

by

G. Volpe and M.J. Siclari  
Grumman Corporate Research Center  
Bethpage, New York 11714-3580

and

A. Jameson  
Department of Mechanical and Aerospace Engineering  
Princeton University  
Princeton, New Jersey 08544

### Abstract

Transonic and supersonic flows over isolated wings and fighter-type aircraft configurations are computed through the numerical solution of the compressible Euler equations. Appropriate single-mesh topologies are used in combination with a new multigrid time-stepping scheme for solving the Euler equations. C-H or C-O meshes are used for the isolated wing. A novel H-O type mesh is introduced to discretize the space about a fighter aircraft. The H-O type mesh is obtained by a sequence of two-dimensional mappings which generate separate O-meshes around successive cross sections of the aircraft. The finite volume method, which has proved quite insensitive to mesh topology, uses a five-stage Runge-Kutta time-stepping scheme to integrate the equations. Acceleration to the steady state solution is obtained by maximizing the local time step, implicit smoothing of the residuals, enthalpy damping, and an efficient multigrid technique. Results are presented for a variety of wing and aircraft configurations.

### I. Introduction

Several numerical methods<sup>1-12</sup> have been developed in the past few years for the solution of the Euler and of the Navier-Stokes equations in two- and three-dimensions. These schemes require large running times, however, resulting in their limited application in the computation of three-dimensional flows. An evaluation of an aircraft's, or an isolated wing's, flight envelope entails the computation of the flow at many free stream Mach numbers and incidences. Even with the Euler methods, such an extensive computation would be prohibitively expensive. An aerodynamic designer must, therefore, fall back on other faster, if cruder, methods, such as potential flow schemes, for the majority of calculations that he needs. In the present paper a new multigrid finite volume method for the solution of the Euler equations is presented. The numerical scheme is very efficient and uses a variety of techniques to accelerate convergence to the steady state. Its structure makes the computer program highly vectorizable. These two features make it possible to obtain converged solutions in just a few minutes of CPU time on a CRAY machine for both transonic and supersonic free stream conditions.

The basic idea is to integrate the time-dependent Euler equations until they reach a steady state. This can be done by first dividing the flow

domain into hexahedral cells and then discretizing the flow equations by writing the conservation laws in integral form for each cell. The set of discretized equations can then be integrated in time with either an explicit or an implicit scheme. In this paper a Runge-Kutta scheme is employed to perform the integration. The space discretization is accomplished in a variety of ways in order to tailor the mesh topology to the configuration to be studied. Some topologies, such as C-H and C-O grids, are better suited to wing-alone geometries. Methods for the generation of such grids have been available for some time (see, for example, Ref. 3, 13). A grid that is ideally suited for aircraft with highly swept wings is a grid of the H-O type. Such a grid is obtained by the union of two-dimensional meshes generated in planes normal to the longitudinal axis of the aircraft. Each 2-D mesh is obtained by mapping the local cross section of the wing-body configuration into a near circle. A polar-like coordinate grid in the transformed plane then gives a grid in the physical plane that not only gives a good concentration of points in regions of high curvature but also is body-conforming. This is a highly desirable feature since it simplifies the application of boundary conditions. Planes can be spaced at will in the longitudinal direction. The resulting 3-D grid is of the O-type if viewed in the normal planes, and of the H-type if viewed in planes parallel to the longitudinal axis.

The compressible Euler equations are discretized by requiring conservation of mass, momentum, and energy in each cell of the mesh, and these are integrated by a finite-volume scheme originally due to Jameson (Ref. 14). The scheme is second-order accurate (except in the vicinity of shock waves) and is very efficient. Dissipative terms, which vanish in the limit of zero cell volume, are added to the discretized equations in order to suppress odd-even point decoupling and oscillations near shock waves. The equations are integrated with a five-stage Runge-Kutta scheme. Very high values of the Courant number can be used by making the time step vary locally and by smoothing of the residuals. Additional acceleration is provided by enthalpy damping, and, finally, a multigrid sequencing of the meshes is invoked to turn this interactive integration scheme into a very rapid method for reaching the steady state. A description of the numerical scheme is given in Section 2. Section 3 discusses the mesh topologies that have been used, and some computed results are presented in Section 4.

## II. Finite-Volume Time-Stepping Scheme

### Finite-Volume Discretization

The conservation laws for mass, momentum and energy over a three-dimensional domain  $\Omega$  enclosed by a surface boundary  $\partial\Omega$  are expressed by the following integral form of the Euler equations:

$$\frac{\partial}{\partial t} \iiint_{\Omega} W^{(i)} d\Omega + \iint_{\partial\Omega} \bar{F}^i \cdot d\bar{S} = 0 \quad (1)$$

where  $W^{(i)}$  is a conserved quantity and  $\bar{F}^{(i)}$  is its corresponding flux term. The first term then is simply the rate of change over time in the quantity, and the second is the total flux into (out of) the domain. For mass conservation

$$W^{(1)} = \rho, \quad \bar{F}^{(1)} = [\rho u, \rho v, \rho w]^T; \quad (2a)$$

for momentum conservation

$$W^{(2)} = \rho u, \quad \bar{F}^{(2)} = [\rho u^2 + p, \rho uv, \rho uw]^T, \quad (2b)$$

$$W^{(3)} = \rho v, \quad \bar{F}^{(3)} = [\rho uv, \rho v^2 + p, \rho vw]^T, \quad (2c)$$

$$W^{(4)} = \rho w, \quad \bar{F}^{(4)} = [\rho uw, \rho vw, \rho w^2 + p]^T; \quad (2d)$$

and for energy conservation

$$W^{(5)} = \rho E, \quad \bar{F}^{(5)} = [\rho Hu, \rho Hv, \rho Hw]^T. \quad (2e)$$

In the expressions above,  $p$ ,  $\rho$ ,  $u$ ,  $v$ ,  $w$ ,  $E$  and  $H$  are, respectively, the pressure, density, Cartesian velocity components, total energy, and total enthalpy. For a perfect gas, whose ratio of specific heats is  $\gamma$ , we have the additional relations

$$E = \frac{p}{(\gamma-1)\rho} + \frac{1}{2} (u^2 + v^2 + w^2), \quad (3a)$$

$$H = E + \frac{p}{\rho}. \quad (3b)$$

Equation (1) can, of course, be applied to any subdomain  $\Omega_N$  of  $\Omega$ . Thus, by dividing the entire domain  $\Omega$  into a large number of hexahedral cells and using Eq (1) to estimate the rate of change of  $W^{(i)}$  in each subdomain, we obtain discrete approximations of the conservation laws expressed in Eq (1) over the entire domain. In the present work, the dependent flow variables  $W^{(i)}$  are assumed to be stored at the nodes of the mesh, in contrast to previous finite volume schemes in which the flow variables were stored at the cell centers. This leads to a revised time stepping and multigrid formulations which have proved particularly efficient in practice. The control volume for each node  $(i, j, k)$  is given by the union of the eight cells meeting at that node. Applying Eq (1) to each of the eight cells, then, for the control volume enclosing the  $(i, j, k)$  node, Eq (1) becomes

$$\frac{d}{dt} \left( \sum_{n=1}^8 V_n \right) W_n^{(i)} + \sum_{n=1}^8 Q_n^{(i)} = 0 \quad (4)$$

where the summation sign runs over the eight cells meeting at  $(i, j, k)$ ,  $V_n$  is the volume of the  $n$ th cell and  $Q_n^{(i)}$  is the flux balance for that cell.  $Q_n^{(i)}$  is approximated by

$$Q_n^{(i)} = \sum_l \bar{F}_l^{(i)} \cdot \bar{S}_l \quad (5)$$

where  $\bar{S}_l$  is the directed face area of the  $l$ th face (of the  $n$ th cell) and  $\bar{F}_l^{(i)}$  is an estimate of the mean flux vector across that face. Since the fluxes across internal faces cancel, contributions to the flux balance through the control volume are given only by its external faces, as one would expect.

Typically, the mesh is generated by the introduction of curvilinear coordinates  $\xi$ ,  $\eta$ ,  $\zeta$ . The cells then correspond to cubes in the transformed space, which is specified by a transformation of the type

$$\begin{aligned} \xi &= \xi(x, y, z) \\ \eta &= \eta(x, y, z) \\ \zeta &= \zeta(x, y, z) \end{aligned} \quad (6)$$

Here we assume that the transformation is constant in time. The volume of each cell can be identified with the Jacobian  $J$  of the inverse transformation, which is given by

$$\begin{aligned} J &= x_{\xi} (y_{\eta} z_{\zeta} - z_{\eta} y_{\zeta}) - y_{\xi} (x_{\eta} z_{\zeta} - z_{\eta} x_{\zeta}) \\ &+ z_{\xi} (x_{\eta} y_{\zeta} - y_{\eta} x_{\zeta}) \end{aligned} \quad (7)$$

and the total flux  $Q_n^{(i)}$  through each cell can be computed from

$$Q_n = \frac{\partial f}{\partial \xi} + \frac{\partial g}{\partial \eta} + \frac{\partial h}{\partial \zeta} \quad (8)$$

where

$$f = J [\rho \tilde{u}, \rho u \tilde{u} + \xi_x p, \rho v \tilde{u} + \xi_y p, \rho w \tilde{u} + \xi_z p, \tilde{u}(E+p)]^T$$

$$g = J [\rho \tilde{v}, \rho u \tilde{v} + \eta_x p, \rho v \tilde{v} + \eta_y p, \rho w \tilde{v} + \eta_z p, \tilde{v}(E+p)]^T$$

$$h = J [\rho \tilde{w}, \rho u \tilde{w} + \zeta_x p, \rho v \tilde{w} + \zeta_y p, \rho w \tilde{w} + \zeta_z p, \tilde{w}(E+p)]^T$$

$\tilde{u}$ ,  $\tilde{v}$ ,  $\tilde{w}$  are contravariant velocity components given by

$$\tilde{u} = \xi_x u + \xi_y v + \xi_z w$$

$$\tilde{v} = \eta_x u + \eta_y v + \eta_z w \quad (9)$$

$$\tilde{w} = \zeta_x u + \zeta_y v + \zeta_z w$$

The metric quantities  $J\xi_x$ ,  $J\xi_y$ ,  $J\xi_z$  can be identified with the directed face areas of the mesh cells. On each face they can be conveniently computed as the components in the Cartesian coordinate directions of the vector obtained by taking the cross-product of the two vectors joining opposite corners of the cell face. The quantities  $u$ ,  $v$ ,  $w$ , needed in Eq (9), are taken as the average of the values stored at the four corners of the cell face. Note that this formulation has the property that the discrete equations are exactly satisfied by uniform flow.

Background dissipation must be added over the entire domain to prevent the appearance of odd and even point oscillations which would give a net contribution of zero to the flux balance in each interior cell. Such a mode might be suppressed in the steady state solution by the boundary conditions, but the convergence rate can be adversely affected by such a mode of oscillations. Dissipation is also needed to eliminate oscillations near shock waves and contact discontinuities. Several non-oscillatory schemes for scalar conservation laws have been designed in the past few years using the concept of total variation diminishing (TVD) difference schemes introduced by Harten,<sup>15</sup> and the ideas have been extended to the treatment of the Euler equations by splitting the flux into components based on the characteristic speeds and by applying the TVD construction separately to each component. Flux splitting allows the construction of terms which introduce just enough dissipation to prevent oscillations. This in turn permits the clear capture of shock spread out over only one or two cells. An alternate approach, which is followed here, is the use of adaptive coefficients for the dissipation terms. In practice, this has been shown to work extremely well. The approach is to use a low background level everywhere in the domain and increase it in the neighborhood of a shock wave. It has been found that the second difference of the pressure provides an effective measure for detecting where the additional dissipation is needed.

The scheme is implemented by substituting for  $Q_n^{(i)}$  in Eq (4) a term  $Q_n^{(i)} - D_n^{(i)}$  where  $D_n^{(i)}$  is a dissipative flux constructed in such a way as to preserve the conservation form of the equations.

$D_n^{(i)}$  is the sum of three dissipation terms constructed for each of the three coordinate directions. Thus, at a node  $(i, j, k)$  the dissipation term in the  $i$  (or  $x$ ) direction is given by

$$D_n^{(i)} = d_{i+1,j,k} - d_{i,j,k}$$

where

$$d_{i,j,k} = r[\epsilon^{(2)} - \epsilon^{(4)}\delta_x^2](w_{i,j,k}^{(i)} - w_{i-1,j,k}^{(i)}) \quad (10)$$

and  $\delta_x^2$  is a second difference central operator.

In Eq (10)  $\epsilon^{(2)}$  and  $\epsilon^{(4)}$  are adaptive coefficients and  $r$  is a scaling factor proportional to an estimate of the maximum local wave speed. In an explicit scheme such as the one we use, a measure of the time it takes for the fastest wave to cross a mesh interval is given by the local time step  $\Delta t^*$ . Thus,  $r$  can be made proportional to  $1/\Delta t^*$ .  $\epsilon^{(4)}$  provides the background dissipation, which is of third order in smooth regions of the flow.  $\epsilon^{(2)}$  controls the dissipation near shock waves and is made proportional to a normalized second difference of the pressure

$$\epsilon^{(2)} = v_{i,j,k} = \frac{|p_{i+1,j,k} - 2p_{i,j,k} + p_{i-1,j,k}|}{|p_{i+1,j,k} + 2p_{i,j,k} + p_{i-1,j,k}|}$$

This part of the dissipation is thus of third order everywhere except near regions of steep pressure gradients. Dissipation terms in the  $j$  (or  $y$ ) and  $k$  (or  $z$ ) directions are constructed in a similar manner.

The low-order dissipative terms effectively introduce upwind biasing. Instead of blending low- and high-order dissipative terms, one may produce a nonoscillatory scheme by eliminating the low-order terms entirely and introducing flow limiters into the high-order terms.<sup>16</sup> This approach leads to a class of TVD schemes.

#### Time Stepping

Since we are working with a mesh that is independent of time, Eq (4) can be re-written in the form

$$\frac{d}{dt} w_{i,j,k}^{(i)} + R(w^{(i)}) = 0 \quad (11)$$

where the residual  $R(w^{(i)})$  is given by

$$R(w^{(i)}) = \frac{1}{V_{i,j,k}} (Q_{i,j,k} - D_{i,j,k}) \quad (12)$$

Here,  $V_{ijk}$ ,  $Q_{ijk}$ ,  $D_{ijk}$  are, respectively, the value of the control volume, the total flux, and the total dissipative flux through the control volume. The steady state solution is obtained by integrating Eq (11) using a five-stage Runge-Kutta scheme with a locally varying time step  $\Delta t$ . This scheme has been selected for its stability and damping properties rather than its accuracy, since the objective is the attainment of a steady state as rapidly as possible. The stability properties of the class of multi-stage scheme to which the present scheme belongs are discussed in detail by Jameson.<sup>17</sup>

Using the superscript notation now to denote a particular stage of the time step the advance of each variable  $W$  from time  $t$  to time  $(t + \Delta t)$  is given by

$$w^{(0)} = w(t) \quad (13a)$$

$$W^{(1)} = W(t) - \alpha_1 \Delta t R(W^{(0)}) \quad (13b)$$

$$W^{(2)} = W(t) - \alpha_2 \Delta t R(W^{(1)}) \quad (13c)$$

$$W^{(3)} = W(t) - \alpha_3 \Delta t R(W^{(2)}) \quad (13d)$$

$$W^{(4)} = W(t) - \alpha_4 \Delta t R(W^{(3)}) \quad (13e)$$

$$W^{(5)} = W(t) - \Delta t R(W^{(4)}) \quad (13f)$$

$$W(t + \Delta t) = W^{(5)} \quad (13g)$$

with  $\alpha_1 = \frac{1}{4}$ ,  $\alpha_2 = \frac{1}{6}$ ,  $\alpha_3 = \frac{3}{8}$ ,  $\alpha_4 = \frac{1}{2}$ . In practice it has been possible to freeze the artificial dissipation terms at the values computed at the second stage, a strategy which noticeably reduces the computational effort and also increases the stability margin for the dissipative terms (Ref. 16).

Computational efficiency is greatly increased by smoothing the residuals. The stability limit on the local Courant number sets restrictions on the maximum permissible time step. As discussed in Ref. 14, smoothing of the residuals increases the support of the numerical scheme, thus relaxing the restriction on the time step. Thus, at each point the residual is replaced by an average of its neighbors. This average is computed implicitly by solving

$$(1 - \epsilon_x \delta_x^2)(1 - \epsilon_y \delta_y^2)(1 - \epsilon_z \delta_z^2) \bar{R} = R \quad (14)$$

where  $\delta_x^2, \delta_y^2, \delta_z^2$  are second difference operators, and  $\epsilon_x, \epsilon_y, \epsilon_z$  are smoothing coefficients.

For a one-dimensional problem, and in the absence of dissipation, it can be shown (see Ref. 17) that stability can be maintained for any Courant number  $\lambda$  if the smoothing parameter is chosen in such a way that

$$\epsilon \geq \frac{1}{4} \left[ \left( \frac{\lambda}{\lambda^*} \right)^2 - 1 \right]$$

where  $\lambda^*$  denotes the stability limit in the unsmoothed scheme. In practice it has been found that the fastest rate of convergence is usually obtained by setting  $\lambda$  to about three times the value of  $\lambda^*$  and the smallest possible values of the smoothing parameters which still maintain stability.

Another useful device for accelerating convergence is enthalpy damping. Assuming that the flow is homoenthalpic, we can assume the time rate of change of each dependent variable  $W$  to be

proportional to the difference between the local value of the enthalpy and its (constant) value at infinity, which is also its steady state value. Thus, we set

$$\frac{\partial W}{\partial t} + \beta W(H - H_\infty) = 0 \quad (15)$$

where  $\beta$  is a user-defined constant. We can apply this implicitly at the end of each time step through the use of

$$W = \tilde{W} + \beta \Delta t W(\tilde{H} - H_\infty) = 0 \quad (16)$$

where  $\tilde{W}$  denotes the value of  $W$  computed at the end of the time step and  $W$  is the new, improved estimate. Equation (15) is applied for each dependent variable, with only a slight modification for ( $\rho E$ ). For this last variable Eq (16) is modified to the form

$$(\rho E) - (\tilde{\rho E}) + \beta \Delta t [\rho E + \tilde{p} - \tilde{\rho H}_\infty] = 0$$

### Multigrid Scheme

The objective of a successful multigrid scheme is to transfer part of the effort of tracking the evolution of a system of discrete equations to a sequence of successively coarser meshes. On the coarser grids, the use of larger control volumes permits the tracking of the evolution on the larger scale. In the context of the Runge-Kutta time-stepping scheme being used, this is made possible by the use of larger time steps on the coarser grids without violating the stability bounds. An additional benefit of performing some of the time steps on coarser grids is the fact that the computational effort required is greatly reduced on the coarser meshes. This more than offsets the cost of passing information up and down through the grids.

In the present scheme the coarser meshes are generated by eliminating alternate points in each coordinate directions. This simplifies the task of transferring information through the several mesh levels. On passing from grid level  $k$  to a coarser grid level  $k + 1$  the first step of the Runge-Kutta operation described by Eq (10a) is modified to

$$W_{k+1}^{(0)} = T_{k+1,k} W_k \quad (17)$$

where  $T_{k+1,k}$  denotes a transfer operator. Since each point on the coarser mesh coincides with a point on the finer mesh, the transfer rule implied by Eq (17) is simply to assign to  $W_{k+1}^{(0)}$  the value on the coincident point in mesh level  $k$ . The subsequent steps of the Runge-Kutta operation are also modified in the sense that the integration is driven by a modified residual

$$W_{k+1}^{(1)} = W_{k+1}^{(0)} - \alpha_1 \Delta t (R_{k+1}^{(0)} + P_{k+1}) \quad (18a)$$

$$W_{k+1}^{(2)} = W_{k+1}^{(0)} - \alpha_2 \Delta t (R_{k+1}^{(1)} + P_{k+1}) \quad (18b)$$

and so on. In Eq (18)  $R_{k+1}$  is the residual computed on the mesh  $k + 1$ , and

$$P_{k+1} = Q_{k+1,k} R_k(W_k) - R_{k+1}^{(0)} \quad (19)$$

where  $Q_{k+1,k}$  is another transfer operator. In the present scheme  $Q_{k+1,k} R_k(W_k)$  is a weighted average of the residuals at the 27 points nearest the coincident point on mesh level  $k$  (including the point itself).

Typically, one time step is performed on each mesh level until the coarsest (desired) mesh is reached. At this point, the end result of the Runge-Kutta step

$$W_{k+1}(t + \Delta t) = W_{k+1}^{(5)} \quad (20)$$

is transferred back to grid level  $K$  to provide an improved value. Thus, denoting by  $W_{k+1}^+$  the value of  $W_{k+1}$  resulting from both the correction calculated by Eq (20) at the end of the time step and any correction transferred from a still coarser grid  $k + 2$ , the value of  $W$  at grid level  $k$  is set to

$$W_k^+ = W_k + I_{k,k+1}(W_{k+1}^+ - W_{k+1}^{(0)})$$

where  $I_{k,k+1}$  is some interpolation operator. At points that are coincident on mesh levels  $k$  and  $k+1$  the value of  $W$  is simply transferred. At intermediate points the correction is obtained by trilinear interpolation. Usually, a V-cycle, in which correction data are transferred up through the meshes without invoking a time-integration step is performed. Occasionally better convergence rates have been obtained with a W-cycle, in which time steps are performed on each of the meshes on the way up.

### III. Mesh Generation

Several mesh topologies have been used with this numerical scheme which has proved quite robust on all. In each case the space discretization results in a number of hexahedral cells. The only effect of topology on the application of the scheme is with regard to boundary condition.

#### Wing Alone

An often used mesh topology for an isolated wing is a C-H type grid. This body conforming mesh, which is of C type in streamwise vertical planes, is generated by the introduction of sheared parabolic coordinates. This is accomplished by a two-stage mapping procedure. The first stage introduces parabolic coordinates by the transformation

$$(\bar{x} + i\bar{y})^2 = [x - x_0(z) + i(y - y_0(z))]/t(z)$$

$$\bar{z} = z$$

where  $z$  is the spanwise coordinate,  $t(z)$  is a scaling factor which can be used to control the number of cells covering the wing, and  $x_0(z)$  and  $y_0(z)$  are the coordinates of a singular line lying just inside the leading edge. The effect of this transformation is to unwrap the wing to a shallow bump  $\bar{Y} = S(\bar{X}, \bar{Z})$ . The second stage is a shearing transformation

$$X = \bar{X}, \quad Y = \bar{Y} - S(\bar{X}, \bar{Z}), \quad Z = \bar{Z}$$

which maps the wing to the coordinate surface  $Y = 0$ . The mesh is then constructed by the reverse sequence of mappings from a rectangular grid in the  $X, Y, Z$  coordinate system. Meshes of this type contain badly distorted cells in the neighborhood of the singular line where it passes into the flowfield beyond the wing tip. These cells, which have a very high aspect ratio and a triangular cross section, present a severe test of the robustness of the multigrid scheme.

Typically, we assume the wing to be attached to a mirror image of itself. Thus, symmetry-type boundary conditions are imposed along one face of the computational cube. Flow tangency is enforced at node points lying on the wing and continuity is enforced on the slit in the mesh behind the wing and past the wing tip. These boundary conditions are enforced during each stage of the Runge-Kutta time step. In addition, during the computation of the residuals, fluxes through cell faces lying on the wing are set to zero. In the far field, appropriate inflow or outflow conditions are imposed using Riemann invariants to attenuate the spurious reflection of outgoing waves into the field. A more complete presentation of far field boundary conditions is contained in Ref. 3.

A C-H type grid concentrates mesh points near the leading edge of the wing, but adequate resolution of the wing tip region is obtained only by concentrating a sufficient number of vertical planes there. However, this would not be an efficient use of grid points. Alternate topologies, which give proper concentration of points at the leading edge and at the wing tip are meshes of the C-O and O-O type. Such meshes, which are of the C or O type in the streamwise direction and of the O type in the spanwise direction, can be generated by a transfinite interpolation technique due to Ericsson.<sup>13</sup> An efficient code for generating such meshes has been developed by B. Wedan.\* The numerical scheme has been exercised on a mesh of the C-O type. Boundary conditions with such a topology are similar to those for C-H mesh. The major difference lies in the fact that with the C-O topology the mesh has a fold at the wing tip, and the top and bottom half of the mesh coincide there. Continuity of the flow variables, thus, has to be enforced along an entire side of the computational cube.

#### Aircraft Configurations

The three-dimensional mesh around a general aircraft configuration can be obtained by

-----  
\*The authors are indebted to Mr. Wedan of NASA Langley Research Center for making a copy of the code available to them.

generating two-dimensional meshes around pre-selected cross sections of the aircraft in planes normal to the longitudinal axis of the aircraft. Upstream and downstream of the aircraft, the mesh is a continuation of the first and last stations on the aircraft, respectively. The scheme was originally developed by Moretti<sup>18</sup> and has been used by Siclari<sup>19</sup> in the computation of supersonic full potential flows by space marching techniques.

An arbitrary cross section may consist of the fuselage body alone, the wing and body, or even the wing alone with or without an intervening slit between it and the plane of symmetry. Each of these cross sections  $z = x + iy$  is mapped into a near circle by removing sequentially all singularities (corners) of the cross-section via a Karman-Trefftz mapping of the type

$$\left( \frac{z - z_0}{z + \bar{z}_0} \right) = \left( \frac{\zeta - z_0}{\zeta + \bar{z}_0} \right)^\tau \quad (21)$$

where  $\zeta$  denotes the ordinate in the mapped plane,  $z_0$  the location of the singularity (and  $\bar{z}_0$  the conjugate value).  $\tau$  is the external angle of the corner to be removed. Equation (21) is applied as many times as there are singularities. On mappings subsequent to the first,  $z$  and  $\tau$  in Eq (21) are to be interpreted as the values in the most recently mapped plane. Once all the corners are removed, the cross section is a near-circle in the final mapped plane and a polar-like coordinate grid can be generated around the body in a straightforward manner. Grid lines wrapping around the body are gradually distorted into a perfect circle at a predetermined distance denoting the outer boundary of the mesh. Radial grid lines emanating from a point equidistant from the vertical and horizontal extremes of the mapped cross section will then give the second family of lines for the mesh. Since the mapped cross section is a near circle, the mesh will usually automatically provide adequate clustering near wing tips. Additional clustering in the azimuthal direction, as well as in the radial direction, can be obtained by the use of stretching functions.

The distance to which the 2-D mesh in each plane extends is kept constant and is typically set to 10 times the body length. Upstream of the aircraft's nose, the 3-D grid is generated by extending the first mesh plane of the aircraft upstream to a pre-determined distance with a gradually increasing spacing between planes. Since the aircraft nose eventually comes to a point, all the points around a cross section degenerate into a single point at the nose. The grid will therefore contain a degenerate axis upstream of the aircraft, a topological feature that requires attention in the solution of the discretized equations. The grid past the last station on the aircraft is similarly extended downstream of the last 2-D mesh on the aircraft. This last mesh usually wraps around a slit since the last cross section typically includes the wing's trailing edge and/or the wake.

A representative grid is shown in Fig. 1-3. Typically one-half of the planes are stationed between the nose and the tail of the aircraft, one

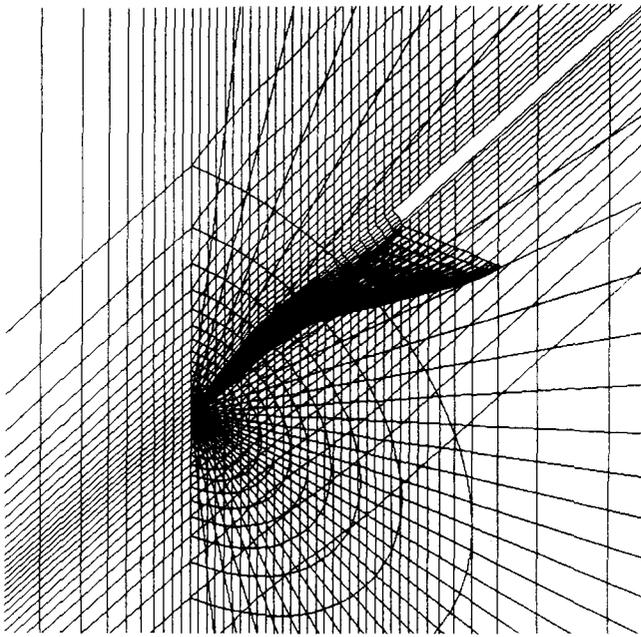
quarter upstream, and one quarter downstream. The planes are clustered near the nose and tail. Figure 1a, b depict the grid in the symmetry plane, along with the surface grid and in representative cross-sectional planes at the leading edge (Fig. 1a) and at a wing-body station (Fig. 1b). A grid surface wrapping around the body is shown in Fig. 2 and a surface running along the aircraft is shown in Fig. 3.

#### IV. Results

Calculations have been carried out on both isolated wings and fighter-like aircraft configurations. In all topologies involved, the numerical scheme has demonstrated remarkable speed and robustness. The speed is mostly due to the numerical scheme which exhibits fast convergence rates. Usually the maximum feasible number of grid levels was used in the multigrid cycle. An average reduction per cycle as low as 0.785 in the density residual has been achieved for cases with a small supersonic zone with a C-H mesh. Convergence rates of 0.8-0.9 are more typical, however, in cases with larger supersonic regions. The convergence rate has shown little sensitivity to the number of mesh cells in a particular grid, as would be expected with a multigrid scheme.

Another factor contributing to the speed is the structure of the code itself which is highly vectorizable. Some speed has been gained at the expense of storing in memory some recurring quantities, such as cell volumes and cell face areas. A C-H mesh, consisting of 96 cells in the streamwise direction and 16 cells in each of the spanwise and normal directions (roughly 25,000 cells), requires approximately 1.7 million 64-bit words of memory. The storage needed for a 192x32x32 mesh (approximately 200,000 cells) is nearly 10.8 million words. On a 96x16x16 C-H mesh, one full multigrid cycle is performed in approximately 1.2 sec. of CPU time on a CRAY-XMP machine, and this time increases linearly with the number of mesh cells, obviously. On such a mesh, force coefficients typically converge to within 1% of their final values in 15 multigrid cycles. All pressures (including shock regions) converge to better than three decimal places in 20-50 cycles. On a 192x32x32 mesh, force coefficients converge to within 1% of their final values in roughly 20 cycles and pressures are converged in fewer than 50 iterations. Most of the results that will be shown have been computed on a CRAY-XMP-14 with 4 million words of memory. An SSD version of the code is under development; however, the code is currently being run entirely within the available core of the machine. A limited number of runs have been made on a CRAY-XMP-216 which has 16 million words of memory available.

The first example in Fig. 4-6 shows results computed for the ONERA M6 wing at  $M_\infty = 0.840$  and  $\alpha = 3.06^\circ$ . This is a well-known case and in Fig. 4 experimental pressure distributions are compared with numerical results computed on both C-H and C-O mesh. In both cases the mesh consisted of 128 cells in the streamwise direction, 24 cells in the direction normal to the wing surface, and 16 cells in the spanwise direction. The grids were such that mesh spacings were approximately equal near the leading edge and the trailing edge at inboard sections. The C-O grid gives better resolution of the wing tip region obviously, and this is



a) Plane at Leading Edge of Aircraft

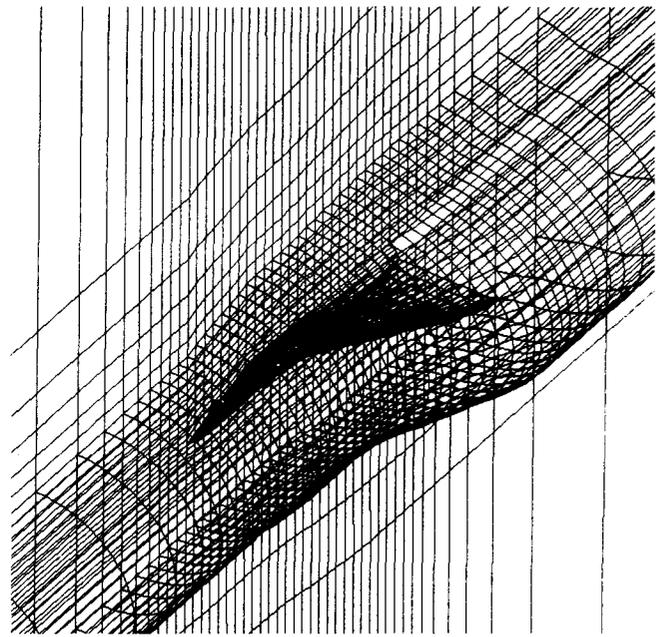
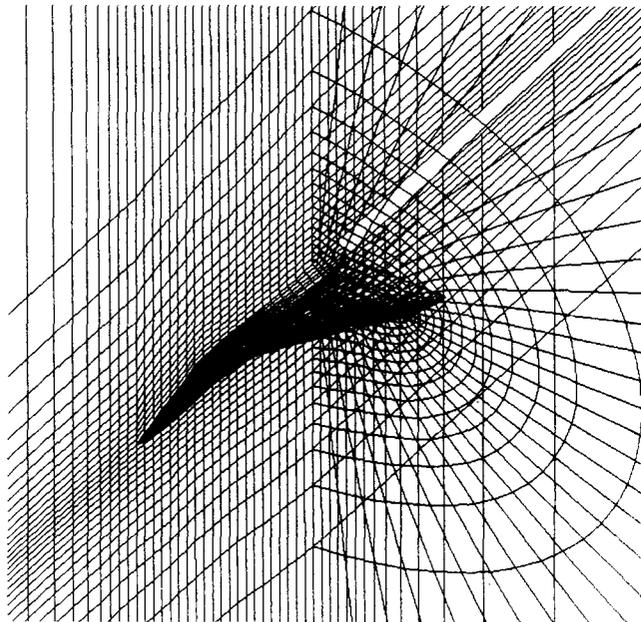


Fig. 2 Grid Lines in Symmetry Plane, on Wing/Fuselage/ Wake Surface and on a Shell around the Aircraft



b) Plane Through the Wing

Fig. 1 Grid Lines in Symmetry Plane, on Wing/Fuselage/ Wake Surface and in Cross-Sectional Plane

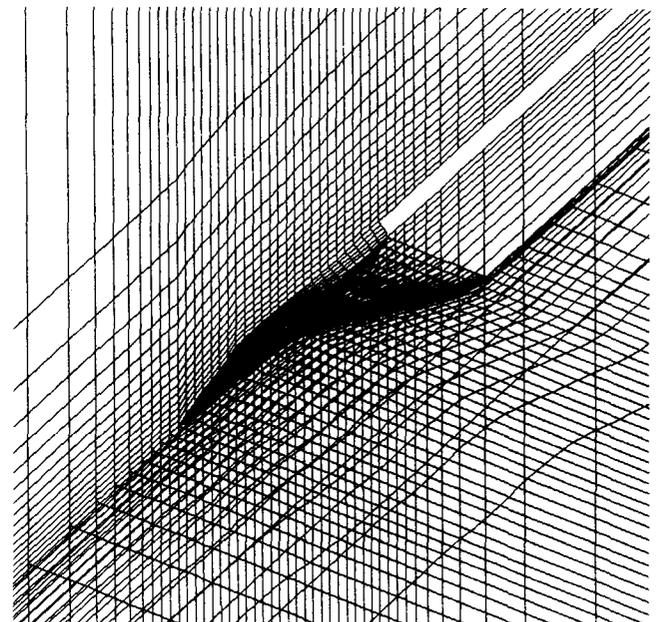


Fig. 3 Grid Lines in Symmetry Plane, on Wing/Fuselage/Wake Surface and in Longitudinal Plane Touching the Wing Tip

reflected in better agreement with data at outboard stations ( $z/b = 0.90$  in Fig. 4). Elsewhere, agreement between experimental data and numerical results is adequate and comparable with both meshes. It should be mentioned that, since mesh planes do not necessarily coincide with data planes, the computed results that are shown are interpolated from neighboring mesh points. Figure 5 depicts the same experimental data and numerical results obtained on a  $192 \times 32 \times 32$  C-H mesh. This mesh gives much better resolution of the streamwise

direction than was possible in the previous case, and this results in improved agreement with the data. Notice in particular, in Fig. 5, that the leading edge expansion is correctly predicted at all four experimental stations. The locations of the leading edge shock and the mid-chord shock are also predicted quite well, even if there is some smearing of the shocks. Shocks are smeared over four mesh cells, typically. The shock pattern is more evident in Plate 1 which depicts isomach contours on the upper wing surface in two vertical planes. Notice the merging of the shocks near the wing tip.

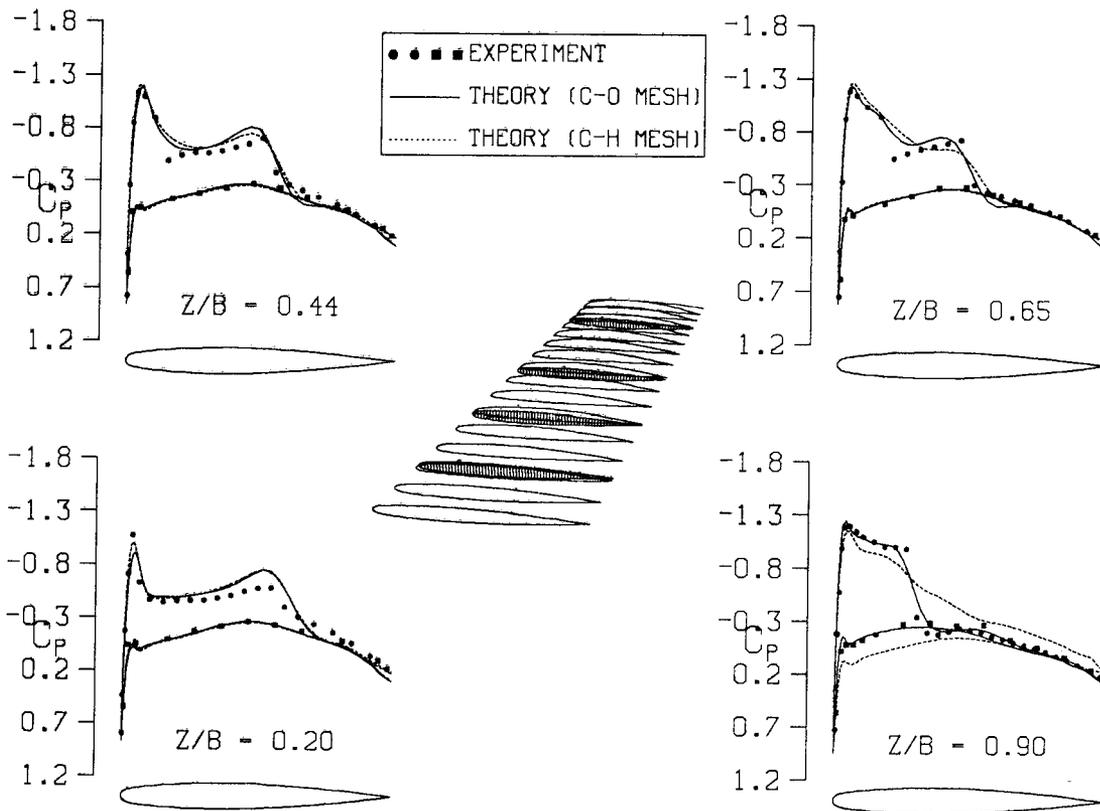


Fig. 4 Computed and Experimental Pressure Distributions; ONERA M6 Wing, 128x24x16 Grid;  $M_\infty = 0.840$ ,  $\alpha = 3.06^\circ$

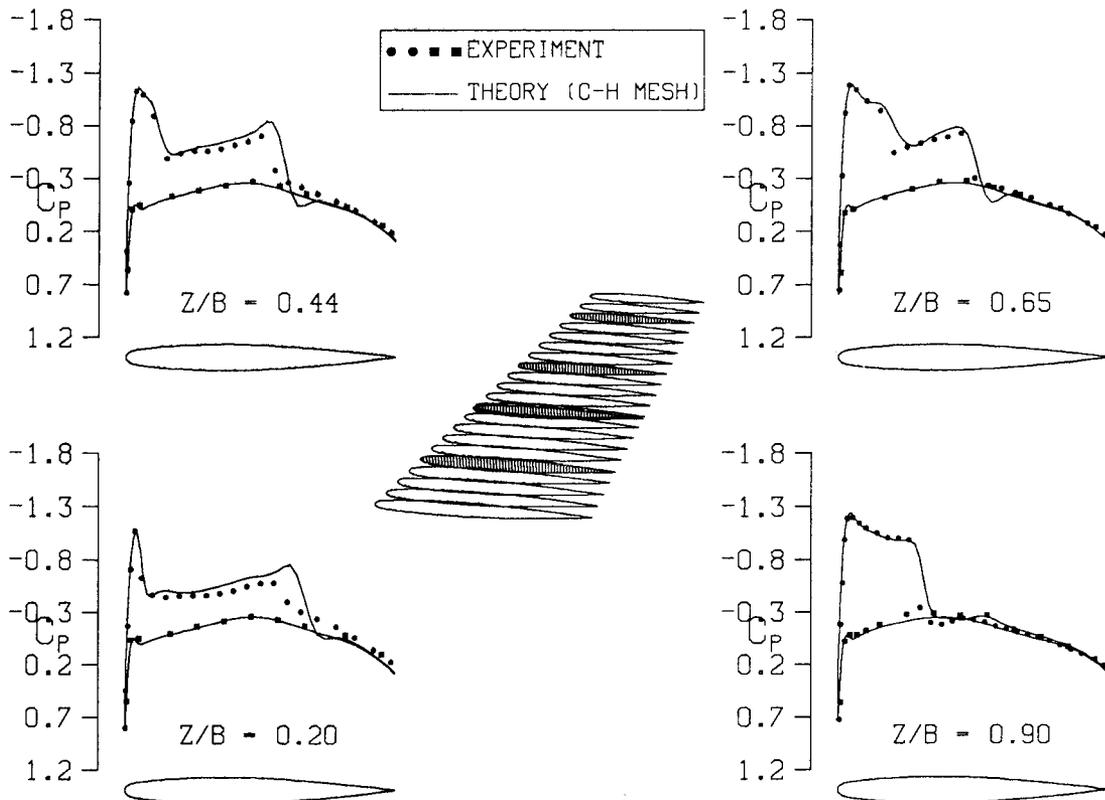


Fig. 5 Computed and Experimental Pressure Distributions; ONERA M6 Wing, 192x32x32 Grid;  $M_\infty = 0.840$ ,  $\alpha = 3.06^\circ$

A second example in Fig. 6 shows pressure distributions computed on a  $192 \times 32 \times 32$  C-H mesh wrapped around a wing which exhibits a very high taper ratio and large wing twist. Evidence of two shocks is again visible here and the two shocks are more clearly seen in the isomach pattern depicted in Plate 2.

Calculations have also been carried out for the fighter-type aircraft configuration whose grid is depicted in Fig. 1-3. These calculations have been carried out with a mesh consisting of 48 longitudinal planes, each containing 48 points in the azimuthal direction around the wing/body and 16 points in the radial direction. Since we are assuming that the flow is symmetric about the midplane (symmetry plane) of the aircraft, the mesh (as well as the computation) covers only one-half of the configuration. A mesh of this size requires approximately 3.5 million 64-bit words of memory. All calculations were performed first on a coarser  $24 \times 24 \times 8$  mesh with three levels of multigrid. The solution on this mesh was then used as a starting point for the finer mesh where calculations were done with four levels of multigrid. On the finer mesh it took approximately 3.1 sec. of CPU time to perform a multigrid cycle. With this topology, convergence rates are in the 0.90-0.95 range typically. Force and pressure coefficients are usually converged in 50-100 cycles. Plate 3 depicts the isomach contours computed for  $M_\infty = 0.900$ ,  $\alpha = 10^\circ$  on the surface and in the trailing edge plane. Since computations were carried out over only one side of the aircraft, the other side shown in the figure is obtained by mirroring the

solution through the symmetry plane. Notice the expansion around the leading edge and the strong swept shock on the outer portion of the wing.  $c_p$  distributions computed at these same flow conditions are shown for several stations in Fig. 7. The isomach contours computed for a supersonic free stream case ( $M_\infty = 1.5$ ) and an angle of attack of  $5^\circ$  are shown in Plate 4 and the corresponding surface pressure distributions are given in Fig. 8.

Results for another fighter-like configuration are shown in Fig. 9-11. This configuration, in addition to a low aspect ratio and a slightly forward-swept trailing edge, features a sharp leading edge and a "chime" ahead of the wing. It should be mentioned that this configuration, like the preceding one, is an analytical model constructed for the purpose of demonstrating the class of geometries to which the H-O grid generator and the multigrid flow solver could be applied. Some cross sections through the wing-fuselage are depicted in Fig. 9 along with the pressure distributions computed for  $M_\infty = 0.90$ ,  $\alpha = 5^\circ$ . Note the sharp expansion around the tip, at all stations. This expansion is followed immediately by a shock on the upper surface. As can be seen in Plate 5, the upper surface of the wing is almost entirely subsonic except for the tip of the wing itself. The isomach contours in a vertical plane over the wing show that a bubble of supersonic flow extends from the tip into the flow field above the upper surface of the wing. Extensive supersonic flow exists on the wing surface at  $M_\infty = 0.95$ ,  $\alpha = 15^\circ$ , as can be seen in Plate 6. There is a shock near the wing tip, but it is not sufficiently

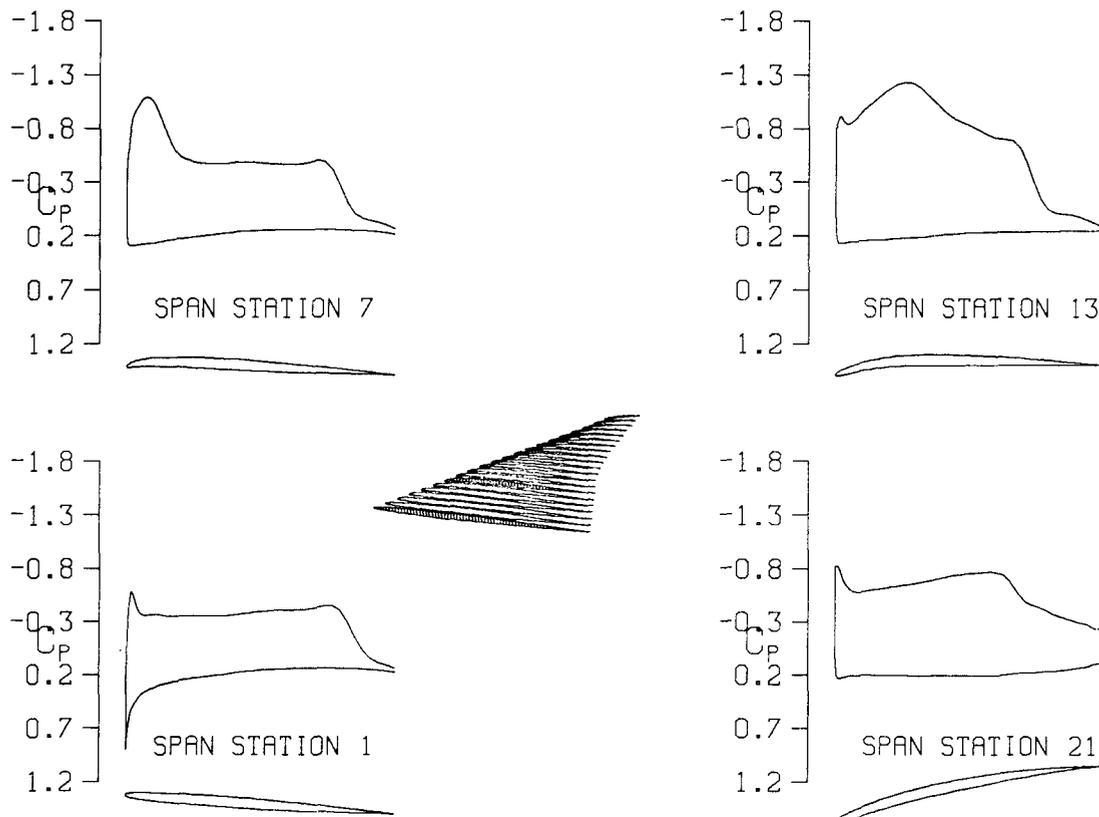
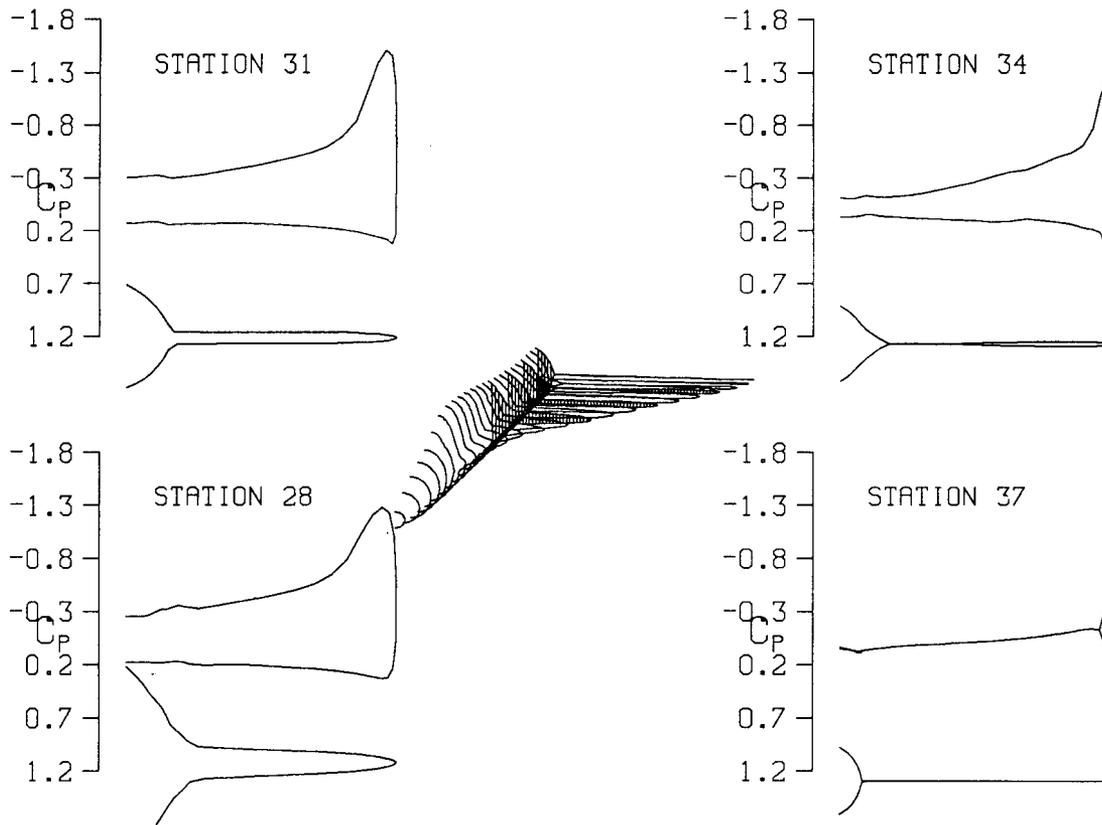
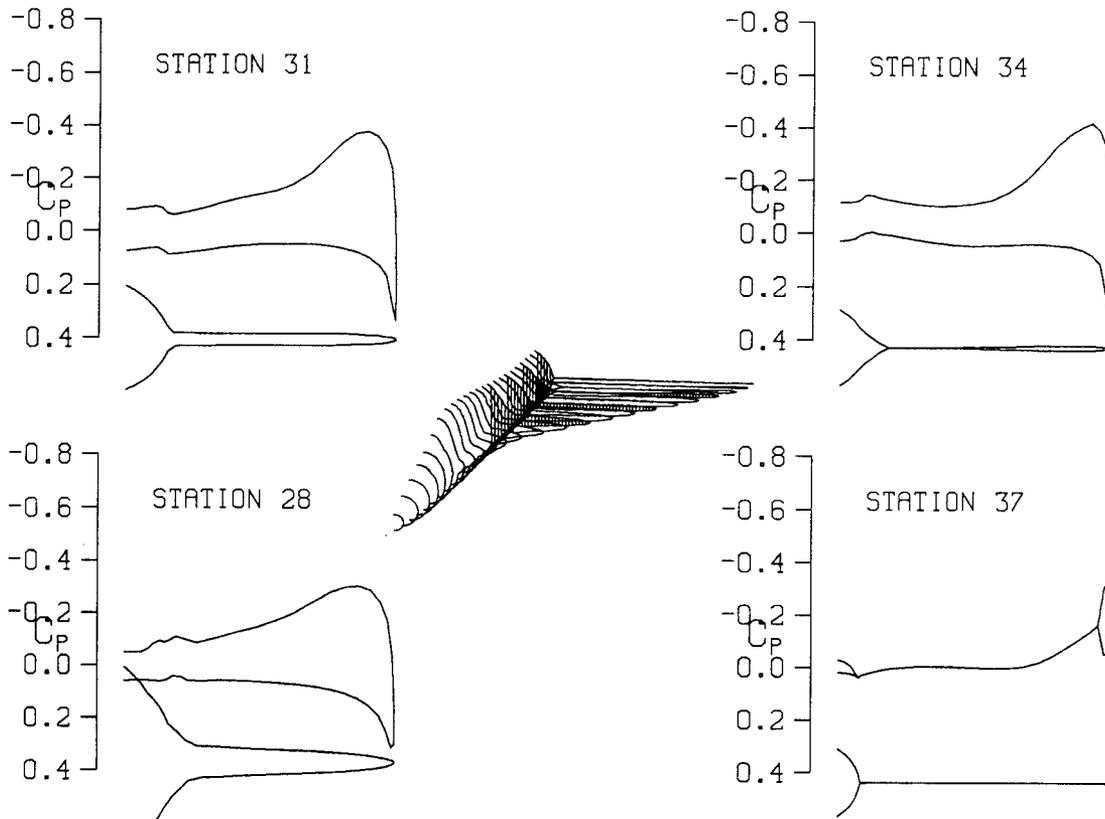


Fig. 6 Computed Pressure Distributions; RFC Wing;  $M_\infty = 0.900$ ,  $\alpha = 8.75^\circ$



**Fig. 7 Computed Pressure Distributions; Fighter Configuration 1;  $M_\infty = 0.900$ ,  $\alpha = 10^\circ$**



**Fig. 8 Computed Pressure Distributions; Fighter Configuration 1;  $M_\infty = 1.50$ ,  $\alpha = 5^\circ$**

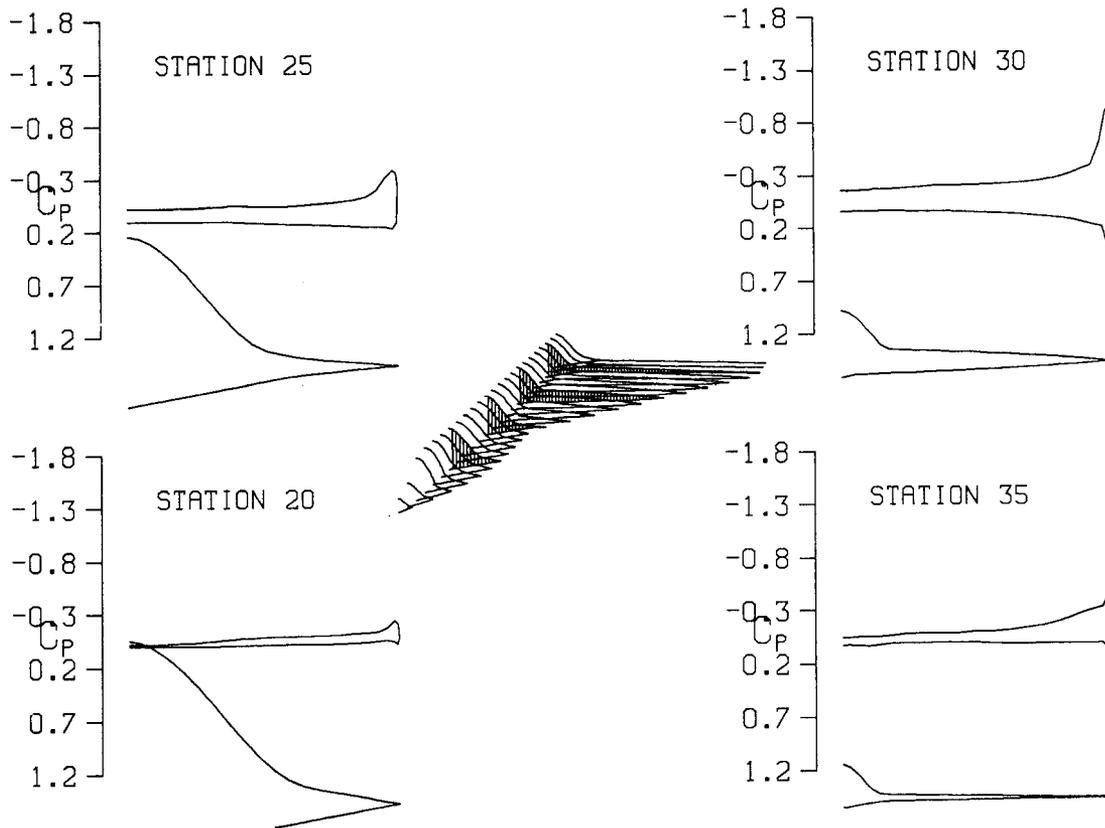


Fig. 9 Computed Pressure Distributions; Fighter Configuration 2;  $M_\infty = 0.900$ ,  $\alpha = 5^\circ$

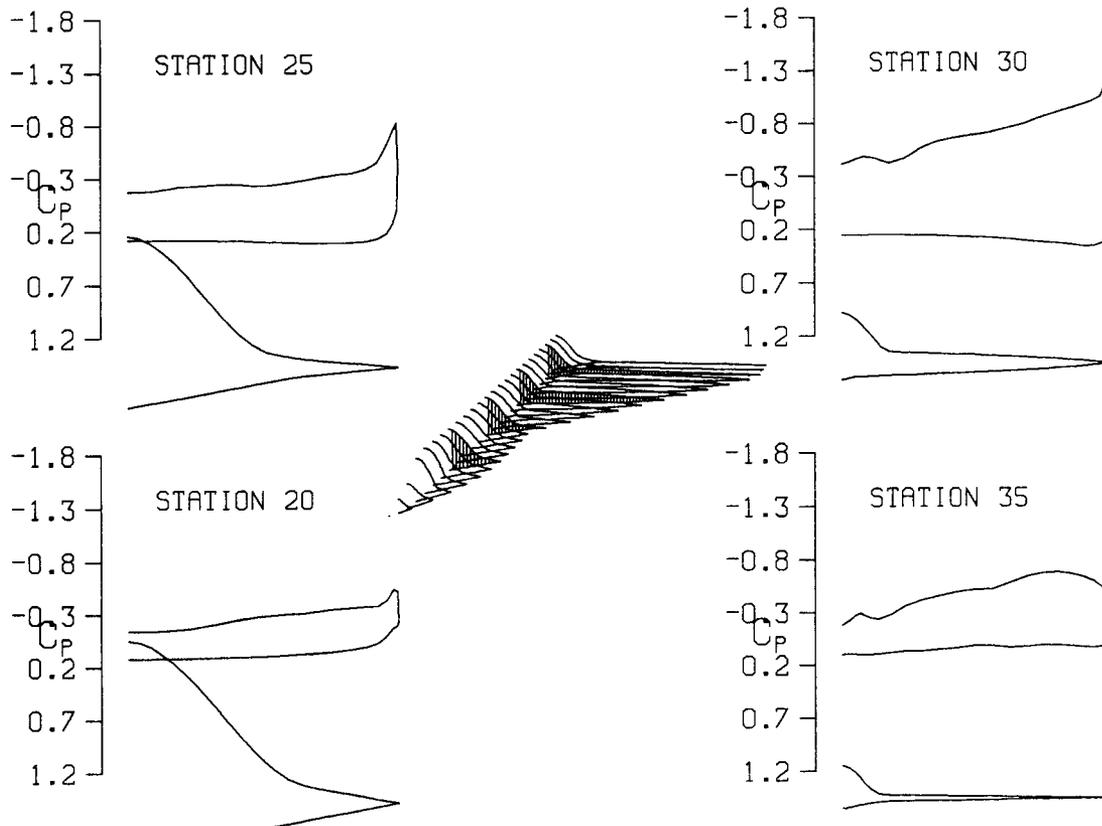


Fig. 10 Computed Pressure Distributions; Fighter Configuration 2;  $M_\infty = 0.950$ ,  $\alpha = 15^\circ$

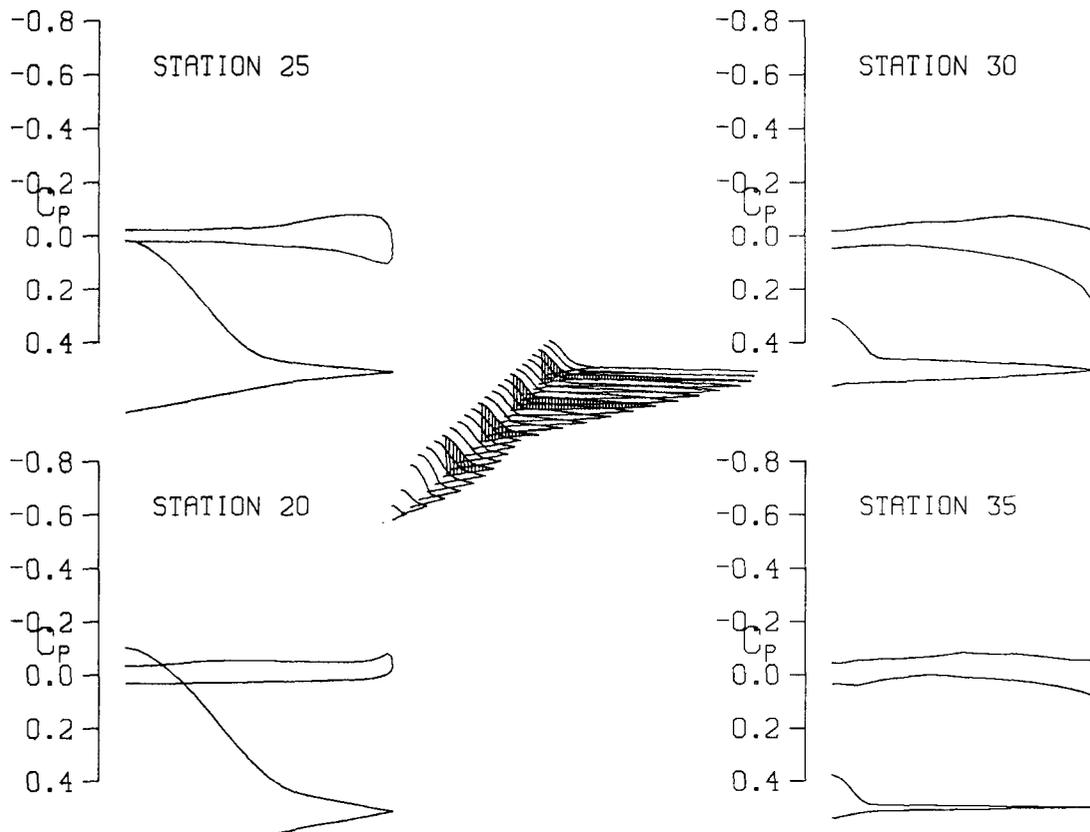


Fig. 11 Computed Pressure Distributions; Fighter Configuration 2;  $M_\infty = 3.00$ ,  $\alpha = 5^\circ$

strong to recompress the flow to subsonic conditions. Figure 10 shows some of the pressure distributions along the body. Plate 7 depicts contours of constant entropy on the surface and in a vertical plane near the trailing edge. The considerable entropy gains indicated in a bubble touching the wing tip points to flow separation. In Plate 8 some streamline "ribbons" computed by joining pairs of streamlines traced from several points near the wing surface clearly indicate flow separation midway along the chime and again one-third of the way along the wing. The code has also been exercised to compute the flow on this configuration for  $M_\infty = 3.0$  and  $\alpha = 5^\circ$ . The computed pressure distributions are given in Fig. 11, for this case, and the corresponding isomach patterns on the surface and in a vertical cross-plane are depicted in Plate 9.

A final example in Fig. 12 shows pressure distributions computed at several stations along a semi-infinite missile-type body. Since the body extends to downstream infinity, only calculations done with a supersonic free stream would be meaningful because of the far-field boundary conditions we apply. The results in Fig. 12 were obtained at  $M_\infty = 1.5$  and  $\alpha = 15^\circ$ . A shock can be discerned near the tip of the body. It starts on the upper surface and gradually moves outboard and around to the lower surface as one proceeds down along the body. The isomach pattern depicted in Plate 10 shows this pattern more clearly. Entropy contours at several body stations are shown in Plate 11 and, again, the pattern is indicative of flow separation. In fact streamline "ribbons" depicted in Plate 12 clearly show the flow separating near the tip.

#### V. Conclusions

A new multigrid time-stepping method has been developed for the computation of steady flows over isolated wings and over aircraft configurations in transonic and supersonic free streams. Several topologies for space discretization can be used in conjunction with the numerical scheme, which in all cases proved to be very efficient and reliable. The novel H-O topology described is well suited for fighter-type aircraft configurations. The method is to be extended to aircraft with canards and tails and with engine nacelles. The results obtained demonstrate that calculations over realistic aircraft configurations can now be obtained quickly and inexpensively enough to make possible the large number of calculations that are required to explore an aircraft's flight envelope or to (re-)design an aircraft's shape.

#### VI. References

1. Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes," AIAA Paper 81-1259, AIAA 14th Fluid Dynamics and Plasma Dynamics Conf, Palo Alto, CA, 1981.
2. Ni, Ron Ho, "A Multiple Grid Scheme for Solving the Euler Equations," *AIAA Journal*, Vol. 20, 1982, pp 1565-1571.
3. Jameson, A., and Baker, T., "Solution of the Euler Equations for Complex Configurations," Proc. AIAA 6th Computational Fluid Dynamics Conf, Danvers, MA, 1983, pp 293-302.

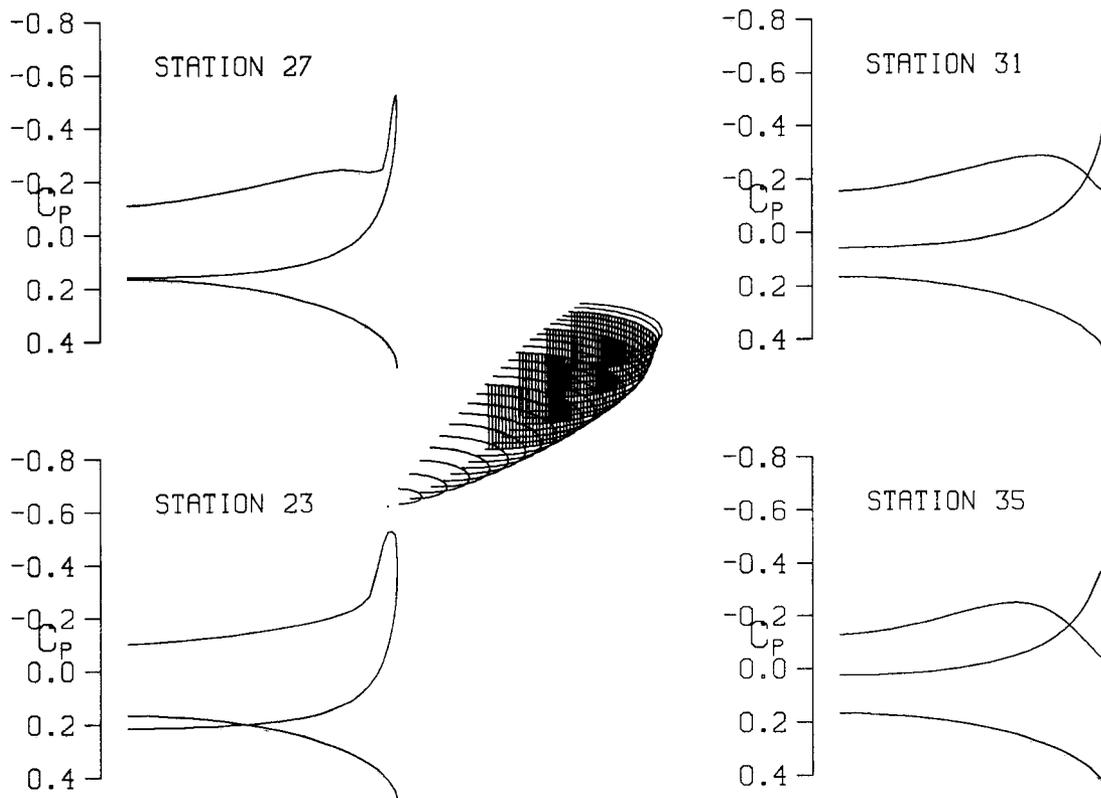
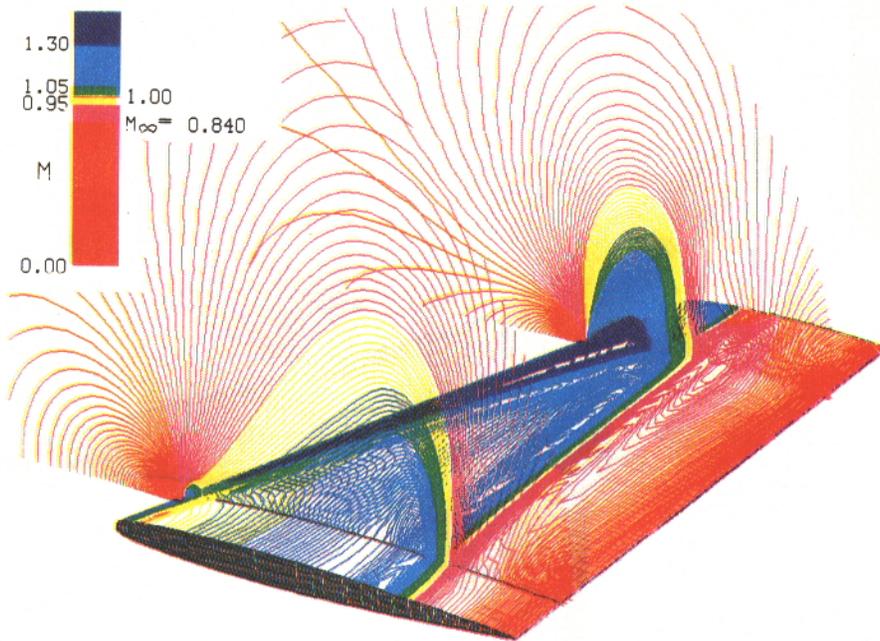


Fig. 12 Computed Pressure Distributions; Missile Configuration;  $M_\infty = 1.50$ ,  $\alpha = 15^\circ$

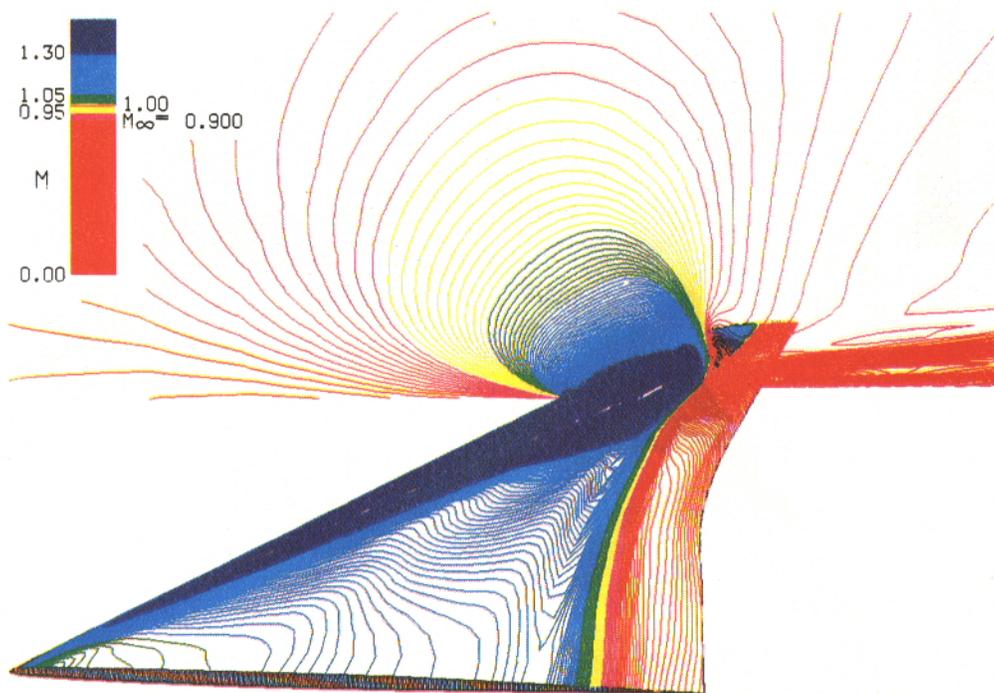
4. Jameson, A., "Solution of the Euler Equations by a Multigrid Method," Applied Math and Computation, Vol. 13, 1983, pp 327-356.
5. Hemker, P.W., and Spekrijse, S.P., "Multigrid Solution of the Steady Euler Equations," Proc. Oberwolfach Meeting on Multigrid Methods, Dec 1984.
6. McCormack, R.W., "Current Status of Numerical Solutions of the Navier-Stokes Equations," AIAA Paper 85-0032, AIAA 23rd Aerospace Sciences Mtg, Reno, NV, Jan 1985.
7. Pulliam, T.H., and Steger, J.L., "Recent Improvements in Efficiency, Accuracy and Convergence for Implicit Approximate Factorization Algorithms," AIAA Paper 85-0360, AIAA 23rd Aerospace Sciences Mtg, Reno, NV, Jan 1985.
8. Hall, M.G., "Cell Vertex Multigrid Schemes for Solution of the Euler Equations," IMA Conf on Numerical Methods for Fluid Dynamics, Reading, England, April 1985.
9. Jameson, A., and Schmidt, W., "Recent Developments in Numerical Methods for Transonic Flows," Proc. 3rd International Conf on Finite Elements in Nonlinear Mechanics, FENOMECH.84, Stuttgart, 1984, ed. by J. St. Doltsinis, North Holland, 1985, pp 467-493.
10. Jameson, A., Baker, T.J., and Weatherill, N.P., "Calculation of Inviscid Transonic Flow Over a Complete Aircraft," AIAA Paper 86-0103, AIAA 24th Aerospace Sciences Mtg, Reno, NV, Jan 1986.
11. Thomas, J.L., Van Leer, B., and Walters, R.W., "Implicit Flux-Split Schemes for the Euler Equations," AIAA Paper 85-1680, presented at AIAA 18th Fluid Dynamics and Plasma Dynamics and Lasers Conf, Cincinnati, OH, July 1985.
12. Anderson, W.K., Thomas, J.L., and Whitfield, D.L., "Multigrid Acceleration of the Flux Split Euler Equations," AIAA paper 86-0105, AIAA 24th Aerospace Sciences Mtg, Reno, NV, Jan 1986.
13. Ericsson, L.E., "Generation of Boundary-Conforming Grids Around Wing-Body Configurations Using Transfinite Interpolation," AIAA J., Oct 1982, pp 1313-1320.
14. Jameson, A., "A Vertex Based Multigrid Algorithm for Three-Dimensional Compressible Flow Calculations," ASME Symp on Numerical Methods for Compressible Flow, Anaheim, CA, Dec 1986.
15. Harten, A., "High Resolution Schemes for Hyperbolic Conservation Laws," J. Comp. Phys., Vol. 49, 1983, pp 357-393.
16. Jameson, A., "A Nonoscillatory Shock Capturing Scheme Using Flux Limited Dissipation," Lectures in Applied Mathematics, Vol. 22, Part 1, Large Scale Computations in Fluid Mechanics, ed. by B.E. Engquist, S. Osher, and R.C.J. Somerville, A.M.S., 1985, pp 345-370.
17. Jameson, A., "Transonic Flow Calculations for Aircraft," Lecture Notes in Mathematics, Vol. 1127, Numerical Methods in Fluid Dynamics, ed. by F. Brezzi, Springer-Verlag, 1985, pp 156-242.

18. Moretti, G., "Conformal Mappings for Computations of Steady, Three-Dimensional, Supersonic Flows," Numerical/Laboratory Computer Methods in Fluid Mechanics, presented at Annual Winter Mtg of the ASME, New York, Dec 1976.
19. Siclari, M.J., "The Computational Treatment of Supersonic Wake Flows in NCOREL," AIAA Paper No. 85-0304, presented at AIAA 23rd Aerospace Sciences Mtg, Reno, NV, Jan 1985.



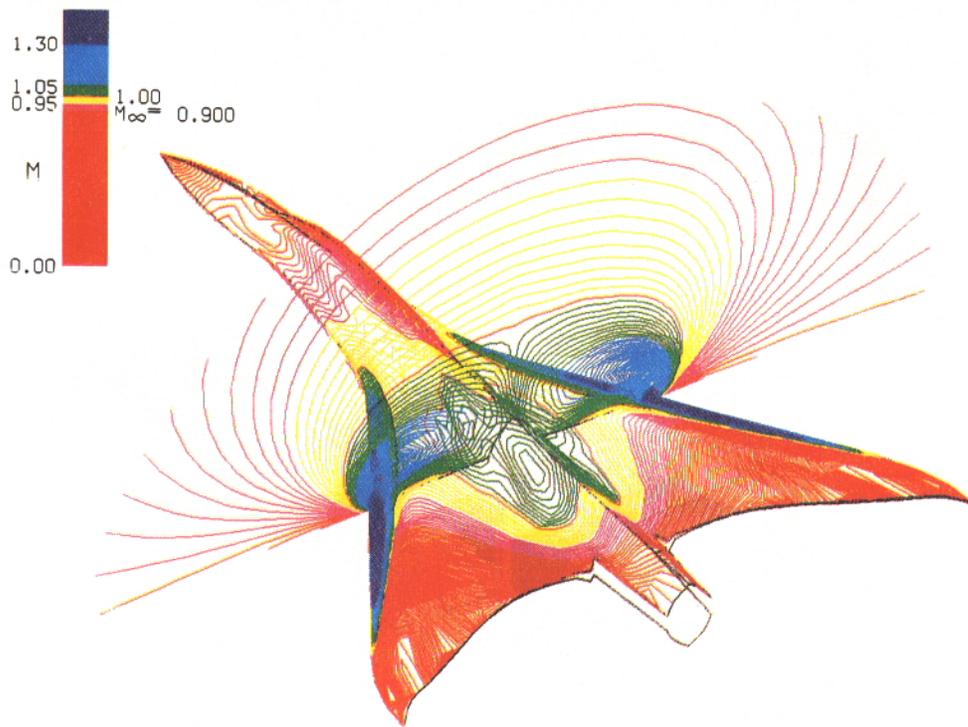
VOLPE, SICLARI AND JAMESON: PLATE 1

Isomach contours on upper wing surface and two vertical planes; ONERA M6 wing:  
 $M_\infty = 0.840$ ,  $\alpha = 3.06^\circ$



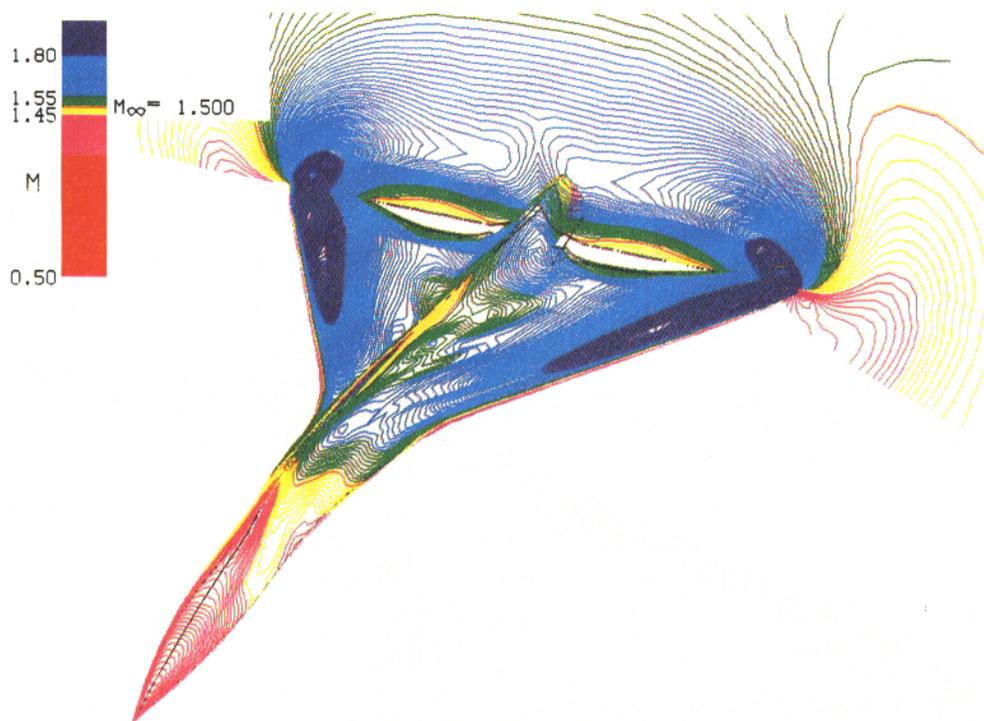
VOLPE, SICLARI AND JAMESON: PLATE 2

Isomach contours on upper wing surface and in an outboard vertical plane; RFC wing;  
 $M_\infty = 0.900$ ,  $\alpha = 8.75^\circ$



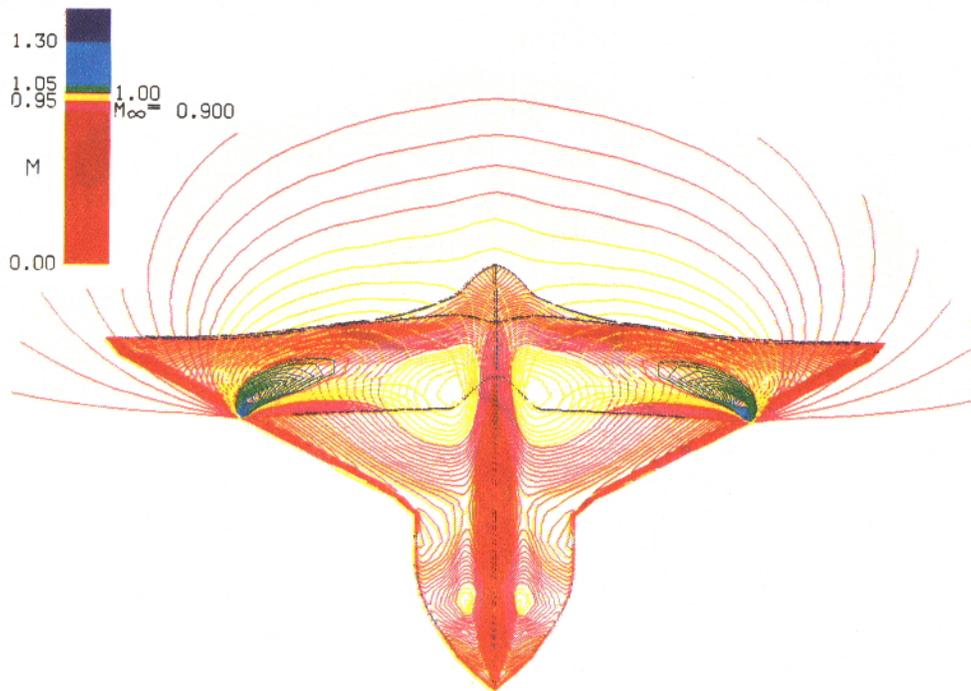
VOLPE, SICLARI AND JAMESON: PLATE 3

Isomach contours on surface and in a cross-sectional plane; fighter configuration 1;  
 $M_\infty = 0.900$ ,  $\alpha = 10^\circ$



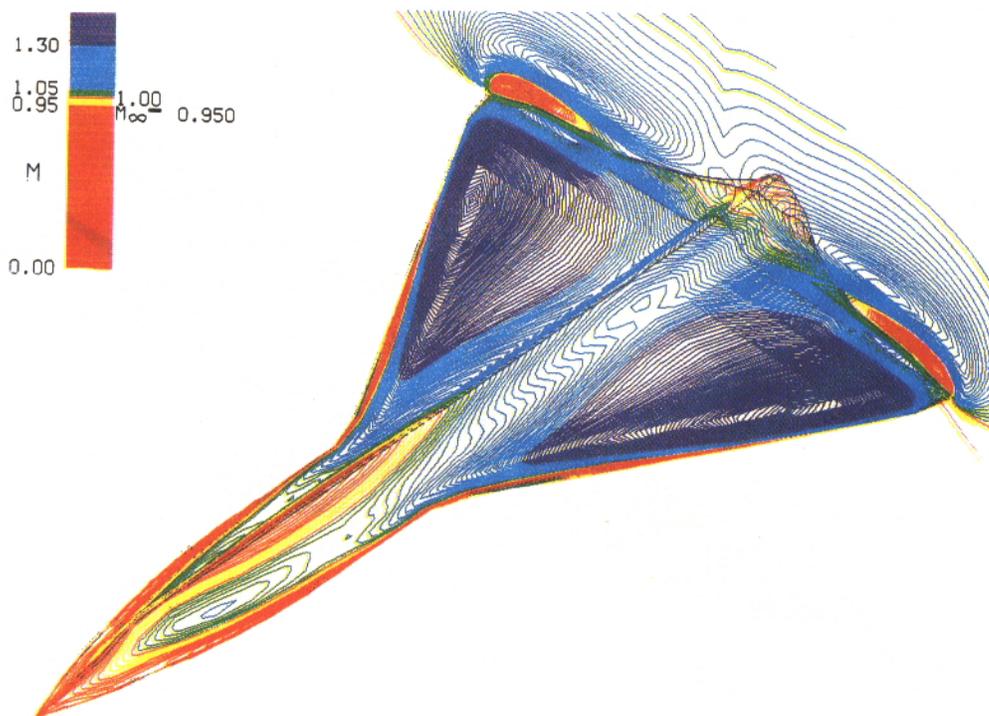
VOLPE, SICLARI AND JAMESON: PLATE 4

Isomach contours on surface and in a cross-sectional plane; fighter configuration 1;  
 $M_\infty = 1.50$ ,  $\alpha = 5^\circ$



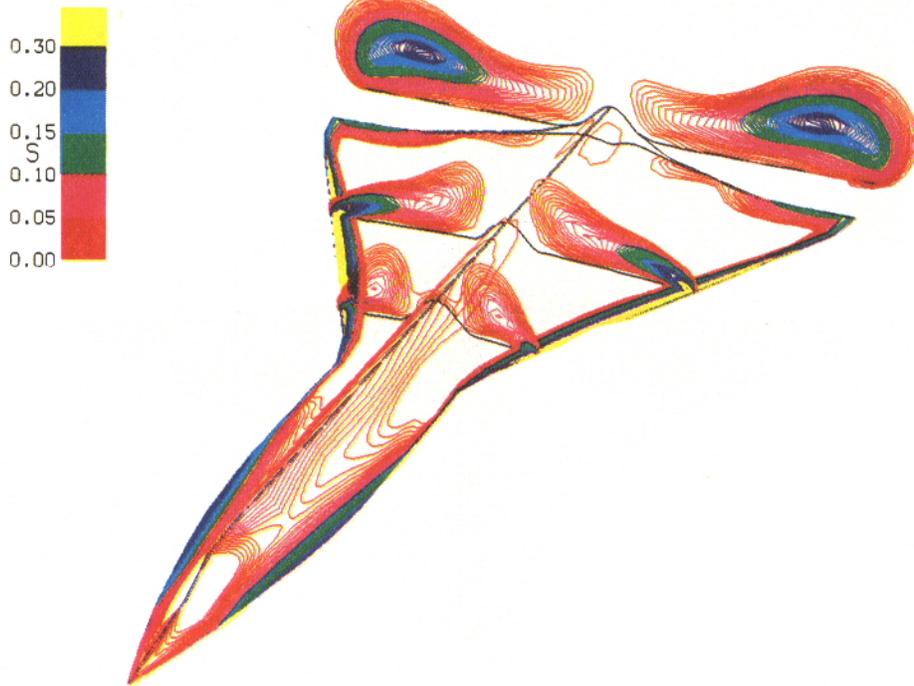
VOLPE, SICLARI AND JAMESON: PLATE 5

Isomach contours on surface and in a cross-sectional plane; fighter configuration 2;  
 $M_\infty = 0.900$ ,  $\alpha = 5^\circ$



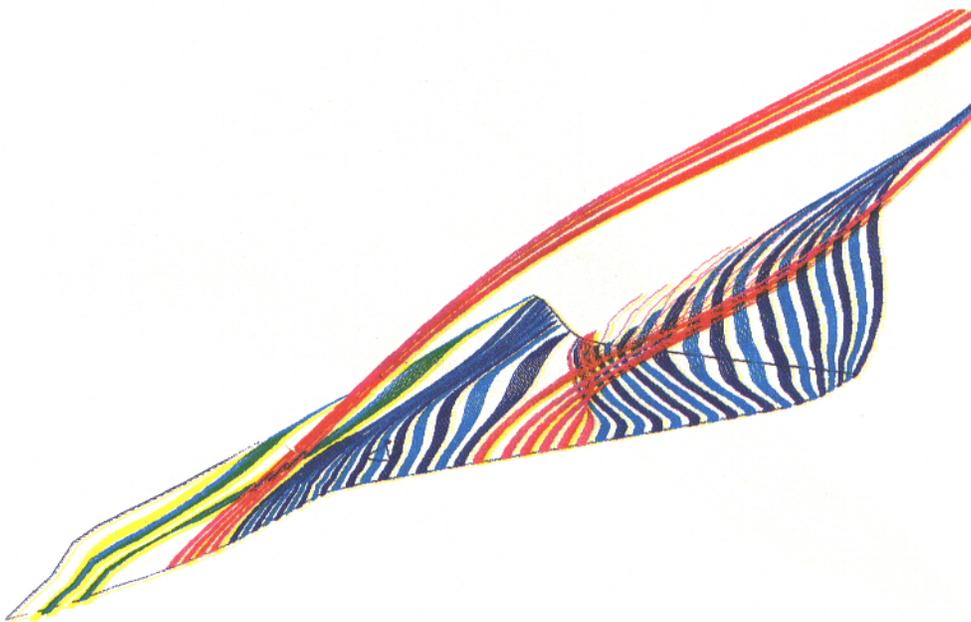
VOLPE, SICLARI AND JAMESON: PLATE 6

Isomach contours on surface and in a cross-sectional plane; fighter configuration 2;  
 $M_\infty = 0.950$ ,  $\alpha = 15^\circ$



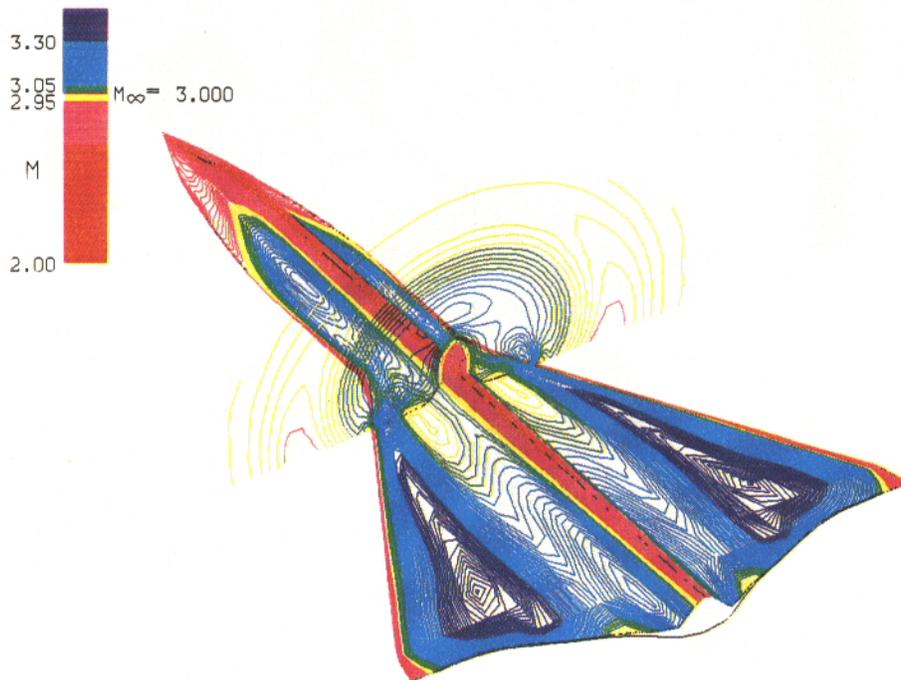
VOLPE, SICLARI AND JAMESON: PLATE 7

Entropy contours on surface and in three cross-sectional planes; fighter configuration 2;  $M_\infty = 0.950$ ,  $\alpha = 15^\circ$



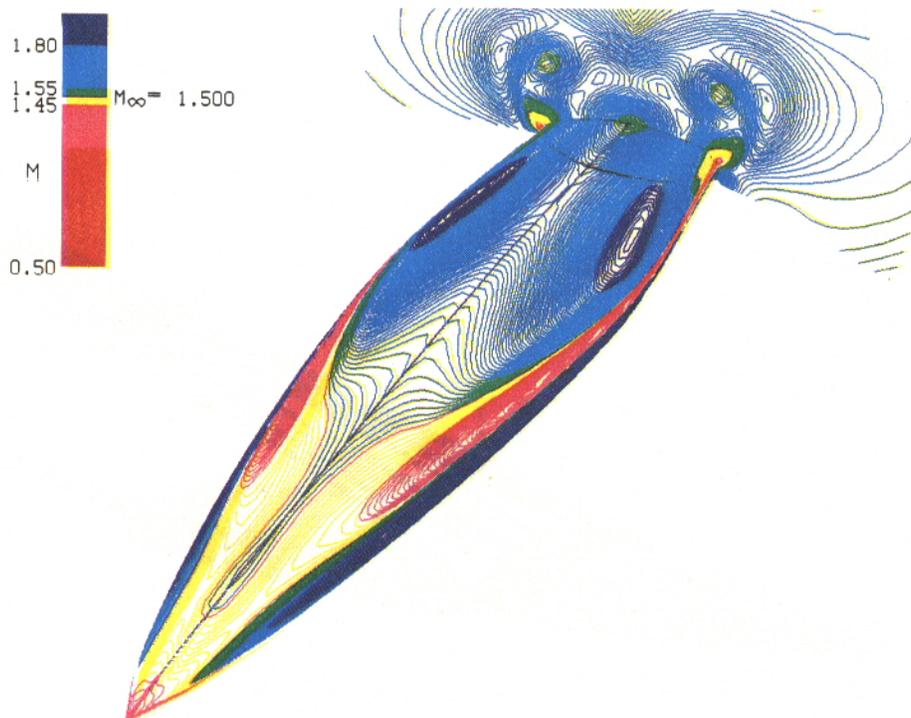
VOLPE, SICLARI AND JAMESON: PLATE 8

Streamline ribbons over fighter configuration 2;  $M_\infty = 0.950$ ,  $\alpha = 15^\circ$



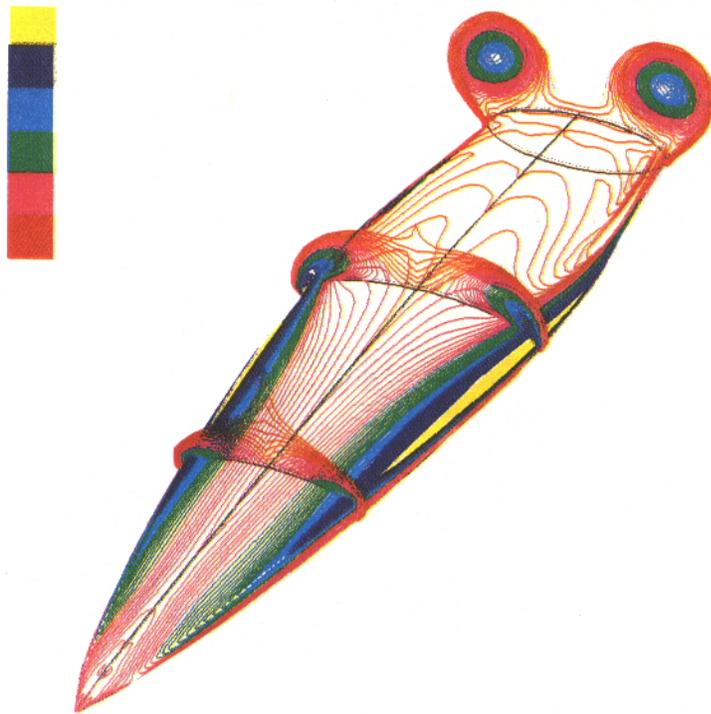
VOLPE, SICLARI AND JAMESON: PLATE 9

Isomach contours on surface and in a cross-sectional plane; fighter configuration 2;  
 $M_\infty = 3.00$ ,  $\alpha = 5^\circ$



VOLPE, SICLARI AND JAMESON: PLATE 10

Isomach contours on surface and in a cross-sectional plane; missile configuration;  
 $M_\infty = 1.50$ ,  $\alpha = 15^\circ$



VOLPE, SICLARI AND JAMESON: PLATE 11

Entropy contours on surface and in three cross-sectional planes; missile configuration;  $M_\infty = 1.50$ ,  $\alpha = 15^\circ$



VOLPE, SICLARI AND JAMESON: PLATE 12

Streamline ribbons over missile configuration;  $M_\infty = 1.50$ ,  $\alpha = 15^\circ$