SEMI-ANNUAL STATUS REPORT ON NASA GRANT NAG 1-837, "AERODYNAMICS OF THRUST VECTORING": JUNE 11, 1989 - DECEMBER 10, 1989

By

J. B. Tseng and C. Edward Lan

Flight Research Laboratory
The University of Kansas Center for Research, Inc.
Lawrence, Kansas 66045
December 1989

CONTENTS

1. INTRODUCTION

Thrust vectoring has been considered as a means to enhance maneuverability and aerodynamic performance of a tactical aircraft. This concept usually involves the installation of a multifunction nozzle. With the nozzle, the engine thrust can be changed in direction without changing the attitude of the aircraft. Change in the direction of thrust induces a significant change in the aerodynamic forces acting on the aircraft. Therefore, this device can be used for lift-augmenting as well as stability and control purposes. When the thrust is deflected in the longitudinal direction, the lift force and pitching stability can be manipulated, while yawing stability can be controlled by directing the thrust in the lateral direction.

Experimental investigation of the thrust vectoring concept has been under way for decades (Refs. 1-3). However, computational methods have not kept pace with experiments. One computational method based on a panel method was developed in Ref. 4. For configurations having multifunction nozzles, the flow field about the nozzle exit becomes very complicated. Jet entrainment as well as shock wave induced by the jet stream will be present. Also, pressure difference across the jet exit will produce vortices. Due to the limitation of flow singularity methods, these critical flow characteristics cannot be predicted by a panel code. Another drawback of panel methods is the absence of energy and temperature terms in the computation. Also, jet exit conditions cannot be included in the calculation. Therefore, the jet stream is frequently modeled as a flap or wake. To fully understand the induced aerodynamic effects from the jet stream deflection, methods other than a panel approach would be needed.

Methods in computational fluid dynamics (CFD) are more suitable for the present problem. CFD applications can be described as applying the laws of flow physics to mathematical equations and solving them with a computer. The concept of CFD can be traced back to the invention of digital computers. However, applications to complicated problems did not succeed until recently. In this decade, successful methods have been developed in scheme stability, solution techniques, grid generation, etc. Thanks to the progress of CFD techniques, flow properties related to thrust vectoring such as shock wave and vorticity can be captured by various methods. Also, with the development of numerical grid generation, complicated configurations can be modeled accurately.

Two CFD methods, coupled with a numerical grid generation program, have been used in this study. These two CFD methods, Euler and thin-layer Navier-Stokes algorithms (Refs. 5, 6), both utilize the finite volume scheme, which is suitable to model the jet momentum transfer. The numerical grid generation program EAGLE (Ref. 7), with the options of algebraic and elliptic grid generation systems, has been used to produce grids in this study.

Theoretical approaches of these three programs will be summarized in this report. Some computational results will also be presented.

2. THEORETICAL APPROACH

Generally, tasks in developing CFD programs include derivation of governing equations, solution of the flow equations, accuracy and stability analysis of the scheme, grid generation, etc. In this section, theoretical outlines and comparisons of the two programs, FLO-59 and CFL3D, will be discussed.

The program FLO-59 solves Euler equations, which feature conservation of mass, momentum, and energy in inviscid flow. Because of the omission of viscous terms, this program has the advantage of saving computer resources and CPU time. However, since this formulation does not include viscous effects, phenomena such as flow separation and secondary vortex cannot be predicted accurately. The program CFL3D utilizes thin-layer Navier-Stokes equations, thus more detailed flow properties can be obtained. Yet this program requires more CPU time and memory space. From experience, choice of these two programs depends on the purpose of applications. For example, Euler codes have great success in predicting some primary vortex features. It is mainly due to the fact that the primary vortex is generated by pressure differences between two regions. In predicting secondary vortex flow, Navier-Stokes equations are needed because the secondary flow is caused by viscous effect. Therefore, if the flow phenomena to be predicted are caused by viscous effect, Navier-Stokes equations should be used. Otherwise, the Euler approach should be able to provide a satisfactory solution.

In terms of solving governing equations, both programs use the finite volume approach. In both cases, the accuracy of the solution depends on the smoothness of the grids.

In discretizing a differentiation into finite difference forms, central difference is usually used. This formulation ensures a second order accuracy. From the consideration of accuracy, a central difference is better

than either the backward or forward difference. However, in some particular situation, backward difference has to be used to satisfy the physics of the flow field. For example, in a supersonic flow region, the Mach lines drawn from a point divide the flow into zones of influence. Properties at this point can only be affected by flow in the zone of influence. Therefore, while formulating differentiation with difference forms, only the points in the zone of influence should be taken into consideration. In this situation, the backward difference has to be used. For subsonic flow, characteristics of each point can be affected by any other points in the flow field. Thus, central difference is suitable for subsonic flow calculation.

In transonic flow computation, the formulation of the differential equations becomes more difficult due to the combination of sub- and supersonic regions. As stated earlier, in the subsonic region, the central difference should be used. On the other hand, in the supersonic region, backward difference has to be used. Upwind difference was developed to solve this problem. Take the differentiation $\partial F/\partial \xi$ as an example. The formulation can be written as

$$\frac{\partial F}{\partial \xi} = \mu\left(\frac{F_{i+1} - F_{i-1}}{2\Delta\xi}\right) + (1 - \mu)\left(\frac{F_i - F_{i-1}}{\Delta\xi}\right)$$

$$\mu = 1 \qquad \text{subsonic} \tag{1}$$

$$\mu = 0 \qquad \text{supersonic}$$

Therefore, central difference is used in the subsonic region. In a supersonic region, backward difference is used.

FLO-59 adds dissipative terms to the Euler equations to achieve upwind differencing. On the other hand, CFL3D has the options of flux difference splitting and flux vector splitting to separate the flow field into different regions. Details of upwind differencing of these two programs will be discussed later.

The most distinct difference between these two programs is the single-block and multiblock operations. FLO-59 utilized the single-block concept. Therefore, applications are limited to simple configurations. On the other hand, CFL3D has the capability of multiblock computation. The development of multiblock computation enables CFD applications to complicated configurations. Also, this concept makes the grid generation an easier task. For example, an individual block can be formed of the region where smooth continuation of the grids is not possible. Therefore, smooth grids can be constructed in each block. Flow properties can also be solved accurately in each block. Transmission of flow characteristics among blocks can be achieved through interpolations. Even the grid spacings are different among blocks.

2.1 Euler Method (FLO-59)

2.1.1 Governing Equations

The equations can be expressed in an integral form as follows.

$$\frac{\partial}{\partial t}\iiint_{\Omega} \vec{q}\, d\Omega + \iint_{S} \vec{N} \cdot \vec{n}\, dS = 0 \tag{2}$$

where

$$\vec{N} = N_x \vec{e}_x + N_y \vec{e}_y + N_z \vec{e}_z$$

$$\vec{q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e \end{bmatrix}, \quad N_x = \begin{bmatrix} \rho u \\ \rho u^2+p \\ \rho uv \\ \rho uw \\ \rho uH \end{bmatrix}, \quad N_y = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2+p \\ \rho vw \\ \rho vH \end{bmatrix}, \quad N_z = \begin{bmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2+p \\ \rho wH \end{bmatrix} \tag{3}$$

2.1.2 Finite Volume Scheme

These Euler equations can be discretized using the finite volume concept. The physical domain is divided into hexahedral cells. At each finite volume cell, the Euler equations can be written as

$$\frac{d}{dt}\left(J_{ijk}q_{ijk}\right) + F_{ijk}q_{ijk} = 0 \tag{4}$$

where J_{ijk} is the cell volume and F_{ijk} is the surface integral. The second term $F_{ijk}q_{ijk}$ represents the net flux out of the cell which is balanced by the rate of change of q in the cell volume. The net flux term is given by

$$\begin{aligned} F_{ijk}q_{ijk} = {} & N_{i+1/2,j,k}S_{i+1/2,j,k} - N_{i-1/2,j,k}S_{i-1/2,j,k} \\ & + N_{i,j+1/2,k}S_{i,j+1/2,k} - N_{i,j-1/2,k}S_{i,j-1/2,k} \\ & + N_{i,j,k+1/2}S_{i,j,k+1/2} - N_{i,j,k-1/2}S_{i,j,k-1/2} \end{aligned} \tag{5}$$

where S denotes the area of the cell surface.

2.1.3 Upwind Differencing

The use of central difference in Equation (5) ensures that this scheme is second order accurate in case the grid spacing is uniform. However, this scheme is suitable for subsonic flow computation only. For supersonic flow, backward differencing should be used in the discretization. An alternative form of Equation (4) can be written as

$$\frac{d}{dt}\left(J_{ijk}q_{ijk}\right) + F_{ijk}q_{ijk} - D_{ijk}q_{ijk} = 0 \tag{6}$$

where

$$Dq_{ijk} = D_x q_{ijk} + D_y q_{ijk} + D_z q_{ijk} \tag{7}$$

$$D_x q_{ijk} = d_{i+1/2,j,k} - d_{i-1/2,j,k}$$

$$D_y q_{ijk} = d_{i,j+1/2,k} - d_{i,j-1/2,k}$$

$$D_z q_{ijk} = d_{i,j,k+1/2} - d_{i,j,k-1/2} \tag{8}$$

$$d_{i+1/2,i,k} = \frac{J_{i+1/2,j,k}}{\Delta t} [\varepsilon^{(2)}_{i+1/2,j,k}(q_{i+1,j,k} - q_{i,j,k})$$

$$- \varepsilon_{i+1/2,j,k}(q_{i+2,j,k} - 3q_{i+1,j,k} + 3q_{ijk} - q_{i+j,k})] \tag{9}$$

$$\text{define } \nu_{ijk} = \frac{|\bar{P}_{i+1,j,k} - 2\bar{P}_{ijk} + \bar{P}_{i+j,k}|}{|\bar{P}_{i+1,j,k} + 2\bar{P}_{ijk} + \bar{P}_{i+j,k}|} \tag{10}$$

$$\varepsilon^{(2)}_{i+1/2,j,k} = k^{(2)} \max(\nu_{i+1,j,k}, \nu_{ijk}) \tag{11}$$

$$\varepsilon^{(4)}_{i+1/2,j,k} = \max[0,(k^{(4)} - \varepsilon^{(2)}_{i+1/2,j,k})] \tag{12}$$

In Equation (11), the artificial sensor ν_{ijk} is used to sense the existence of shock wave. Theoretically, the pressure distribution is smooth in the flow field if no shock wave exists. Therefore, ν_{ijk} is close to zero. Central difference is used in this situation. If the shock occurs in the flow field, the pressure rises across the shock. The sensor is not zero; thus, backward difference is brought into the computation.

2.1.4 Time-Stepping Scheme

At each point, Equation (5) is written as

$$\frac{dq}{dt} + \frac{1}{J}(Fq - Dq) = 0 \tag{13}$$

For the integration of Equation (6), the time-stepping scheme used is a fourth-order Runge-Kutta approach. At the time level n + 1, the solution q^{n+1} can be obtained as

$$q^{(0)} = q^n$$

$$q^{(1)} = q^{(0)} - \frac{\Delta t}{8J}(Fq^{(0)} - Dq^{(0)})$$

$$q^{(2)} = q^{(0)} - \frac{\Delta t}{6J}(Fq^{(1)} - Dq^{(0)})$$

$$q^{(4)} = q^{(0)} - \frac{\Delta t}{2J}\left(Fq^{(3)} - Dq^{(0)}\right)$$
$$q^{n+1} = q^{(4)} \tag{14}$$

where Δt is the time step.

2.1.5 Boundary Conditions

The boundary conditions on the solid boundaries are

$$\bar{v} \cdot \bar{n} = 0$$

In other words, there is no convected flow across the solid surfaces. The far-field boundaries are posed as free-stream conditions with one-dimensional gas equation correction.

2.1.6 Solution Iterations

The flow properties are initialized as

$$\bar{q} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e \end{Bmatrix} = \begin{Bmatrix} \rho_o \\ \rho_o u_\infty \cos\alpha \\ \rho_o u_\infty \sin\alpha \\ 0 \\ \rho_o e_o \end{Bmatrix} \tag{15}$$

where α is the angle of attack. For time-dependent calculation, the properties can be updated after each finite volume and Runge-Kutta iteration. For steady-state calculation, no time differencing is needed. Therefore, only the finite-volume iteration is used.

2.1.7 Thrust Vectoring Application

For a thin jet stream, the relation between pressure difference and the jet-path curvature can be expressed as

$$\frac{\rho v^2}{r} = \frac{\partial p}{\partial r} \tag{16}$$

This is obtained by balancing the centrifugal force with the pressure force. Solving for the radius of curvature, it can be obtained as

$$r \simeq \frac{\rho_j v_j^2 t_j}{\Delta p} \tag{17}$$

where t_j is the jet stream thickness. Also, the curvature 1/r can be written as

$$\frac{1}{r} = \frac{z_j''}{[1 + (z_j')^2]^{3/2}} = \frac{\Delta p}{\rho_j v_j^2 t_j} \tag{18}$$

The thrust vectoring application with FLO-59 is based on Equation (18). Initially, the calculation starts with an assumed jet shape. The grid has to be constructed following this initially guessed path. Then from the flow solver, pressure difference across the jet can be obtained. A new jet path can be updated with the relation described in Equation (18). Another set of grids should be built based on the new jet path. The iteration goes on until the jet path does not change.

This approach is suitable for thin-jet calculation. For a thick jet, the formulation of Equation (17) is not valid. Another drawback of this approach is that the nozzle exit geometry cannot be accurately modeled. The main reason for this drawback is the limitation of single-block grids. The geometric unwrapping operation in FLO-59 is along the streamwise wing surface. Following this orientation, there would be a geometric discontinuity at the nozzle exit. This problem can be solved with the multiblock operation.

2.2 CFL3D (Thin-Layer Navier-Stokes Method)

2.2.1 Governing Equations

From conservation of mass, momentum, and energy, the Navier-Stokes equations can be written as

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}(F - F_v) + \frac{\partial}{\partial y}(G - G_v) + \frac{\partial}{\partial z}(H - H_v) = 0 \tag{19}$$

where

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}$$

Flux in x-direction:

$$F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (e + p)u \end{bmatrix} \qquad F_v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - \dot{q}_x \end{bmatrix} \tag{20}$$

Flux in g-direction:

$$G = \begin{bmatrix} \rho u \\ \rho uv \\ \rho u^2 + p \\ \rho uw \\ (e + p)v \end{bmatrix} \qquad G_v = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - \dot{q}_y \end{bmatrix} \tag{21}$$

Flux in z-direction:

$$H = \begin{pmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ (e + p)w \end{pmatrix} \qquad H_v = \begin{pmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - \dot{q}_z \end{pmatrix} \tag{22}$$

2.2.2 Coordinate Transformation

Because of the irregular grid spacing in the physical domain (x-, y-, z-coordinates), the order of computational accuracy is difficult to predict. From a scheme-stability point of view, the computation will be better performed in a generalized coordinate domain (so-called computational domain). The generalized coordinate transformation can be described as

$$\begin{aligned} \xi &= \xi(x, y, z, t) \\ \eta &= \eta(x, y, z, t) \\ \zeta &= \zeta(x, y, z, t) \end{aligned} \tag{23}$$

The Jacobian of this generalized transformation can be obtained as

$$\begin{aligned} J &= \frac{\partial(\xi, \eta, \zeta)}{\partial(x, y, z)} \\ &= [x_\xi(y_\eta z_\zeta - y_\zeta z_\eta) - y_\xi(x_\eta z_\tau - x_\zeta z_\eta) + z_\xi(x_\eta y_\zeta - y_\eta x_\zeta)]^{-1} \end{aligned} \tag{24}$$

The governing equations become

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial}{\partial \xi}(\hat{F} - \hat{F}) + \frac{\partial}{\partial \eta}(\hat{G} - \hat{G}_v) + \frac{\partial}{\partial \tau}(\hat{H} - \hat{H}_v) = 0 \tag{25}$$

where

$$\hat{F} - \hat{F}_v = \frac{|\nabla\xi|}{J}\left[\hat{\xi}_x(F - F_v) + \hat{\xi}_y(G - G_v) + \hat{\xi}_z(H - H_v) + \hat{\xi}_t Q\right]$$

$$\hat{G} - \hat{G}_v = \frac{|\nabla\eta|}{J}\left[\hat{\eta}_x(F - F_v) + \hat{\eta}_y(G - G_v) + \hat{\eta}_z CH - C_v) + \hat{\eta}_t Q\right]$$

$$\hat{H} - \hat{H}_v = \frac{|\nabla\xi|}{J}\left[\hat{\zeta}_x\,(F - F_v) + \hat{\zeta}_y\,(G - G_v) + \hat{\zeta}_z(H - H_v) + \hat{\zeta}_t Q\right]$$

$$(\hat{K}_x,\ \hat{K}_y,\ \hat{K}_z,\ \hat{K}_t) = (K_x,\ K_y,\ K_z\ K_t)/|\nabla K|$$

$$|\nabla K| = (K_x^2 + K_y^2 + K_\zeta^2)^{1/2}$$

$K = (\xi,\ \eta,\ \zeta)$, respectively. (26)

2.2.3 Finite-Volume Scheme

The governing equations, while written in generalized coordinates, are used in the finite volume formulation. Integrating equations over control volume bounded by lines of constant in ξ-, η-, ζ-directions yields

$$\left(\frac{\partial\hat{Q}}{\partial t}\right)_{i,j,k} + (\hat{F} - \hat{F}_v)_{i+1/2,j,k} - (\hat{F} - \hat{F}_v)_{i-1/2,j,k}$$
$$+ (\hat{G} - \hat{G}_v)_{i,j+1/2,k} - (\hat{G} - \hat{G}_v)_{i,j-1/2,k}$$
$$+ (\hat{H} - \hat{H}_v)_{i,j,k+1/2} - (\hat{H} - \hat{H}_v)_{i,j,k-1/2} = 0 \qquad (27)$$

2.2.4 Thin-Layer Navier-Stokes Approximation

It is known that the velocity and pressure vary rapidly in the direction normal to the surface. On the other hand, flow characteristics are generally smooth along the surface, unless singularities exist. Retaining only the viscous terms corresponding to the direction normal to the surface can save

computing resources, yet maintaining enough computational accuracy. In this manner, the thin-layer Navier-Stokes equations can be written as

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial}{\partial \xi}(\hat{F}) + \frac{\partial}{\partial \eta}(\hat{G}) + \frac{\partial}{\partial \zeta}(\hat{H} - \hat{H}_v) = 0 \tag{28}$$

Certainly, viscous terms can also be retained in ξ, η directions because, in CFL3D, solid surfaces can be placed on any block surfaces. The laminar viscous terms can be included in all ξ, η, ζ directions simultaneously. However, turbulence viscous terms can only be added at most in two directions simultaneously.

2.2.5 Approximate Factorization

The thin-layer Navier-Stokes equations can be solved by approximate factorizations. CFL3D has the options of two-factor and three-factor approximate factorizations. Taking the three-factor approximate factorization, for example, at time step n + 1, Equation (28) can be written as

$$[\frac{I}{J\Delta t} + \delta_\xi \frac{\partial \hat{F}}{\partial Q} + \delta_\eta \frac{\partial \hat{G}}{\partial Q} + \delta_\zeta(\frac{\partial \hat{H}}{\partial Q} - \frac{\partial \hat{H}_v}{\partial Q})]^n \Delta Q = -R^n \tag{29}$$

where R^n is the residual. Equation (29) can be factorized into three equations along ξ, η, ζ sweep directions.

$$[\frac{I}{J\Delta t} + \delta_\xi \frac{\partial \hat{F}}{\partial Q}]^n \Delta Q^* = -R^n \tag{30}$$

$$[\frac{I}{J\Delta t} + \delta_\eta \frac{\partial \hat{G}}{\partial Q}]^n \Delta Q^{**} = (\frac{I}{J\Delta t})\Delta Q^* \tag{31}$$

$$[\frac{I}{J\Delta t} + \delta_\zeta(\frac{\partial \hat{H}}{\partial Q} - \frac{\partial \hat{H}_v}{\partial Q})]^n \Delta Q = (\frac{I}{J\Delta t})\Delta Q^{**} \tag{32}$$

The solution can be updated as

$$Q^{n+1} = Q^n + \Delta Q \tag{33}$$

2.2.6 Upwind Differencing Scheme

Two distinct methods are provided in this program, i.e. flux difference splitting and flux vector splitting.

2.2.6.1 Flux Difference Splitting: The generalized fluxes F, G, and H are split into forward and backward contributions according to the signs of eigenvalues of the Jacobian matrices. For example, considering the flux in the ξ direction,

$$A = \frac{\partial \hat{F}}{\partial Q} = T\Lambda T^{-1} = T(\Lambda^+ + \Lambda^-)T^{-1} \tag{34}$$

where $\Lambda^{\pm} = \frac{\lambda \pm |\lambda|}{2}$, Λ are diagonal matrices formed from the eigenvalues of A;

i.e., $\Lambda = \text{diag}(\frac{\tilde{\bar{u}}|\nabla\xi|}{J} , \frac{\tilde{\bar{u}}|\nabla\xi|}{J} , \frac{\tilde{\bar{u}}|\Delta\xi|}{J} , \frac{(\tilde{\bar{u}} + \tilde{a})|\nabla\xi|}{J} , \frac{(\tilde{\bar{u}} - \tilde{a})|\nabla\xi|}{J})$

and the symbol "~" indicates a Roe-averaged variable. The left-hand side of Equation (30) can be expressed as

$$[\frac{I}{J\Delta t} + \delta_\xi \frac{\partial \hat{F}}{\partial Q}]^n \Delta Q^* \approx T[\frac{I}{J\Delta t} + \delta_\xi^- \Lambda^+ + \delta_\xi^+ \Lambda^-]T^{-1}\Delta Q^* \tag{35}$$

In the same manner, the flux in the η- and ζ-directions can be split accordingly.

2.2.6.2 Flux Vector Splitting: The generalized fluxes $\hat{F}$, $\hat{G}$, and $\hat{H}$ are split into forward and backward contributions according to the flow speed.

For example, considering the flux in the ξ-direction in Equation (28):

$$\delta_\xi \hat{F} = \delta_\xi^- \hat{F}^+ + \delta_\xi^+ \hat{F}^- \tag{36}$$

where δ_ξ^- and δ_ξ^+ denote backward and forward difference.

Define:

$$M_\xi = \bar{u}/a \text{ and } \bar{u} = u/|\nabla\xi| \tag{37}$$

for supersonic flow ($|M_\xi| > 1$)

$$\hat{F}^+ = \hat{F},\ \hat{F}^- = 0,\ M_\xi > 1$$

$$\hat{F}^+ = 0,\ \hat{F}^- = \hat{F},\ M_\xi < -1 \tag{38}$$

for subsonic flow ($|M_\xi| < 1$)

$$\hat{F}^\pm = \frac{|\nabla\xi|}{J} \begin{bmatrix} f^\pm_{mass} \\ f^\pm_{mass}[\hat{k}_x(-\bar{u} \pm 2a)/\gamma + u] \\ f^\pm_{mass}[\hat{k}_y(-\bar{u} \pm 2a)/\gamma + v] \\ f^\pm_{mass}[\hat{k}_z(-\bar{u} \pm 2a)/\gamma + w] \\ f^\pm_{energy} \end{bmatrix}$$

$$f^\pm_{mass} = \pm\rho a(M_\xi \pm 1)^2/4 \qquad (M_\xi = \frac{\bar{u}}{a})$$

$$f^\pm_{energy} = f^\pm_{mass}\{[-(\gamma-1)\bar{u}^2 \pm 2(\gamma-1)\bar{u}a + 2a^2]/(\gamma^2+1) + (u^2+v^2+w^2)/2\}$$

$$\hat{k}_x = \xi_x/|\nabla\xi|\ ,\ \hat{k}_y = \xi_y/|\nabla\xi|\ ,\ k_z = \xi_z/|\nabla\xi| \tag{39}$$

The flux in the η- and ζ-directions can be obtained in the same manner.

2.2.7 Multiblock Operation

For complicated configurations, the traditional single-block flow solver cannot predict the flow properties without sacrificing accuracy at some portion of the flow field. For example, the unwrapping type grid system can only describe a simple configuration, such as wing alone or body alone cases. Applying the same grid system to a wing-body configuration, grid smoothing is needed. Also, this grid system usually produces coarse grids on the fuselage while using the same number of grid points on the wing in the unwrapping direction. For a configuration with integrated nozzles, the single-block calculation (grid generation and flow solver) cannot produce accurate results. Therefore, multiblock calculation becomes inevitable in this research.

In the multiblock calculation, the computational domain is divided into several hexahedral blocks. Basically, the program solves the thin-layer Navier-Stokes equations in each individual block. Boundary conditions on each boundary surface of each block has to be posted. If a boundary surface is designated as a patch surface, flow properties will be provided from neighboring blocks connected to this patch surface. With this useful technique, complicated geometry can be solved without smoothing out the geometry or losing accuracy.

2.2.8 Jet Application

In the program CFL3D, eight standard boundary conditions are available. A list of the standard boundary conditions can be found in the Appendix. Special purpose boundary conditions can be added following the multiblock coordinate orientation. In the current work, a jet block has to be formed to simulate jet exit conditions. The jet exit plane is treated as one boundary

surface of the jet block. A special routine has been developed to specify the jet exit conditions and include the jet in the computation. On the other boundary surfaces, boundary conditions can be specified as patching or in/out flow surfaces. Examples of designating boundary conditions on block surfaces will be included in this report.

For thrust vectoring applications, to separate the jet momentum from the free stream, jet blocks should be constructed following the jet trajectory. The jet path depends on the strength of the jet as well as the free-stream velocity. Investigation of the jet deformation shape and effects of jet conforming grids will be continued in the next period of this study.

2.3 Numerical Grid Generation

Generally, flow equations describe the flow physical properties; and the grids represent the discretized flow field that the flow equations influence. Transformation type grid generation can be used to describe simple geometries. However, for complicated configurations, numerical grid generation becomes an indispensable tool. Currently, the program EAGLE (Eglin Arbitrary Geometry Implicit Euler) has been used for this thrust vectoring research. Features of this program include two-dimensional surface generation and three-dimensional grid generation. In CFD applications, an individual computational domain can be seen as a hexahedral block. The surface generation produces coordinates on each boundary surface and/or any surface in the block. Consequently, the three-dimensional grid generation collects information from two-dimensional surface generation and produces the entire grids. Two grid generation mechanisms, algebraic and elliptic generation systems, are available in this program.

2.3.1 Surface Generation

If the boundaries of a surface are specified, the coordinates on the surface can be obtained by various interpolation methods. Popular interpolation methods include Lagrange interpolation, Hermite interpolation, spline interpolation, etc. In the EAGLE program, the Lagrange interpolation is recommended, although other interpolation functions are available.

The general form of one-dimensional (ξ) Lagrange interpolation is

$$\gamma(\xi) = \sum_{n=1}^{N} \phi_n(\frac{\xi}{I})\gamma_n \ , \ \phi_n(\frac{\xi}{I}) = \prod_{\ell=1}^{N} \frac{\xi - \xi_\ell}{\xi_n - \xi_\ell} \tag{40}$$

where N is the number of point in the ξ interpolation domain and $\xi \in [0, \ I]$. For example, for a two-point Lagrange interpolation,

$$\gamma(\xi) = (1 - \frac{\xi}{I})\gamma + \frac{\xi}{I}\gamma_2 \tag{41}$$

The general form of two-dimensional (ξ, η) Lagrange interpolation is

$$\gamma(\xi, \eta) = \sum_{n=1}^{N} \phi_n(\frac{\xi}{I})\gamma(\xi_n, \eta) + \sum_{m=1}^{M} \phi_m(\frac{\eta}{J})\gamma(\xi, \eta_m)$$

$$- \sum_{n=1}^{N} \sum_{m=1}^{M} \phi_n(\frac{\xi}{I})\phi_m(\frac{\eta}{J})\gamma(\xi_n, \eta_m) \tag{42}$$

where $\phi_m(\frac{\eta}{J}) = \prod_{\ell=1}^{M} \frac{\eta - \eta_\ell}{\eta_m - \eta_\ell}$

M is the number of points in the η interpolation domain, and $\eta \in [0, \ J]$.

From Equation (3), the coordinates of the grid points can be obtained.

2.3.2 Three-Dimensional Grid Generation

2.3.2.1 Algebraic Grid Generation: In a three-dimensional domain (ξ, η, ζ), the interpolation becomes

$$\gamma(\xi, \eta, \zeta) = \sum_{n=1}^{N} \phi_n(\frac{\xi}{I})\gamma(\xi_n, \eta, \zeta) + \sum_{m=1}^{M} \phi_m(\frac{\eta}{J})\gamma(\xi, \eta_m, \zeta)$$

$$+ \sum_{t=1}^{T} \theta_t(\frac{\zeta}{k})\gamma(\xi, \eta, \zeta_t) - \sum_{n=1}^{N}\sum_{m=1}^{M} \phi_n(\frac{\zeta}{I})\phi_m(\frac{\eta}{J})\gamma(\xi_n,\eta,\zeta)$$

$$- \sum_{n=1}^{N}\sum_{t=1}^{T} \phi_n(\frac{\xi}{I})\theta_t(\frac{\zeta}{k})\gamma(\xi_n, \eta, \zeta_t) - \sum_{m=1}^{M}\sum_{t=1}^{T} \phi_m(\frac{\eta}{J})\theta_t(\frac{\zeta}{k})\gamma(\xi, \eta_m, \zeta_t)$$

$$+ \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T} \phi_n(\frac{\xi}{I})\phi_m(\frac{\eta}{J})\theta_t(\frac{\zeta}{k})\gamma(\xi_n, \eta_m, \zeta_t) \qquad (43)$$

The grids can be produced by using Equation (43). These are the so-called algebraic grids. Usually this type of grid generation system produces smooth grid coordinates. The main advantages of this algebraic grid generation are simplicity and saving of computing time. From Equation (43) the grids are produced based on the boundary spacing. Therefore, boundary point spacing is critical in this system. For example, in flow calculation the spacing near abruptly changing contour should be small. Therefore, the spacing should be small while specifying the boundary points for surface generation in EAGLE. In other words, users need good judgement while using the algebraic grid generation system. One way to avoid the disadvantages inherent in using the algebraic grid generation system is to use the elliptic grid generation system.

2.3.2.2 Elliptic Grid Generation: The purposes of elliptic grid generation system are controlling coordinate line distribution, and orthogonality in the field. Elliptic grids can be generated by solving the Poisson equations:

$$\nabla^2 \xi^i = P^i \qquad (i = 1, 2, 3) \tag{44}$$

in which the control function P^i is to control the spacing and orthogonality. Detailed formulation of the control functions can be found in Reference 7. While solving the Poisson equations, the algebraic grids obtained from transfinite interpolation are used for initial guesses. Then an iterative procedure is used to solve Equation (44) until satisfactory results are obtained.

3. RESULTS

Five configurations have been tested in this study. The first three test cases are configured to test the FLO-59 program. They are

(1) A rectangular wing integrated with a nozzle

(2) A 30-degree swept wing integrated with a nozzle

(3) A 58-degree delta wing.

To verify the new routines added in CFL3D, two rectangular nozzle configurations are tested.

3.1 Rectangular Wing Integrated with a Nozzle

As shown in Figure 1, this rectangular ONERA wing (Ref. 8) is modified with the jet nozzle distributed between 40% and 70% of the semispan. The thickness of the nozzle is 4% of the root chord. The jet is deflected at an angle of 30 degrees relative to the chord. The thrust coefficient is 0.2. Pressure distribution at 25%, 45%, and 75% of the semispan is shown in Figure 2.

3.2 30-Degree Swept Wing Integrated with a Nozzle

This configuration also has the ONERA wing section distribution (Fig. 3). The jet nozzle is distributed between 40% and 70% of the semispan. The thickness of the nozzle is 4% of the root chord. The jet is deflected at an angle of 30 degrees relative to the chord. The thrust coefficient is 0.2. Two different grid systems are tested with the same jet exit condition. The first grid system is constructed with the cells following the jet path; i.e., the cut line of the mesh being identical to the jet path centerline. In this manner, the calculation direction follows the physical flow. Hence, the Kutta

condition can be satisfied. For the purpose of comparison, another grid system with a straight cut line representing a straight wake has also been tested. The pressure coefficient distributions are shown in Figure 4. It is seen from the calculated results that the jet effect is diminished in the straight-wake case. At present, no experimental data for this test case are available. However, the results from the deformed wake formulation represents a more realistic solution. Therefore, it is concluded that the grids should follow the jet path to correctly model the jet effect.

3.3 A 58-Degree Delta Wing

One of the objectives in this study is modeling a wing-body configuration with a vectored jet, as shown in Figure 5. This aerospace plane has a highly swept lifting surface for high-speed flight conditions. Highly swept wings have the tendency to produce vortices. As known, the vortex is mainly generated by pressure difference. In the situation of thrust vectoring, the pressure difference across the afterbody is largely increased. Therefore, even at low angle of attacks, vortex flow is likely to occur.

A 58-degree swept delta wing is modeled to investigate the leading-edge flow (Fig. 6). Pressure distribution at 25%, 45%, and 75% are shown in Figure 7. The leading-edge vortex is sensitive to the grid spacing near the leading edge (Ref. 9). In the calculation, the grid spacing is very dense near the leading edge. Reference 9 also demonstrates the success of capturing leading-edge vortex with the program CFL3D.

3.4 A Rectangular Nozzle without Turning Vanes

A new routine has been added in the program CFL3D to simulate the jet thrust. The jet exit plane is modeled as a block boundary surface. The jet velocity, temperature, and pressure can all be specified on this plane. A rectangular nozzle without turning vanes is modeled to test this new routine.

This nozzle is featured as a 2D nozzle with an aspect ratio of 4.6. As shown in Figure 8, the jet exit Mach number is 0.9 with the free-stream Mach number equal to 0.1. No similar experimental data are available for comparison. However, from the theory of jet expansion, the calculated results match the entrainment and expansion concept. In this case, only the jet velocities are specified at the nozzle exit. For comparison with a real jet, the jet pressure as well as energy should also be specified.

3.5 A Rectangular Nozzle with Thrust Vectoring Effect

This configuration is designed to test the thrust vectoring concept. As shown in Figure 9, the nozzle exit is deflected at 30 degrees. Figure 9 shows the block arrangement and specified boundary conditions on the block surfaces. A jet block is formed following the jet trajectory, which contains the jet momentum. From multiblock formulation point of view, the boundary surface positions of this jet block are not critical because the interpolations among blocks can capture the momentum and energy transition. However, the accuracy of the interpolations critically depends on the grid orthogonality and spacing on both sides of the interface. To eliminate the possibility of interpolation error, the jet block is constructed containing most of the jet momentum in the region near the exit.

The jet exit Mach number is 0.8, and the free-stream Mach number equals 0.25. The angle of attack in this situation is zero. Again, no similar experimental data are available for this configuration. The symmetric plane u-w velocity profile shown in Figure 10 does demonstrate all features of a deflected jet. Also, pressure distribution in the longitudinal direction is shown in Figure 11. The sectional lift coefficient due to the deflected thrust is approximately 1.0. Under the same situation, the lift coefficient should be nearly zero without thrust vectoring effect. The calculation demonstrates the concept of induced lift force due to the deflected thrust. Owing to the pressure difference across the jet stream, vortices are expected to occur. Figure 12 shows the v-w velocity profile in a cross flow plane. The cross-flow plane is located at ten nozzle heights from the exit. As expected, vortices do occur in the jet stream.

4. CONCLUDING REMARKS

The programs FLO-59 and CFL3D were used to predict the thrust vectoring effects. Computational grids are generated by the program EAGLE. Complicated geometries, such as nozzle and conforming jet boundaries, can be modeled accurately by this program.

The Euler flow solver FLO-59 produces reasonable results for wings integrated with jet-flap type exit. The pressure difference due to thrust vectoring near the trailing edge can be reasonably calculated by this program. For afterbody nozzle configurations, the multiblock flow solver CFL3D must be used. The multiblock function enables users to model the nozzle in detail. Computational results demonstrate the flow field asociated with a deflected jet. Also, the induced lift can be predicted by CFL3D.

The computational results presented in this report demonstrate the capability of the flow solvers and the grid generation program. These tools will be applied to the study of wing-body configurations with thrust vectoring in the remaining period of this study.

5. REFERENCES

1. Capone, F. J.; Hunt, B. L.; and Poth, G. E. "Subsonic/ Supersonic Aeropropulsive Characteristics of Nonaxisymmetric Nozzles Installed on an F-18 Model." Journal of Aircraft, Volume 20, October 1983, pp. 853-858.

2. Bowers, D. L. "Aerodynamic Effects Induced by a Vectored High Aspect Ratio Nonaxisymmetric Exhaust Nozzle." Journal of Aircraft, Volume 16, August 1979, pp. 515-520.

3. Berrier, B. L.; and Re, R. J. "A Review of Thrust-Vectoring Schemes for Fighter Aircraft." AIAA Paper 78-1023, July 1978.

4. Carlson, J. B. "Evaluation and Application of VSAERO to a Nonaxisymmetric Afterbody with Thrust Vectoring." Presented at SAE Aerotech 1987 Conference, October 1987.

5. Jameson, A.; Schmidt, W.; and Turkel, E. "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes." AIAA Paper 81-1259, June 1981.

6. Thomas, J. L.; Taylor, S. L.; and Anderson, W. K. "Navier-Stokes Computations of Vortical Flows Over Low Aspect Ratio Wings." AIAA Paper 87-0207, 1987.

7. Thompson, J. F. "Numerical Grid Generation System User's Manual." USAF Armament Laboratory Technical Report 87-15, 1987.

8. Agarwal, R. K.; and Deese, J. E. "Transonic Wing-Body Calculations Using Euler Equations." AIAA Paper 83-0501, 1983.

9. Krist, S. L.; Thomas, J. L.; Sellers, W. L.; and Kjelgaard, S. O. "An Embedded Grid Formulation Applied to Delta Wings." Tentative publication.

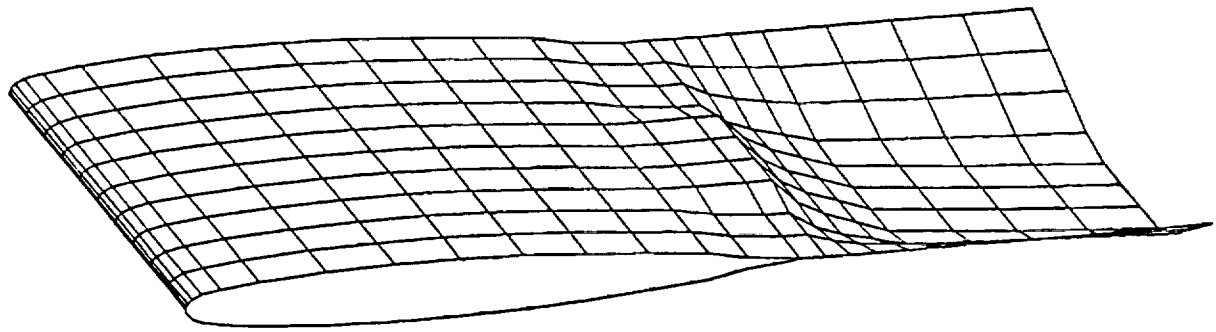
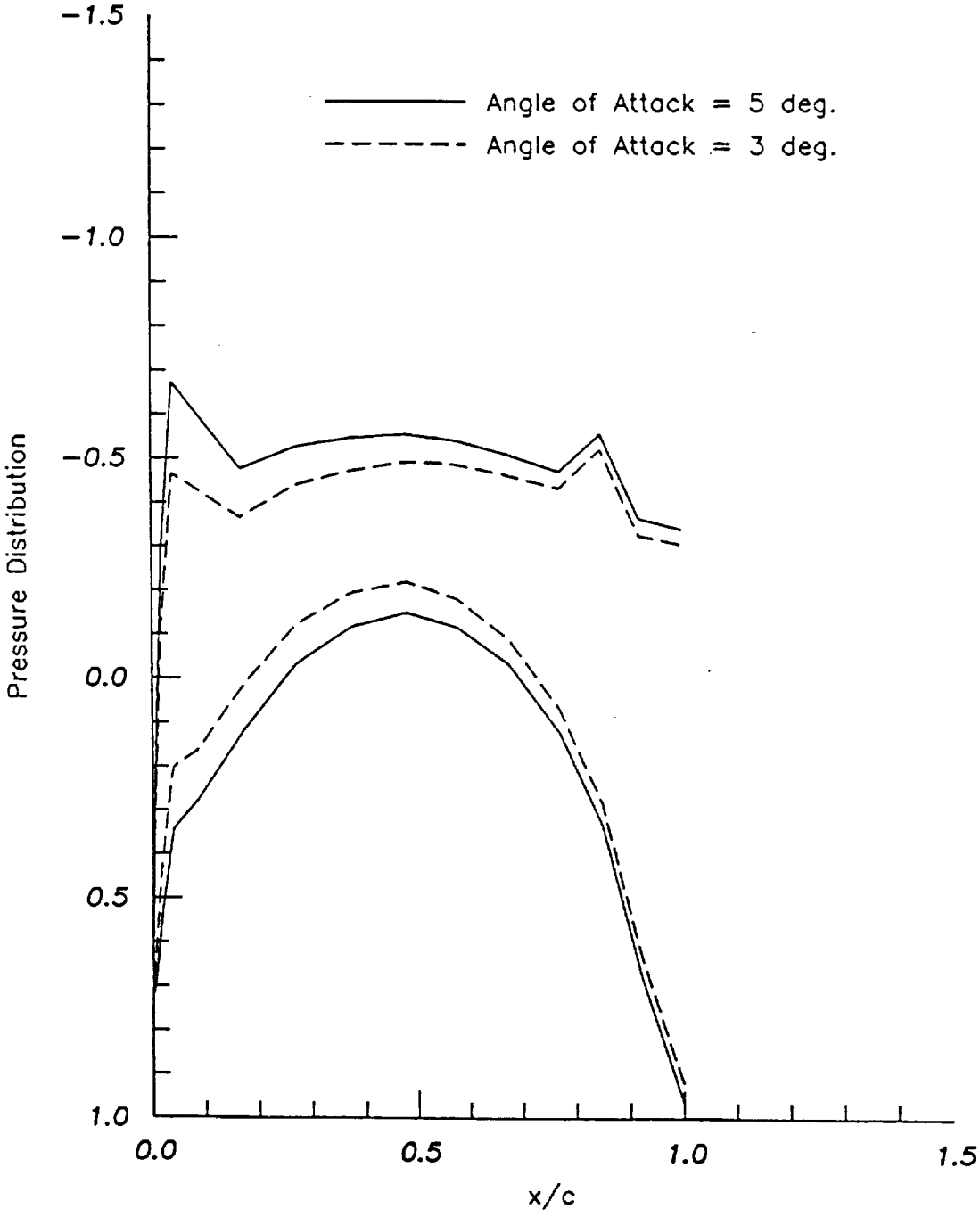
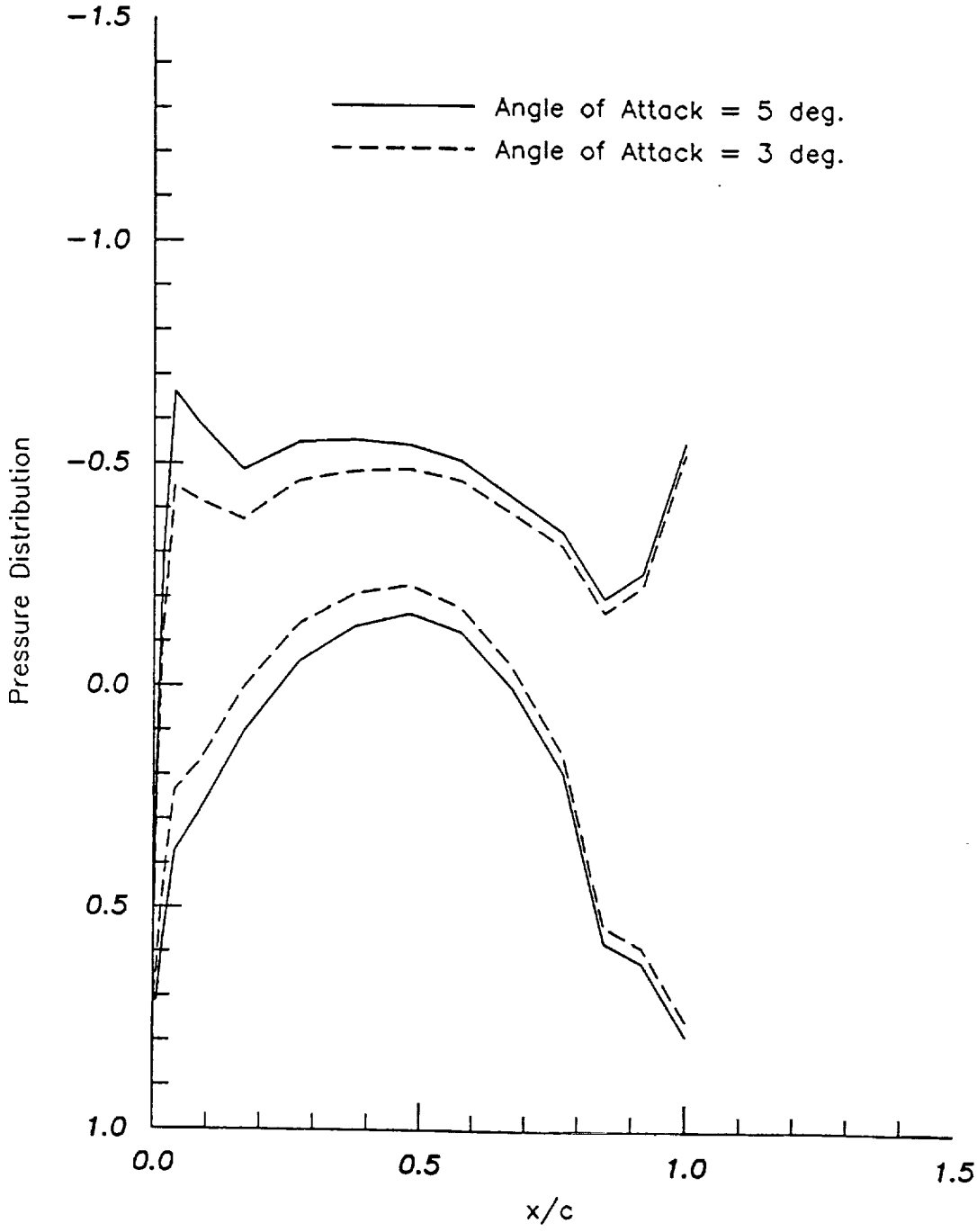


Figure 1: Grids on the Rectangular Wing and Jet-Path Surface.



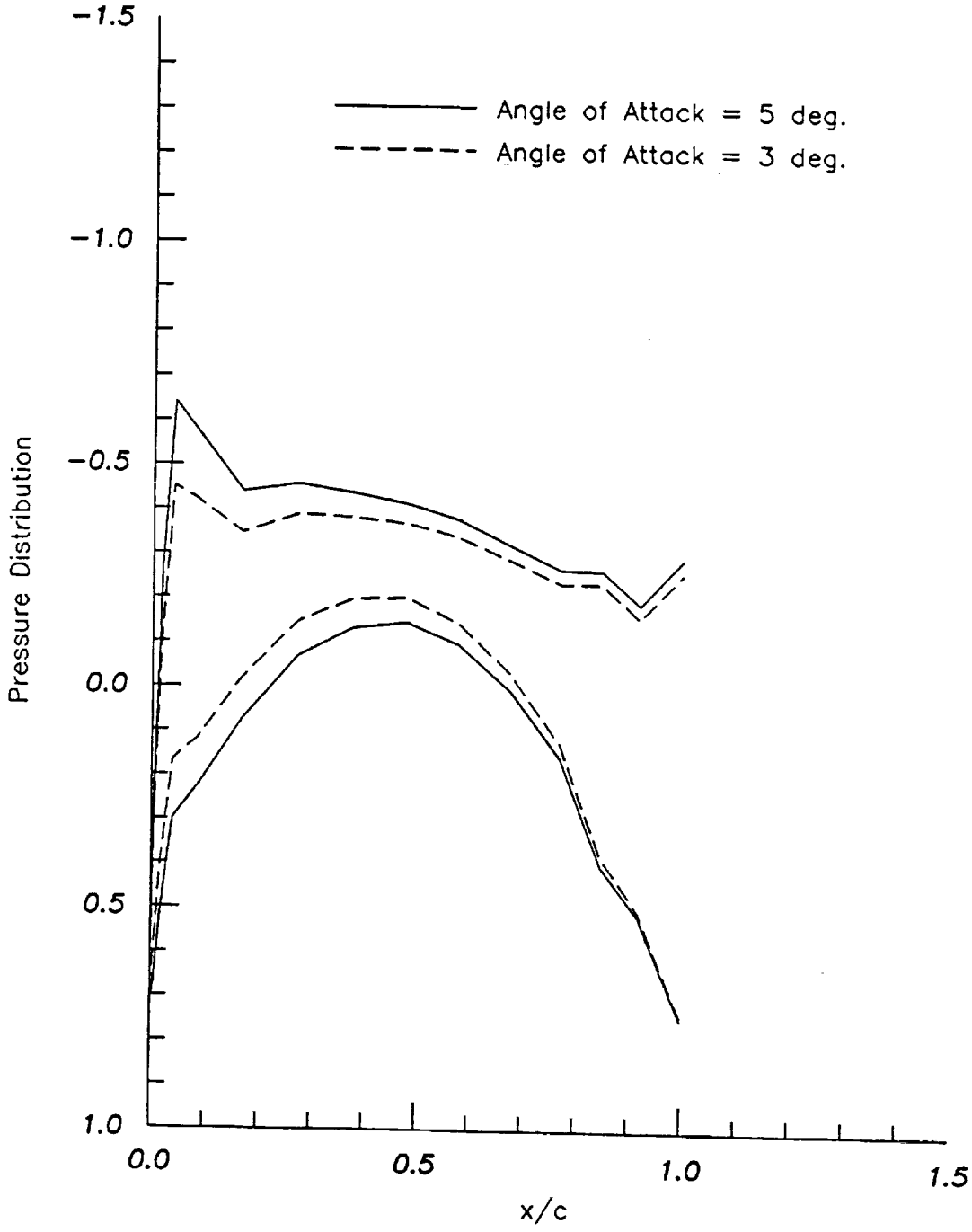
(a) 2y/b = 0.25

Figure 2: Rectangular Wing Chordwise Pressure Distribution at M = 0.84.



(b) 2y/b = 0.45

Figure 2. Continued



(c) 2y/b = 0.75

Figure 2. Continued

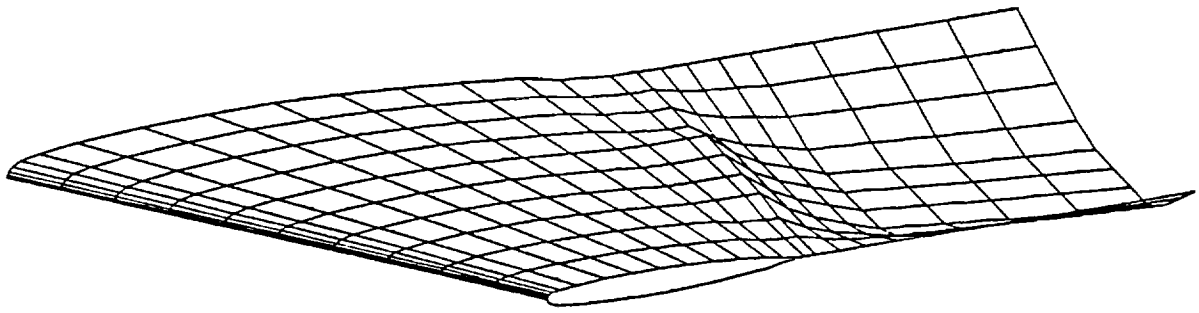
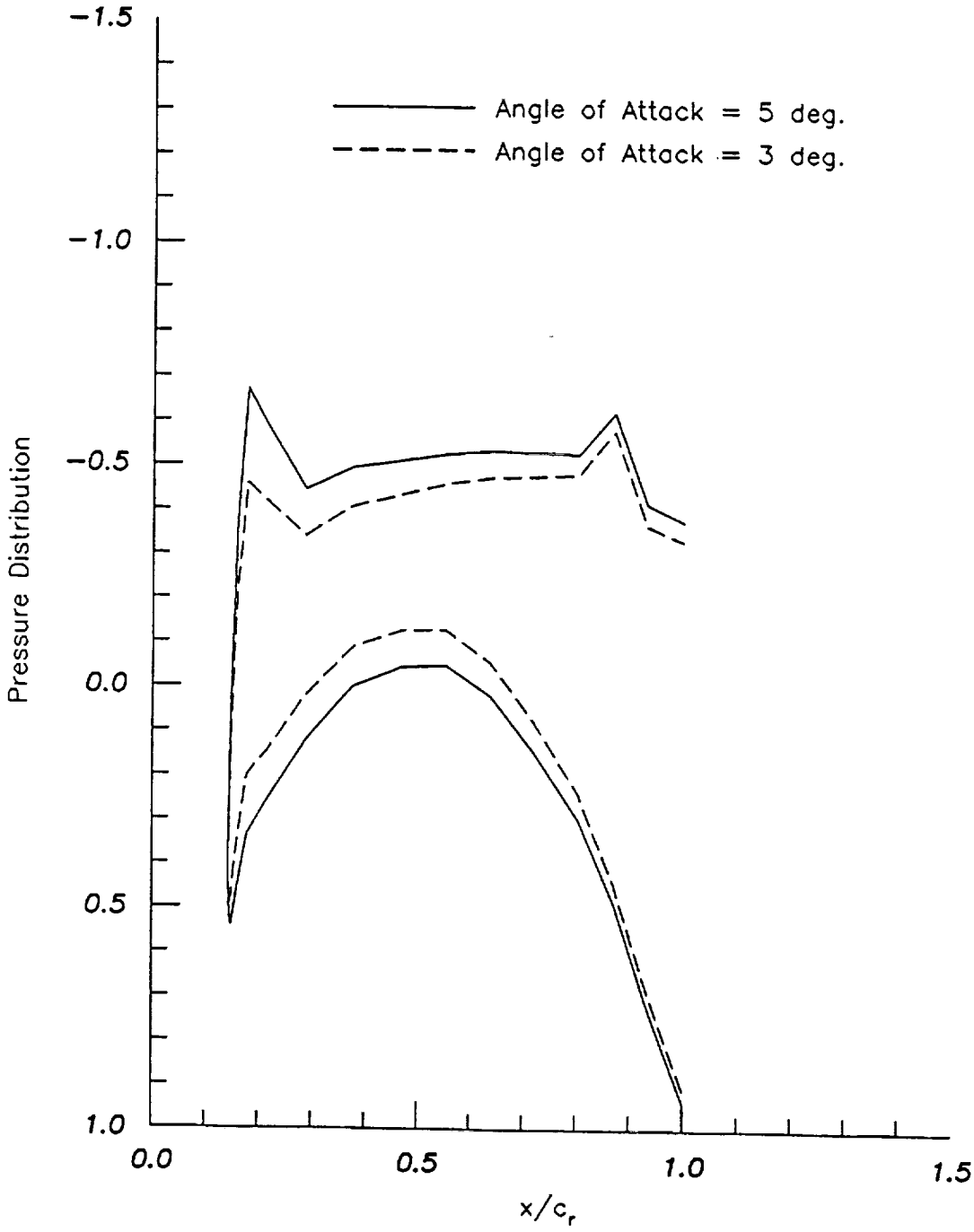
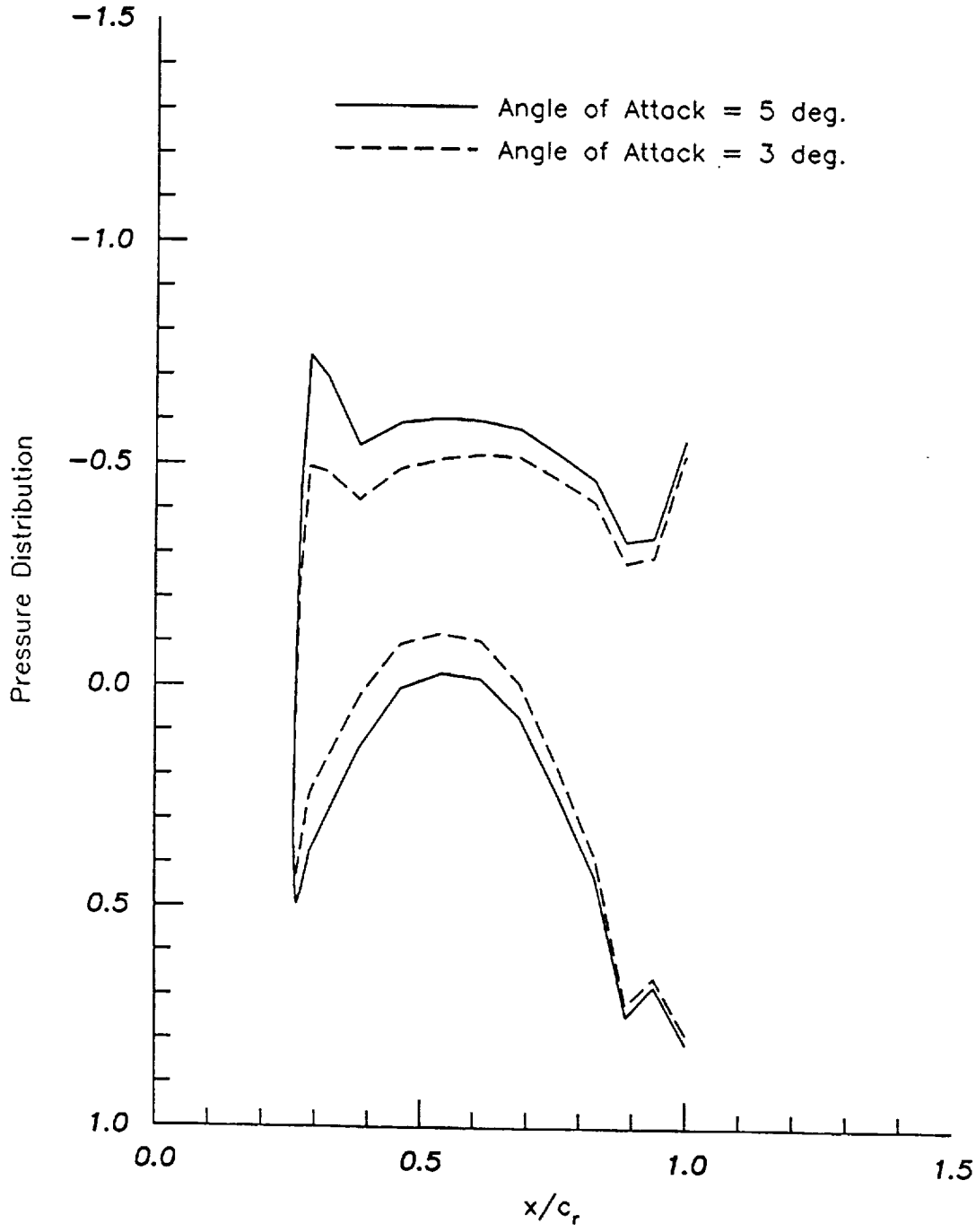


Figure 3. Grids on the Swept Wing and the Jet–Path Surface



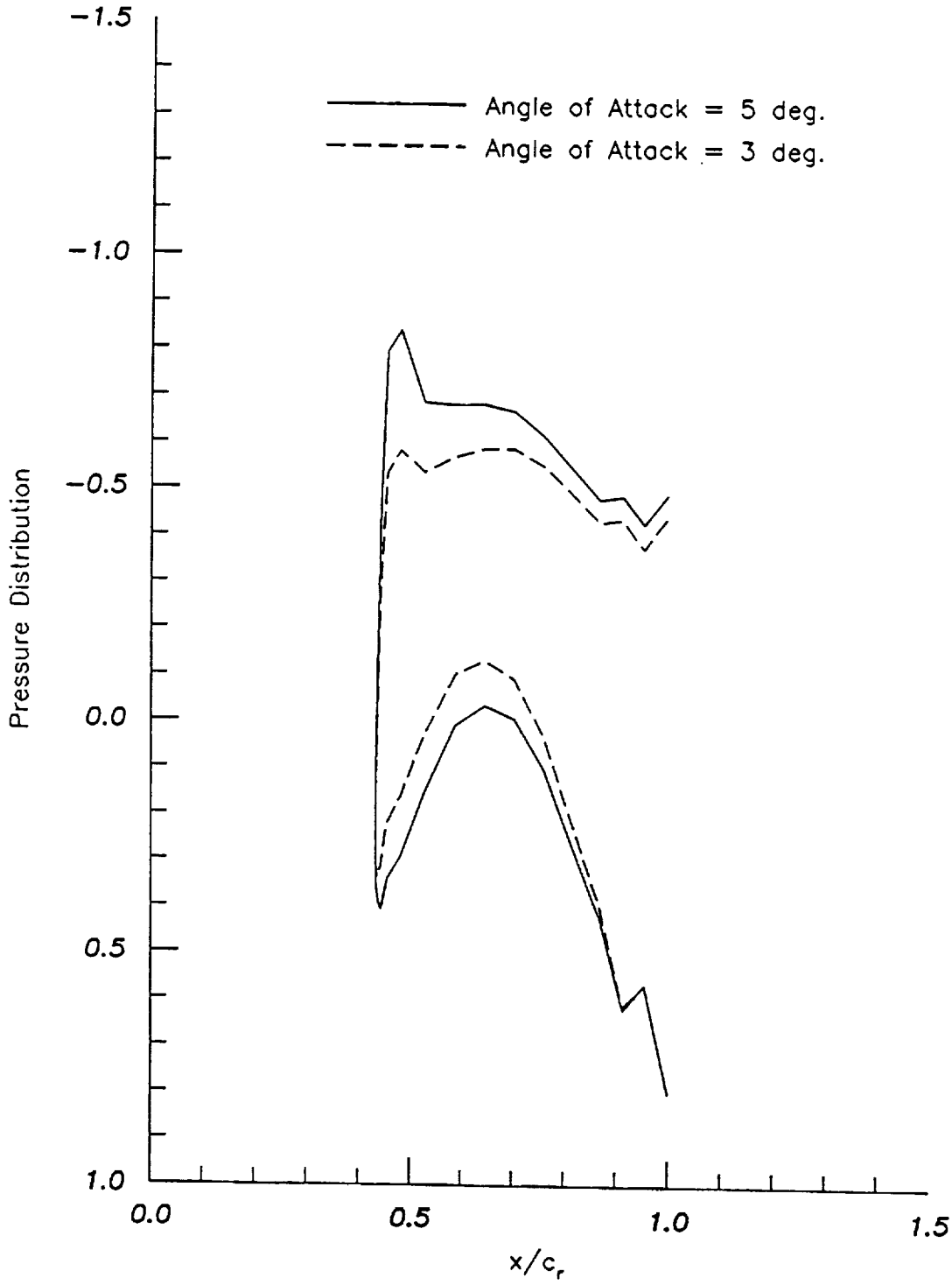
(a) 2y/b = 0.25

Figure 4. Swept Wing Chordwise Pressure Distribution at M = 0.84



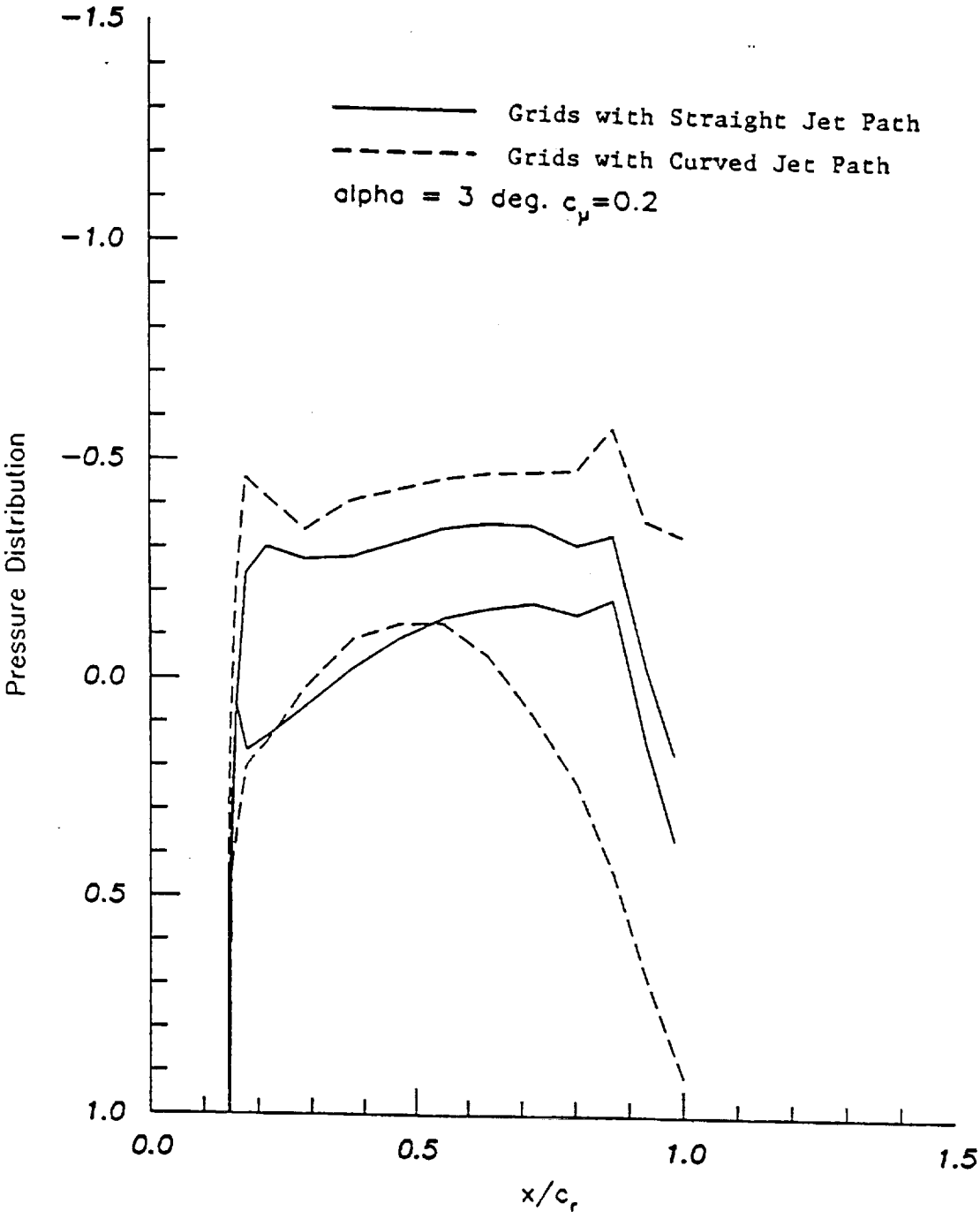
(b) 2y/b = 0.45

Figure 4. Continued



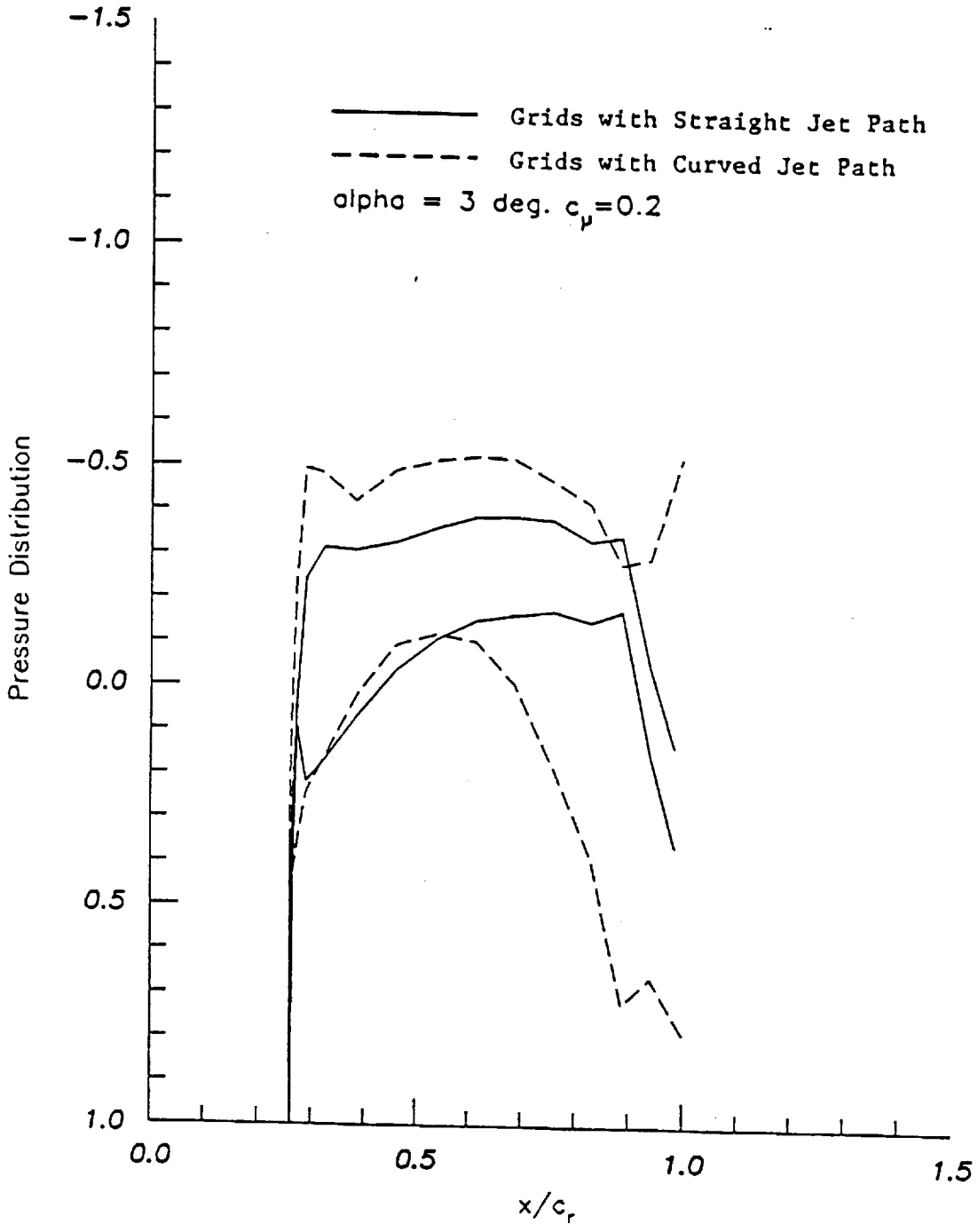
(c) 2y/b = 0.75

Figure 4. Continued



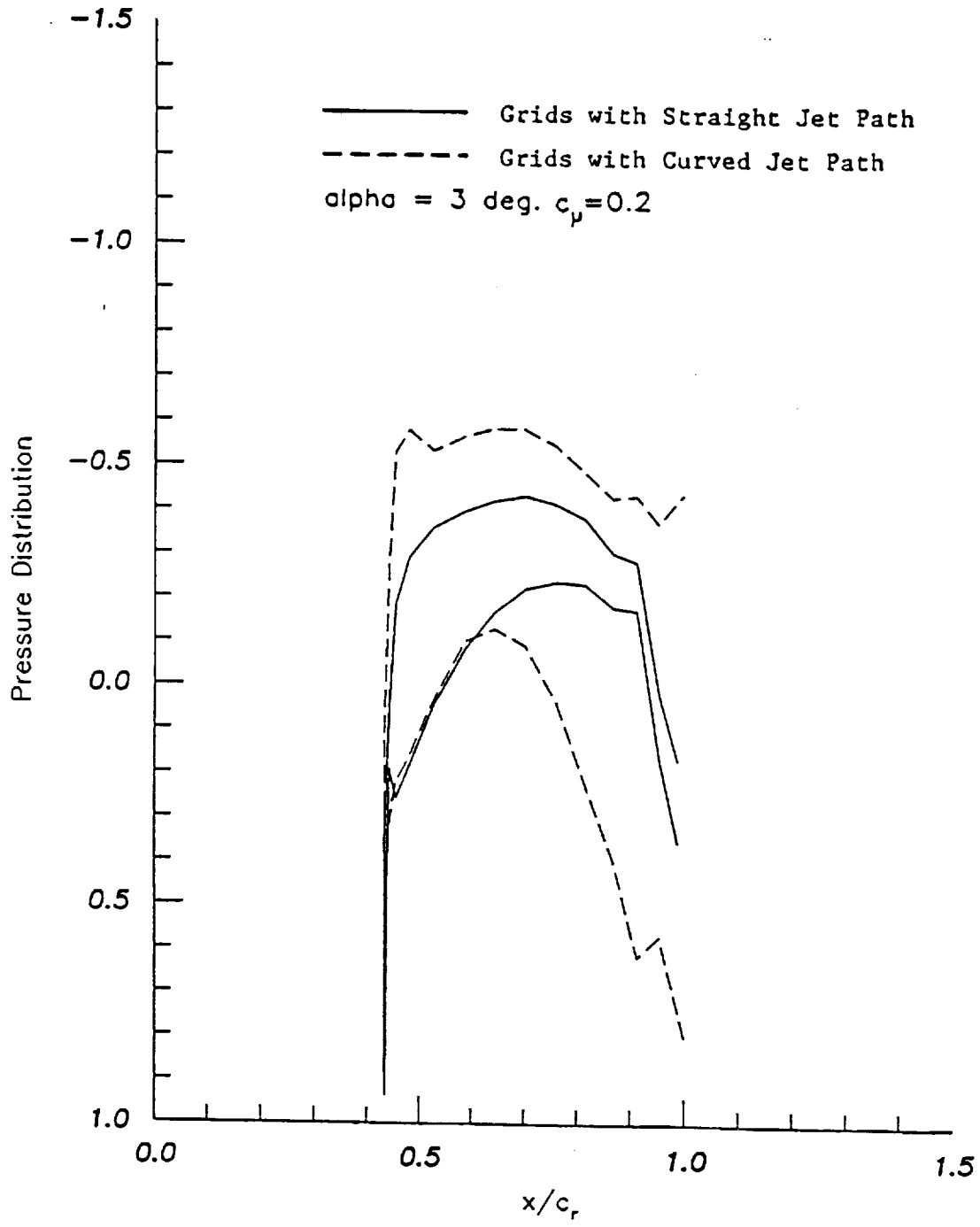
(d) 2y/b = 0.25

Figure 4. Continued



(e) 2y/b = 0.45

Figure 4. Continued



(f) 2y/b = 0.75

Figure 4. Concluded

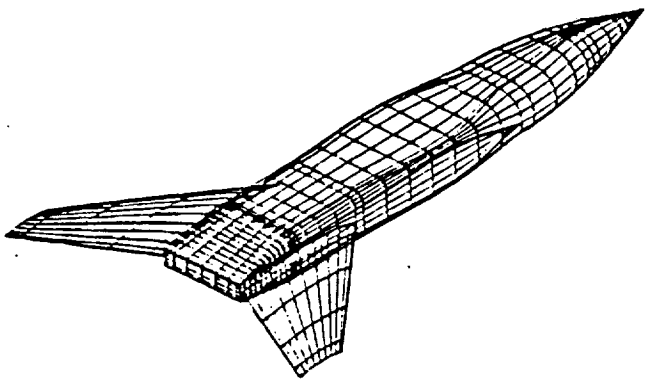


Figure 5: An Aerospace Plane Configuration (taken from Reference 4).

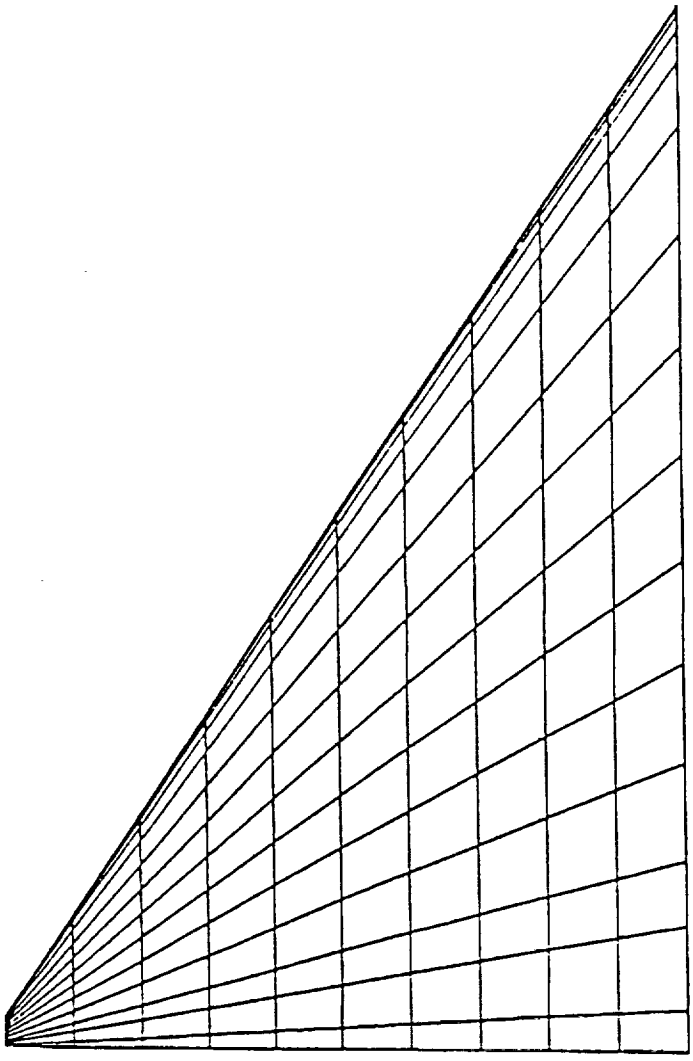
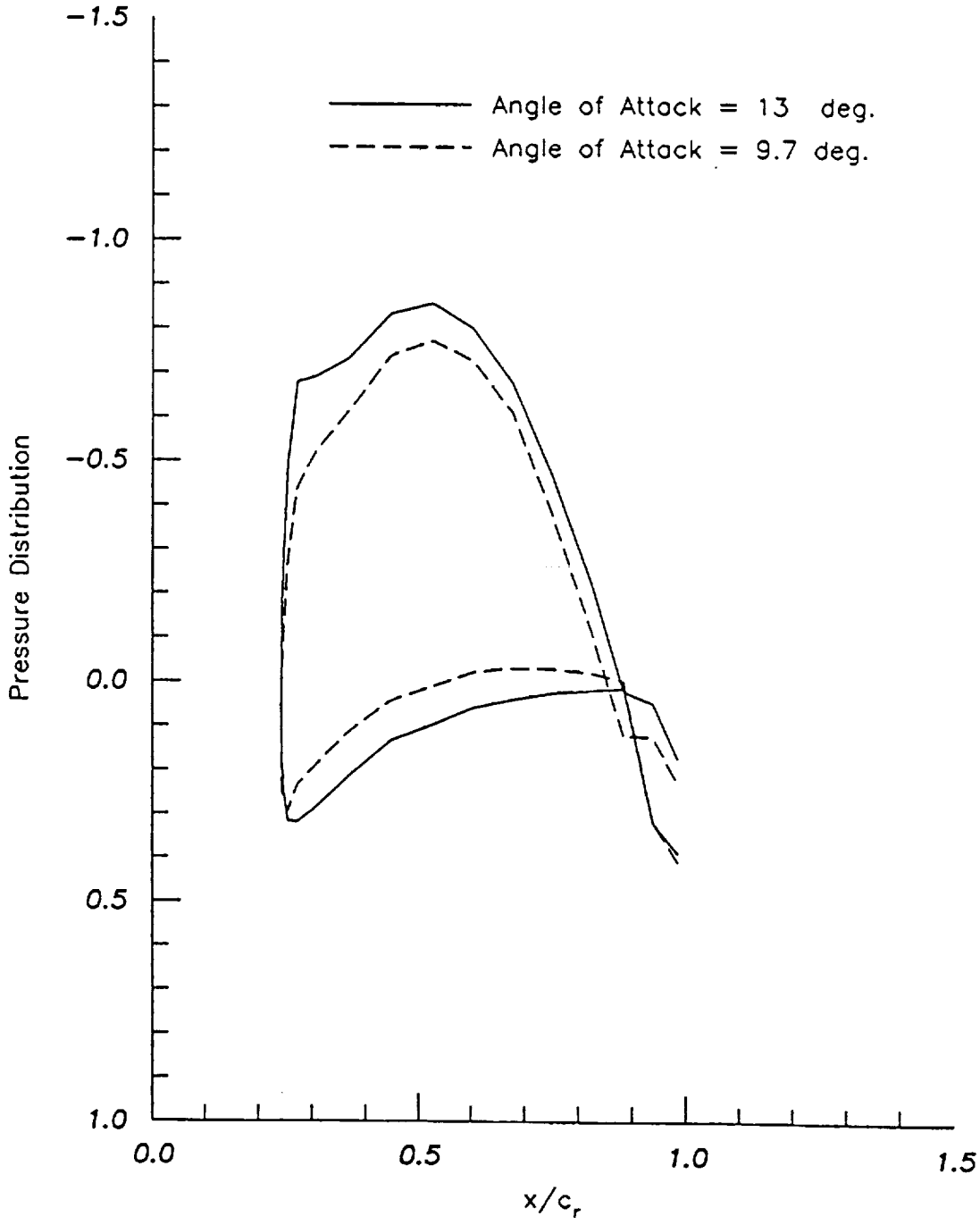
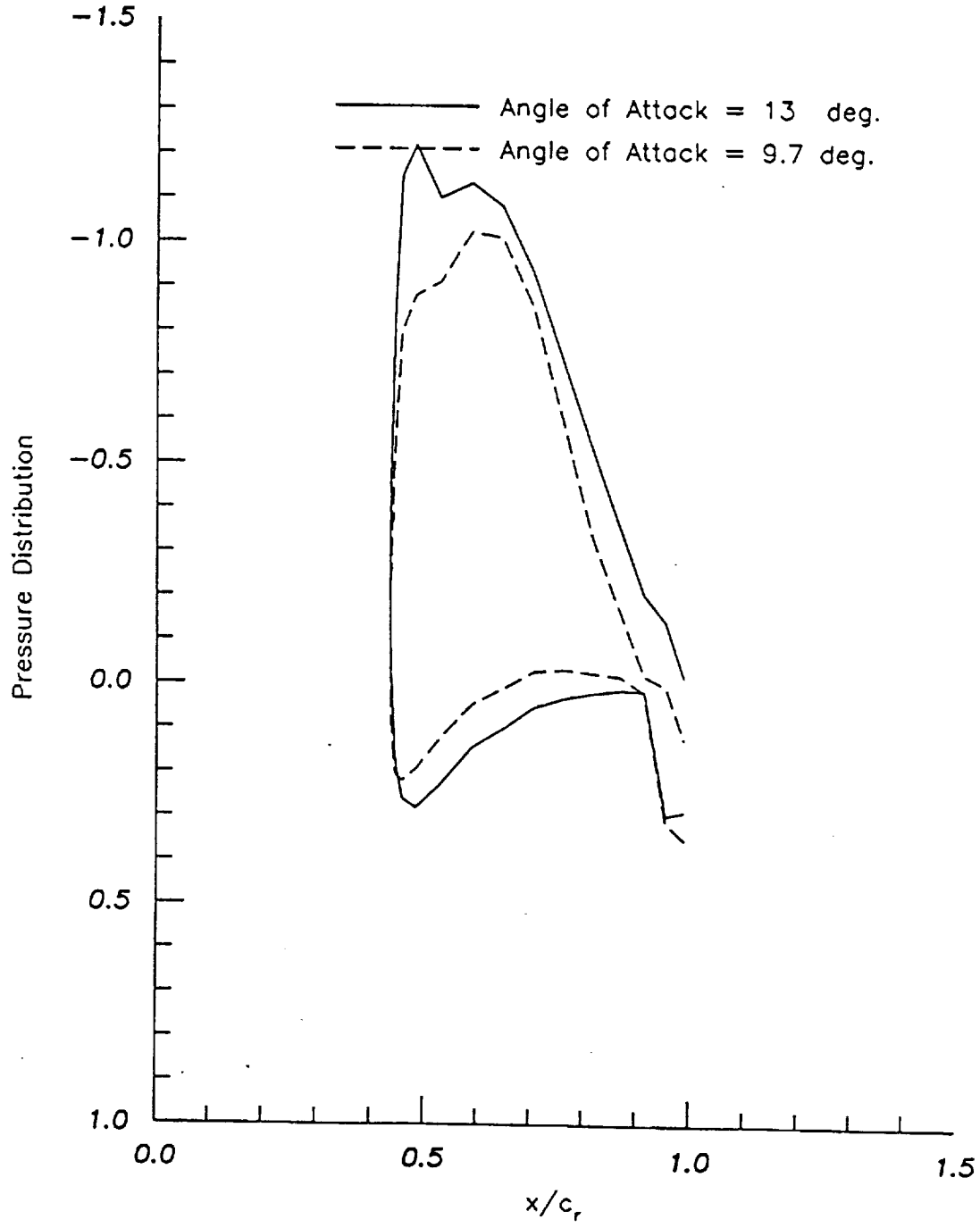


Figure 6: Grids on the 58-Degrees-Sweep Delta Wing.



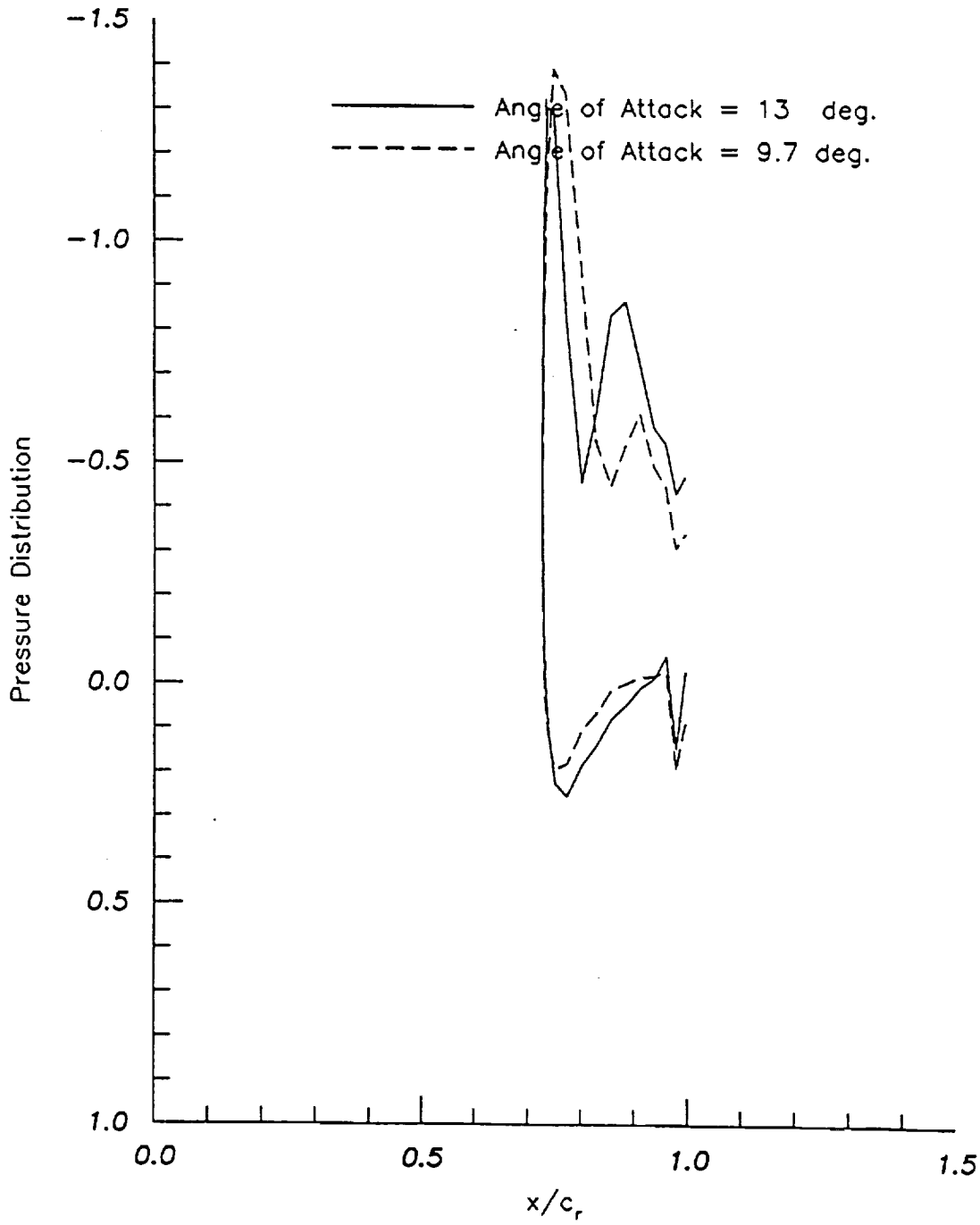
(a) 2y/b = 0.25

Figure 7: Thick Delta Wing Chordwise Pressure Distribution at M = 0.84.



(b) 2y/b = 0.45

Figure 7: Continued



(c) 2y/b = 0.75

Figure 7: Concluded.

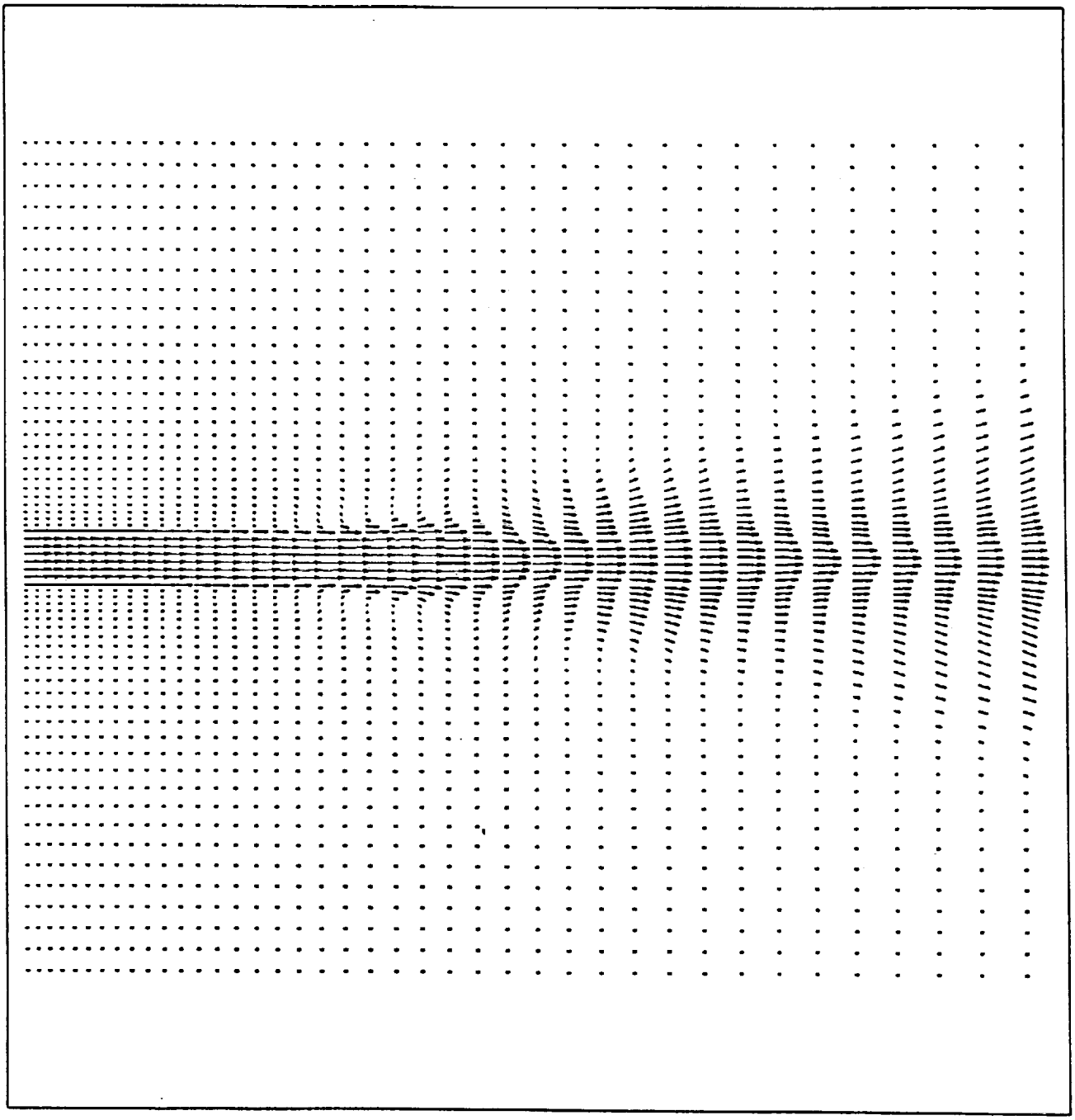
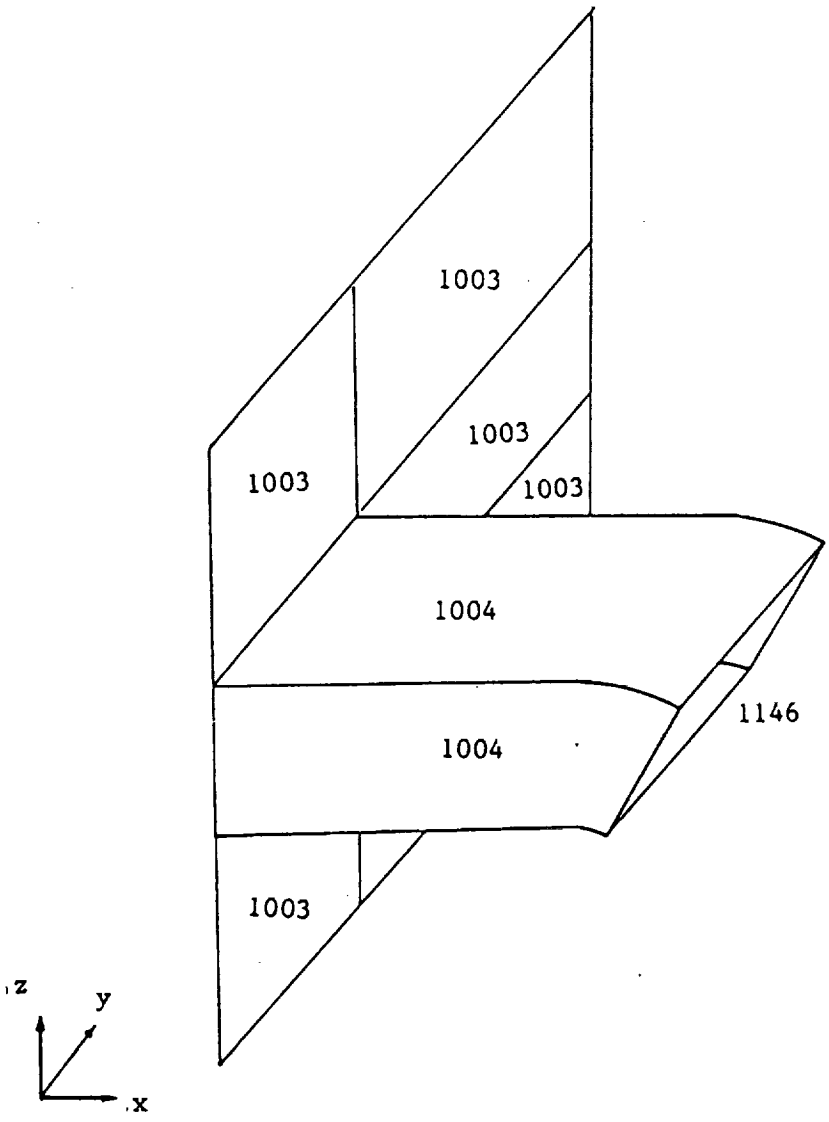
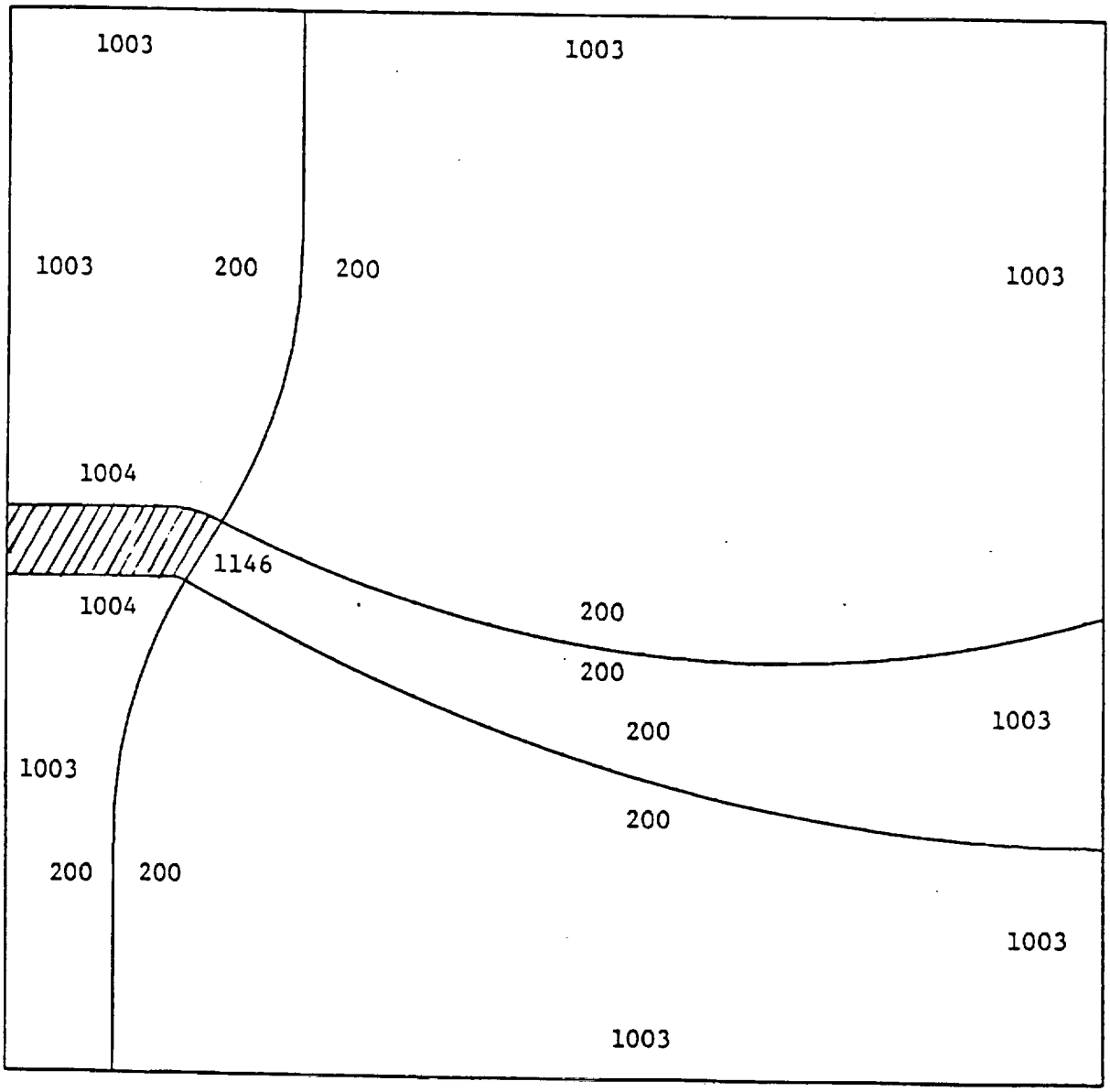


Figure 8: Section Velocity Profile of a Straight Jet.
$M_\infty = 0.1$; $M_{jet} = 0.9$; $2y/b = 0.$



(a) y-z Plane

Figure 9: Computational Block Arrangement and Boundary Condition Specification.



(b) x-z Plane

Figure 9: Computation Block Arrangement and Boundary Condition Specification.

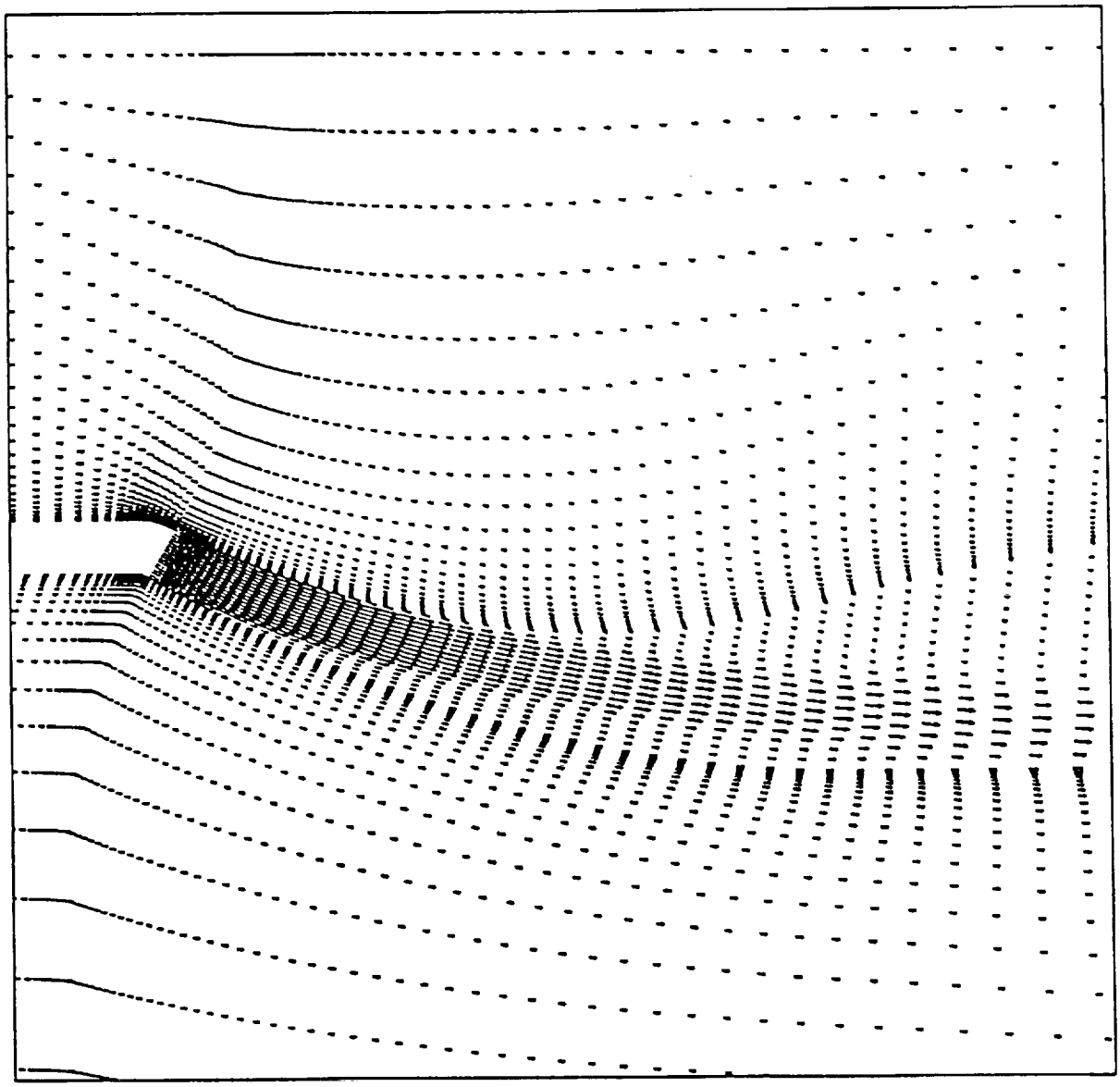


Figure 10: Sectional u-w Velocity Profile of a Jet Deflected at 30 Degrees. $2y/b = 0$; $M_{\infty} = 0.25$; $M_{jet} = 0.8$.

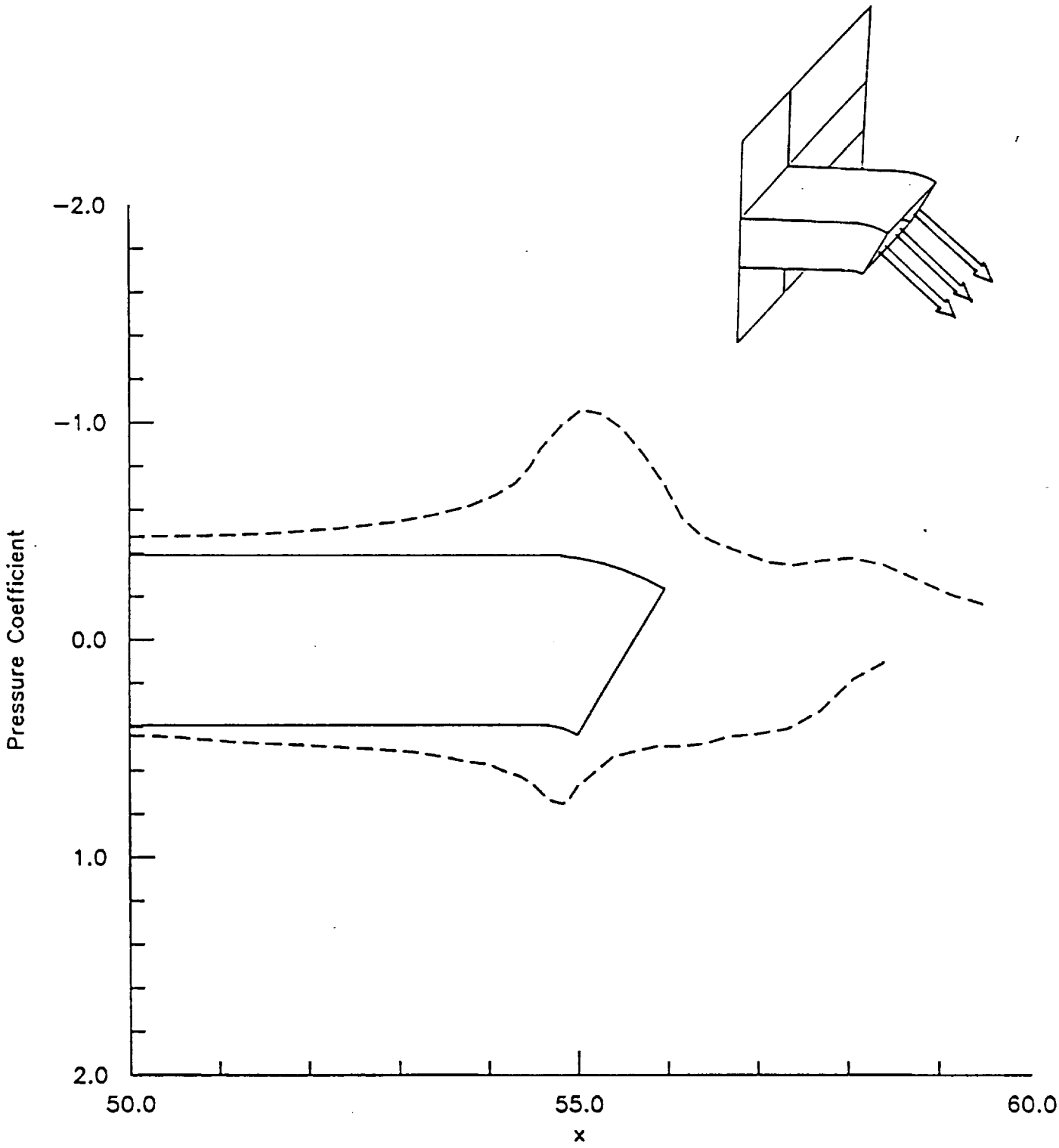


Figure 11: Pressure Distribution along a Thrust Vectoring Nozzle.
$2y/b = 0$; $M_\infty = 0.25$; $M_{jet} = 0.8$.

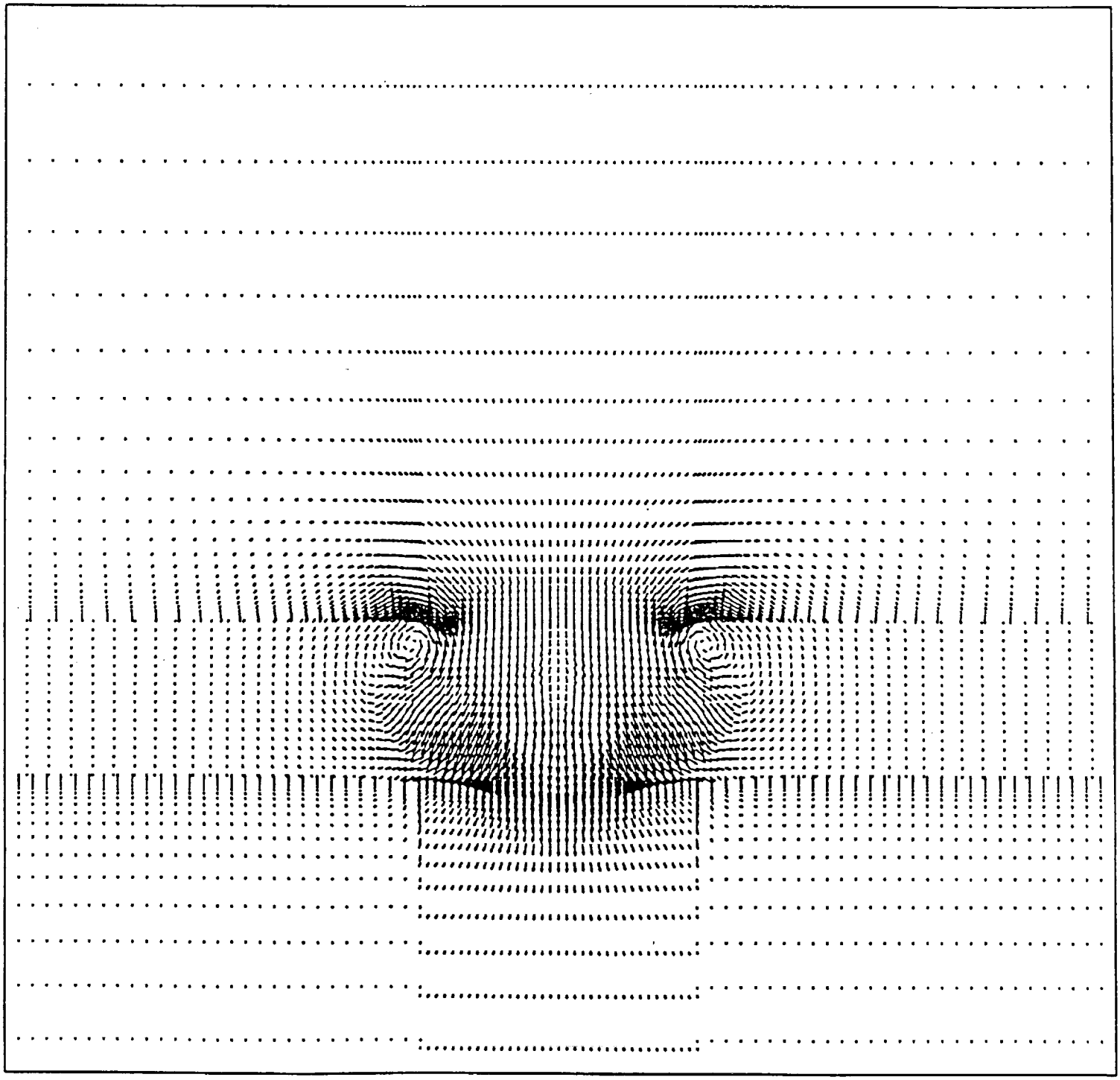


Figure 12: Cross Plane v-w Velocity Profile at x/h_{nozzle} = 10 from the Exit.

APPENDIX

Standard Boundary Conditions:

1000	free stream
1001	reflection x-z plane
1002	extrapolation
1003	inflow/outflow
1004	viscous surface
1005	inviscid surface
1006	x-y plane
1007	axisymmetric

Special Purpose Conditions:

1146	jet exit plane