. A value of $\varepsilon = 0.25$ was used to illustrate the clustering function on the grid generation. This value of ε spreads or smears out the clustering to yield a more smoothly varying grid. As shown in Fig. 47, mapping the wing and corners yields a nearly orthogonal grid. Further clustering the grid about the wing-body juncture points enhances the resolution in this area. The radial clustering could also have been implemented enhancing the radial resolution in the juncture area yielding a nearly perfect grid.

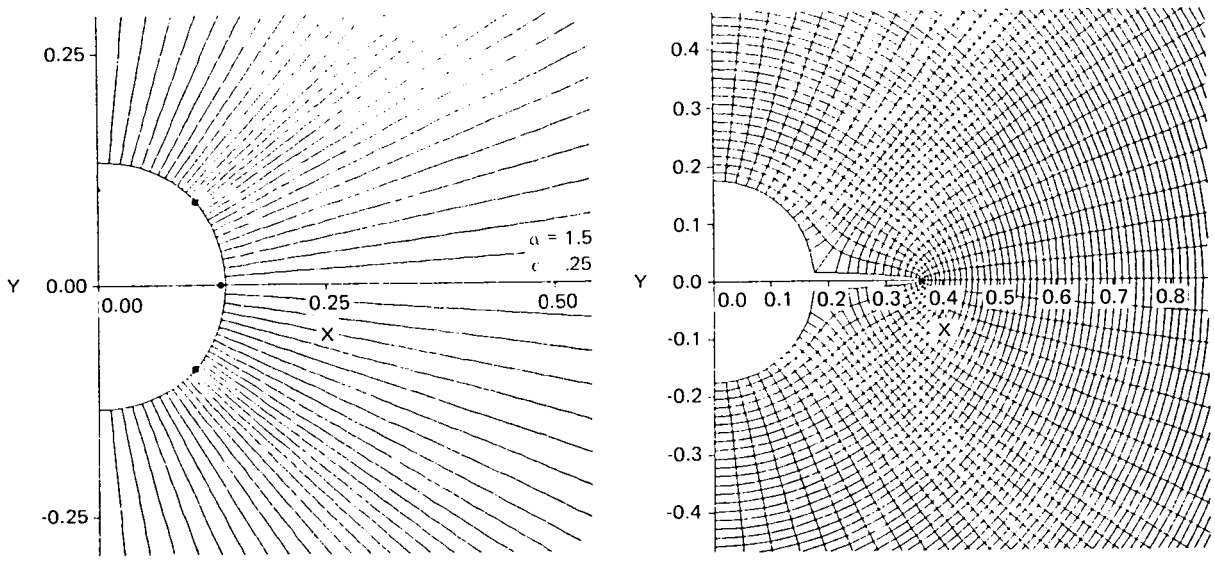
The grid generation package will be applied to a variety of cross sections to test its effectiveness. For the current applications, the corner mappings and clustering function will not be implemented. Figure 48a shows the physical mesh generated for a symmetric wing-body cross section with a vertically elliptic centerbody. The automatic mapping of the centerbody generates a reasonable grid for this cross section. Figure 48b further shows an asymmetric wing-body cross section. Both grids shown in Fig. 48 could be improved by mapping the corners and clustering.

Figure 49 shows another type of cross section, a two-finned body. The fins are located at ± 45°. The grid is a 57x57 mesh with two wing mappings implemented; the resulting grid is reasonable. For a computation, this grid is too sparse for a configuration with two wings. A 57x57 mesh is usually adequate for a single wing. A more realistic mesh for this type of configuration might be a 114x57 grid in order to resolve each wing. Figure 50 shows the same fins located at ±30° on the body; because of the proximity of the wing mappings, the grid between the two fins becomes somewhat sparse.

Figure 51 shows the same geometry as Fig. 49, but with a third fin added at $\theta = 0°$. In addition, the mesh has been increased for better resolution. It is interesting to see how the mappings generate a partition effect on the grid. Three wing mappings and the vertical ellipse mapping are used to generate this grid.

9.3 SAMPLE CONICAL COMPUTATIONS

Before embarking on a full implementation of a new grid generation package in NCOREL, some sample computations were carried out for conical flows



(a) THETA CLUSTERING IN MAPPED SPACE ABOUT WING-BODY JUNCTURE POINTS

(b) COMPUTATIONAL MESH IN PHYSICAL SPACE

Fig. 47 Example of Clustering in Theta

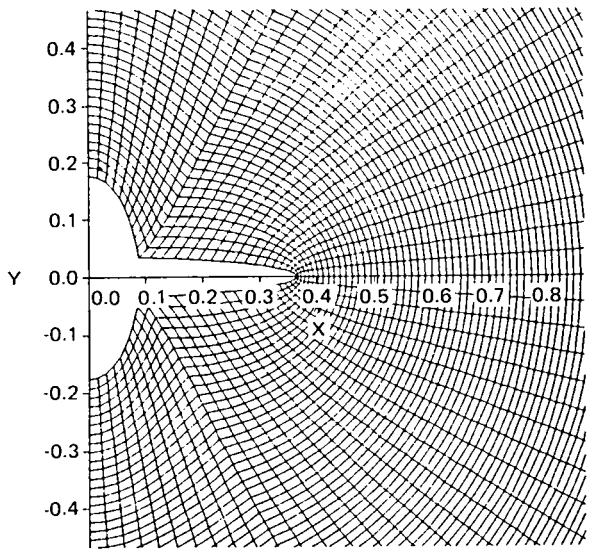


Fig. 48a Symmetric Wing-Body Mesh

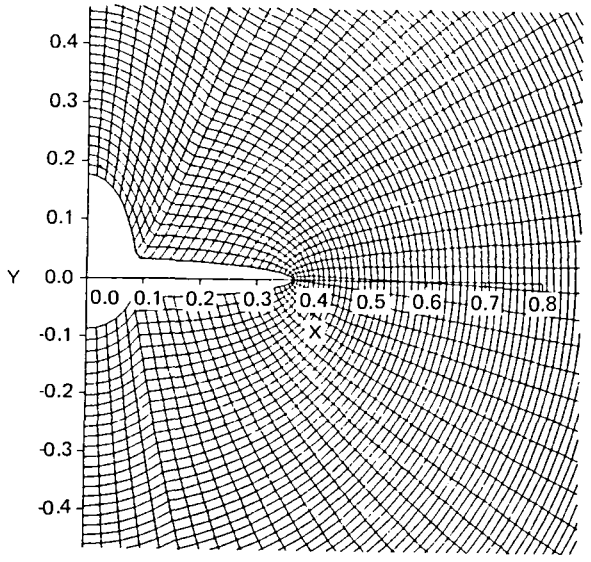


Fig. 48b Asymmetric Wing-Body Mesh

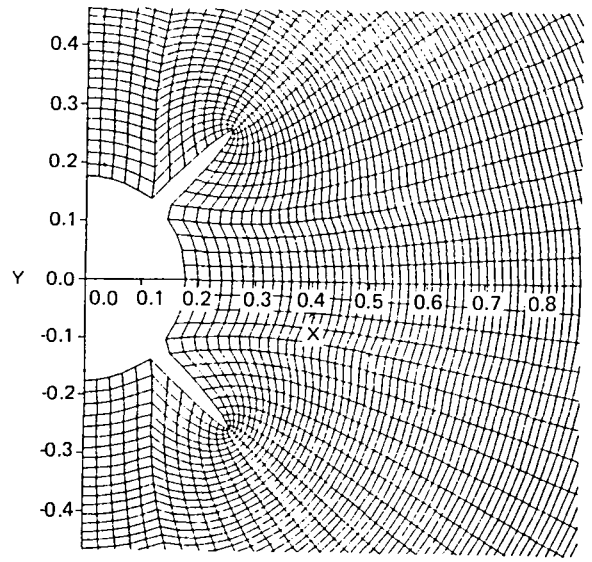


Fig. 49 Two-finned Body Cross Section ($\theta_F = \pm 45°$)

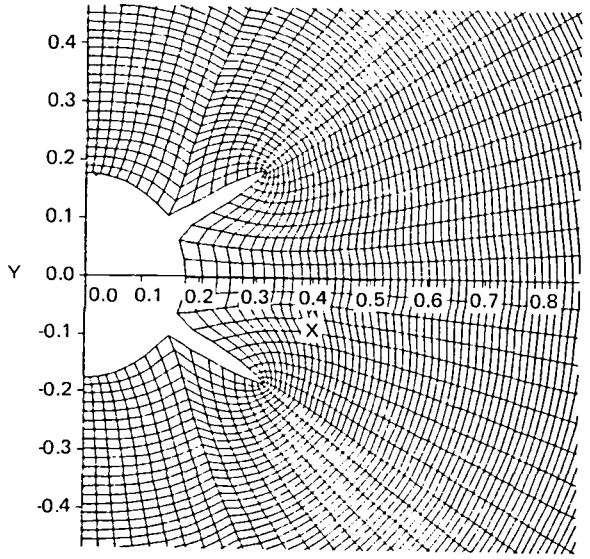


Fig. 50 Two-finned Body Cross Section ($\theta_F = \pm 30°$)

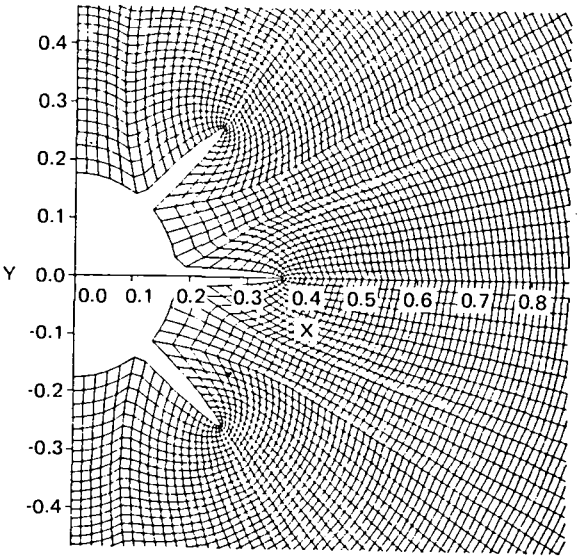


Fig. 51 Three-finned Body Cross Section

that will demonstrate the effectiveness of the new grids. These and other applications were presented in Ref 29.

Figures 52a and 52b show the final computational grids generated for a two- and three-finned body cross section at $M_\infty = 2$, $\alpha = 10°$. Since the bow shock is fit as an external boundary, the grid changes with the computation. The body spherical angle is 10° and the fins have 20° angles measured from the centerline or 70° sweep (subsonic leading edges). The fin geometry was generated by sinusoidal perturbations of the body radius and are single-valued in the physical polar angle. The computation was carried out on a 130x58 grid. In these computations no attempts were made to map the corners or internal angles. As a result, the grids generated have a partitioned appearance with rapid changes in slope in the circumferential grid lines. Figure 52a required three mappings and Fig. 52b was generated with four mappings. The singularities or control points of the mappings were placed near the leading edges of the fins. This results in a clustering of grid points in the vicinity of the fin tips which is necessary and essential to resolve large flow gradients in these regions. Figure 53 shows the computed crossflow sonic lines. Supercritical or supersonic crossflow regions develop on the leeward surfaces of all the fins. It is interesting to note that when the middle fin is added to the configuration, the supercritical region on the lower fin becomes much smaller. The presence of the supersonic crossflow regions indicate the existence of crossflow shocks on all of the fins. Figure 54 shows the isobar patterns for the two configurations. Large gradients are generated about the leading edge of each of the fins. Large gradients in crossflow Mach number also exist on the fins, as indicated by Fig. 55. The behavior of the flow field isobars and crossflow isomachs is relatively smooth, indicating that the abrupt circumferential grid changes are not having an adverse effect for conical flow computations. Figure 56 shows the surface pressure distributions for the two cross sections plotted as a function of the physical polar angle θ. The pressure distributions about the leading edge of the fins appear as spikes when plotted in this fashion. Some variation in body pressure occurs between the two fins. When the middle fin is added, the pressure distribution appears to become step-like, having various relatively constant pressure levels. Evidently, the reason for the shrinkage of the supersonic crossflow region on the lower fin, when the middle fin is added, is

the occurrence of another stagnation point on the lower surface of the middle fin. This causes the flow to shock earlier on the lower fin in response to the presence of the middle fin.

Figures 57 and 58 show the surface pressure distributions on the fins plotted as a function of the polar radius. Since the body angle is half the fin angle, the pressure distributions start at 0.5 and terminate at unity. When plotted in this fashion, the pressure distributions take on a different character and yield a clearer picture of the fin behavior. Figure 57 shows the pressure distribution for the two-fin configuration. A large supersonic crossflow region is shown on the lower fin terminating in a strong crossflow shock. The upper fin exhibits a smaller region of supersonic crossflow as indicated by the low-suction pressure plateau near the leading edge, but it also terminates in a crossflow shock. The general behavior of these fins is typical of wing or wing-body flows. When the middle fin is added (see Fig. 58), the supercritical region on the lower fin reduces significantly, but the strength of the crossflow shock does not diminish. The lift of the lower fin does diminish significantly, as indicated by the reduced ΔC_p. The mesh used yields about 42 points on each of the fins for the two-fin body and 34 points for each fin on the three-fin configuration. This type of mesh resolution is barely adequate for thin fins, and finer grids may be used in the future. Second-order accurate finite difference approximations were used for the supersonic crossflow regions for these configurations.

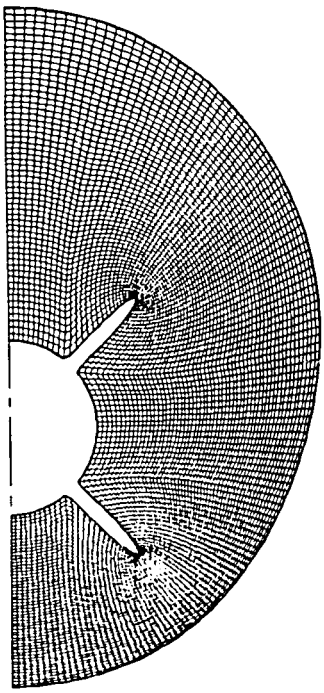


Fig. 52a Computational Grid for a Two-finned Body

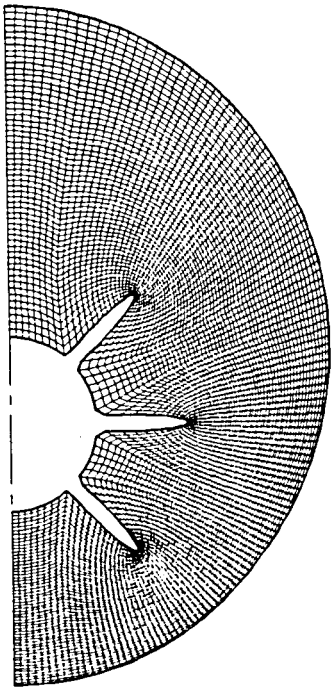
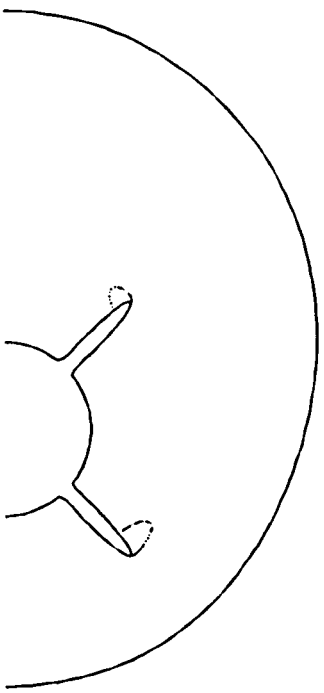
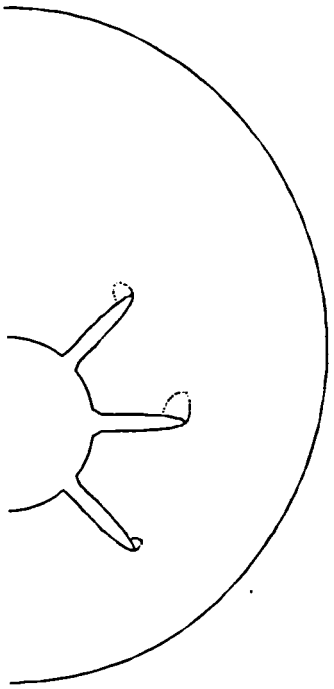


Fig. 52b Computational Grid for a Three-finned Body

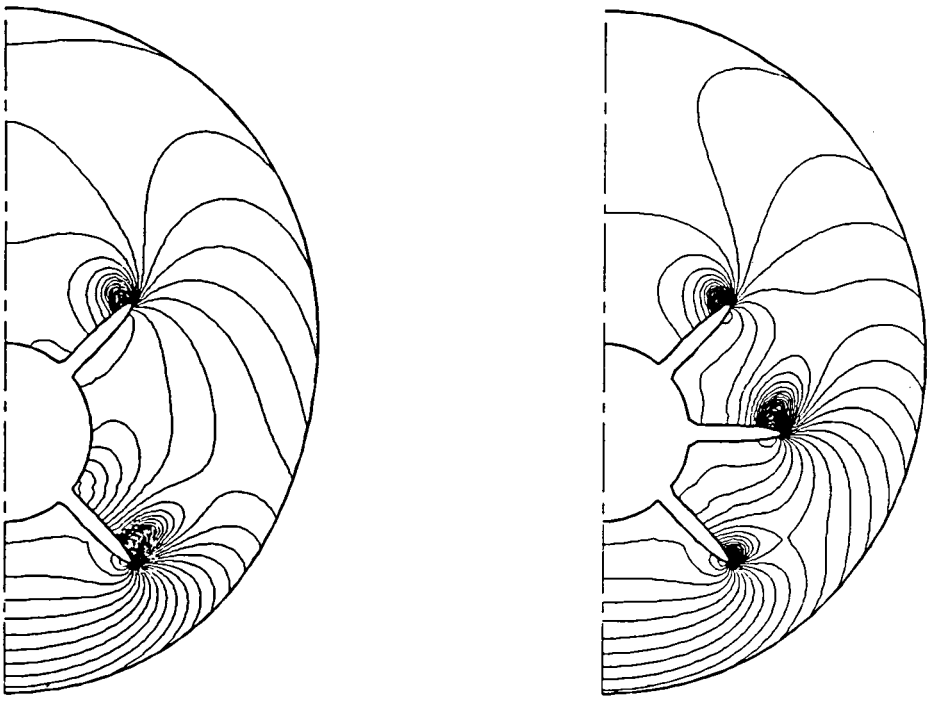


(a) TWO-FINNED BODY



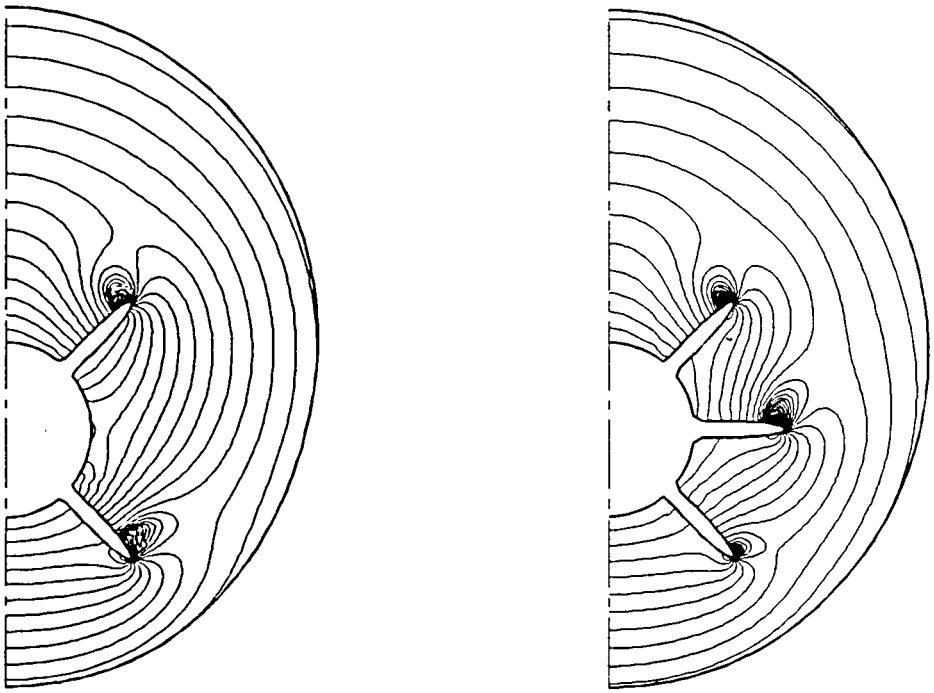
(b) THREE-FINNED BODY

Fig. 53 Computed Crossflow Sonic Lines ($M_\infty = 2.0$, $\alpha = 10°$)



(a) TWO-FINNED BODY (b) THREE-FINNED BODY

Fig. 54 Computed Isobar Patterns for Body-Fin Configurations ($M_\infty = 2.0$, $\alpha = 10°$)



(a) TWO-FINNED BODY (b) THREE-FINNED BODY

Fig. 55 Computed Crossflow Isomach Patterns for Body-fin Configurations ($M_\infty = 2.0$, $\alpha = 10°$)

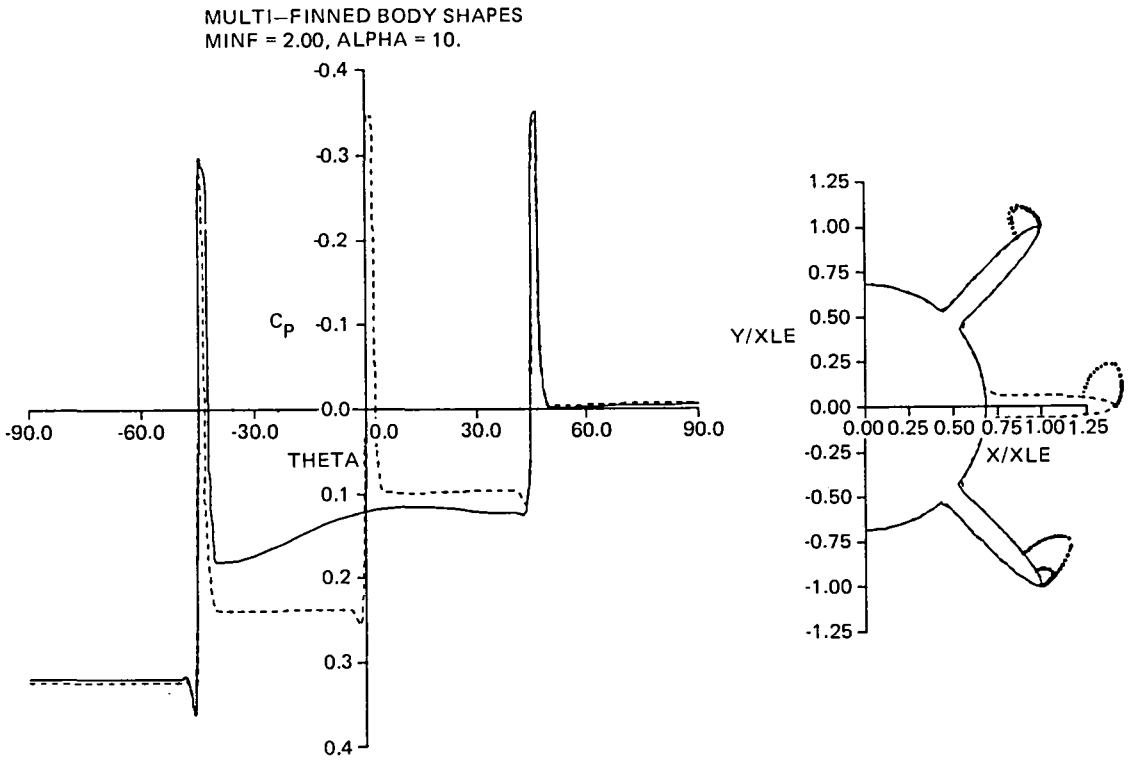


Fig. 56 Surface Pressure Distributions on Multi-finned Body Shapes as a Function of Physical Polar Angle

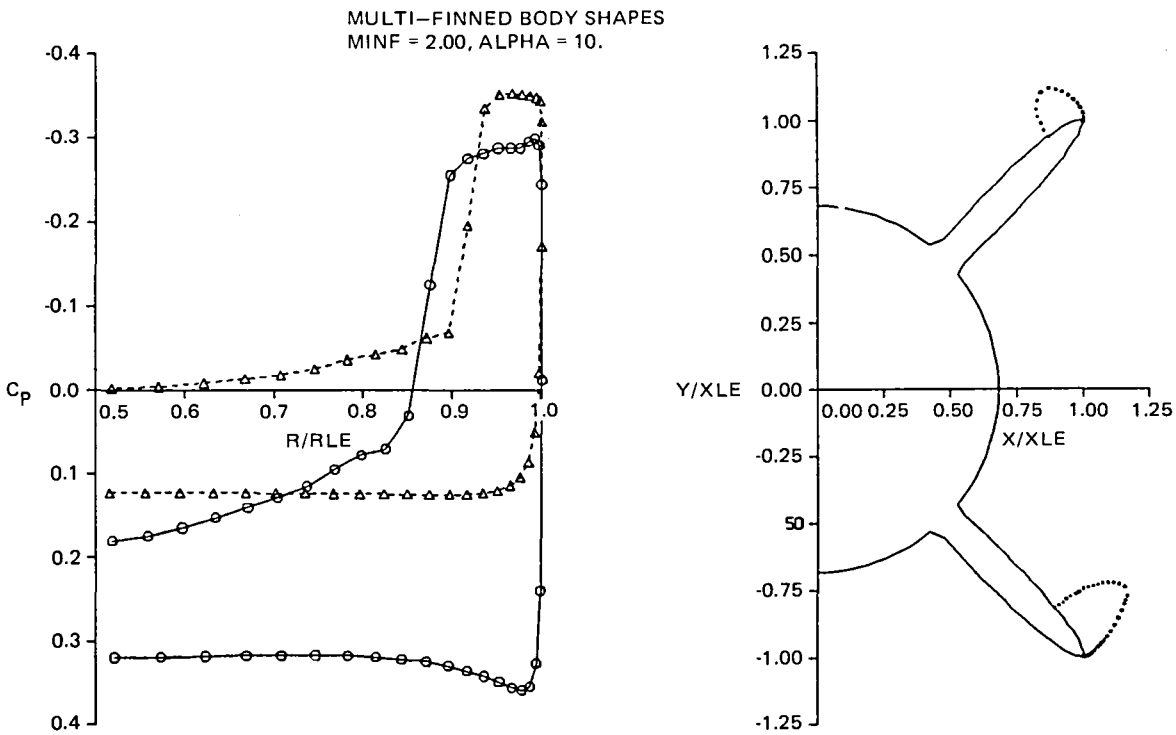


Fig. 57 Surface Pressure Distribution on Two-finned Body Shape as a Function of Physical Polar Radius

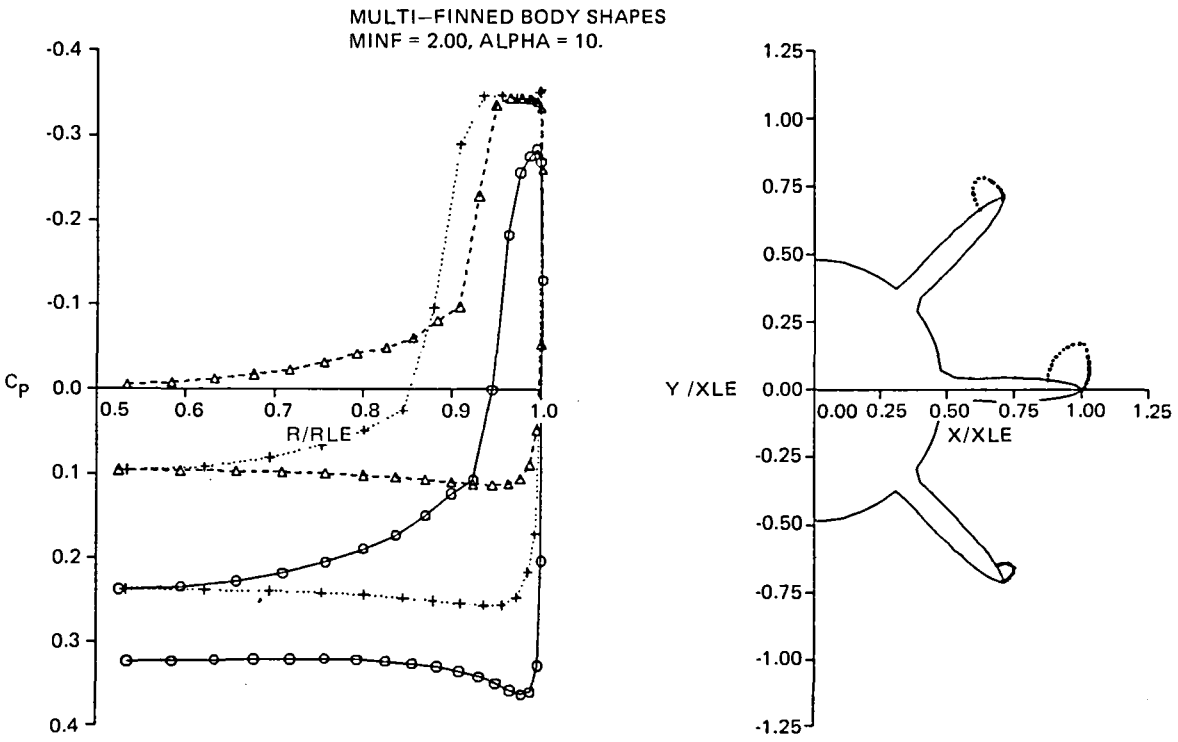


Fig. 58 Surface Pressure Distributions on Three-finned Body Shape as a Function of Physical Polar Radius

This Page Intentionally Left Blank

10. WING-BODY MODIFICATIONS

10.1 NUMERICAL SCHEME

Early versions of NCOREL were limited to relatively smooth or blended wing-body geometries. Real aircraft geometries do not always fall into this category. To be able to treat relatively arbitrary wing-body configurations, two aspects of NCOREL were investigated. First, a grid must be generated that can model abrupt wing-body cross sections. If the two intersecting sides of the wing and body geometries at the juncture are close to perpendicular, a single wing mapping is not adequate to resolve the juncture area and may lead to instabilities as the two coordinate directions of the mesh collapse in the wing-body juncture area. Hence, the grid generation procedure outlined in Section 9.2 must be implemented in 3-D to generate reasonable wing-body grids. Second, the numerics must be capable of capturing embedded wing shocks whose origin differs greatly from the apex or nose of the aircraft. Also, sweep angles as low as 40° may be of interest. These wing shocks will be highly supersonic in nature and will wrap around the leading edge of the wing. Some difficulties in capturing these shocks have occurred in the past with NCOREL, and it is the intent of this section to further modify the numerics in NCOREL to solve this problem.

The modifications included in NCOREL mainly apply to the three-dimensional terms constituting the governing equation (Ref 30). These are the terms that depend on R and vanish for conical flows.

For subsonic crossflow at a point (i,j) in the crossflow plane, the finite difference form of the residual can be written as

$$Res_{ij} = Resc_{i,j} + Resz_{i,j} \tag{87}$$

where

$$Resz_{i,j} = B_1 F_{XX} + B_2 F_{XY} + B_3 F_{YY} + B_9 F_{XZ} + B_{10} F_{YZ} + \cdots \tag{88}$$

The first part of Eq (87), i.e., $Resc_{i,j}$, refers to the terms that are independent of R. The second part, $Resz_{i,j}$, refers to the part of the

governing equation that depends upon the R coordinate and, hence, is nonzero only for three-dimensional flows. If the crossflow is subsonic, Eq (88) was originally treated using centered differences for the crossflow X and Y terms, and upwind differences for the Z-marching coordinate terms. For supersonic crossflow points, Eq (88) was formulated using upwind finite differences. The two forms of the Z residual (i.e., centered and upwind) were then smoothly switched from centered to upwind, commencing at sonic crossflow.

Difficulties arose in capturing embedded wing shocks whose virtual origin differed significantly from the apex of the spherical coordinate system at R = 0. The difficulty arises from the fact that the switching is linked to the crossflow Mach number whose definition is dependent upon the coordinate system. For example, for a wing emerging midway axially on a fuselage and at zero angle of attack, the computed crossflow would probably be subsonic and no switching would be in effect as the embedded wing shock emerges. For the same wing in a wing-alone computation, crossflow switching would occur across the wing generated bow shock. This undesirable feature occurs primarily due to the translation of the wing-centered coordinate system to a fuselage-centered coordinate system which affects the crossflow velocities. This problem is most apparent for less highly swept leading edge wings which emerge abruptly from the fuselage.

Hence, the original formulation of the three-dimensional terms in NCOREL has been modified to take into account the translation in coordinate systems.

A switch crossflow Mach number is defined as M_{CFo}, which is less than or equal to one. Then, for $M_{CF} < M_{CFo}$, Eq (88) is rewritten as

$$Resz_c = B_1 F_{XX_c} + B_2 F_{XY_c} + B_3 F_{YY_c} + B_9 F_{XZ_c} + B_{10} F_{YZ_c} + \ldots . \qquad (89)$$

where the subscript C refers to centered difference representations of the X and Y terms. For $M_{CF} > M_{CFo}$, Eq (88) is written as

$$Resz_u = \frac{B_1}{Q_{CF}^2}\left[U^2 F_{XX_u} + V^2 F_{XX_c}\right] + \frac{B_2}{Q_{CF}^2}\left[2|UV| F_{XY_u} + (Q_{CF}^2 - 2|UV|) F_{XY_c}\right]$$

$$+ \frac{B_3}{Q_{CF}^2}\left[U^2 F_{YY_c} + V^2 F_{YY_u}\right] + \frac{B_9}{Q_{CF}}\left[|U| F_{XZ_u} + [Q_{CF} - |U|] F_{XZ_c}\right]$$

$$+ \frac{B_{10}}{Q_{CF}} \left[|V| F_{YZ_u} + (Q_{CF} - |V| F_{YZ_c} \right] + \ldots . \tag{90}$$

Equation (90) represents an upwind biasing in the direction of the total crossflow velocity vector for points that exceed the switch Mach number M_{CFo}. For convergence stability, the two forms of the Z residual (Eq (89) and (90)) must be smoothly switched in a similar fashion as the original formulation. Equations (89) and (90) are written more compactly as,

$$Resz_c = R_{XY_c} + R_{XZ_c} + R_{YZ_c} + \ldots . \tag{91}$$

$$Resz_u = R_{XY_u} + R_{XZ_u} + R_{YZ_u} + \ldots . \tag{92}$$

where the subscript XY refers to the three second-order derivative terms in X and Y only. Therefore, a smooth switch commencing at the prescribed switch Mach number M_{CFo} can be written as, for $M_{Cf} > M_{CFo}$,

$$Res_z = \frac{M_{CFo}^2}{M_{CF}^2} R_{XY_c} + \left(\frac{M_{CF}^2 - M_{CFo}^2}{M_{CF}^2}\right) R_{XY_u} \tag{93}$$

$$+ \frac{M_{CFo}}{M_{CF}} (R_{XZ_c} + R_{YZ_c}) + \left(\frac{M_{CF} - M_{CFo}}{M_{CF}}\right)(R_{XZ_u} + R_{YZ_u}).$$

A similar smooth switching technique was used in the original formulation for the Z residual but was implemented at sonic crossflow conditions. The procedure outlined by Eq (89) to (90) allows switching to occur at an earlier crossflow Mach number, and reduces to the same switch if M_{CFo} is set equal to unity. Equation (90) is distinct from the earlier formulation in that it allows upwind biasing primarily in the crossflow velocity vector direction. In some sense, it rotates the finite difference equation in a similar fashion to a rotated difference scheme in conical or two-dimensional flows. Hence, Eq (90) does not exactly duplicate the original formulation even if the switch crossflow Mach number is set to unity. No consistent study of the differences between this formulation Eq (93) and the original numerical scheme has been carried out.

In addition to the above mentioned modifications, the coefficients of the tridiagonal matrix have been modified to duplicate the exact combined centered and upwind coefficients with the smooth switching. In the original formulation, two forms of the tridiagonal matrix were used reflecting either centered or upwind coefficients without the effect of the smooth switching. By including the effect of smooth switching on the coefficients in the tridiagonal terms, the tridiagonal terms become continuous and do not exhibit abrupt changes and lead to a more stable convergence.

10.2 AIRCRAFT COMPUTATIONS

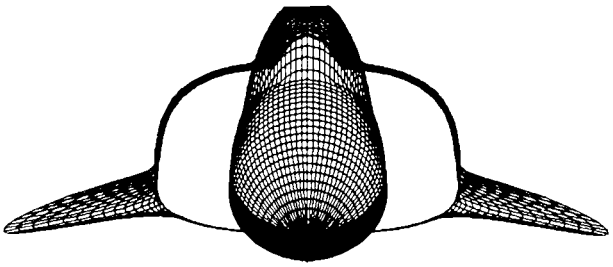
10.2.1 Grid Generation

The original single-wing conformal mapping method used in NCOREL has been modified to include the application of a series of Karman-Trefftz mappings. This type of analytic conformal mapping method was originally proposed by Moretti (Ref. 3) and applied by Marconi (Ref. 28) for Euler computations. The method has been applied successfully to a variety of cross-sectional shapes to generate grids for conical full potential flow computations as reported in Ref. 29. In this study, two of these mappings were used in conjunction with vertical shift mappings to generate reasonable grids about a wing-body inlet configuration. A horizontal slit wing mapping and a vertical slit body mapping in the symmetry plane is used for a general wing-body cross section. Both the body and wing can have camber. The shift mappings account for the body camber. As reported in Ref. 29, corner mappings could also have been implemented at the wing-body juncture. This would have required additional clustering in the circumferential direction about both wing-body juncture points. Since the corner mappings and clustering are not yet fully operational for three-dimensional flows in the code, the wing-body juncture geometry was faired to smooth out the wing-body intersection points. The aircraft geometry is comprised of piecewise analytic surfaces to represent the body, inlet, canopy and wing.

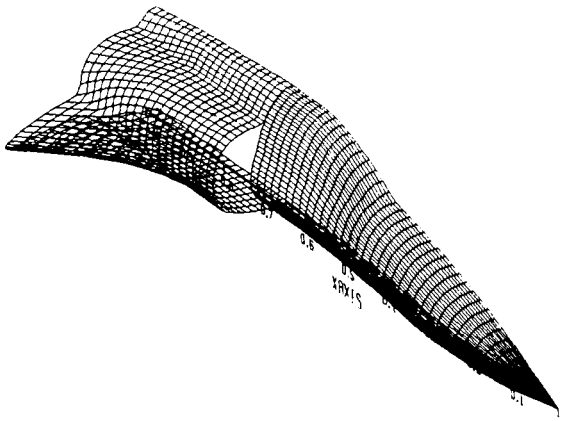
The mappings are applied in progression to a particular cross section. Only the wing mapping singularity location or control point need be specified as a function of the wing geometry. The cross section is first shifted to align the wing mapping singularity to the X-axis. The wing mapping is then applied and another shift takes place to center the cross section vertically. This shift typically removes the body camber. The vertical slit

mapping is then used to eliminate vertical eccentricity and results in a more orthogonal grid around the body. The resulting grids are fairly orthogonal on the wing and the body with the largest nonorthogonality occurring in the vicinity of the wing-body juncture. The application of the vertical slit mapping can be adjusted to cluster more points around the body or to improve body orthogonality. Care must be taken to try to insure continuous mapping derivatives in the marching direction. To do this the mappings are tied directly to the geometry. This doesn't always assure continuity as in the case of the canopy emergence. The grid generation is done automatically in the code with the user specifying only the control points for the wing mapping and one other parameter that adjusts the vertical slit body mapping. The computations to follow consisted of approximately 100 spherical marching planes and, hence, a hundred cross-sectional grids were generated automatically by the code during the marching computation. After mapping the body cross section the physical grid is generated by a shearing transformation. The shearing transformation assures that the body and outer bow shock boundaries are constant coordinate lines in the final computational space. Two types of grids are currently available from the shearing transformation. A standard grid that evenly subdivides the shock layer and a highly stretched grid that clusters points in the vicinity of the body surface. For the wing-body problem, the stretched grid was found to yield higher accuracy near the leading edge in the small layer between the body surface and embedded wing shock.

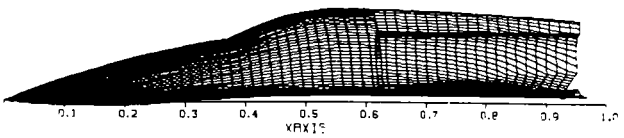
Figure 59 shows several views of the surface grid and geometry for the forward half of the configuration. The body has a large amount of camber in terms of nose droop to alleviate the canopy shock. The canopy gradually merges back into the fuselage to suppress the formation of a canopy recompression shock. The inlet is mounted on the side of the fuselage where two discontinuous geometry cross sections are defined with inlet off and inlet on. The wing shown in Fig. 59 has span-wise camber with a maximum camber angle of 20° at the leading edge. The wing emerges slightly downstream of the inlet station. The initial wing sweep varies rapidly until it reaches a minimum of 57° where upon it remains constant. The surface grids shown in Fig. 59 have 42 circumferential points.



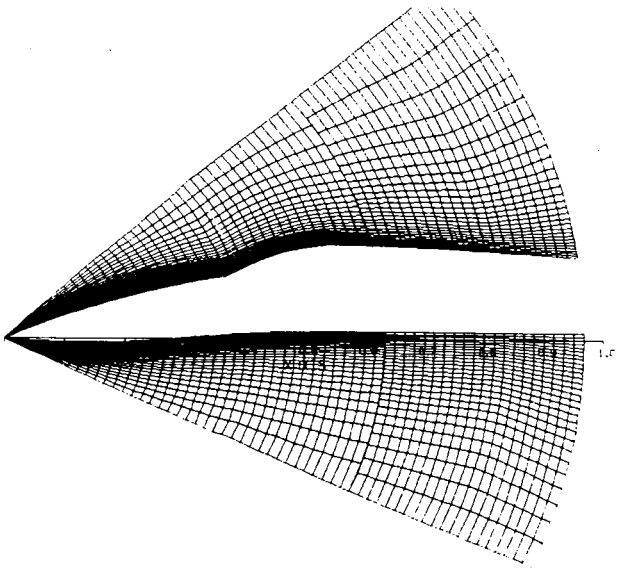
FRONT



ISOMETRIC



SIDE SURFACE GRID



SYMMETRY PLANE GRID

Fig. 59 Surface and Symmetry Plane Grids for Forward Half of Aircraft Configuration

symmetry. The discontinuity in the grid mapping at the inlet station is evident and is also apparent in the isometric view shown in Fig. 59c. Figure 59d also shows the radial stretched grid available to the user that clusters points in the vicinity of the body surface. Several variations of the crossflow grids are available to the user that are controlled by a couple of simple input parameters.

Figure 60 illustrates the effect of varying the length of the vertical slit body mapping on the body and juncture area grid. In Fig. 60b, the height or length of the vertical slit body mapping has been increased. This clusters more points around the body and improves body orthogonality in the juncture area at some expense to wing orthogonality.

As mentioned earlier, a discontinuous mapping is permitted at the inlet station. The cross sections overlap exactly except at the inlet face. The mass flow is captured by the inlet, new mappings and grid are computed, potentials and their derivatives are interpolated onto the new grid and the marching then continues.

Figure 61 shows a sampling of the cross-sectional spherical surface grids that are representative of the entire aircraft configuration. The outer boundary corresponds to the bow shock of the fuselage, which is fit using Rankine-Hugoniot shock jump conditions. The entire length of the configuration is approximately 30. The grids shown in Fig. 61 represent the stretched radial crossflow mesh for 58 circumferential by 30 grid. The computation was carried out on grids twice as fine as those illustrated in Fig. 61. At R = 6, the cross section is slightly elliptic in nature. The canopy has emerged at R = 7.5. The inlet station and mass capture occurs at R = 11.4. Both inlet-off and inlet-on grids are shown at this station. The wing has emerged by R = 15.0. The wing emerges more highly swept and then has a constant 57° sweep angle. At R = 18.0, the wing has fully emerged. The grid lines are clustered near the leading edge of the wing as a result of the wing mapping. The vertical slit body mapping causes the grid lines to remain fairly orthogonal around the body causing some loss in orthogonality on the wing surface. At further stations along the configuration, the wing mapping dominates. Since the number of points on a cross section remain constant, as the body becomes smaller relative to the wing, the body slit mapping has to be adjusted to retain enough points to define the body. In addition, the

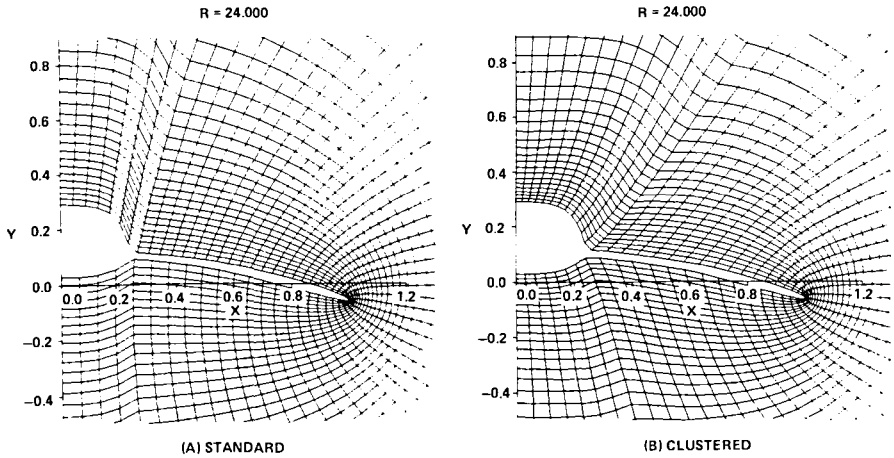


Fig. 60 Effect of Body Slit Mapping on Crossflow Grid

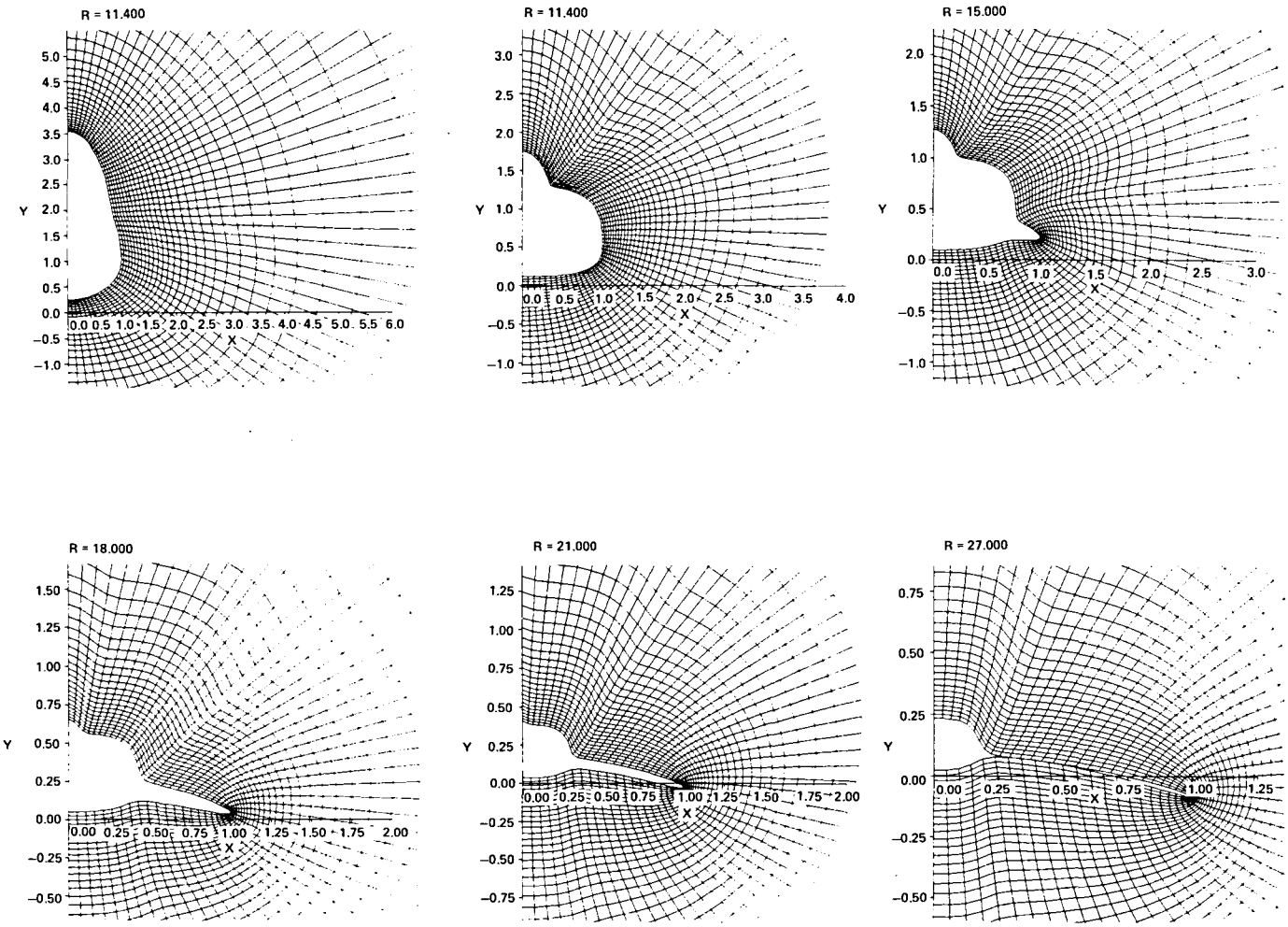


Fig. 61 Representative Sampling of Cross-Flow Plane Grids for Aircraft Configuration

geometry included a fairing of the wing-body juncture to try to retain a smooth grid and consistent definition of the juncture area. R = 27 shows a wing-body wake grid. The wake cut is represented as an infinitesimal slit where pressure and normal velocity continuity is imposed by computing values for the jumps in potential across the slit.

Figure 62 shows a planform projection of the surface grid lines. The actual trailing edge of the wing is shown by the dashed line. The wake is not distinguishable in the grid, except that the thickness of the wing vanishes in the wake. Special wake boundary conditions are imposed for grid points that lie on the wake cut of the wing. Hence, the wing trailing edge is approximated by the nearest grid point to the wake that lies on the wing surface. No attempt was made to clip the wing tips, or turn the leading edge, parallel to the freestream direction. The tip of the wing was extended to a point.

Figure 63 shows an overall composite isometric view of the basic grid topology. The computations were carried out on several grids including both the standard and stretched grids with various degrees of body clustering achieved with the vertical slit body mapping. As shown in Fig. 63, the axial deformation of the grids is quite large. The number of crossflow stations was held fixed at approximately 100 in order to resolve the large axial variations of the geometry and resultant flow field. Several crossflow grids were computed, the finest of the grids being 114 circumferential by 58 highly stretched in the radial crossflow direction. The results to follow were computed on these fine grids.

Figure 64 shows computed isobars projected on a side view of the isolated fuselage, with a side-mounted inlet at $M_\infty = 2.0$, $\alpha = 8^\circ$.

Figure 65 shows a planform view of the external isobars for the entire configuration. Since the configuration is highly three-dimensional and has both body and wing camber, Fig. 65 corresponds to the isobars computed on a three-dimensional grid surface near the wing leading edge or the middle of the circumferential grid. There is a slight break in the isobars at the inlet. This is due to the change in the mapping or grid at the inlet stations. The grid surface before and after the inlet do not match at the inlet. Since the inlet is capturing the flow, isobars disappear or are swallowed by the inlet. The inlet shock is indicated somewhat in this view. The strong

embedded wing shock is clearly visible in this view. The wing shock is somewhat detached from the wing leading edge.

Figure 66 shows the external isobars for the complete configuration in the two symmetry planes. The canopy shock is quickly attenuated by the large expansion aft of the canopy. In the windward plane, the inlet shock appears quite strong. The shock downstream of the inlet which appears in both leeward and windward planes is the wing shock. In the windward plane, the inlet and wing shock merge to form a single shock.

Figure 67 shows the projected isobars on the surface of the configuration in plan view. As mentioned earlier, the grid extends into the wake and is only distinguished by the different boundary conditions and the computation of a jump in potential for grid points that lie on the wake cut of the wing. Figure 67 includes the portion of the wing grid and isobars that lies in the wake cut of the wing. The wing crossflow shock is evident and has its origin from the initial wing emergence. The trailing edge shock on the upper surface and trailing edge expansion on the lower surface is also clearly visible delineating the trailing edge of the wing. The fuselage or body is extended into the wake and the origin of the trailing edge shock occurs right from the intersection of the wing trailing edge and fuselage. One of the more interesting features of Fig. 67 is the interaction of the crossflow shock on the upper surface of the wing with the trailing edge shock. In inviscid flow, the crossflow shock can not terminate. Instead, it interacts with the trailing edge shock of the wing and passes through it. This is clearly shown in Fig. 67 where the crossflow shock is shown on the upper surface of the wake. Its appearance on the lower surface of the wake indicates that it passes through the wake out the windward side. The gradients that exist in the wing-body juncture area indicate that a crossflow stagnation point may persist in this area. In other words, the incoming streamlines on the upper surface probably run along the wing-body juncture and may not run over the surface of the body to the centerline.

Figure 68 shows the computed axial pressure distribution on the upper and lower surface centerlines at $M_\infty = 2.0$, $\alpha = 8°$. The apex of the configuration is dropped at 10°. A parabolic camber is used for the body which vanishes at approximately where the canopy reaches maximum height. As a result, the angle of attack is almost completely negated on the fuselage, as indicated by the

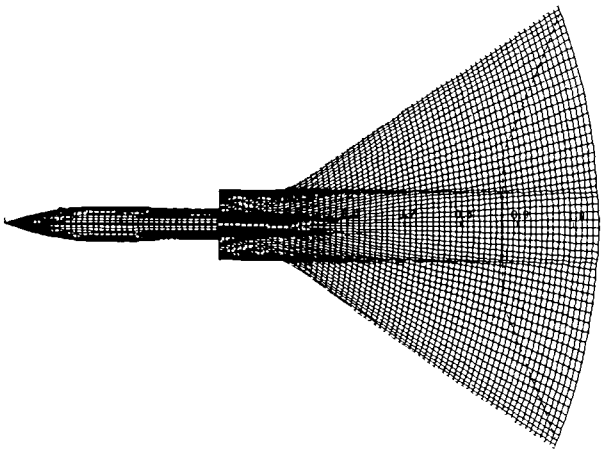


Fig. 62 Plan View of Axial Grid Lines Showing Trailing Edge Definition

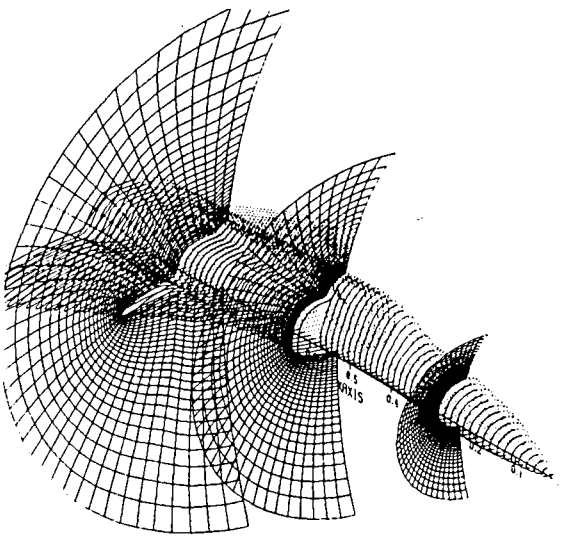


Fig. 63 Isometric View of Overall Grid Topology

Fig. 64 Side View of Isolated Fuselage Isobar Pattern at $M_\infty = 2.0$, $\alpha = 8^\circ$

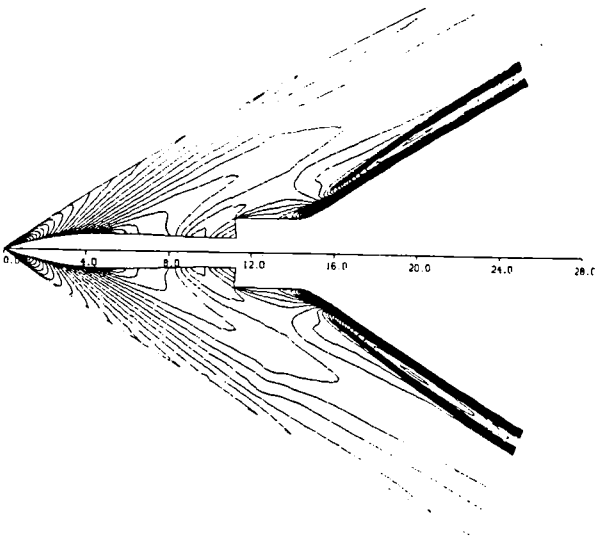


Fig. 65 Plan View of External Isobar Pattern Near Midsurface, $M_\infty = 2.0$, $\alpha = 8^\circ$

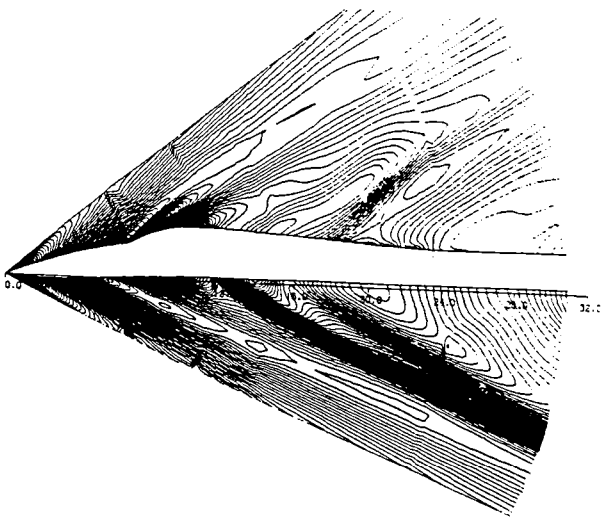


Fig. 66 Symmetry Plane External Isobar Pattern, $M_\infty = 2.0$, $\alpha = 8^\circ$

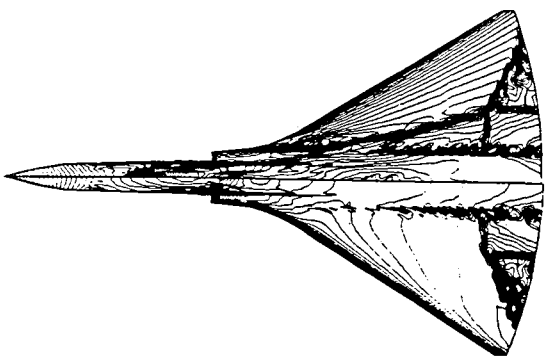


Fig. 67 Plan View of Surface Isobar Pattern, $M_\infty = 2.0$, $\alpha = 8^\circ$

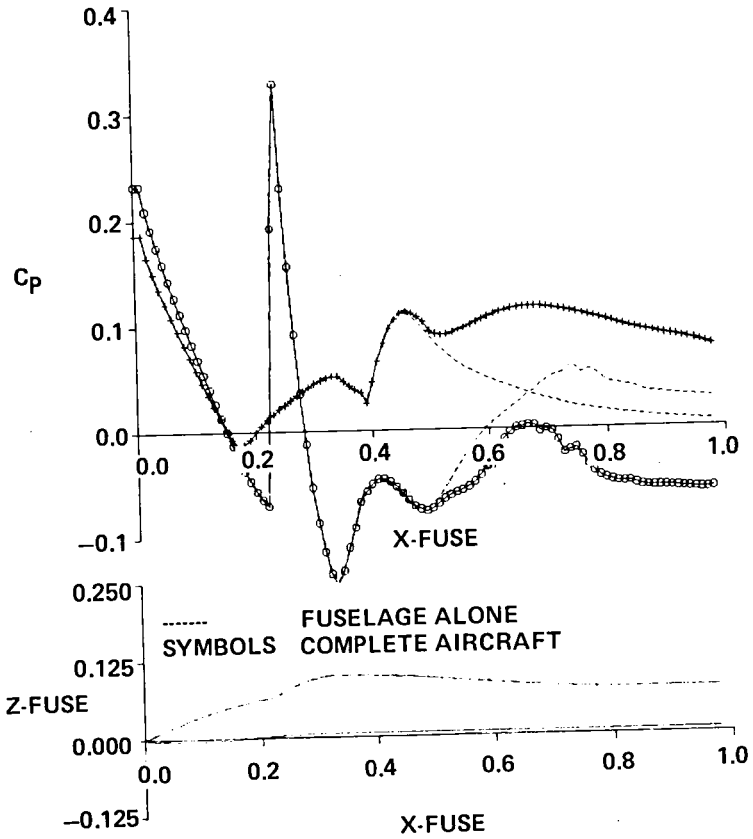


Fig. 68 Symmetry Plane Axial Surface Pressure Coefficient Distributions at $M_\infty = 2.0$, $\alpha = 8^\circ$

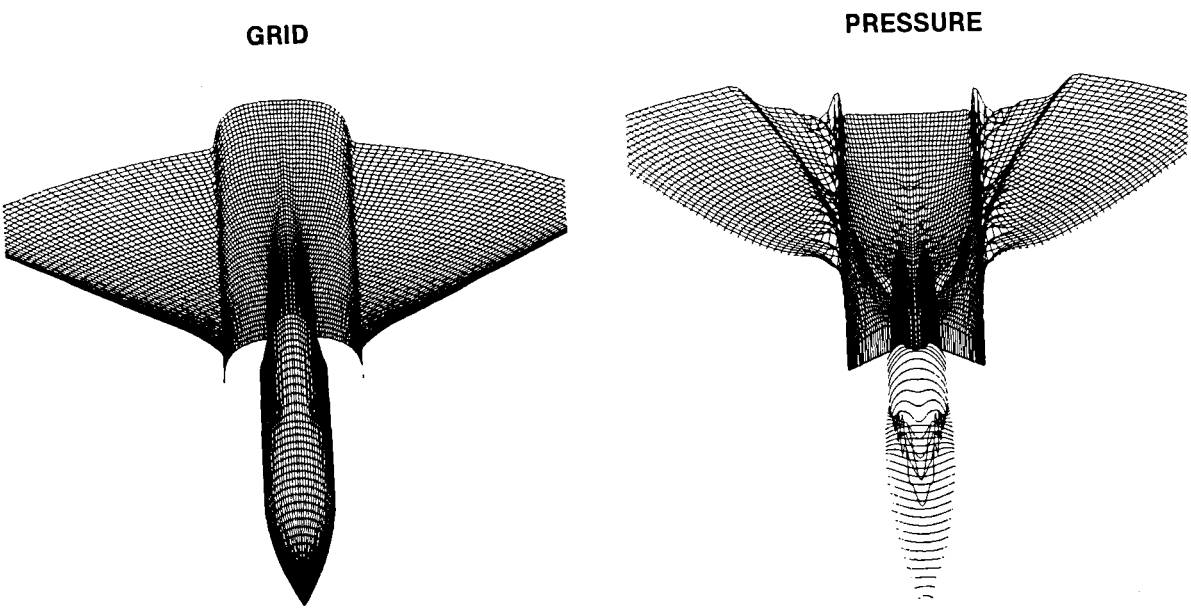


Fig. 69 Three-dimensional View of Upper Surface Grid and Pressure Coefficient Distributions at $M_\infty = 2.0$, $\alpha = 8^\circ$

pressure distributions. The dashed lines represent the fuselage inlet alone configuration. The symbols represent the complete configuration with the cambered wing. Obviously, for supersonic flow, the pressures do not change until the wing emerges from the fuselage. The addition of the wing dramatically increases the lower surface centerline pressures and decreases the upper surface pressures. The canopy shock followed by an even larger canopy expansion is evident in the upper surface pressures. The inlet shock is indicated in both the upper and lower surface pressures.

Figure 69a shows a three-dimensional view of the actual upper surface fine grid used for this computation; 114 circumferential points were used for the fine grid. Even with this number of grid points the wing-body juncture area was still difficult to define. A cross sectional smoothing was used in the geometry to slightly fair the corner. In Fig. 69a, some wiggles or inconsistencies in the juncture geometry in the axial direction in the vicinity of the wing-body area are still evident.

Figure 69b shows the same view for the entire set of upper surface pressure distributions. An initial pressure plateau and leading edge shock develops as the wing first emerges. Sharp gradients exist and persist throughout the wing-body juncture area. As the wing sweep diminishes, the leading-edge pressure plateau quickly disappears. As the cambered wing sweep becomes a constant at 57°, a gradual expansion occurs on the leeward surface, culminating in a crossflow shock well inboard on the wing surface.

Figure 70 shows a sampling of the crossflow isobars computed for the entire configuration at $M_\infty = 2.0$, $\alpha = 8°$. At R = 13.2, the inlet shock has developed and appears to emanate from both the upper and lower surface shoulder regions. On the lower surface, it surrounds the fuselage; while on the upper surface, it appears to attach to the remnants of the canopy. At R = 16.2, the wing has emerged. At R = 18 and further downstream, the embedded wing shock is clearly indicated which wraps around the leading edge. The wing shock quickly attenuates on the lee side of the configuration and merges with the inlet shock in the lower plane. A crossflow shock develops on the lee surface of the wing very close to the wing body juncture. At R = 22.2 and further downstream, the crossflow shock has moved outboard of the wing-body juncture. A stagnation region appears to exist in both lee and windward wing-body juncture points. At R = 28.2, only a small

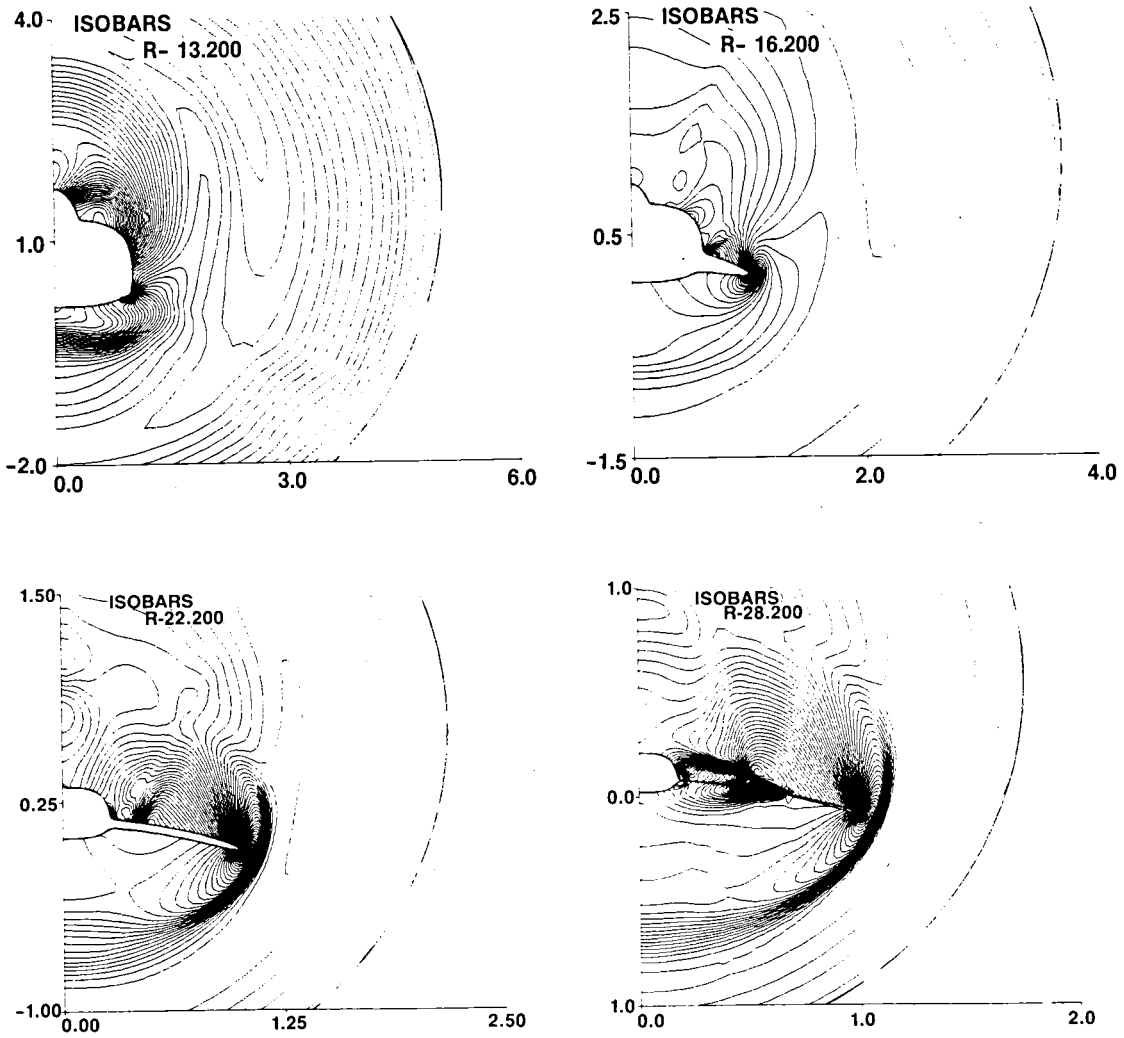


Fig. 70 Representative Sampling of Crossflow Plane Isobar Patterns at $M_\infty = 2.0$, $\alpha = 8^\circ$

piece of the wing remains in the outboard section of this wing-body wake cut. A trailing edge shock emanates from the upper trailing edge of the wing. The trailing edge shock appears to attach to the side of the fuselage. A lower surface trailing edge expansion exists which interacts with the cross flow shock that cuts through the wake cut. The shock interaction that occurs between trailing edge shock, expansion and crossflow shock appears to be quite complicated and confined to a region very close to the surface of the cross section. Even with the fine grid used, it is questionable whether this interaction has been adequately resolved.

Figure 71 shows a sampling of the detailed surface pressure distributions for the entire configuration at $M_\infty = 2.0$, $\alpha = 8°$. At R = 16.8, the wing has fully emerged. A crossflow shock exists on the upper surface of the wing which immediately compresses or stagnates in the wing-body juncture. The flow then expands around the upper shoulder region. This double compression persists on the upper surface as indicated by the pressure distributions further downstream on the vehicle. First, the flow compresses through a crossflow shock and then further compresses or shocks to the wing-body juncture. This behavior appears to be still somewhat unresolved even with the fine grids used for these computations. The wing-body juncture behavior also appears to be very sensitive to the degree of geometry fairing used in this region. The pressure distribution at R = 26.7 is indicative of the type of surface pressures obtained for a wing-body wake cut. The pressures are matched along the wake cut. The trailing edge of the wing shows an upper surface shock, lower surface expansion behavior typical of supersonic trailing edges at angle of attack.

A flat wing was also computed using the same fuselage at $M_\infty = 2.0$, $\alpha = 8°$. Figure 72 shows some examples of the effect of the spanwise camber on the surface pressure distributions. The newly emerged wing is more highly swept and a larger expansion takes place on the upper surface for the flat wing in comparison to the cambered wing. This effect persists throughout the entire configuration since the spanwise camber was kept at a constant. The flat wing does have a stronger crossflow shock as indicated at both R = 18.9 and 24.0. The flat wing produces more lift at these conditions both on the upper and lower surfaces. The double compression caused by the crossflow shock and wing-body juncture is also readily apparent. The flat wing is also

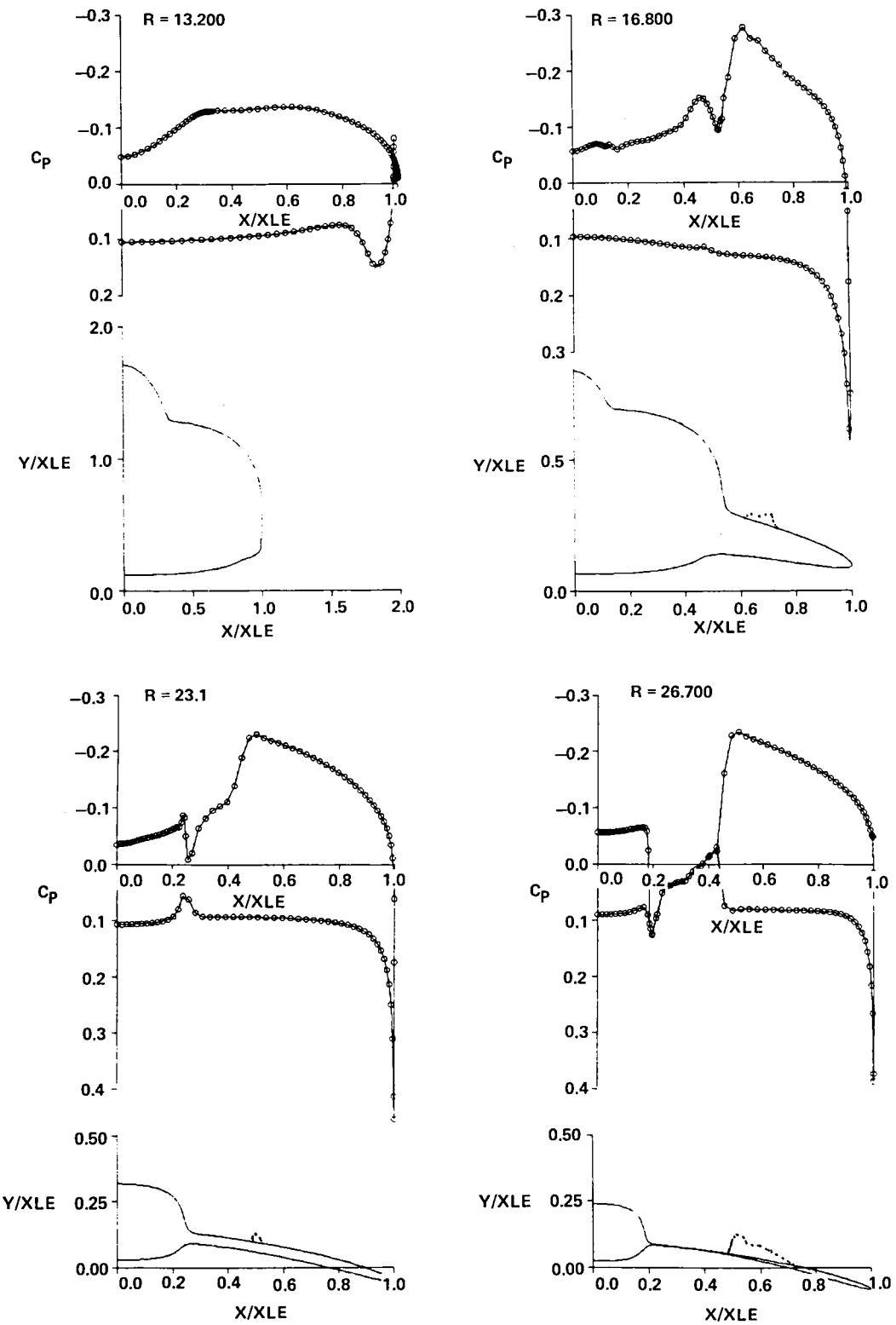


Fig. 71 Representative Sampling of Surface Pressure Coefficient Distributions at $M_\infty = 2.0$, $\alpha = 8°$

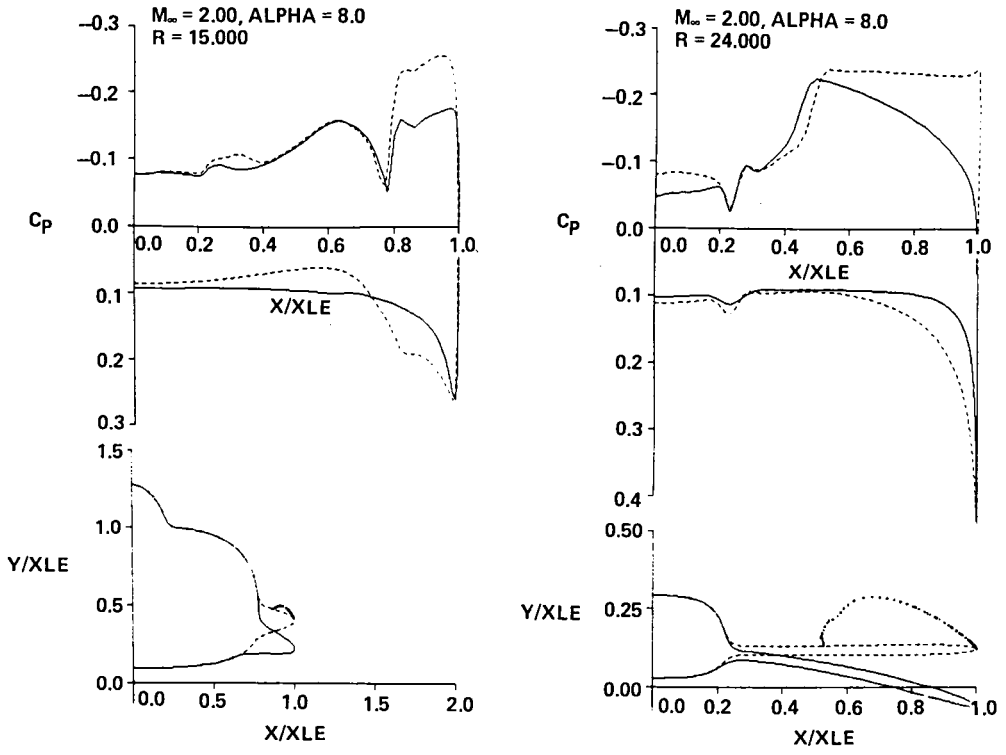


Fig. 72 Cambered Versus Flat Wing Effect on Surface Pressure Coefficient at $M_\infty = 2.0$, $\alpha = 8°$

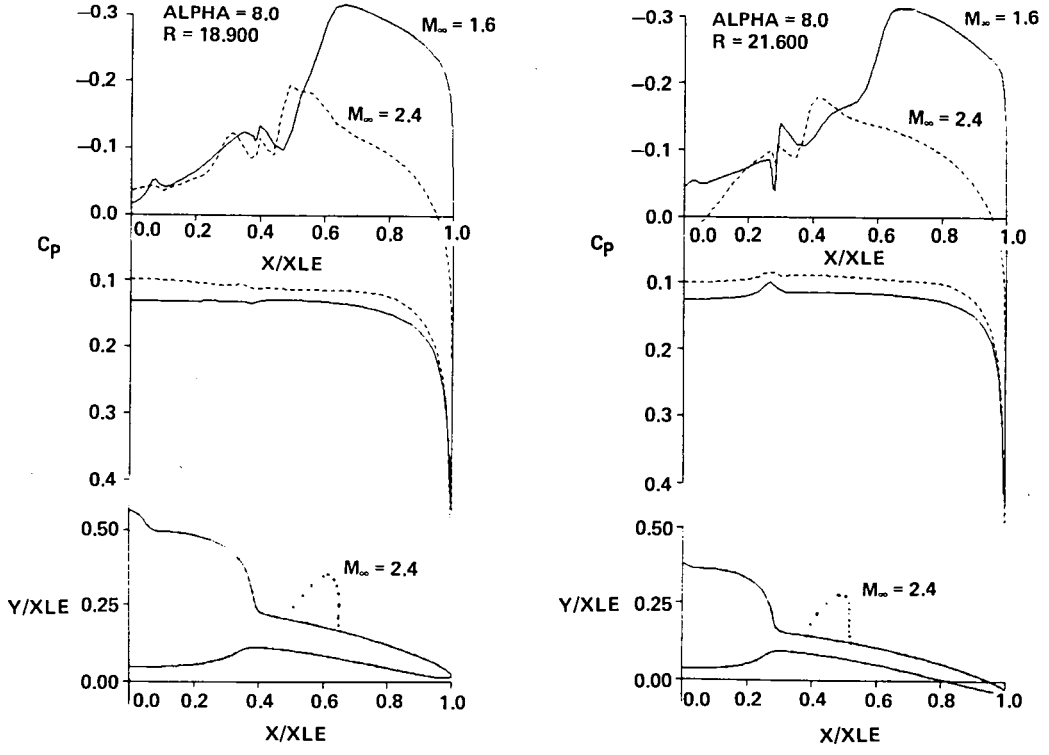


Fig. 73 Mach Number Effect on Surface Pressure Coefficient at $\alpha = 8°$

shock and wing-body juncture is also readily apparent. The flat wing is also more supercritical in terms of the crossflow Mach number as indicated by the crossflow sonic lines although the sonic crossflow condition has less meaning in three-dimensional flow.

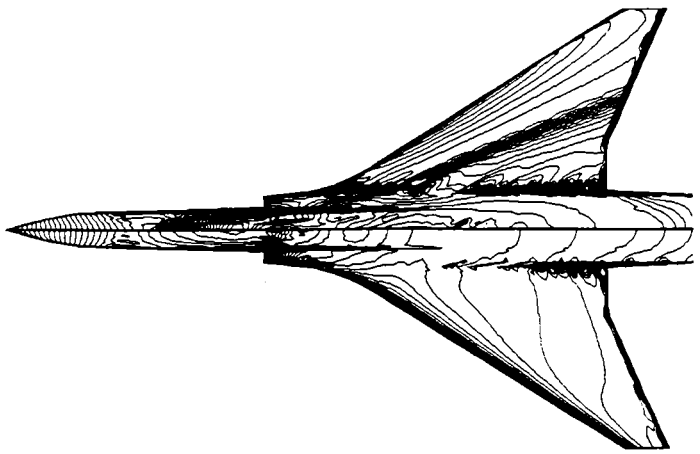
Figure 73 shows some examples of the effect of Mach number on the surface pressure distributions at $\alpha = 8°$ for the cambered wing configuration. The lower the freestream Mach number, the more subsonic the leading edge of the wing. Subsonic leading edge wings typically exhibit larger leeward expansions and usually tend to develop more apparent crossflow shocks. Figure 73 shows pressure distributions at $M_\infty = 1.6$ and 2.4. At $M_\infty = 1.6$, the entire wing has a subsonic leading edge; while at $M_\infty = 2.4$, except for the initial part of the wing, the leading edges are supersonic. As a result, larger expansions occur at the lower freestream Mach number. Crossflow shocks exist at both freestream Mach numbers. At $M_\infty = 2.4$, the crossflow shock is located well inboard near the wing body juncture. The crossflow shock at the lower Mach number moves outboard with increasing station. Figure 74 shows the computed isobar patterns on the surface planform at both $M_\infty = 1.6$ and 2.4. In this figure, the wake was not included in the isobar patterns. The crossflow shock is readily apparent at $M_\infty = 1.6$. At $M_\infty = 2.4$, the crossflow shock is barely distinguishable and lies well inboard near the wing-body juncture.

10.2.2 F-106B Computation

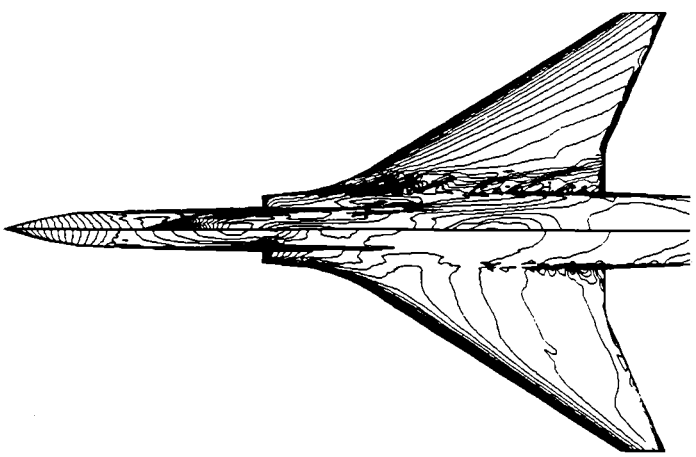
A Craidon or Harris Wave Drag data set for the geometry of the F-106B aircraft was available. The data set consists of a series of point wise descriptions for the fuselage cross section and a series of chordwise airfoils.

Figure 75 shows various views of the surface mesh model or cross sections generated by NCOREL using approximately 60 steps in the marching direction. The wing is mounted low on the side of the inlet. The grid changes discontinuously at the inlet interface. Two blocks of continuous meshes are used. One before the inlet and the other aft of the inlet with two non matching crossflow plane meshes at the inlet interface. The surface models shown in Fig. 75 were generated by a 42x30 grid in the crossflow plane.

This configuration was computed by NCOREL at $M_\infty = 1.5$ and for a range of incidence up to 12°. Forces and moments were also computed by NCOREL and compared to wind tunnel data. Figure 76 shows the resultant comparison. Figure 76a shows the C_L vs α curve. Excellent agreement with measured data is



$M_\infty = 1.60, \alpha = 8°$



$M_\infty = 2.40, \alpha = 8°$

Fig. 74 Comparison of Surface Isobar Patterns at $M_\infty = 1.6$ and 2.4 and $\alpha = 8°$

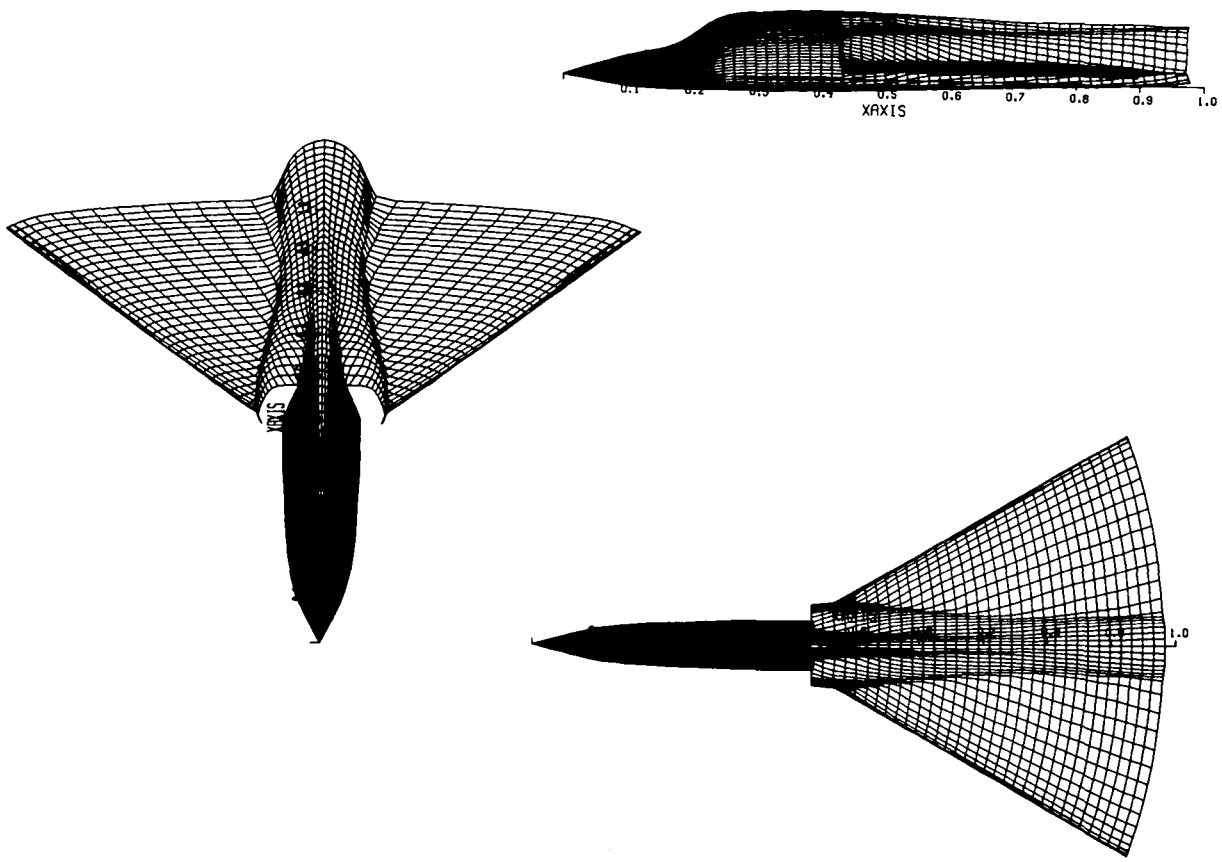


Fig. 75 Several Views of F-106B Surface Grid

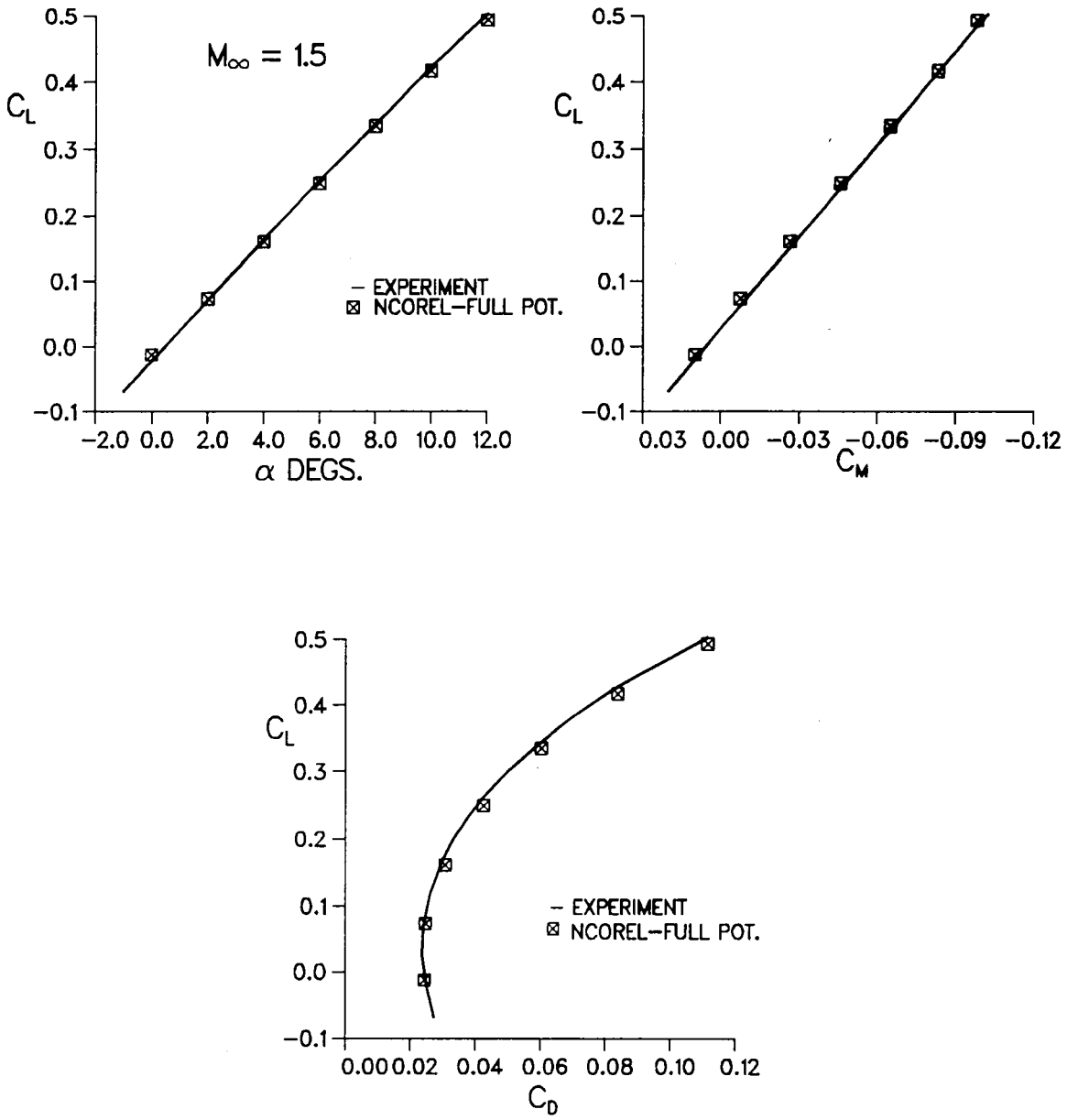


Fig. 76 Aerodynamic Forces and Moment for the F-106B Aircraft

shown up to a lift coefficient of 0.5. NCOREL predicts a slightly higher C_{L_α}. Figure 76b shows the Lift vs Pitching Moment comparison. Excellent agreement is also achieved for pitching moment. Figure 76c shows the drag polar or C_L vs C_b curve. The NCOREL results were plotted by extracting C_{D_0} from the experimental data. Very good comparison is also achieved for the shape of the drag polar. NCOREL predicts slightly less drag at the higher lift coefficients.

11. REFERENCES

1. Grossman, B., "Numerical Procedure for the Computation of Irrotational Conical Flows," AIAA J., Vol 17, No. 8, Aug 1979, p 828-837.

2. Weatherburn, C.E., Differential Geometry of Three Dimensions, Vol 1, Cambridge, England: University Press, 1955, p 167-173.

3. Moretti, G., "Conformal Mappings for the Computation of Steady Three-Dimensional Supersonic Flows," in Numerical/Laboratory Computer Methods in Fluid Mechanics, eds. A.A. Pouring, V.I. Shah, NY: ASME, 1979, p 13-28.

4. Grossman, B., and Siclari, M.J., "The Nonlinear Supersonic Potential Flow Over Delta Wings," AIAA J., Vol 20, No. 7, July 1982, p 924,.

5. Jameson, A., "Iterative Solution of Transonic Flow Over Airfoils and Wings, Including Flows at Mach 1," Comm. Pure Appl. Math., Vol 27, May 1974, p 283-309.

6. Murman, E.M. and Cole, J.D., "Calculation of Plane Steady Transonic Flows," AIAA J, Vol 9, Jan. 1971, p 114-121.

7. Siclari, M.J., "Supersonic Nonlinear Potential Flows with Subsonic Regions and Implicit Isentropic Shock Fitting," AIAA-81-1201, presented at AIAA 14th Fluid and Plasma Dynamics Conference, Palo Alto, CA, June 1981.

8. Siclari, M.J., "Supersonic Nonlinear Potential Flow with Implicit Isentropic Shock Fitting," AIAA J., Vol 20, No. 7, July 1982, p 924.

9. Siclari, M.J., "Computation of Nonlinear Supersonic Potential Flow Over Three-Dimensional Surfaces," Journal of Aircraft, Vol 20, No. 5, p 462, May 1983.

10. Siclari, M.J., "The NCOREL Computer Program for 3-D Nonlinear Supersonic Potential Flow Computations," NASA CR-3694, Aug 1983.

11. Siclari, M.J., "Approximate Factorization Schemes for 3D Nonlinear Supersonic Potential Flow," AIAA Paper No. 83-0376, presented at AIAA 21st Aerospace Sciences Meeting, Reno, NV, Jan 1983.

12. Peaceman, D.W. and Rachford, H.H., "The Numerical Solution of Parabolic and Elliptic Differential Equations," J. Soc Indust Ap Mech, Vol 3, 1955, p 28-41.

13. Ballhaus, W.F., Jameson, A. and Albert, J., "Implicit Approximate Factorization Schemes for the Efficient Solution of Steady Transonic Flow Problems," AIAA J., Vol 16, June 1978, p 573-579.

14. Holst, T.L., "Fast, Conservative Algorithm for Solving the Transonic Full-Potential Equation," AIAA J., Vol 18, No. 12, Dec 1980, p 1431-1439.

15. Baker, T.J., "A Fast Implicit Algorithm for the Nonconservative Potential Equation," Open Forum Presentation at AIAA 4th Computational Fluid Dynamics Conf, July 1979.

16. Baker, T.J., "Potential Flow Calculation by the Approximate Factorization Method," J. of Comp Physics, 42, p 1-19 (1981).

17. Catherall, D., "Optimum Approximate-Factorization Schemes for 2D Steady Potential Flows," AIAA Paper No. 81-1018, presented at AIAA Computational Fluid Dynamics Conf, June 1981.

18. Jameson, A., "Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid Method," AIAA paper No. 79-1458, presented at AIAA Computational Fluid Dynamics Conf, July 1979.

19. Pittman, J.L., Miller, D.S. and Mason, W.H., "Supersonic, Attached-Flow Wing Design for High Lift with Experimental Validation," NASA TP-2336, Aug 1984.

20. Mangler, K.W., and Smith, J.H.B., "Behavior of the Vortex Sheet at the Trailing Edge of a Lifting Wing," The Aeronautical J. of the Royal Aeronautical Soc, Vol 74, Nov 1970, p 906-908.

21. Shankar, V. and Szema, K.Y., "A Conservative Type-Dependent Full Potential Method for the Treatment of Supersonic Flows with Embedded Subsonic Regions," AIAA Paper No. 83-1887, presented at AIAA Computational Fluid Dynamics Conf, Danvers, MA, July 1983.

22. Siclari, M.J., "The Computational Treatment of Supersonic Wake Flows in NCOREL," AIAA Paper No. 85-0304, presented at the AIAA 23rd Aerospace Sciences Meeting, Reno, NV, Jan 1985.

23. Townsend, J.C., "Pressure Data for Four Analytically Defined Arrow Wings in Supersonic Flow," NASA TM-81835, Sept 1980.

24. Siclari, M.J., and Rubel, A., "Entropy Corrections to Supersonic Conical Nonlinear Potential Flows." Computer & Fluids, Vol 13, No. 3, 1985, p 337-359.

25. Zakkay, V., and Visich, M., "Experimental Pressure Distributions on Conical Elliptic Bodies at M_∞ = 3.09 and 6.0," Polytechnic Institute of Brooklyn, PIBAL Report No. 467, AFOSR TN59-10, March 1959.

26. Townsend, J.C., Collins, I.K., Howell, D.T., and Hayes, C., "Surface Pressure Data in a Series of Conical Forebodies at Mach Numbers from 1.70 to 4.50 and Combined Angles of Attack and Sideslip," NASA TM-78808, March 1979.

27. Landrum, E.J., "Wind-Tunnel Pressure Data at Mach Numbers from 1.6 to 4.63 for a Series of bodies of Revolution at Angles of Attack from -4° to 60°," NASA TMX-3558, Oct 1977.

29. Siclari, M.J., "Application of NCOREL to Conical Multi-Finned and Multi-Faceted Configurations," AIAA Paper No. 86-0441, presented at AIAA 24th Aerospace Sciences Meeting, Reno, NV, Jan 1986.

30. Siclari, M.J., and Pittman, J.L., "Application of NCOREL to Aircraft Configurations," AIAA Paper No. 86-1830, presented at the AIAA 4th Applied Aerodynamics Conf, San Diego, CA, June 1986.

31. Craidon, C.B., "A Computer Program for Fitting Smooth Surfaces to an Aircraft Configuration and Other Three-Dimensional Geometries," NASA TM-X-3206, June 1975.

32. Hall, J.F., Neuhart, D.H., and Walkley, K.B., "An Interactive Graphics Program for Manipulation and Display of Panel Method Geometry," NASA CR-166098, March 1983.

This Page Intentionally Left Blank

12. NCOREL USER'S MANUAL

12.1 INPUT PARAMETERS

Card No. 1 — Description Title of Geometry

Card No. 2 — Title Describing Freestream Conditions

Note: Cards No. 1 and 2 are alphanumeric title cards and must be less than 80 characters.

Note: The remainder of the input is in namelist format. Recommended default values of certain parameters are suggested.

Namelist Variable	Description
Freestream Conditions:	
EMINF	Mach Number ($M_\infty > 1$)
ALP	Angle of Attack (degrees)
GAMMA = 1.4	Ratio of Specific Heat
Grid Size Controls:	
KREF	Conical grid refinements. Maximum two refinements. Refinement given by 2(IC-1) and 2(JC-1). ($1 \leq KREF \leq 3$).
IC	Circumferential Mesh Points (IC-1), must be even number
JC	Radial Crossflow Mesh Points (JC-1), must be even number
ISYM = 0	(Currently not supported.)
Modified Potential Method:	
IENTRY = 1	The code will fit the bow shock isentropically or with the Rankine-Hugoniot shock conditions. With the Rankine-Hugoniot shock, the flow field pressures are corrected by the computed bow shock entropy. Only used in conjunction with the bow shock fitting option (IFIT=1).
IENTRY = 0 = 1	Isentropic bow shock Entropy corrections with R-H bow shock

Namelist Variable	Description
Iteration & Convergence Controls:	
Conical:	
KMAX (1) KMAX (2) KMAX (3)	Maximum number of iterations for each conical mesh up to a maximum of three meshes or two mesh refinements.
DMIN (1) = 1.E-2 DMIN (2) = " DMIN (3) = "	Convergence maximum residual tolerances for each conical grid.
ITCONC = 50	Convergence checking. Maximum residual is checked every ITCONC iteration for convergent sequence. If divergent sequence occurs, temporal damping is increased. Implemented in SLOR schemes only.
Nonconical (3D):	
KMAXNC	Maximum nonconical iterations.
DMINR = 1.E-2	Maximum residual tolerance for convergence at nonconical stations.
ITCONR = 25	Same as ITCONC but for nonconical stations.
Relaxation Scheme:	
IAF IAF = 0 (SLOR) IAF = 1 (AF1Z)	Optional switch for relaxation schemes. Line Relaxation Scheme. Approximate Factorization Scheme.
General Relaxation Parameters:	
Conical:	
W(1) = 1.5 W(2) = " W(3) = "	Over-relaxation parameter for both SLOR and AF schemes. Above the maximum of 1.7, the schemes become unstable.
Nonconical:	
OMEGR = 1.5	Over-relaxation parameter for potential at nonconical stations, must be $\leq$ 1.70.

Namelist Variable	Description

Approximate Factorization (AF) Parameters:

The following parameters are only used if IAF=1,

Conical:

NCYC(1) = 3 to 6 NCYC(2) NCYC(3)	NCYC is the number of cycles from the minimum to maximum acceleration parameter for each conical mesh.
AFMIN(1) = .5 AFMIN(2) AFMIN(3)	Minimum of factorization parameter.
AFMAX(1) = 2.0 AFMAX(2) ARMAX(3)	Maximum of factorization parameter. AFMIN and AFMAX govern the range of factorization parameters. Suggested ranges can be from 0.5 to 2.0 to 2.0 to 4.0, for example.

Note: Starting bow shock fitting on crude grids may require 6-8 cycles and AFMIN = 3.0, AFMAX = 6.0. Sensitivity mostly to AFMIN. The larger the number of cycles and the higher the range of AFMIN to AFMAX, the more stable but slower convergence rate.

Nonconical (3D):

NCYCR AFMINR AFMAXR	These parameters treated similar to conical parameters.

Temporal Damping:

EST = -0.1 = -0.25	Coefficient of Temporal Artificial Viscosity (-5.0 < EST < 0.0).
ESTMAX = -5.0	Maximum value of temporal damping. If divergence occurs, the temporal damping will be automatically set to this value.

Note: Very small EST values (<1.E-2) or zero should be used with AF schemes. This parameter does not affect the final solution only the stability and convergence rate. The larger the negative value of EST, the slower the convergence rate.

Namelist Variable	Description
Grid Cluster Parameters:	
NPOW	Two types of crossflow radial meshes are available: NPOW = 0; evenly spaced radial mesh; NPOW = 1; hyperbolic stretching which clusters grid points near the body surface.
BMAP1 = 1.0 BMAP2 = 0.5 to 1.0 RBMAP1 RBMAP2	Controls body mapping for wing-body configurations. Standard body mapping will occur for r<RBMAP1. Between r-stations, RBMAP1 and RBMAP2, body mapping constant will vary from BMAP1 to BMAP2. For r-stations greater than RBMAP2, body mapping will be held constant at BMAP2. Note: Used mostly for wing-body configurations where body radius is small compared to wing. Will cluster more points around body as BMAP2 is made less than unity. For R > RBMAP2 map cluster is held constant at BMAP2 value. BMAP1 must be set to unity. BMAP = 1 corresponds to standard body mapping.
Bow Shock Options and Parameters:	
IFIT	IFIT = 0, Bow Shock is captured. IFIT = 1, Bow Shock is fit.
Note: NS must be zero for IFIT = 1.	
IOPTS	Bow Shock capture options IOPTS = 0, Outer boundary not conformed to shock shape. IOPTS = 1, after first conical mesh, bow shock is conformed to captured shock shape for further conical mesh refinements. IOPTS = 2, Nonconical meshes are conformed to previous captured bow shock shape. Not recommended for complex configurations.
NS = 4 (IOPTS = 0) = 4-8 (IOPTS = 1,2)	Used in bow shock capturing scheme to position mesh beyond initial outer boundary.
EP > 1.0 (IFIT = 0) ≥ 1.0 (IFIT = 1)	If EP = 1.0, initial conical guess for bow shock shape is the rotated Mach cone.

Namelist Variable	Description
	For IFIT = 1, shock guess is generated by a combination of rotated Mach cone and a surface normal displacement (EP - 1.0) times the lower half thickness of the body shape. For IFIT = 0, EP should be > 1. This will set outer boundary to a shape larger than the rotated Mach cone on first conical mesh.
Conical (IF1T = 1):	
RELSHK(1) RELSHK(2) RELSHK(3)	Conical Bow Shock relaxation parameters. Bow shock fitting only. 0 < RELSHK < 1.0
ITSHKC = 0 to 15	The number of initial iterations for which the bow shock shape is held fixed at shock guess.

Note: ITSHKC can be used to determine if shock guess is good for difficult bow shock fit cases. Internal solution should converge prior to update of bow shock shape or for iterations less than ITSHK. If not, adjust EP.

	For iterations greater than ITSHK, the bow shock shape will be updated until shock conditions are satisfied.
ISMOOC = 0 to 3	This parameter smooths the initial shock guess on first conical mesh in the mapped space for bow shock fit option. Can be useful for conical supersonic edge conditions, otherwise set ISMOOC = 0.
Nonconical (3D) , IF1T = 1:	
RELNC = 1.5 (SLOR) = 3.0 (AF)	Nonconical bow shock fit relaxation parameter. If second-order bow shock computation is used (IBOW=2), RELNC must be under-relaxed or RELNC < 1.
ITSHKR = 0 to 5	Same as ITSHKC.
ISMOO = 0 to 3	Smoothing applied to shock angle at nonconical stations. This parameter only used in special cases where bow shock instability occurs. Normally, ISMOO = 0.

Namelist Variable	Description
Marching Step Size Controls:	
DZ	Marching step size
REND	Solution terminates at R = REND.
NT	Alternate termination of program after NT steps.
KRR (1) KRR (2) KRR (3) KRR (4) KRR (5)	If KRR = 0, marching step size will not be refined. If KRR > 0, marching step size will be decreased. New step size will be DZ = DZ/KRR. If KRR < 0, marching step size will be doubled. New step size will be DZ = DZ * |KRR|. KRR constrained to -2.
RZNEW (1) RZNEW (2) RZNEW (3) RZNEW (4) RZNEW (5)	R stations at which marching step size is changed.
Second-Order Accuracy Options:	
IBOW	An additional option has been included to implicitly fit the bow shock in 3D. This option yields a fully second order bow shock computation, but increases computational time significantly.
IBOW = 1	First-order bow shock. Marching meshes held fixed.
IBOW = 2	Second-order implicit bow shock. 3D mesh position updated in a similar fashion to conical. Not recommended for general use.
RBOW	Second-order bow shock can be turned off at R = RBOW. Reduces to first-order bow shock marching for R > RBOW.
ISUP	
ISUP = 1	First-order accurate treatment of supersonic crossflow points.

Namelist Variable	Description
ISUP = 2	Second-order accurate treatment of supersonic crossflow points.
Wake Parameters:	
IWAKE	
IWAKE = 0	Wake cut treated as impermeable flat plate extension of chordwise wing camber line.
IWAKE = 1	Turns on wake computation. Wake should be modeled as smooth extension of trailing edge.
IBODY	Only used if IWAKE $\neq$ 0
IBODY = 0	Wing alone
IBODY = 1	Wing-body configuration
RELWK = 0.25 to 1.0	Relaxation factor for wake pressure match
Geometry Options:	
IOPTG	If IOPTG = 1, axisymmetric body subroutine CONBOD is used to generate geometry (user supplied or altered). IOPTG = 2, 4 not supported. If IOPTG = 3, subroutine CONUSE will be used to generate geometry. User supplied. Input data can be read from unit LIN1 or LIN2. If IOPTG = 5, Harris wave drag input required on LIN1.
Inlet and Wing-Body Parameters:	
RINLET	Inlet computation turned on at R=RINLET. Turns on inlet switch (INLETT) from zero to unity. R-Station will be remapped with new inlet on geometry cross section. Solution is interpolated to new mesh and then marching continues.

Note: The step size ΔZ must be chosen such that a station occurs at R=RINLET. The following parameters are only used for IOPTG=5.

Two basic types of wing-body-inlet configurations can be handled with the parameter RWING. If RWING < 0, the code will assume that the wing exists commencing at the inlet station. If RWING is set to the true apex of the wing, the code will only seek the wing cross section when R > RWING and assumes wing emergence is independent of the inlet.

In other words, RWING < 0 is used for configurations where the wing planform cuts across the inlet station.

Namelist Variable	Description
RWING > 0	Radial or axial location of wing apex on centerline.
RWING < 0	Used for certain configurations where leading edge of wing cuts across inlet station.

Note: In the wing input, the wing should be extended to centerline. Care should be taken with cambered wings to insure that centerline wing airfoil lies entirely within the fuselage.

Note: The following parameters are only used for special cases.

IBOPT = 0	Normal internal default for body cross sections and singularity placement for grid generation.
= 1	Chined body or wing-body cross section assumed as body input. Singularity location for body grid must be specified as input.

Note: IBOPT = 1 can be used for two special purposes. The code will internally assume that the body is blunt and smooth, unless IBOPT = 1.

Purpose 1: If the wing-body configuration has a chined body, or for a smooth blended wing-body configuration, where the wing has been chopped at some span station, leaving a portion of the wing attached to the body.

Purpose 2: If cross-sectional data for complete wing-body cross sections are available, the wing body configuration can be treated as body input.

If IBOPT = 1, the code will bias the geometry interpolation towards the leading edge of the cross section and assume either a sharp-edged body or wing is present.

The user can then override the internal placement of the mesh singularity with the following parameters:

$0 \leq$ XBLOC1 < 1. RBLOC1	XBLOC1 and XBLOC2 are the locations of the mapping singularity in terms of the maximum half width.

Namelist Variable	Description
$0 \leq$ XBLOC2 $< 1.$ RBLOC1	RBLOC1 and RBLOC2 are the radial stations where the mapping location will vary linearly from XBLOC1 to XBLOC2. For R$\leq$ RBLOC1, XBLOC = XBLOC1; for R $\geq$ RBLOC2, XBLOC = XBLOC2.

Note: For example, if XBLOC2 = 0.99, the mapping singularity will be placed at 0.99 times the maximum half width. XBLOC should be specified close to unity if a sharp-edge body or wing-body cross section is to be treated.

IWOPT = 0	Assumes a blunt leading edge wing. Singularity location is determined internally.
= 1	Used for sharp leading edge configurations. User can specify location of singularity.
XWOPT < 1	Location of wing mapping singularity as a function of leading edge.

Output Control Parameters:

NOUT = -1 = 0 = 1	Controls output print If NOUT = 0, no mapped plane variables and derivatives are output. Minimum output. If NOUT = 1, maximum mapped plane output. If NOUT = -1, same output as NOUT = 0, but convergence history will be printed.
JOUT = -1 = 0 = 1	If JOUT = -1, minimum output for solution (only on body surface). If JOUT = 0, body surface and shock surface (IFIT = 1) output and three coordinate lines are output from body-to-bow shock. If JOUT = 1, maximum output of entire crossflow mesh. If JOUT > 1, every JOUT coordinate line from body surface-to-bow shock will be output.
ROUTCF (K), K=1,10	R-stations at which entire cross flow plane mesh and solutions are punched. If ROUTCF is set to -1.0, or other negative number, no output will occur.

Restart Parameters:

Namelist Variable	Description
IPUN	If IPUN = 0, the solution will not be punched out to restart. If IPUN = 1, the solution will be punched out on unit LPUN2.
IRUN	If IRUN = 0, new run If IRUN = 1, restart run, reads data from LPUN2
Geometry Controls	
ILIN	Optional mapped space body shape interpolation.
ILIN = 0	Linear Interpolation. Should be used when cross sections have abrupt changes. For example, unfaired wing-body cross sections or nearly rectangular body shapes. The cubic spline fitting can produce unpredictable results. The maximum number of geometry points (e.g., 99) should be used with ILIN = 0.
ILIN = 1	Cubic spline fitting.
IBSMOO = 0 to 3	Circumferential smoothing of input geometry (NG points) in mapped space. Should be used with caution since this option will alter input geometry. The amount of smoothing depends on number of input geometry points (NG) and the value of IBSMOO. The greater IBSMOO, more smoothing will take place. IBSMOO = 0 removes smoothing. Can be useful to smooth rough input data, otherwise set IBSMOO = 0.
NG	Number of geometry points generated in cross flow plane. Generally, NG > IC after mesh refinement. (NG must be odd).

Aerodynamic Coefficient Parameters:

IAERO	If IAERO = 0, aerodynamic coefficients are not computed. If IAERO = 1, aerodynamic coefficients are computed at end of output.
ZCG	Axial moment center.
YCG	Vertical moment center.

Namelist Variable	Description
CBAR	Aerodynamic chord used for nondimensional moment coefficient.

Note: Currently, aerodynamic forces can be computed both by spanwise and chordwise integrations. The default is spanwise integration. If chordwise values (IXWI>0) are specified also as input for interpolation, a chordwise integration of forces will also be carried out if IAERO = 1.

Three trailing edge subroutines are required. Subroutine XTES is used for the wake computation and should always correspond to the true trailing edge of the wing. Subroutine XTESX and XTESZ are used in the chordwise and spanwise force computations, and can include body and clipped tip as part of the trailing edge planform in the chordwise interpolation. The arguments of XTES are XTES (X,ZTE, DUMMY). Given the spanwise coordinate X, the routine must output the axial coordinate of the trailing edge ZTE.

Note: There is no provision in the spanwise force integration to include body forces aft of the trailing edge of the wing or clipped tips. Spanwise forces are primarily used for bodies or wing-alone computations.

These subroutines are generated automatically with IOPTG = 5 from wing data. User must write these subroutines for IOPTG = 3. See Subsection 12.2.

Namelist Variable	Description
ITE	If ITE = 0, trailing edge subroutine XTES will not be called. Force calculation will terminate at z = ZTE, assuming a trailing edge with zero sweep. If ITE > 0, trailing edge subroutine XTES will be called and used by force subroutines. ITE = 1, aft-swept trailing edge will be assumed and used in wake computation. ITE = 2, forward-swept trailing edge will be assumed and used in wake computation.
ZTE	Default location of trailing edge. If ITE = 0, a straight trailing edge with zero sweep or a body will be assumed where forces are computed to z = ZTE.
IZW	If IZW = 0, interpolated z-station data used in spanwise force calculation will not be printed.

Namelist Variable	Description
	If IZW = 1, interpolated z-station data will be printed.

Spanwise and Chordwise Interpolations:

IXWI	If IXWI = 0, no chordwise stations are read for interpolation output. If IXWI > 0, IXWI chordwise interpolation stations will be read from LIN and x-station interpolated output will be printed. Also, chordwise integration of forces will occur.
IZWI	If IZWI = 0, no spanwise stations are read for interpolation output. If IZWI > 0, IZWI spanwise interpolation stations will be read from LIN and z-station interpolated output will be printed. Note: These span stations are independent of spanwise force integration.

Note: If IXWI and IZWI are made negative at the above values, no printed output will occur.

Auxillary Input on LIN if IZWI or IXWI $\gtrless$ 0:

ZI(I) , I = 1 , IZWI

XI(I) , I = 1 , IXWI

Note: The format for XI or ZI is 10F7.

12.2 GEOMETRY INPUT OPTIONS AND MESH GENERATION

This version of NCOREL supports two basic geometry options. IOPTG = 1 or 3 are user-supplied geometry subroutines. IOPTG = 1 is a simple axisymmetric body and IOPTG = 3 is a general user-supplied analytic definition. IOPTG = 5 supports the pointwise type of input sometimes referred to as Harris wave drag input, as defined in Ref. 31.

It should be noted that NCOREL requires only pointwise Harris wave drag input or cross-sectional analytic definitions. The code will internally iterate to find the appropriate spherical geometry.

NCOREL is set up to be able to compute body or wing alone, wing-body and wing-body inlet configurations. The mesh is generated internally with the use of analytic conformal mappings and shearing transformations. Two conformal mappings are utilized. One conformal mapping is used for wing-like configurations, and requires the location of the wing mapping singularity. The second mapping is a body-vertical slit mapping primarily used for wing-body cross sections and is specified internally. Hence, the user need only specify one control point for the wing mapping.

Geometry (IOPTG = 1)

This geometry allows the user to get the feel of the program with a simple axisymmetric body definition. Subroutine CONBOD requires only one equation yielding the body polar radius as a function of the axial coordinate z.

Geometry (IOPTG = 3)

This geometry option allows the user to specify a complete configuration analytically. The basic subroutine for this geometry option is called CONUSE. For a complete wing-body definition, CONUSE can call BODYU and WINGU to describe the individual body and wing components. The I/O requirements for subroutine CONUSE are:

```
CONUSE (Z,X,IFLAG,Y)

Z is the axial coordinate.
X is the spanwise coordinate.
Y is the vertical coordinate.
```

Given Z,X the subroutine outputs Y, except if IFLAG = 0, then the subroutine computes the leading edge or maximum half-width coordinates as XTIP, YTIP. The singularity location used for the mesh generation is also computed as XSIN and YSIN and stored in COMMON/TIP/XTIP,YTIP,XSIN,YSIN. The subroutine then returns without computing Y.

IFLAG = 1	The subroutine, given Z and X, computes the Y-coordinate of the upper surface.
IFLAG = 2	The subroutine, given Z and X, computes the Y-coordinate of the lower surface.

Note: When IFLAG = 1 or IFLAG = 2, the logic for IFLAG = 0 is skipped.

The following listing illustrates a simple analytic symmetric wing body configuration where CONUSE calls BODYU and WINGU for the body and wing geometry, respectively. The final wing-body cross section is merged within subroutine CONUSE.

The body has a 2:1 elliptic cross section with a parabolic arc/cylindrical area/radius distribution. The wing has 60° leading edge sweep and a 10° aft swept trailing edge with a 4% thick symmetrical airfoil shape prescribed in subroutine AIRFOL.

```
      SUBROUTINE CONUSE (Z,X,IFLAG,Y)                                   CON00010
C                                                                       CON00020
C***********************************************************************CON00030
C                                                                       CON00040
C    S U B R O U T I N E   C O N U S E                                  CON00050
C                                                                       CON00060
C***********************************************************************CON00070
C                                                                       CON00080
      COMMON/TIP/XTIP,YTIP,XSIN,YSIN                                    CON00090
      COMMON/TIPB/XTIPB,YTIPB,XSINB,YSINB                               CON00100
      COMMON/TIPW/XTIPW,YTIPW,XSINW,YSINW                               CON00110
      ZWING=20.0                                                        CON00120
      IF (IFLAG.NE.0) GO TO 100                                         CON00130
      XD=0.                                                             CON00140
      CALL BODYU(Z,XD,IFLAG,YB)                                         CON00150
      XTIP=XTIPB                                                        CON00160
      YTIP=YTIPB                                                        CON00170
      XSIN=XSINB                                                        CON00180
      YSIN=YSINB                                                        CON00190
      IF (Z.LT.ZWING) RETURN                                            CON00200
      ZW=Z-ZWING                                                        CON00210
      CALL WINGU(ZW,XD,IFLAG,YW)                                        CON00220
      IF (XTIPW.GT.XTIPB) YTIP=YTIPW                                    CON00230
      IF (XTIPW.GT.XTIPB) XTIP=XTIPW                                    CON00240
      IF (XSINW.GT.XSINB) YSIN=YSINW                                    CON00250
      IF (XSINW.GT.XSINB) XSIN=XSINW                                    CON00260
      RETURN                                                            CON00270
  100 IF (Z.GT.ZWING) GO TO 200                                         CON00280
      CALL BODYU(Z,X,IFLAG,YB)                                          CON00290
      Y=YB                                                              CON00300
      RETURN                                                            CON00310
  200 IFF=0                                                             CON00320
      CALL BODYU(Z,XD,IFF,YBDUM)                                        CON00330
      ZW=Z-ZWING                                                        CON00340
      CALL WINGU(ZW,XD,IFF,YWDUM)                                       CON00350
      IF (XTIPB.GT.XTIPW) GO TO 300                                     CON00360
      IF (XTIPW.GE.XTIPB) GO TO 400                                     CON00370
  300 CALL BODYU(Z,X,IFLAG,YB)                                          CON00380
      Y=YB                                                              CON00390
      YW=0.                                                             CON00400
      IF (X.LE.XTIPW) CALL WINGU(ZW,X,IFLAG,YW)                         CON00410
      IF (IFLAG.EQ.1.AND.YW.GT.Y) Y=YW                                  CON00420
      IF (IFLAG.EQ.2.AND.YW.LT.Y) Y=YW                                  CON00430
      GO TO 500                                                         CON00440
  400 CALL WINGU(ZW,X,IFLAG,YW)                                         CON00450
      Y=YW                                                              CON00460
      YB=0.                                                             CON00470
      IF (X.LE.XTIPB) CALL BODYU(Z,X,IFLAG,YB)                          CON00480
      IF (IFLAG.EQ.1.AND.YB.GT.Y) Y=YB                                  CON00490
      IF (IFLAG.EQ.2.AND.YB.LT.Y) Y=YB                                  CON00500
```

```
  500 CONTINUE                                                          CON00510
      RETURN                                                            CON00520
      END                                                               CON00530
      SUBROUTINE BODYU(Z,X,IFLAG,Y)                                     BOD00010
C                                                                       BOD00020
C*****************************************************************************BOD00030
C                                                                       BOD00040
C    S U B R O U T I N E   B O D Y U                                    BOD00050
C                                                                       BOD00060
C*****************************************************************************BOD00070
C                                                                       BOD00080
      COMMON/TIPB/XTIPB,YTIPB,XSINB,YSINB                               BOD00090
      COMMON/INLET/INLETT,RINLET,RWING,BP(120),DUMP(120,60)             BOD00100
      PI=4.0*ATAN(1.0)                                                  BOD00110
      Z1=20.0                                                           BOD00120
      THNOSE=15.0                                                       BOD00130
      TANTHN=TAN(THNOSE*PI/180.)                                        BOD00140
      IF (Z.LE.Z1) RADB=Z*TANTHN*(1.-.5*Z/Z1)                           BOD00150
      IF (Z.GT.Z1) RADB=.5*Z1*TANTHN                                    BOD00160
      AREAB=PI*RADB**2                                                  BOD00170
      BRATIO=2.                                                         BOD00180
      BEL=SQRT(AREAB/(PI*BRATIO))                                       BOD00190
      AEL=BRATIO*BEL                                                    BOD00200
      IF (IFLAG.NE.0) GO TO 100                                         BOD00210
      XTIPB=AEL                                                         BOD00220
      YTIPB=0.0                                                         BOD00230
      ARG=XTIPB**2-BEL**2                                               BOD00240
      IF (ARG.LT.0.) ARG=0.                                             BOD00250
      XSINB=SQRT(ARG)                                                   BOD00260
      YSINB=0.0                                                         BOD00270
      RETURN                                                            BOD00280
  100 ARGB=1.-(X/AEL)**2                                                BOD00290
      IF (ARGB.LT.0.) ARGB=0.0                                          BOD00300
      YB=BEL*SQRT(ARGB)                                                 BOD00310
      IF (IFLAG.EQ.1) Y=YB                                              BOD00320
      IF (IFLAG.EQ.2) Y=-YB                                             BOD00330
      RETURN                                                            BOD00340
      END                                                               BOD00350
      SUBROUTINE WINGU(Z,XX,IFLAG,Y)                                    WIN00010
C                                                                       WIN00020
C*****************************************************************************WIN00030
C                                                                       WIN00040
C    S U B R O U T I N E   W I N G U                                    WIN00050
C                                                                       WIN00060
C*****************************************************************************WIN00070
C                                                                       WIN00080
      COMMON/TIPW/XTIPW,YTIPW,XSINW,YSINW                               WIN00090
      PI=4.*ATAN(1.)                                                    WIN00100
      X=XX                                                              WIN00110
      THLE=30.                                                          WIN00120
```

```
      THTE=80.00                                                        WIN00130
      TANTHL=TAN(THLE*PI/180.)                                          WIN00140
      TANTHT=TAN(THTE*PI/180.)                                          WIN00150
      XT=Z*TANTHL                                                       WIN00160
      CR=20.                                                            WIN00170
      ZA=CR*TANTHT/(TANTHT-TANTHL)                                      WIN00180
      XA=ZA*TANTHL                                                      WIN00190
      IF (X.GT.XT) X=XT                                                 WIN00200
      IF (IFLAG.EQ.0) GO TO 500                                         WIN00210
      ZLE=X/TANTHL                                                      WIN00220
      ZTE=CR+X/TANTHT                                                   WIN00230
      C=ZTE-ZLE                                                         WIN00240
      ZC=(Z-ZLE)/C                                                      WIN00250
      CALL AIRFOL(ZC,YTHC)                                              WIN00260
      YTH=C*YTHC                                                        WIN00270
      IF (IFLAG.EQ.1) Y=YTH                                             WIN00280
      IF (IFLAG.EQ.2) Y=-YTH                                            WIN00290
      RETURN                                                            WIN00300
  500 YTIPW=0.                                                          WIN00310
      XTIPW=Z*TANTHL                                                    WIN00320
      ZC=Z/CR                                                           WIN00330
      CALL AIRFOL (ZC,YTHC)                                             WIN00340
      YTH=CR*YTHC                                                       WIN00350
      ARG=XTIPW**2-YTH**2                                               WIN00360
      IF (ARG.LT.0.) ARG=0.0                                            WIN00370
      XSINW=SQRT(ARG)                                                   WIN00380
      YSINW=0.                                                          WIN00390
      RETURN                                                            WIN00400
      END                                                               WIN00410
      SUBROUTINE AIRFOL (XC,YTH)                                        AIR00010
C                                                                       AIR00020
C***********************************************************************AIR00030
C                                                                       AIR00040
C    S U B R O U T I N E   A I R F O L                                  AIR00050
C                                                                       AIR00060
C***********************************************************************AIR00070
C                                                                       AIR00080
      T=.04                                                             AIR00090
      IF (XC.GT.1.) X=1.0                                               AIR00100
      IF (XC.GE.0.AND.XC.LE.1.) X=XC                                    AIR00110
      IF (XC.LT.0.) X=0.                                                AIR00120
      YTH=(T/.20)*(.2969*SQRT(X)-.126*X-.3516*X*X                       AIR00130
     1+.2843*X*X*X-.1015*X*X*X*X)                                       AIR00140
      RETURN                                                            AIR00150
      END                                                               AIR00160
      SUBROUTINE XTES (X,ZTE,DUMMY)                                     XTE00010
C                                                                       XTE00020
C***********************************************************************XTE00030
C                                                                       XTE00040
C    S U B R O U T I N E   X T E S                                      XTE00050
```

```
C                                                                       XTE00060
C*************************************************************************XTE00070
C                                                                       XTE00080
      COMMON/TE/IZW,ITE,ZTEC,XTEBK,ZTIP,XTIP                            XTE00090
      XTEBK=0.                                                          XTE00100
      ZWING=20.                                                         XTE00110
      PI=4.*ATAN(1.)                                                    XTE00120
      CR=20.                                                            XTE00130
      THLE=30.00                                                        XTE00140
      THTE=80.00                                                        XTE00150
      TANTHL=TAN(THLE*PI/180.)                                          XTE00160
      TANTHT=TAN(THTE*PI/180.0)                                         XTE00170
      ZTIPW=CR*TANTHT/(TANTHT-TANTHL)                                   XTE00180
      ZTIP=ZWING+ZTIPW                                                  XTE00190
      XTIP=ZTIPW*TANTHL                                                 XTE00200
      ZTEW=CR+X/TANTHT                                                  XTE00210
      ZTE=ZTEW+ZWING                                                    XTE00220
      RETURN                                                            XTE00230
      END                                                               XTE00240
      SUBROUTINE BODYW (Z,X,IFLAG,Y)                                    CON02150
C                                                                       CON02160
C*************************************************************************CON02170
C                                                                       CON02180
C    S U B R O U T I N E   B O D Y W                                    CON02190
C                                                                       CON02200
C*************************************************************************CON02210
C                                                                       CON02220
      COMMON/RWUNIT/LIN,LIN1,LIN2,LOUT,LPUN,LPUN1,LPUN2,LPUN3,LPUN4,    CON02230
     1LPUN5,LPUN6,LPUN7,LPUN8,LPUN9,LPUN10,LPUN11,IFLO                  CON02240
      COMMON/TIPB/XTIPB,YTIPB,XSINB,YSINB                               CON02250
      COMMON/BODTIP/XBODT                                               CON02260
      CALL BODYU (Z,X,IFLAG,Y)                                          CON02270
      XBODT=XTIPB                                                       CON02280
      RETURN                                                            CON02290
      END                                                               CON02300
      SUBROUTINE XTESX (X,ZTE,DUMMY)                                    XTE00250
C                                                                       XTE00260
C*************************************************************************XTE00270
C                                                                       XTE00280
C    S U B R O U T I N E   X T E S X                                    XTE00290
C                                                                       XTE00300
C*************************************************************************XTE00310
C                                                                       XTE00320
      COMMON/TE/IZW,ITE,ZTEC,XTEBK,ZTIP,XTIP                            XTE00330
      XTEBK=0.                                                          XTE00340
      ZWING=20.                                                         XTE00350
      PI=4.*ATAN(1.)                                                    XTE00360
      CR=20.                                                            XTE00370
      THLE=30.00                                                        XTE00380
      THTE=80.00                                                        XTE00390
```

```
      TANTHL=TAN(THLE*PI/180.)                                          XTE00400
      TANTHT=TAN(THTE*PI/180.0)                                         XTE00410
      ZTIPW=CR*TANTHT/(TANTHT-TANTHL)                                   XTE00420
      ZTIP=ZWING+ZTIPW                                                  XTE00430
      XTIP=ZTIPW*TANTHL                                                 XTE00440
      ZTEW=CR+X/TANTHT                                                  XTE00450
      ZTE=ZTEW+ZWING                                                    XTE00460
      RETURN                                                            XTE00470
      END                                                               XTE00480
      SUBROUTINE XTESZ (X,ZTE,DUMMY)                                    XTE00490
C                                                                       XTE00500
C*****************************************************************************XTE00510
C                                                                       XTE00520
C    S U B R O U T I N E   X T E S Z                                    XTE00530
C                                                                       XTE00540
C*****************************************************************************XTE00550
C                                                                       XTE00560
      COMMON/TE/IZW,ITE,ZTEC,XTEBK,ZTIP,XTIP                            XTE00570
      XTEBK=0.                                                          XTE00580
      ZWING=20.                                                         XTE00590
      PI=4.*ATAN(1.)                                                    XTE00600
      CR=20.                                                            XTE00610
      THLE=30.00                                                        XTE00620
      THTE=80.00                                                        XTE00630
      TANTHL=TAN(THLE*PI/180.)                                          XTE00640
      TANTHT=TAN(THTE*PI/180.0)                                         XTE00650
      ZTIPW=CR*TANTHT/(TANTHT-TANTHL)                                   XTE00660
      ZTIP=ZWING+ZTIPW                                                  XTE00670
      XTIP=ZTIPW*TANTHL                                                 XTE00680
      ZTEW=CR+X/TANTHT                                                  XTE00690
      ZTE=ZTEW+ZWING                                                    XTE00700
      RETURN                                                            XTE00710
      END                                                               XTE00720
```

Note: Care must be taken in specifying certain parameters in the COMMON/TE/ of subroutines XTES, etc. They are primarily used for the wake computation logic or in the spanwise force integration. They are described as follows:

XTEBK: This is used only in spanwise force integration for a cranked trailing edge where a portion of the inboard trailing edge has zero sweep. XTEBK specifies the crank point. For all other constant or variable sweep trailing edges, set XTEBK = 0.0.

ZTIP:
XTIP These two parameters should specify the tip or outermost point of the wing. ZTIP being axial and XTIP being spanwise. Only used in wake computation logic (IWAKE = 1). These parameters allow the program to internally determine if the trailing edge is aft swept or forward swept. It also determines when wing no longer exists and only wake for $R > \sqrt{XTIP^{**2} + YTIP^{**2}}$.

Geometry (IOPTG = 5):

IOPTG = 5 currently supports a geometry input similar to that of Ref. 31. NCOREL can compute wing-body configurations with fuselage-mounted inlets. The inlet is defined by a discontinuous cross section of the fuselage. RINLET in the namelist input must be consistent with the geometry input data (e.g., RINLET must correspond to the X-station in the geometric input that has a discontinuous cross section).

The numerical description of the initial configuration geometry is input in a form which is similar to Ref. 31 input. This input data format is similar to typical linear theory panel method input. Reference 32 describes an interactive graphics program which manipulates other types of geometry input into a form compatible with the NCOREL input.

The configuration is defined to be symmetrical about the x-z plane; therefore, only one side (the positive y-side) of the configuration need be described. The geometric input is in typical aircraft coordinates, "y" being spanwise and "x" being axial, as opposed to the computational coordinates which are "x" being spanwise and "z" being axial.

Input on LIN1: Format is 10F7. for real numbers and I3 for integers.

Note: Fin and Canard input are not currently supported.

Columns	Variable	Value	Description
1-80	TITLE1		This card contains any desired identifying information.
		Control Integers	
1-80	J0	0	No reference area.
		1	Reference area to be read.
4-6	J1	0	No wing data.
		1	Cambered wing data to be read.
		-1	Uncambered wing data to be read.
7-9	J2	0	No fuselage data.
		1	Data for arbitrarily shaped fuselage.
10-12	J3	-	Not used at this time. The code is not currently capable of modeling pods.
13-15	J4	-	Not used.
16-18	J5	0	No canard (horizontal tail) data.
		1	Cambered canard data to be read.
		-1	Uncambered canard data to be read.
19-21	J6	0	No fin data.
		NFIN = 0-3	Cambered fin data to be read, where NFIN = the number of fins to be input.
		-NFIN = -(0-3)	Uncambered fin data to be read, where NFIN = the number of fins to be input.
22-24	NWAF	2-20	Number of airfoil sections used to describe the wing.
25-27	NWAFOR	3-30	Number of ordinates used to define each wing airfoil section. If the value of NWAFOR is negative, the program will expect to read lower surface ordinates also; otherwise, the airfoil is assumed to be symmetrical.
28-30	NFUS	1-4	Number of fuselage segments.

Columns	Variable	Value	Description
31-33	NRADX(1)	3-99	Number of points used to represent half-section of first fuselage segment. If fuselage is circular, the program computes the indicated number of Y- and Z-coordinates.
34-36	NFORX(1)	2-99	Number of stations for first fuselage segement.
37-39	NRADX(2)	3-99	Same as NRADX(1), but for the second fuselage segment.
40-42	NFORX(2)	2-99	Same as NFORX(1), but for the second fuselage segment.
43-45	NRADX(3)	3-99	Same as NRADX(1), but for the third fuselage segment.
46-48	NFORX(3)	2-99	Same as NFORX(1), but for the third fuselage segment.
49-51	NRADX(4)	3-99	Same as NRADX(1), but for the fourth fuselage segment.
52-54	NFORX(4)	2-99	Same as NFORX(1), but for the fourth fuselage segment.
55-57	NCAN	2-20	Number of airfoil sections used to describe the canard.
58-60	NCANOR	3-30	Number of ordinates used to define each canard airfoil section. If the value of NCANOR is negative, the program will expect to read lower surface ordinates also; otherwise, the airfoil is assumed to be symmetrical.
61-63	NFAF(1)	2-20	Number of airfoil sections used to describe the first fin.
64-66	NFAFOR(1)	3-30	Number of ordinates used to define each airfoil section on the first fin. If the value of NFAFOR(1) is negative, the program will expect to read lower surface ordinates also; otherwise, the airfoil is assumed to be symmetrical.
67-69	NFAF(2)	2-20	Same as NAFA(1), but for the second fin.
70-72	NFAFOR(2)	3-30	Same as NFAFOR(1), but for the second fin.

Columns	Variable	Value	Description
73-75	NFAF(3)	2-20	Same as NFAF(1), but for the third fin.
76-78	NFAFOR(3)	3-30	Same as NFAFOR(1), but for the third fin.
		Reference Area	
1-7	REFA	-	Reference Area Card. This is a dummy card which is not used in this code, but often appears in Craidon input decks. This dummy card should only be input if JO = 1.
		Wing	
If J1 = 0, do not input wing cards.			
1-7	XAF		Cards, each containing up to 10 values of percent chord, at which ordinates of airfoils are to be specified. Total of NWAFOR values. Each card may be identified in Columns 73-80 by XAFJ, where J denotes the last location specified on that card.
1-7 8-14 15-21 22-28	WAFORG		NWAF cards, each containing values of: X-coordinate of wing airfoil leading edge, Y-coordinate of wing airfoil leading edge, Z-coordinate of wing airfoil leading edge, wing airfoil streamwise chord length. Each card may be identified in Columns 73-80 by WAFORGJ, where J denotes the airfoil number, starting from the most inboard airfoil.
1-7	TZORD		NWAF sets of cards, one set for each airfoil. Each card in a set containing up to 10 values of DELTAZ (mean 8-14 camber line). A total of NWAFOR values will be read per airfoil. Each card may be identified in Columns 73-80 by TZORDJ, where J denotes the last location on that card. These values will be input only if J1 = 1.

Columns	Variable	Value	Description
1-7 8-14 etc	WAFORD		Cards, each containing up to 10 values of wing half-thickness, (each specified as percent of the chord) specified for each wing airfoil. A total of NWAFOR values will be read per airfoil. If NWAFOR < 0, the same number of values will be read for the lower surface.

Body (Fuselage)

If J2 = 0, do not input body cards.

Repeat the following body cards once for each body segment where the number of segments = NFUS.

1-7 8-14 etc	XIN	Cards, each containing up to 10 values of X-coordinates of body axial stations specified for each body segment. Total number of values per segment is specified by NFORX. Each card may be identified in columns 73-80 by XFUSJ, where J denotes the last location on that card.

Sets of cards YIN and ZIN are repeated NFORX times, one set for each XIN station of the fuselage segment.

1-7 8-14 etc	YIN	Cards, each containing up to 10 values of Y-ordinates of half-cross section points. A total of NRADX values are input.
1-7 8-14 etc	ZIN	Cards, each containing up to 10 values of Z-ordinates of half-cross section points. A total of NRADX values are input.

Note: The following illustrates a sample wing-body with inlet data set corresponding to an F-106 configuration.

```
  F106B CONFIGURATION WITH INLET AT X=26.325
   1   1   1      0        16 17   2 17   8 18 17   0   0                  0   0
200.75 14.327 18.93
   0.0     0.840   3.680   8.300 14.530 22.120 30.780 40.180 49.950 59.730
 69.130 77.780 85.360 91.580 96.190 99.040100.000
21.1944  0.0    1.226135.6911
24.6894  2.0    1.226132.0211
28.2048 4.0117 1.226128.3296
29.2533 4.6117 1.223327.2286
31.3244 5.8117 1.182425.0525
33.4014 7.0117 1.125522.8705
35.6823 8.3283 1.084420.4744
37.9187 9.6200 1.046318.1250
40.085810.8700 1.016515.8485
42.250012.1200 0.974013.5750
44.416313.3700 0.934211.2993
45.499113.9950 0.918810.1619
48.392915.6667 0.8693 7.1218
50.416816.8333 0.8272 5.0011
52.434318.0000 0.7927 2.8820
54.483219.1250 0.7523 0.5562
 0.0     0.0467 0.0949 0.1048 0.1020 0.1020 0.1020 0.1020 0.1006 0.1006
 0.1020 0.1020 0.1020 0.1020 0.1020 0.1006 0.1020
 0.0     0.0467 0.0949 0.1048 0.1020 0.1020 0.1020 0.1020 0.1006 0.1006
 0.1020 0.1020 0.1020 0.1020 0.1020 0.1006 0.1020
 0.0     0.0467 0.0949 0.1048 0.1020 0.1020 0.1020 0.1020 0.1006 0.1006
 0.1020 0.1020 0.1020 0.1020 0.1020 0.1006 0.1020
 0.0     0.0300 0.0871 0.1062 0.1035 0.1035 0.1048 0.1048 0.1035 0.1035
 0.1048 0.1035 0.1048 0.1048 0.1035 0.1035 0.1035
 0.0     0.0326 0.1065 0.1428 0.1466 0.1453 0.1441 0.1441 0.1453 0.1441
 0.1453 0.1453 0.1453 0.1453 0.1453 0.1453 0.1453
 0.0     0.0549 0.1372 0.1887 0.2024 0.2013 0.2024 0.2024 0.2024 0.2013
 0.2024 0.2024 0.2024 0.2013 0.2024 0.2013 0.2013
 0.0     0.0614 0.1474 0.2058 0.2385 0.2436 0.2426 0.2436 0.2436 0.2436
 0.2426 0.2436 0.2436 0.2426 0.2436 0.2436 0.2436
 0.0     0.0480 0.1332 0.2139 0.2619 0.2809 0.2809 0.2809 0.2809 0.2809
 0.2818 0.2818 0.2818 0.2809 0.2809 0.2809 0.2809
 0.0     0.0333 0.1204 0.2060 0.2647 0.2987 0.3114 0.3106 0.3106 0.3106
 0.3114 0.3114 0.3106 0.3114 0.3114 0.3114 0.3106
 0.0     0.0319 0.1167 0.2023 0.2722 0.3197 0.3434 0.3543 0.3536 0.3536
 0.3543 0.3543 0.3536 0.3536 0.3536 0.3536 0.3529
 0.0     0.0271 0.1085 0.1915 0.2588 0.3181 0.3627 0.3842 0.3927 0.3932
 0.3932 0.3927 0.3932 0.3932 0.3932 0.3938 0.3932
 0.0     0.0213 0.0965 0.1738 0.2535 0.3171 0.3587 0.3846 0.3983 0.4085
 0.4090 0.4085 0.4085 0.4090 0.4090 0.4085 0.4085
 0.0     0.0164 0.0588 0.1278 0.1994 0.2717 0.3269 0.3714 0.4049 0.4294
 0.4448 0.4526 0.4562 0.4583 0.4583 0.4558 0.4501
 0.0     0.0228 0.0645 0.1198 0.1770 0.2368 0.2886 0.3326 0.3723 0.4061
 0.4308 0.4491 0.4624 0.4706 0.4764 0.4794 0.4806
 0.0     0.0082 0.0274 0.0618 0.1148 0.1666 0.2087 0.2487 0.2860 0.3190
```

```
0.3464 0.3686 0.3860 0.3994 0.4090 0.4131 0.4141
0.0    0.0036 0.0135 0.0224 0.0339 0.0479 0.0628 0.0778 0.0924 0.1060
0.1183 0.1289 0.1378 0.1448 0.1497 0.1527 0.1538
0.0    0.5150 0.9050 1.1900 1.4300 1.6300 1.7800 1.8900 1.9350 1.9250
1.8400 1.3800 1.0300 0.6100 0.2600 0.0650 0.0000
0.0    0.5150 0.9050 1.1900 1.4300 1.6300 1.7800 1.8900 1.9350 1.9250
1.8400 1.3800 1.0300 0.6100 0.2600 0.0650 0.0000
0.0    0.5150 0.9050 1.1900 1.4300 1.6300 1.7800 1.8900 1.9350 1.9250
1.8400 1.3800 1.0300 0.6100 0.2600 0.0650 0.0000
0.0    0.5300 0.9000 1.2000 1.4300 1.6300 1.7750 1.8850 1.9300 1.9100
1.8050 1.4900 1.0550 0.6050 0.2700 0.0700 0.0000
0.0    0.5100 0.8950 1.1900 1.4250 1.6200 1.7650 1.8750 1.9200 1.9050
1.8000 1.4900 1.0600 0.6100 0.2700 0.0700 0.0
0.0    0.4900 0.8900 1.1850 1.4250 1.6100 1.7650 1.8750 1.9150 1.9000
1.7950 1.4850 1.0650 0.6100 0.2750 0.0700 0.0
0.0    0.5100 0.8600 1.1850 1.4050 1.6100 1.7550 1.8600 1.9100 1.9000
1.7850 1.4900 1.0600 0.6150 0.2800 0.0700 0.0
0.0    0.5050 0.8950 1.1700 1.4150 1.6000 1.7500 1.8500 1.9000 1.8900
1.7750 1.4950 1.0650 0.6200 0.2800 0.0700 0.0
0.0    0.5000 0.8800 1.1700 1.3900 1.5850 1.7350 1.8400 1.8800 1.8700
1.7650 1.5050 1.0600 0.6250 0.2950 0.0750 0.0
0.0    0.4950 0.8800 1.1600 1.3950 1.5650 1.7200 1.8200 1.8650 1.8550
1.7700 1.5100 1.0650 0.6350 0.3050 0.0750 0.0
0.0    0.5100 0.8900 1.1450 1.4100 1.6150 1.6900 1.7800 1.8250 1.8100
1.7500 1.4950 1.0700 0.6400 0.3200 0.0850 0.0000
0.0    0.5300 0.8800 1.1300 1.3350 1.5400 1.6500 1.7850 1.8000 1.8100
1.7350 1.5000 1.0700 0.6450 0.3250 0.0900 0.0000
0.0    0.4500 0.7250 0.8050 1.0900 1.4050 1.5300 1.6950 1.6950 1.7400
1.6750 1.4350 1.0650 0.6350 0.2950 0.1200 0.0
0.0    0.5650 0.9500 0.9950 1.1400 1.3750 1.5200 1.5700 1.5550 1.6100
1.5150 1.3100 1.0550 0.6800 0.3750 0.1050 0.0
0.0    0.5250 1.0000 1.0550 1.0950 1.1900 1.2300 1.3000 1.3750 1.3600
1.2200 1.0100 0.7650 0.5400 0.3400 0.1050 0.0
0.0    0.3300 1.0650 0.9850 0.9100 0.9500 0.9650 0.9950 1.0200 1.0450
1.0550 1.0500 1.0350 1.0150 1.0550 0.4000 0.0
 0.0    5.408  7.149  9.908 15.075 18.825 23.408 26.325
 0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
 0.0    0.0    0.0    0.0    0.0    0.0    0.0
 0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
 0.0    0.0    0.0    0.0    0.0    0.0    0.0
 0.0    0.223  0.440  0.643  0.829  0.990  1.121  1.220  1.283  1.306
 1.258  1.140  0.959  0.726  0.453  0.230  0.0
-0.914 -0.894 -0.838 -0.743 -0.616 -0.458 -0.275 -0.071  0.148  0.450
 0.751  1.033  1.280  1.480  1.621  1.680  1.702
 0.0    0.273  0.539  0.788  1.013  1.207  1.361  1.470  1.535  1.553
 1.489  1.353  1.147  0.875  0.548  0.279  0.0
-1.063 -1.036 -0.972 -0.857 -0.701 -0.508 -0.281 -0.030  0.237  0.601
 0.961  1.299  1.605  1.846  1.994  2.061  2.111
 0.0    0.326  0.647  0.953  1.230  1.461  1.632  1.742  1.801  1.804
 1.716  1.550  1.324  1.024  0.641  0.326  0.0
```

-1.165 -1.146 -1.088 -0.974 -0.802 -0.570 -0.292 0.015 0.336 0.771
1.196 1.598 1.970 2.283 2.497 2.593 2.647
0.0 0.503 1.001 1.486 1.885 2.105 2.183 2.185 2.105 1.897
1.646 1.361 1.108 0.847 0.576 0.352 0.0
-1.190 -1.181 -1.116 -0.981 -0.680 -0.230 0.268 0.771 1.267 1.904
2.524 3.133 3.705 4.273 4.837 5.251 5.537
0.0 0.554 1.106 1.647 2.105 2.346 2.392 2.382 2.285 2.023
1.752 1.490 1.301 1.081 0.782 0.449 0.0
-1.163 -1.153 -1.108 -0.979 -0.673 -0.179 0.376 0.930 1.474 2.165
2.853 3.540 4.189 4.829 5.434 5.815 6.033
0.0 0.606 1.209 1.793 2.289 2.546 2.575 2.462 2.244 1.947
1.651 1.354 1.216 1.016 0.722 0.412 0.0
-1.079 -1.066 -1.001 -0.828 -0.483 0.058 0.666 1.259 1.826 2.578
3.331 4.081 4.665 5.230 5.752 6.079 6.243
0.0 0.606 1.209 1.793 2.289 2.546 2.575 2.462 2.244 1.947
1.651 1.354 1.216 1.016 0.722 0.412 0.0
-1.079 -1.066 -1.001 -0.828 -0.483 0.058 0.666 1.259 1.826 2.578
3.331 4.081 4.665 5.230 5.752 6.079 6.243
26.325 27.575 28.100 28.825 31.158 32.575 35.075 38.992 41.742 43.825
46.325 49.908 55.158 56.643 57.575 60.075 63.105
0.0 0.792 1.572 2.276 2.608 2.784 3.480 3.480 3.844 3.706
3.430 3.054 2.355 1.621 1.141 0.966 0.630 0.0
-1.003 -0.960 -0.827 -0.461 0.259 0.904 1.289 1.289 1.918 2.653
3.339 3.987 4.199 4.096 4.341 5.162 5.856 6.226
0.0 0.829 1.641 2.378 2.560 2.994 3.754 3.754 3.983 3.833
3.592 3.240 2.578 1.846 1.149 0.934 0.629 0.0
-0.967 -0.931 -0.776 -0.359 0.473 0.915 1.250 1.250 1.935 2.678
3.390 4.054 4.398 4.365 4.355 5.097 5.822 6.204
0.0 0.828 1.641 2.370 2.623 3.149 3.940 3.940 3.943 3.831
3.602 3.231 2.604 1.844 1.153 0.949 0.630 0.0
-0.951 -0.913 -0.757 -0.348 0.439 0.910 1.185 1.185 1.936 2.677
3.396 4.047 4.453 4.429 4.346 5.093 5.792 6.192
0.0 0.811 1.610 2.341 2.629 3.099 3.905 3.932 3.913 3.784
3.523 3.076 2.462 1.740 1.090 0.863 0.593 0.0
-0.929 -0.893 -0.751 -0.369 0.384 0.871 1.063 1.533 2.242 2.942
3.606 4.160 4.504 4.529 4.425 5.091 5.811 6.173
0.0 0.779 1.548 2.272 2.654 2.977 3.764 3.764 3.750 3.601
3.318 2.896 2.351 1.710 1.044 0.779 0.538 0.0
-0.849 -0.812 -0.699 -0.402 0.270 0.833 0.946 1.711 2.367 3.006
3.598 4.097 4.463 4.618 4.558 5.091 5.750 6.090
0.0 0.757 1.505 2.220 2.663 2.884 3.659 3.652 3.614 3.456
3.174 2.774 2.258 1.667 1.014 0.709 0.511 0.0
-0.790 -0.764 -0.670 -0.413 0.187 0.831 0.890 1.768 2.398 3.007
3.572 4.060 4.425 4.642 4.606 5.036 5.695 6.032
0.0 0.682 1.359 2.025 2.569 2.814 3.447 3.448 3.390 3.224
2.968 2.618 2.186 1.678 1.102 0.735 0.447 0.0
-0.713 -0.689 -0.629 -0.459 -0.064 0.574 0.825 1.834 2.406 2.957
3.473 3.930 4.309 4.580 4.665 5.063 5.599 5.916
0.0 0.617 1.231 1.845 2.427 2.713 3.134 3.107 3.016 2.843
2.604 2.294 1.904 1.444 0.905 0.528 0.429 0.0

-0.577	-0.563	-0.529	-0.440	-0.197	0.338	0.761	1.897	2.428	2.938
3.423	3.862	4.234	4.513	4.633	4.852	5.422	5.717		
0.0	0.580	1.159	1.733	2.286	2.708	2.925	2.881	2.790	2.638
2.431	2.149	1.798	1.391	0.918	0.472	0.387	0.0		
-0.480	-0.469	-0.444	-0.368	-0.193	0.206	0.741	1.917	2.412	2.893
3.353	3.771	4.132	4.430	4.614	4.757	5.275	5.560		
0.0	0.559	1.118	1.674	2.215	2.616	2.769	2.742	2.665	2.526
2.337	2.077	1.751	1.365	0.922	0.448	0.382	0.0		
-0.407	-0.400	-0.378	-0.321	-0.177	0.199	0.736	1.921	2.406	2.878
3.331	3.750	4.116	4.421	4.634	4.754	5.244	5.529		
0.0	0.535	1.070	1.604	2.122	2.515	2.649	2.645	2.571	2.449
2.281	2.043	1.743	1.374	0.939	0.440	0.346	0.0		
-0.320	-0.315	-0.296	-0.247	-0.114	0.240	0.757	1.900	2.386	2.863
3.325	3.756	4.144	4.468	4.701	4.812	5.264	5.552		
0.0	0.529	1.058	1.586	2.094	2.501	2.658	2.674	2.624	2.533
2.388	2.190	1.909	1.571	1.167	0.649	0.351	0.0		
-0.194	-0.189	-0.166	-0.114	0.034	0.367	0.869	1.785	2.287	2.782
3.265	3.729	4.146	4.522	4.819	4.956	5.252	5.554		
0.0	0.537	1.070	1.596	2.098	2.525	2.726	2.750	2.753	2.699
2.603	2.437	2.194	1.864	1.476	1.015	0.465	0.0		
0.034	0.046	0.100	0.208	0.398	0.728	1.221	1.435	1.982	2.526
3.066	3.585	4.077	4.512	4.899	5.196	5.316	5.552		
0.0	0.526	1.049	1.563	2.049	2.467	2.708	2.708	2.751	2.728
2.634	2.468	2.229	1.915	1.524	1.068	0.550	0.0		
0.147	0.169	0.229	0.344	0.543	0.864	1.327	1.327	1.882	2.438
2.987	3.518	4.021	4.482	4.879	5.196	5.403	5.479		
0.0	0.488	0.974	1.450	1.900	2.287	2.552	2.552	2.676	2.670
2.602	2.447	2.214	1.905	1.520	1.065	0.552	0.0		
0.251	0.268	0.322	0.432	0.621	0.916	1.327	1.327	1.868	2.428
2.980	3.516	4.023	4.487	4.891	5.213	5.431	5.508		
0.0	0.361	0.719	1.072	1.411	1.727	1.996	1.996	2.283	2.403
2.402	2.310	2.123	1.848	1.487	1.048	0.544	0.0		
0.663	0.676	0.717	0.793	0.914	1.087	1.327	1.327	1.794	2.331
2.881	3.423	3.940	4.416	4.831	5.162	5.381	5.461		
0.0	0.111	0.222	0.332	0.441	0.549	0.654	0.654	1.153	1.558
1.845	1.984	1.972	1.803	1.501	1.064	0.555	0.0		
1.220	1.224	1.233	1.247	1.268	1.294	1.327	1.327	1.582	1.969
2.448	2.991	3.549	4.084	4.554	4.906	5.139	5.216		

12.3 PRINTED OUTPUT

Some sample printed output can be found in Ref. 10.

Note: The first set of printed output echoes the input parameters.

Output Block	Description
Spherical Geometry	NG Input Geometry points are printed at new radial station (R = R + DR).

Variable	Description
IG	IG = 1, lower symmetry plane IG = NG, upper symmetry plane
R	Spherical radius
x	Spanwise coordinate
y	Vertical coordinate
z	Axial coordinate
OMEG	Spherical angle (degrees)
PSI	Spherical angle (degrees)

Output Block	Description
Map Singularity	The location of the mapping singularity and derivatives at the current R and new R + Δ R station are printed for both the physical and mapped space.

Output Block	Description
Body Mesh Data (Not printed if NOUT = 0)	Blocks of variables are printed for the body surface (J = 2) used in solution at current R station.

Block #1

Variable	Description
X	Mapped space circumferential coordinate (X = θ)
B	Mapped space body radius (B = ρ)
BPR	Mapped space body derivatives B_θ
BSEC	$B_{\theta\theta}$

Variable	Description
BR	B_R
BRR	B_{RR}
BRTH	B_R

Block #2

Variable	Description
X	Mapped space circumferential coordinate (X = 0)
C	Mapped space shock (IFIT = 1) or outer boundary (IFIT = 0) radius
	Mapped space shock or outer boundary derivatives
CPR	C_θ
CSEC	$C_{\theta\theta}$
CR	C_R
CRR	C_{RR}
CRTH	C_R

Block #3

Variable	Description
X/Z Y/Z	Physical space body mesh Coordinates
H UI VI WI	Mapping metric Mapped space freestream velocities
H1	Radial mesh derivatives ρ_r
H2	Radial mesh derivatives $\rho\theta_r$

Block #4

Variable	Description
	Metric derivatives in computational space
HX	h_x
HY	h_y

Variable	Description
HR	h_r
H1X	h_{1x}
H1Y	h_{1y}
H1R	h_{1r}
H2X	h_{2x}
H2Y	h_{2y}
H2R	h_{2R}

Output Block

Solution Convergence History
(Not printed if NOUT = 0)

Variable	Description
ITER	Iteration
DELMX	Maximum correction to potential
I J	Mesh point where maximum correction occurs
DELAVG	Average correction
RESMX	Maximum residual
I J	Mesh point where maximum residual occurs
RESAVG	Average residual
KSUP	Number of supersonic crossflow points
DELS	(Only printed if IFIT = 1.) Maximum residual in shock jump equation
IS	Shock Mesh point where maximum residual occurs. (Only printed if IFIT = 1.)
NCYC	Cycle number for AF scheme. NCYC = 1 corresponds minimum value of AF parameter. (Only printed if IAF = 1.)
AF	Value of cyclic AF parameter. Varies between AFMIN and AFMIN. (Only printed if IAF = 1.)

Output Block

Bow Shock Mesh Location

Only printed if IFIT = 0. The JS = J-mesh location of the last supersonic point that occurs, starting with J = JC-2 and decreasing J. The JS-1 mesh point corresponds to the subsonic side of the captured bow shock.

Note: The J-mesh location must not exceed JC-3 or erroneous results may be obtained.

Output Block

Mesh and Solution

Block #1

Body coordinates and surface data (J = 2, I = 2 to IC).

Variable	Description
RHO	Cross-flow mapped radius
F	Reduced potential
UMAP VMAP WMAP	Map space velocities
X/Z Y/Z	Physical space spherical coordinates
M	Total Mach number
M_R	Radial Mach number
MC	Cross-flow Mach number
CP	Pressure coefficient

Note: If NOUT = 0, or -1, RHO, $\dot{F}$, UMAP, VMAP, and WMAP are not printed.

Block #2

Bow shock coordinates and surface data (J = JC, I = 2 to IC).
Printed only if IFIT = 1.

Above output is printed for the bow shock surface.

Block #3

RADIAL OUTPUT DATA IN CROSSFLOW PLANE

This output is printed only if JOUT ≥ 0. Above printout occurs for I = constant and J = 2 to JC. If JOUT = 0, printout occurs at I = 2, $I = 1 + \frac{IC}{2}$ and I = IC.

Block #4

INTERPOLATED CROSSFLOW SONIC LINE

If JOUT ≥ 0, the interpolated physical coordinates (X/Z, Y/Z) of the embedded cross flow sonic line are printed.

INTERPOLATED BOW SHOCK

Occurs only if IFIT = 0 and JOUT ≥ 0, interpolated physical coordinates of the outer sonic line.

Note: The above printout occurs for each R-station computed.

Special Messages:

Note: The code will terminate at R = REND or after NT steps. If the message,

SOLUTION HAS DIVERGED AT R =

occurs, the solution has diverged and the run may terminate. Automatic divergence contingency procedures will take place at first divergence for any R-station. The program will attempt to automatically adjust your parameters for a second attempt at convergence. If the second attempt to convege on a solution fails, the code will automatically terminate and net output block will occur.

Output Block

END-OF-RUN ITERATION SURVEY

This set of output summarizes the number of iterations required for convergence at each R-station.

TOTAL NUMBER OF ITERATIONS =

This number reflects the total number of iterations required for the computation, except if divergence has occurred at a particular station with a successful second attempt. The iterations leading to divergence will be neglected in this summary.

Output Block

If IZWI > 0, spanwise data at the specified ZI locations are interpolated and printed.

Output Block

AERODYNAMIC OUTPUT DATA

IF IAERO = 1, aerodynamic coefficients are computed. Two types of force integration are available; a chordwise integration from centerline to tip and an axial integration of forces from the apex to the end of the configuration. The chordwise force integration uses the chordwise interpolated stations specified by IXWI. If IXWI = 0, chordwise integration will not occur even if IAERO = 1. The chordwise force integration is output first, followed by the axial integration for forces. The axial integration of forces occurs automatically from z = constant interpolated data. The interpolated Z-data will only be output if IZW > 0.

Output Block

Chordwise interpolated data at IXWI stations at XI = constant specified as input for use by the force integration program.

Output Block

CHORDWISE FORCE INTEGRATION PROGRAM

Block #1

CHORDWISE SECTIONAL FORCES AND MOMENTS

This output prints the sectional increments at each chordwise segment, or span station.

x = constant.

Variable	Description
XSTAT	Chordwise station
SREF(DX)	Planform area
SAREA (DX)	Surface area
FN(DX)	Normal force
FA(DX)	Axial force
CN(DX)	Normal force coefficient

Variable	Description
CA(DX)	Axial force coefficient
CL(DX)	Lift coefficient
CD(DX)	Drag coefficient
CM(DX)	Pitching moment coefficient

Block #2

CHORDWISE INTEGRATION OF FORCES AND MOMENTS

Variable	Description
XSTAT	Chordwise station
SREF(X)	Planform
STOTX(X)	Surface area
CN(X)	Normal force coefficient
CAX(X)	Axial force coefficient
CL(X)	Lift coefficient
CO(X)	Drag coefficient
CM(X)	Pitching moment coefficient

Block #3

The freestream conditions, total lift, drag and moment coefficients, and the aerodynamic constants used in this computation are printed.

Output Block

AXIAL INTEGRATION OF FORCES AND MOMENTS

Block #1

HALF CROSS-SECTIONAL AREA

This output prints the cross-sectional area for each of the interpolated z-stations.

Block #2

SECTIONAL OR INCREMENTAL AREAS, FORCES, AND COEFFICIENTS

This output prints the forces and moments on each spanwise increment for half of the symmetric configuration.

Variable	Description
Z	Axial station
Incremental:	
SREF(DZ)	Planform area
SAREA (DZ)	Surface area
FN(DZ)	Normal force
FA(DZ)	Axial force
CN(DZ)	Normal force coefficient
CA(DZ)	Axial force coefficient
CL(DZ)	Lift coefficient
CD(DZ)	Drag coefficient
CM(DZ)	Moment coefficient

Note: The above coefficients are nondimensionalized by the incremental span areas.

Block #2

AXIAL INTEGRATION OF AREAS AND COEFFICIENTS

Variable	Description
Z	Axial station
XMIN	Location of centerline or wing trailing edge
XMAX	Maximum spanwise coordinate or wing leading edge
SREF(Z)	Planform area
STOT(Z)	Surface area
CN(Z)	Normal force coefficient
CA(Z)	Axial force coefficient
CL(Z)	Lift coefficient
CD(Z)	Drag coefficient
CM(Z)	Moment coefficient

Note: The last set of numbers represent the total areas, forces, and moments of the configuration. Coefficients are nondimensionalized by the computed planform area.

Block #2

The freestream conditions, total lift, drag and moment coefficients, and the aerodynamic constants used in this computation are printed.

12.4 OUTPUT DEVICES

A diverse array of disk output or punch data is available in NCOREL for plotting or graphical postprocessing. The following table outlines the output device units and their purpose in the program.

Note: The actual unit number is specified in the main program of NCOREL and can be easily changed by the user. The unit number currently used is also indicated after the unit variable name.

TABLE 1

ROUTINE	READS	WRITES	INPUT/OUTPUT DESCRIPTION
MAIN	LIN=1		Namelist input
		LIN1=5	Pointwise Craidon or Harris wave drag input (IOPTG=5)
		LIN2=25	Can be used for additional geometric input
INITN	LPUN1=14		Restart file for continuation runs
OUTPN		LPUN=7	Surface coordinates and pressure data (CP) Interpolated crossflow sonic line Bow shock coordinates
		LPUN1=14	Restart file used by [INITN] to restart a continuation run.
		LPUN2=8	Surface coordinates and pressure data (CP) used by [chordi:stream], [spani:interz], and [aerof: intern] for various interpolations of surface data.
			Outputs mesh coordinates and pressure for all radial stations at:
		LPUN5=18	Lower symmetry plane
		LPUN6=19	Middle plane
		LPUN7=20	Upper symmetry plane
		IFLO=15	Entire crossflow mesh and solution at specified radial stations
STREAM	LPUN2=8	LPUN3=9	Writes interpolated suface coordinates and pressure at specified chordwise stations to be used by [XTERP] for further use in chordwise force integration.

Note: Surface coordinates on R = constant are specified as x/z and y/z. Interpolated data on x = constant or z = constant are actual dimensional coordinates, (y,z) or (x,y), respectively.

INTERZ	LPUN2=8	LPUN11=13	Writes interpolated spanwise data at specified stations.
INTERN	LPUN2=8	LPUN4=2	Writes interpolated spanwise data at Z=R stations to be used by [AEROF] spanwise force integration.
CPOUT		LPUN=7	Writes surface output and flow field at interpolated and remeshed inlet on station IFLO=15.

12.5 GETTING STARTED, HELPFUL HINTS

NCOREL is not suited for treating all types of configurations. The code has been constructed to be as user friendly as possible. If divergence occurs, the code will automatically adjust your parameters to try to achieve convergence with a second try. Some configuration types that are not well suited for treatment by NCOREL are described below.

Blunt Bodies:

This type of configuration will produce subsonic Mach numbers close to the apex of the configuration. Locally blunt bodies can be treated, since NCOREL assumes a conical nose cap between R = 0 and R = ΔR. This type of configuration should be approached with caution since it will inevitably result in a discontinuous change in geometry between R = ΔR and R = 2ΔR, which can lead to instabilities. If instabilities occur, the step size should be reduced immediately after the first nonconical step R = ΔR. Use of ISMOO and second-order bow shock fitting may help, but is not generally recommended.

Conical Supersonic Edge Wings with Bow Shock Fitting:

This type of configuration is generally difficult to start. It relies upon the initial shock guess generator (SHOCKI) to start the computation at R = 0. Generally, these flows should be started on very crude meshes (16x16) and small values (< 0.10) for the bow-shock relaxation parameter. Using ISMOOC>0 may help. This parameter will smooth the initial shock guess.

Sharp Leading Edge Wings:

Limited treatment of sharp leading edge wings can be carried out by NCOREL. Generally, sharp supersonic leading edge configurations from the apex can not be treated with bow shock fitting since the bow shock will be attached. Subsonic sharp leading edge configurations can be treated at low incidence. Crossflow Mach numbers exceeding 10 will generally precede divergence, if the configuration cannot be treated. For wings with sharp or nearly sharp leading edges, the mapping singularity should be placed very close to the leading edge.

Clipped Wing Tips:

Generally, wing leading edges which turn parallel to the freestream will cause problems in the computation at high incidence due to the large

expansions that will occur. If the leading edge is to be turned, it should be done gradually. Typical high crossflow Mach numbers (>10) will precede divergence. Difficulty is encountered because the flow around wing tips generally separates. There is no mechanism in the potential flow equations to predict this behavior. It is probably better and easier not to turn the leading edge, and instead to clip the wing in the chordwise forces computation.

Wakes:

The code runs best when the wake computation is turned off. This is usually acceptable, because most configurations have supersonic trailing edge wings. Hence, the wing wake will have no effect on the wing pressures. The wake computation should be used if body pressures aft of the wing are of interest. The wake computation can have difficulties at high incidence due to strong trailing edge shocks.

Note that force computations will generally include the forces on the flat-plate extension of the wing, if the wake computation is turned off. This can be fixed by using the chordwise integration of forces and including the body as part of the trailing edge in subroutines XTESX.

General Comments:

Many of the above difficulties that are encountered are due to the development of excessively strong shocks, or expansions to vacuum pressure. Since NCOREL is a full potential code, the irrotational flow equations do not allow for the alleviation of strong shocks or expansions. In an Euler or Navier-Stokes code, these phenomena would inevitably lead to separated flow.

12.6 SAMPLE NAMELIST INPUT

12.6.1 Analytical Wing-Body Computation

The following is a listing of the namelist input corresponding to the analytic wing-body geometry described in Section 12.2. The namelist input requires all parameters to be input even if they are not being used. In this way, the user needs only establish one namelist input dataset which can be modified for all future cases.

Note: This namelist input is set to use the AF scheme with bow shock fitting and wake computation. To use SLOR scheme, change IAF = 0, RELNC = 1.5, EST = -1.0, RELSHK(2) = .30. To turn off wake, merely set IWAKE = 0.

```
 SAMPLE ANALYTICAL (IOPTG=3), WING-BODY WITH LOW INLET
    MINF=1.80,ALPHA=5.0 DEGS.
&INPUT
     EMINF=1.8,ALP=5.0,
   IC=26,JC=16,KREF=2,
 KMAX(1)=300,W(1)=1.7,DMIN(1)=1.E-2,NCYC(1)=6,AFMIN(1)=1.0,AFMAX(1)=4.0,
 KMAX(2)=300,W(2)=1.7,DMIN(2)=1.E-2,NCYC(2)=3,AFMIN(2)=.50,AFMAX(2)=2.0,
 KMAX(3)=300,W(3)=1.7,DMIN(3)=1.E-2,NCYC(3)=3,AFMIN(3)=.50,AFMAX(3)=3.0,
   KMAXNC=150,OMEGR=1.7,DMINR=1.E-2,NCYCR=3,AFMINR=.50,AFMAXR=2.0,
    IAF=1,ISUP=2,NPOW=0,ILIN=1,ISYM=0,GAMMA=1.4,
     IOPTG=3,NG=99,RINLET=99.00,
      IBOPT=0,IWOPT=0,RBLOC1=0.0,RBLOC2=50.0,XBLOC1=0.0,XBLOC2=.99,XWLOC=.99,
       EST=0.0,ESTMAX=-5.0,ITCONC=99,ITCONR=25,
        IFIT=1,IENTRY=1,RELSHK(1)=.75,RELSHK(2)=.40,RELSHK(3)=.15,RELNC=3.0,
         ITSHKC=5,ITSHKR=1,EP=1.10,NS=0,IOPTS=0,
          IBOW=1,RBOW=999.,
           IWAKE=1,IBODY=1,RELWK=.50,
           ISMOOC=0,ISMOO=0,
          RSW1=10.,EMCSW1=1.,RSW2=30.,EMCSW2=.50,
         RBMAP1=30.0,BMAP1=1.0,RBMAP2=50.0,BMAP2=0.50,
        DZ=1.00,RWING=20.0,REND=44.,NT=99,
       KRR(1)=0,KRR(2)=0,KRR(3)=0,KRR(4)=0,KRR(5)=0,
      RZNEW(1)=999.,RZNEW(2)=999.,RZNEW(3)=999.,RZNEW(4)=999.,RZNEW(5)=999.,
     IBSMOO=0,ISMOOC=0,ISMOO=0,
    IAERO=1,IZW=0,ITE=1,ZTE=100.,YCG=1.226,ZCG=39.0,CBAR=23.76,
   IXWI=27,IZWI=0,
  NOUT=0,JOUT=0,IRUN=0,IPUN=0,IPLOT=2,
 ROUTCF(1)=-1.,ROUTCF(2)=-1.,ROUTCF(3)=-1.,ROUTCF(4)=-1.,ROUTCF(5)=-1.,
 ROUTCF(6)=-1.,ROUTCF(7)=-1.,ROUTCF(8)=-1.,ROUTCF(9)=-1.,ROUTCF(10)=-1.,
 ROUTCF(11)=-1.,ROUTCF(12)=-1.,ROUTCF(13)=-1.,ROUTCF(14)=-1.,ROUTCF(15)=-1.,
 ROUTCF(1)=-1.,ROUTCF(2)=-1.,ROUTCF(3)=-1.,ROUTCF(4)=-1.,ROUTCF(5)=-1.,
       &END
0.0    0.50   1.00   1.50   2.00   2.50   3.00   3.50   3.78   3.80
4.00   4.50   5.00   5.50   6.00   6.50   7.00   7.50   8.00   8.50
9.00   9.50   10.0   10.5   11.0   11.5   12.0
/EOF


/EOF
```

12.6.2 F-106 Namelist Input (IOPTG = 5)

The following is the namelist input that will run the F-106 configuration with inlet using IOPTG = 5 input listed in Section 12.2. Note that the namelist input is for bow shock fitting (IF1T = 1) with SLOR scheme (IAF = 0) without wake computation. For AF scheme, set IAF = 1 and EST = 0.0, and up RELNC = 3.0.

```
 F-106 WING-BODY WITH INLET AT X=26.325
    MINF=1.50,ALPHA=6.0 DEGS.
&INPUT
    EMINF=1.5,ALP=6.0,
   IC=26,JC=22,KREF=2,
 KMAX(1)=300,W(1)=1.7,DMIN(1)=1.E-2,NCYC(1)=6,AFMIN(1)=3.0,AFMAX(1)=6.0,
 KMAX(2)=300,W(2)=1.7,DMIN(2)=1.E-2,NCYC(2)=3,AFMIN(2)=.50,AFMAX(2)=2.0,
 KMAX(3)=300,W(3)=1.7,DMIN(3)=1.E-2,NCYC(3)=3,AFMIN(3)=1.0,AFMAX(3)=2.0,
   KMAXNC=100,OMEGR=1.7,DMINR=1.E-2,NCYCR=3,AFMINR=1.0,AFMAXR=3.0,
    IAF=0,ISUP=2,NPOW=0,ILIN=1,ISYM=0,GAMMA=1.4,
     IOPTG=5,NG=99,IBSMOO=2,RINLET=26.325,
      IBOPT=0,IWOPT=0,RBLOC1=0.0,RBLOC2=50.0,XBLOC1=0.0,XBLOC2=.99,XWLOC=.99,
       EST=-.10,ESTMAX=-5.0,ITCONC=99,ITCONR=25,
        IFIT=1,IENTRY=1,RELSHK(1)=.40,RELSHK(2)=.20,RELSHK(3)=.15,RELNC=1.5,
         ITSHKC=25,ITSHKR=5,EP=1.10,NS=0,IOPTS=0,
          IBOW=1,RBOW=999.,
           IWAKE=0,IBODY=1,RELWK=.10,
           ISMOOC=0,ISMOO=0,
          RSW1=7.,EMCSW1=1.,RSW2=30.,EMCSW2=.50,
         RBMAP1=26.0,BMAP1=1.0,RBMAP2=45.0,BMAP2=0.50,
        DZ=1.0125,RWING=-1.0,REND=59.,NT=99,
       KRR(1)=0,KRR(2)=0,KRR(3)=0,KRR(4)=0,KRR(5)=0,
      RZNEW(1)=999.,RZNEW(2)=999.,RZNEW(3)=999.,RZNEW(4)=999.,RZNEW(5)=999.,
     ISMOOC=0,ISMOO=0,
    IAERO=0,IZW=0,ITE=1,ZTE=100.,YCG=1.226,ZCG=39.0,CBAR=23.76,
   IXWI=0,IZWI=0,
  NOUT=0,JOUT=0,IRUN=0,IPUN=0,IPLOT=2,
ROUTCF(1)=-1.,ROUTCF(2)=-1.,ROUTCF(3)=-1.,ROUTCF(4)=-1.,ROUTCF(5)=-1.,
ROUTCF(6)=-1.,ROUTCF(7)=-1.,ROUTCF(8)=-1.,ROUTCF(9)=-1.,ROUTCF(10)=-1.,
ROUTCF(11)=-1.,ROUTCF(12)=-1.,ROUTCF(13)=-1.,ROUTCF(14)=-1.,ROUTCF(15)=-1.,
ROUTCF(1)=-1.,ROUTCF(2)=-1.,ROUTCF(3)=-1.,ROUTCF(4)=-1.,ROUTCF(5)=-1.,
       &END
```

NASA National Aeronautics and Space Administration

Report Documentation Page

1 Report No	2 Government Accession No	3 Recipient's Catalog No
NASA CR-4165		

4 Title and Subtitle	5 Report Date
An Improved Version of NCOREL - A Computer Program for 3-D Nonlinear Supersonic Potential Flow Computations	July 1988
	6 Performing Organization Code

7 Author(s)	8 Performing Organization Report No
Michael J. Siclari	RE-744
	10 Work Unit No
	505-61-71-01

9 Performing Organization Name and Address	11 Contract or Grant No
Grumman Corporation Bethpage, NY 11714	NAS1-16758
	13 Type of Report and Period Covered
	Contractor Report

12 Sponsoring Agency Name and Address	14 Sponsoring Agency Code
National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225	

15 Supplementary Notes

Langley Technical Monitor - James L. Pittman
Final Report

16 Abstract

A computer code called NCOREL (for Nonconical Relaxation) has been developed to solve for supersonic full potential flows over complex geometries. The method first solves for the conical flow at the apex and then marches downstream in a spherical coordinate system. Implicit relaxation techniques are used to numerically solve the full potential equation at each subsequent crossflow plane.

Many improvements have been made to the original code including more reliable numerics for computing wing-body flows with multiple embedded shocks, inlet flow through simulation, wake model and entropy corrections. Line relaxation or approximate factorization schemes are optionally available. Improved internal grid generation using analytic conformal mappings, supported by a simple geometric Harris wave drag input that was originally developed for panel methods and internal geometry package are some of the new features.

17 Key Words (Suggested by Author(s))	18 Distribution Statement
Supersonic Flow Full Potential NCOREL	Unclassified - Unlimited Subject Category 02

19 Security Classif (of this report)	20 Security Classif (of this page)	21 No of pages	22 Price
Unclassified	Unclassified	188	A09

NASA FORM 1626 OCT 86

NASA-Langley, 1988

For sale by the National Technical Information Service, Springfield, Virginia 22161-2171

End of Document