An Implementation of a Grid-Independent Upwind Scheme for the Euler Equations

David W. Levy [*]

Kenneth G. Powell [†]

Bram van Leer [‡]

The University of Michigan

Department of Aerospace Engineering

Abstract

A scheme for the two-dimensional Euler equations that uses flow parameters to determine the direction for upwind-differencing is described. This approach reduces the grid-dependence of conventional schemes. Upwinding at the local flow angle and the local pressure-gradient angle is tested. The upwind-biased fluxes are calculated with Roe's approximate Riemann solver. A centered flux is used for the component normal to the upwinding direction. Results for a first-order scheme show significant improvement over conventional grid-aligned upwinding. Specifically, oblique shock waves are less diffused. Preliminary results for a second-order scheme are also included.

Introduction

When approximating the one-dimensional Euler equations by upwind differencing, the choice of the upwind direction is straightforward. Characteristic information can only be transmitted forwards or backwards depending on the sign of the corresponding characteristic speed. In two or three dimensions the choice is more difficult. Although efforts are underway to develop truly multi-dimensional schemes [1,2], most multi-dimensional schemes treat the flow in a direction-split manner, with the upwind direction always normal to the face of the computational cell across which the fluxes are calculated. The problem with this approach is that it has little regard for the physical features of the flow. This becomes particularly evident when computing oblique waves, which tend to become excessively smeared.

Previous efforts to develop grid-independent schemes include the work of Davis [3], who has implemented a scheme with upwind differencing normal to shock waves. Hirsch et al. [4] has tested one with upwind directions normal to certain characteristic waves. The present work initially began as an extension of the QAZ1D formulation of the Euler equations in Verhoff et al. [5], which is a non-conservative upwind scheme in intrinsic coordinates. Moretti [6] has developed the λ-scheme, which also is non-conservative, and determines the proper domain of dependence for each grid point based on the directions of certain characteristics. Preceding the Euler discretizations is the scheme of Jameson [7], which solves the full transonic potential equation in intrinsic coordinates, calculating upwind differences in the flow direction and central differences in the normal direction.

This paper investigates the effects of upwind differencing at a specified angle, one that may be chosen independent of the grid coordinates. There are three major aspects of this problem:

1. Choice of upwind differencing angle;
2. Determination of a left and a right state at each cell interface, as a function of the upwinding angle;
3. Computation of fluxes at an arbitrary angle.

The differences between conventional grid-aligned schemes and the present schemes all relate to these aspects:

1. In a grid-aligned scheme, the direction of upwind differencing is normal to the cell face. In the present scheme, the upwinding angle can be freely chosen, and, in particular, can be based on flow features such as flow and pressure-gradient angles.

[*] Graduate Research Assistant, AIAA Member
[†] Assistant Professor, AIAA Member
[‡] Professor, AIAA Member

2. In a first-order grid-aligned scheme, the left and right states are taken to be the average quantities of the cells on either side of the cell face. In the present scheme, the left and right state vectors are calculated in a manner biased by the upwinding direction. The first-order scheme interpolates between cell centers to obtain these states.

3. In a grid-aligned scheme, the flux normal to the upwinding direction does not contribute to the flux through the cell face, so it is not calculated. In the present scheme, the flux at the chosen angle is computed with upwind bias and a centered flux is computed normal to this direction. The flux normal to the cell face is computed by rotating these two flux components to the coordinate frame aligned with the cell face.

Approximate Riemann Solver

The Euler equations in two dimensions can be written:

$$\tilde{\mathbf{U}}_t + \tilde{\mathbf{F}}_x + \tilde{\mathbf{G}}_x = 0 \tag{1}$$

where:

$$\tilde{\mathbf{U}} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \tilde{\mathbf{F}} = \begin{bmatrix} \rho \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{bmatrix}, \tag{2}$$

$$\tilde{\mathbf{G}} = \begin{bmatrix} \rho \\ \rho uv \\ \rho v^2 + p \\ \rho vH \end{bmatrix}, \tag{3}$$

where $H = E + p/\rho$ is the total enthalpy. The equations are then integrated over a cell Ω:

$$\iint_\Omega \tilde{\mathbf{U}}_t dA + \iint_\Omega \left(\tilde{\mathbf{F}}_x + \tilde{\mathbf{G}}_y\right) = 0. \tag{4}$$

The second integral is converted to a line integral over the cell boundary, $\partial\Omega$, with Gauss' theorem:

$$\iint_\Omega \tilde{\mathbf{U}}_t dA + \oint_{\partial\Omega} \left(\tilde{\mathbf{F}} dy - \tilde{\mathbf{G}} dx\right) = 0. \tag{5}$$

The first integral can be expressed in terms of the change in the *average* state $\mathbf{U}$ in the cell, and the line integral becomes a sum over the four faces of a quadrilateral cell:

$$A\frac{d\mathbf{U}}{dt} + \sum_{k=1}^{4} \left(\tilde{\mathbf{F}}\Delta y - \tilde{\mathbf{G}}\Delta x\right)_k = 0. \tag{6}$$

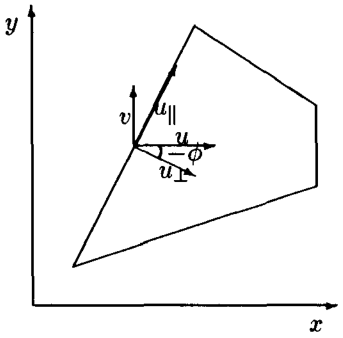

Figure 1: Geometry for Flux Calculation

After introducing the cell-face length Δs,

$$\Delta s^2 = \Delta x^2 + \Delta y^2, \tag{7}$$

equation 6 can be written as:

$$A\frac{d\mathbf{U}}{dt} + \sum_{k=1}^{4} \mathbf{F}_k \Delta s_k = 0, \tag{8}$$

where $\mathbf{F}$ is the flux normal to the cell face (see Figure 1):

$$\mathbf{F} = \begin{bmatrix} \rho u_\perp \\ \rho u_\perp u + p\cos\phi \\ \rho u_\perp v - p\sin\phi \\ \rho u_\perp H \end{bmatrix}, \tag{9}$$

$$\cos\phi = \Delta y/\Delta s, \tag{10}$$

$$\sin\phi = -\Delta x/\Delta s. \tag{11}$$

Equation 8 is exact if $\mathbf{F}_k$ equals the *average* normal flux at the k^{th} cell face. If no angles other than ϕ are defined, the above formulation will automatically lead to grid-aligned upwind differencing.

The basic method of flux computation is essentially the same for both grid-aligned and grid-independent schemes. It is based on Roe's approximate Riemann solution [8] for the Euler equations. The flux across a plane of which the normal is at an angle ϕ with respect to the x-axis is computed from:

$$\mathbf{F} = \frac{1}{2}(\mathbf{F}_L + \mathbf{F}_R) - \frac{1}{2}|\mathbf{A}|^* \Delta\mathbf{U} \tag{12a}$$

$$= \frac{1}{2}(\mathbf{F}_L + \mathbf{F}_R) - \frac{1}{2}\mathbf{R}|\mathbf{\Lambda}|^* \Delta\mathbf{V}; \tag{12b}$$

$\mathbf{F}_L$ and $\mathbf{F}_R$ are the flux vectors computed from the left and right states, $\mathbf{U}_L$ and $\mathbf{U}_R$; 'left' refers to the lower cell index; furthermore,

$$\Delta\mathbf{U} = \mathbf{U}_R - \mathbf{U}_L. \tag{13}$$

The matrix $\mathbf{A}$ is the Jacobian of $\mathbf{F}(\mathbf{U})$, the eigenvalues $\hat{a}_k$ of which are the characteristic speeds $\hat{u} - \hat{c}$, $\hat{u}$, $\hat{u}$, and $\hat{u} + \hat{c}$. The matrix $\mathbf{\Lambda}$ is the diagonal matrix of these eigenvalues. The matrix $\mathbf{R}$ is composed of the right eigenvectors of $\mathbf{A}$:

$$\mathbf{R} = [\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \mathbf{R}_4] \tag{14a}$$

$$\mathbf{R}_1 = \begin{bmatrix} 1 \\ \hat{u} - \hat{c}\cos\phi \\ \hat{v} - \hat{c}\sin\phi \\ \hat{H} - \hat{u}_\perp \hat{c} \end{bmatrix}, \tag{14b}$$

$$\mathbf{R}_2 = \begin{bmatrix} 0 \\ -\hat{u}_\| \sin\phi \\ \hat{u}_\| \cos\phi \\ \hat{u}_\|^2 \end{bmatrix}, \tag{14c}$$

$$\mathbf{R}_3 = \begin{bmatrix} 1 \\ \hat{u} \\ \hat{v} \\ \frac{1}{2}(\hat{u}^2 + \hat{v}^2) \end{bmatrix}, \tag{14d}$$

$$\mathbf{R}_4 = \begin{bmatrix} 1 \\ \hat{u} + \hat{c}\cos\phi \\ \hat{v} + \hat{c}\sin\phi \\ \hat{H} + \hat{u}_\perp \hat{c} \end{bmatrix}. \tag{14e}$$

The hats refer to Roe averages; the Roe average of a quantity is computed by weighting with $\sqrt{\rho}$, for instance:

$$\hat{u} = \frac{u_L\sqrt{\rho_L} + u_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \tag{15}$$

and similarly for $\hat{v}$, $\hat{H}$, and $1/\hat{\rho}$. The other quantities with hats are not averaged independently, but are obtained from the basic Roe-averaged quantities by their normal functional relations.

The weights of $\mathbf{R}_k$ in the $\Delta\mathbf{U}$ form the vector $\Delta\mathbf{V}$:

$$\Delta\mathbf{V} = \begin{bmatrix} (\Delta p - \hat{\rho}\hat{c}\Delta u_\perp)/2\hat{c}^2 \\ \hat{\rho}\Delta u_\| / \hat{u}_\| \\ \Delta\rho - \Delta p/\hat{c}^2 \\ (\Delta p + \hat{\rho}\hat{c}\Delta u_\perp)/2\hat{c}^2 \end{bmatrix}. \tag{16}$$

Since $\hat{u}_\|$ may vanish locally the second element of $\Delta\mathbf{V}$ is preferably multiplied, and $\mathbf{R}_2$ is divided, by $\hat{u}_\|/\hat{c}$.

The matrix $|\mathbf{\Lambda}|^*$ is computed from the diagonal matrix of absolute characteristic speeds $|\hat{a}_k|$:

$$|\mathbf{\Lambda}|^* = \begin{bmatrix} |\hat{u}_\perp - \hat{c}|^* & 0 & 0 & 0 \\ 0 & |\hat{u}_\perp| & 0 & 0 \\ 0 & 0 & |\hat{u}_\perp| & 0 \\ 0 & 0 & 0 & |\hat{u}_\perp + \hat{c}|^* \end{bmatrix} \tag{17}$$

The modified absolute speeds, $|\hat{a}|_k^*$, k=1,4, are computed as follows:

$$\delta a = a_R - a_L, \tag{18}$$

$$\delta a^* = max(\delta a, 0), \tag{19}$$

$$|\hat{a}|^* = \begin{cases} |\hat{a}|, & |\hat{a}| > \frac{1}{2}\delta a^*, \\ \frac{\hat{a}^2}{\delta a} + \frac{1}{4}\delta a, & |\hat{a}| \le \frac{1}{2}\delta a^*. \end{cases} \tag{20}$$

This enforces the entropy condition, which eliminates expansion shocks, and ensures a smooth transition from subsonic to supersonic flow.

In a first-order scheme, the left and right states are chosen to be the cell center values to the left and right of the cell face. Higher-order schemes interpolate or extrapolate state quantities to the cell face using more than one cell, but the choice of cells used is still based on grid geometry.

In the first-order scheme, forward Euler time stepping is used.

Angles for Upwind Differencing

Three choices of angles for upwind differencing are investigated:

1. Arbitrary, fixed angle throughout the grid;
2. Local flow angle;
3. Local pressure-gradient angle.

Choice 1 is included to test the sensitivity of the solution to the upwinding angle. With Choice 2, the flow angle used at an interface is the simple average

$$\theta_s = \tan^{-1}\left(\frac{v_L + v_R}{u_L + u_R}\right). \tag{21}$$

With Choice 3, pressure gradients are computed after averaging the pressure field to avoid noise in the numerical derivatives. For a cell boundary centered at $\left(x_{i-\frac{1}{2}}, y_j\right)$ define:

$$p_{left} = \frac{1}{4}\left(p_{i-1,j-1} + 2p_{i-1,j} + p_{i-1,j+1}\right), \tag{22a}$$

$$p_{right} = \frac{1}{4}\left(p_{i,j-1} + 2p_{i,j} + p_{i,j+1}\right), \qquad (22b)$$

$$p_{upper} = \frac{1}{2}\left(p_{i-1,j+1} + p_{i,j+1}\right), \qquad (22c)$$

$$p_{lower} = \frac{1}{2}\left(p_{i-1,j-1} + p_{i,j-1}\right). \qquad (22d)$$

The derivatives in computational space are

$$\frac{\partial p}{\partial i} = p_{right} - p_{left}, \qquad (23a)$$

$$\frac{\partial p}{\partial j} = \frac{1}{2}\left(p_{upper} - p_{lower}\right). \qquad (23b)$$

Equations 23a and 23b are transformed back to the physical plane using the grid metrics:

$$\frac{\partial p}{\partial x} = \frac{\partial p}{\partial i}\frac{\partial i}{\partial x} + \frac{\partial p}{\partial j}\frac{\partial j}{\partial x}, \qquad (24a)$$

$$\frac{\partial p}{\partial y} = \frac{\partial p}{\partial i}\frac{\partial i}{\partial y} + \frac{\partial p}{\partial j}\frac{\partial j}{\partial y}. \qquad (24b)$$

The pressure gradient angle θ_p then follows from

$$\theta_p = \tan^{-1}\frac{\partial p/\partial y}{\partial p/\partial x}. \qquad (25)$$

To avoid random angles in areas with small pressure gradients, the final upwinding angle is a blend of the pressure gradient angle with the flow angle:

$$\theta = \frac{\epsilon\frac{|\nabla p|}{\rho a^2}\theta_p + (1-\epsilon)\frac{|u|}{a}\theta_s}{\epsilon\frac{|\nabla p|}{\rho a^2} + (1-\epsilon)\frac{|u|}{a}}. \qquad (26)$$

Here, ϵ is a weighting factor. In the present implementation, the value $\epsilon = 0.20$ is the largest value that will allow convergence, as discussed below.

Interpolation Methods for First-Order Scheme

As in the method of Davis [3], the present method determines by interpolation the input states for the flux calculation in the frame of the upwinding angle. Two interpolation methods are tested. The first method searches a cluster of six cells surrounding the cell face of interest to find between which cell centers the upwind and normal directions fall (Figure 2). For an arbitrary grid, implementation of this method is rather cumbersome, although it can be made completely general.

The second method maps the six-cell cluster (shown in Figure 3a) onto $\xi \in [-1,1]$, $\eta \in [-1,1]$ (Figure 3b), through the linear-quadratic coordinate transformations:

$$\xi(x,y) = a_1 + a_2x + a_3y + a_4xy + a_5y^2 + a_6xy^2, \qquad (27a)$$

$$\eta(x,y) = b_1 + b_2x + b_3y + b_4xy + b_5y^2 + b_6xy^2, \qquad (27b)$$

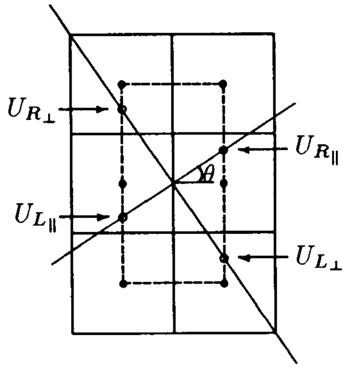

Figure 2: Interpolation in the Physical Plane

A transformation similar to 27a–27b, but quadratic in x and linear y, is used for the horizontal cell faces. The slope of the line $y = mx + b$ in the upwind direction going through the midpoint of the cell face transforms according to:

$$m' = \frac{b_2 + b_3m + b_4t_1 + b_5t_2 + b_6t_3}{a_2 + a_3m + a_4t_1 + a_5t_2 + a_6t_3}, \qquad (27c)$$

where:

$$t_1 = 2m(mx_m + b), \qquad (28a)$$

$$t_2 = 2mx_m + b, \qquad (28b)$$

$$t_3 = 3m^2x_m^2 + 4mbx_m + b^2. \qquad (28c)$$

If the line is vertical ($m \rightarrow \infty$), then:

$$m' = \frac{2b_4x_m + 3b_6x_m^2}{2a_4x_m + 3a_6x_m^2} \qquad (28d)$$

The advantage of this method is that it simplifies the searching procedure, as the geometry in the mapped plane is always the same. To calculate the left state for $m' < -1$, the interpolation is done between the average quantities in lower left and lower right cells. If $-1 \leq m' \leq 0$, the left and lower left cells are used, while for $0 \leq m' \leq 1$, the left and upper left states are used. Finally, if $1 < m'$ the interpolation is done between the upper left and upper right states. A similar procedure is followed for finding the right state in the upwind direction and for finding finding both states in the direction normal to the upwinding angle (cross direction). Note that the interpolation is linear in the (ξ, η) plane.

The methods produce similar results. The cases presented below, with one exception, are computed with the interpolation in the mapped plane.

Flux Formulation in Rotated Frame

The generic fomula for a flux normal to a cell face is

$$F = F\left(U_L, U_R\right); \qquad (29)$$

in a conventional first-order-scheme the left and right input states simply are the average values in the adjacent cells. Convection speeds are based on velocity components parallel and perpendicular to the cell face; the upwinding direction, i.e., the direction in which the Riemann problem is solved, is normal to the cell face.

When an upwind direction is chosen that is not normal to the cell face, the flux component in this direction is

$$F_\perp = F_\perp\left(U_{L_\perp}, U_{R_\perp}\right). \qquad (30)$$

This flux is computed as before, with two changes. The velocity components are now in the new reference frame, and the left and right states are interpolated from the values in the nearest cell centers, as shown in Figure 3. The subscript $\perp$ is chosen to indicate that the direction, although not normal to a cell face, may be normal to another important line, such as a shock front.

Next, a flux normal to this direction, e.g. *along* a shock front, must also calculated; for this, simple averaging will suffice:

$$\begin{aligned} F_{\parallel} &= F_{\parallel}\left(U_{L_{\parallel}}, U_{R_{\parallel}}\right) \qquad (31) \\ &= \frac{1}{2}\left(F\left(U_{L_{\parallel}}\right) + F\left(U_{R_{\parallel}}\right)\right). \end{aligned}$$

In the final scheme, this leads to central differencing of this flux component. Note that this flux component also exists in the conventional formulation, but does not contribute to the flux through the cell face.

The flux normal to the cell face is constructed by rotating the above fluxes back to the coordinate frame normal to the cell face:

$$\begin{aligned} &F\left(U_{L_\perp}, U_{R_\perp}, U_{L_{\parallel}}, U_{R_{\parallel}}\right) = \qquad (32) \\ &\qquad cos\left(\theta - \phi\right) F_\perp - sin\left(\theta - \phi\right) F_{\parallel}. \end{aligned}$$

Here, θ is the upwinding angle and ϕ is the angle normal to the cell face. The computation of the upwind biased fluxes in a rotated frame, followed by rotating back to the computational frame, makes a difference only for the nonlinear waves. For passive convection, such as entropy convection, there is no change.

It is possible to use upwind differencing for both flux components. This would lead to a better approximation of the effect of pressure waves in the cross direction. However, this almost doubles the computational effort of calculating the flux through the cell face.

Boundary Procedures

For the inlet and exit boundaries, conventional characteristic boundary procedures are employed. For cell faces coinciding with a solid wall boundary, the normal velocity is zero, so the mass and energy fluxes can be set to zero. Only the pressure terms contribute to the momentum flux terms. These are calculated based on the cell average value from the adjacent cell. This procedure corresponds to the use of central differencing normal to the wall; this in turn implies upwind differencing *along* the wall, which is also the streamwise direction.

Results and Discussion

The test-case geometry is a two dimensional channel with a 15° wedge on the lower wall. A 15° expansion corner is also included to study expansion waves and wave interactions. The inflow Mach number equals 2. An oblique shock wave is produced by the wedge, and is somewhat weakened by the expansion wave. The shock reflects from the upper wall and passes through the rest of the expansion wave. The expansion wave also reflects from the upper wall slightly downstream of the shock reflection point.

Conventional, analytical calculations predict a post-shock Mach number of 1.454. For this Mach number, any turning angle greater than 10.5° is too large to allow a regular reflection off the upper wall. Thus, a 'Mach reflection' results, with subsonic flow behind the Mach stem and a slip surface trailing from the intersection of the incident, normal, and reflected shock waves [9]. In this test case, however, the incident shock wave interacts with the expansion wave. The shock incident on the upper wall may still be too strong to allow a regular reflection; the numerical solutions seem to support this conclusion.

The grids used are algebraic, with constant Δx and $\Delta y = (y_{max} - y_{min})/N$, where N is the number of cells in the y direction. The three grids are shown in Figures 4–6.

For comparison, grid-aligned upwinding cases are shown first for coarse (60x20), fine (120x40), and extra fine (180x60) grids (Figures 7–9). On the coarse grid, the shock wave is substantially diffused after reflection, as expected from a first-order-scheme. The fine grid case has improved resolution. The extra fine grid case shows little improvement in resolution, besides narrower shocks, indicating proximity to a grid-converged solution.

Sensitivity to the upwinding angle is shown in Figure 10. In this case, an upwinding direction of 45° is specified only for the horizontal cell faces. The upwind angle for the vertical cell faces is fixed at 0° relative to the x-axis. Notice that the reflected shock wave is better resolved than in the grid-aligned case of Figure 7, as the upwinding direction is approximately normal to this wave. Clearly there is reason to upwind in special directions; the question is which direction is best.

The result shown in Figure 10 for fixed-θ is based on interpolation in the physical domain, except in the five rows of cells nearest to the walls, where the angle is gradually turned toward the wall direction ('blending'). All other grid-independent results discussed below use interpolation in the mapped domain without blending.

Figures 11 and 12 show the result of using the local flow direction as the upwinding direction (streamwise upwinding). The expansion wave and the reflected shock wave are well resolved. Comparison with the grid-aligned cases of Figures 7–9 shows that the resolution of the grid-independent case on the coarse grid is comparable to that of the grid-aligned upwinding case on the fine grid, etc. This is especially true of the shock wave reflecting from the lower wall near the exit of the channel. The Mach contours of Figures 11-12, and specifically the subsonic Mach levels after the reflected shock, hint at the presence of the analytically predicted Mach reflection. This effect is not present in even the finest-grid case for grid-aligned upwinding.

One reason for the improved resolution and reduced dissipation is that the grid-independent scheme calculates a part of the flux with central-differencing, which is a more accurate approximation ($\mathcal{O}(\Delta x)^2$) than upwind differencing ($\mathcal{O}(\Delta x)$). In fact, the flux through faces aligned with the local flow direction is entirely composed of the central-differencing component. In the present scheme, the stabilizing effect of using upwind differencing in the dominant direction allows the use of central differencing in the cross direction.

Figures 13 and 14 show the upwind directions for the grid-aligned and grid-independent cases. There is a significant difference for the horizontal cell faces, while there is not much difference for the vertical cell faces.

Pressure-gradient upwinding, as seen from Figure 15, shows little improvement over streamwise upwinding, except near the expansion corner. Figure 16 shows the upwinding angles for the pressure gradient case.

Although the directional upwinding shows improved performance in comparison to the grid-aligned method, it comes at a price. The non-linearities introduced by the method cause convergence difficulties. To remedy this, the upwind angle is recalculated at specific times, and is frozen when the residual falls below 10^{-3}. For streamwise upwinding, the upwinding angle is recalculated every 50 time steps. For pressure gradient upwinding, this process still results in limit cycles, so the upwinding angle is recomputed each time the residual falls below 10^{-3}. Each time the angle is recomputed, ϵ is increased linearly. For the given example, intermediate values of 0.05, 0.10, and 0.15 are used, until the final value of $\epsilon = 0.20$ is reached. Even with this phasing in procedure, limit cycles resulted for $\epsilon > 0.25$. Figure 17 shows the convergence characteristics for grid-aligned, streamwise, and pressure-gradient upwinding.

The calculation of the left and right states requires additional computational effort, so total run time is somewhat increased for fixed grid size. The computation of the centered flux and the rotation back to cell-face direction, plus the interpolation procedure increase computation time by 33% per iteration. The present code, with storage for the mapping coefficients and the states, requires 56 additional words per cell of memory over a grid-aligned scheme. There is a trade-off between memory and computational speed. It may be possible to eliminate the storage for the left, right, upwind and normal states without sacrificing speed. The mapping coefficients, however, require inversion of a six by six matrix, and they never change, so it is worthwhile to save these in memory. There may still be room for speeding up the interpolation process.

The present implementation of grid-independent upwinding also introduces a short-wave instability due to lack of dissipation near the wall boundaries. Pressure oscillations of wavelength $2\Delta y$ propagate from the wall into the domain vertically. The instability is aggravated as the grid is refined. Residual averaging is used to help solve this stability problem, but a solution for streamwise upwinding on the extra-fine grid mesh could not be obtained. A modification of the wall boundary procedures may help solve this problem.

Examination of the total-pressure contours shown in Figures 18 and 19 reveals the pressure loss through the initial shock wave, and its convection downstrwam. Entropy is being generated at the expansion corner for both grid-aligned and streamwise upwinding. There is a larger entropy gain in the streamwise case, although the layers are of comparable thickness.

In general, the results of grid-independent upwinding show more numerical noise, due to the inherent non-linearity of the method. For example, in the grid-independent cases there are overshoots on the lower wall, just ahead of the wedge, and behind the incident and reflected shock waves.

Extension to Second Order Accuracy

To achieve second-order accuracy, a template of 16 cells surrounding the cell face of interest is used (Figure 20). Unit vectors from the midpoint of the face to each of the 16 cell centers are calculated, as well as the unit vector oriented in the desired upwinding direction. The dot-product of each cell direction vector with the upwind vector is calculated. The cells with the four most negative dot products are chosen to form the 'left' state; the four most positive products are chosen to form the 'right' state. The same method is used to calculate the states in the cross direction.

To implement this procedure, the 16 dot products with the upwind direction vector. are put in ascending order with a standard bubble sort. The same method is used to sort the dot products with the vector in the cross direction. Figure 20 shows the selection of the left states for various upwind directions.

The state value at the midpoint of the cell face is a weighted average of the four cell center values. The weights, w_i, are chosen such as to:

1. Preserve a constant solution,
2. Preserve a linear function in the x-direction,
3. Preserve a linear function in the y-direction,
4. Preserve a bilinear function in x and y.

This results in the following equation:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & x_2 & x_3 & x_4 \\ x_1y_1 & x_2y_2 & x_3y_3 & x_4y_4 \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} 1 \\ x_m \\ y_m \\ x_my_m \end{bmatrix} \tag{33}$$

The cell-face midpoint state is then calculated from

$$\mathbf{U}_m = w_1\mathbf{U}_1 + w_2\mathbf{U}_2 + w_3\mathbf{U}_3 + w_4\mathbf{U}_4. \tag{34}$$

This, in effect, is a bilinear curve fit of the data.

For the regular grid used in these calculation, the above matrix is singular for the 'T' shape stencil in Figure 20c. This case occurs frequently, e.g. when the flow is uniformly aligned with the grid. Logic is included to avoid these singularities by replacing the fourth most upwind cell with the fifth most upwind cell.

The second-order interpolation does not depend as smoothly on θ as the interpolation for the first-order scheme. The upwinding direction is used to choose the cells with which to form the curve fit, but the fit is based on grid geometry. The Riemann problem is still solved in a directional manner, however.

This method has yet to be completely implemented. As a preliminary step, a fixed set of cells were used to form the bilinear curve fit. The cells chosen are representative of those that would be chosen for the dominant flow directions present in the test case.

In the test, the wedge angle is reduced to 7.5° to reduce the magnitude of overshoots typical of a second-order scheme (no limiters are used). For this test case, a post-shock Mach number of 1.705 and a regular reflection from the upper wall are predicted. The forward-Euler time-stepping algorithm used for the first-order scheme is generally unstable for a second-order scheme. Instead, the optimally smooth, six-stage scheme described in Van Leer et al.[10] is used.

Results for the second-order, grid-dependent scheme on coarse and fine grids are shown in Figures 21 and 22. Results for the fixed-template, streamwise upwinding on the same grids are shown in Figures 23 and 24.

There is some improved resolution over the grid-aligned cases, but it is not dramatic. The improvement could simply be due to the fact that more cells are used to extrapolate to the cell face. The overshoot ahead of the shock wave is somewhat aggravated.

Conclusions and Recommendations

Upwind schemes have been formulated for the two-dimensional Euler equations with the direction of upwind differencing independent of the grid geometry. A first-order scheme has been implemented with the upwinding angle based on the flow pattern, such as flow angle and pressure-gradient angle. This formulation exploits the multi-dimensional nature of the flow and leads to improved resolution of flow features such as oblique shock waves.

A second-order scheme has also been implemented that may be regarded as a two-dimensional extension of the fully one-sided second-order scheme in one dimension.

The present codes require more memory and have longer run times than the codes based on grid-aligned upwind differencing for comparable grid sizes. There are still possibilities for reducing memory requirements and run times. Future research will focus on improving robustness and testing different flow geometries.

Acknowledgements

This work was funded in part by the McDonnell Aircraft Company under a grant monitored by Dr. August Verhoff, and by the National Science Foundation under Grant #EET-8857500, monitored by Dr. George Lea.

References

[1] K. G. Powell and B. van Leer, "A genuinely multidimensional upwind cell-vertex scheme for the Euler Equations," AIAA Paper 89-0095, 1989.

[2] R. Struijs and H. Deconinck, "A multidimensional upwind scheme for the Euler equations using fluctuation distribution on a grid consisting of triangles." Submitted for the Eighth GAMM Conference, Delft, 1989.

[3] S. F. Davis, "A rotationally-biased upwind difference scheme for the Euler equations," *Journal of Computational Physics*, vol. 56, 1984.

[4] C. Hirsch, C. Lacor, and H. Deconinck, "Convection algorithm based on a diagonalization procedure for the multidimensional Euler equations," in *AIAA 8th Computational Fluid Dynamics Conference*, 1987.

[5] A. Verhoff and P. J. O'Neil, "A natural formulation for numerical solutions of the Euler equations," Tech. Rep. MCAIR 83-031, McDonnell Aircraft Company, 1983.

[6] G. Moretti, "The λ-scheme," *Computers and Fluids*, vol. 7, 1979.

[7] A. Jameson, "Iterative solution of transonic flows over airfoils and wings," *Communications on Pure and Applied Mathematics*, vol. 28, 1974.

[8] P. L. Roe, "Approximate Riemann solvers, parameter vectors and difference schemes," *Journal of Computational Physics*, vol. 43, 1981.

[9] H. W. Liepmann and A. Roshko, *Elements of Gasdynamics.* John Wiley & Sons, Inc., 1957.

[10] B. van Leer, C. H. Tai, and K. G. Powell, "Design of optimally-smoothing multi-stage schemes for the Euler equations," in *AIAA 9th Computational Fluid Dynamics Conference*, 1989.

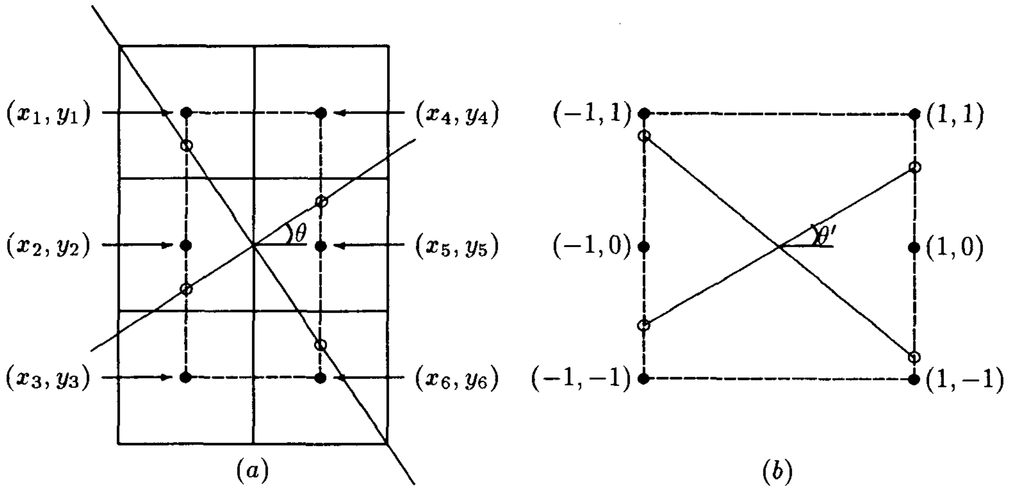

Figure 3: Interpolation in the Mapped Plane

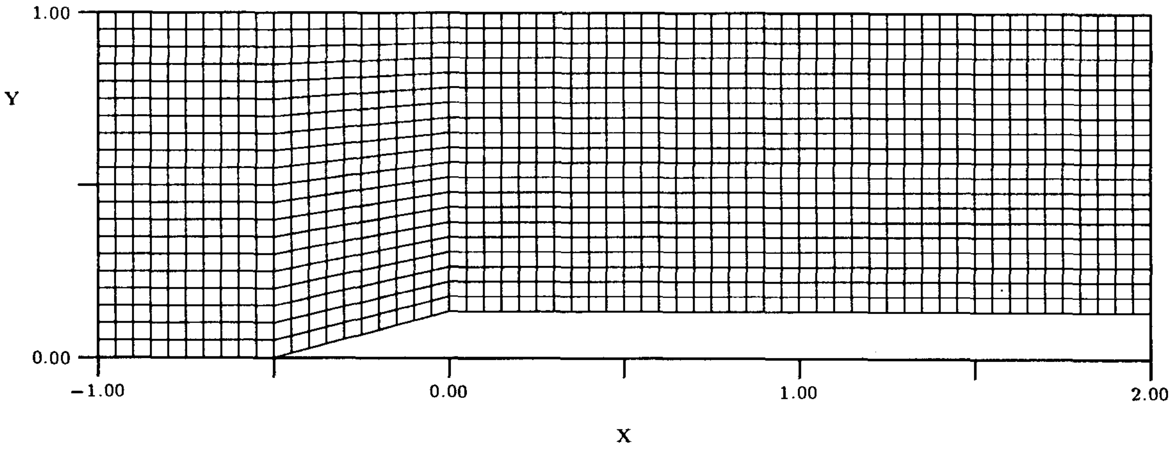

Figure 4: Coarse (60x20) Grid for the Test Geometry

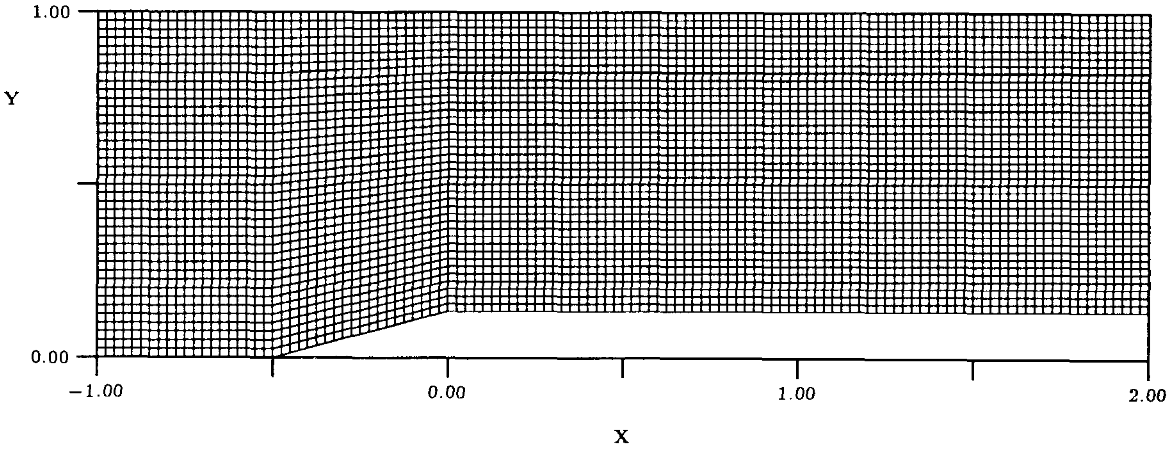

Figure 5: Fine (120x40) Grid for the Test Geometry

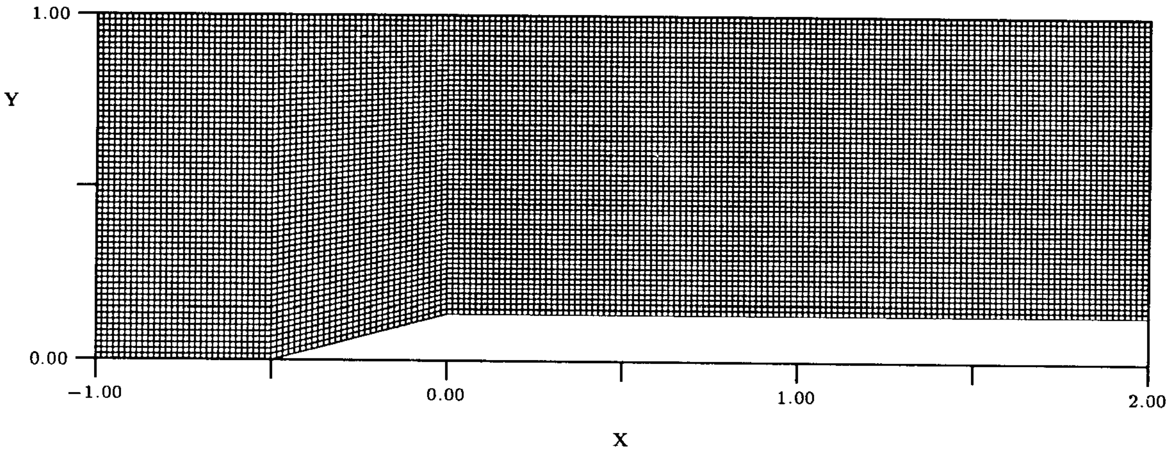

Figure 6: Extra-Fine (180x60) Grid for the Test Geometry

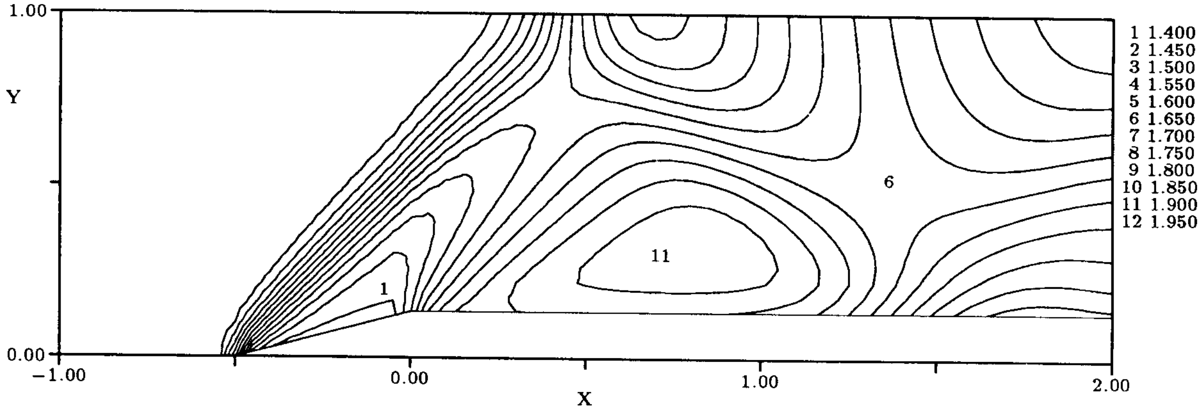

Figure 7: Mach Contours for Grid-Aligned Upwinding, Increment = 0.05, Coarse Grid (60x20)

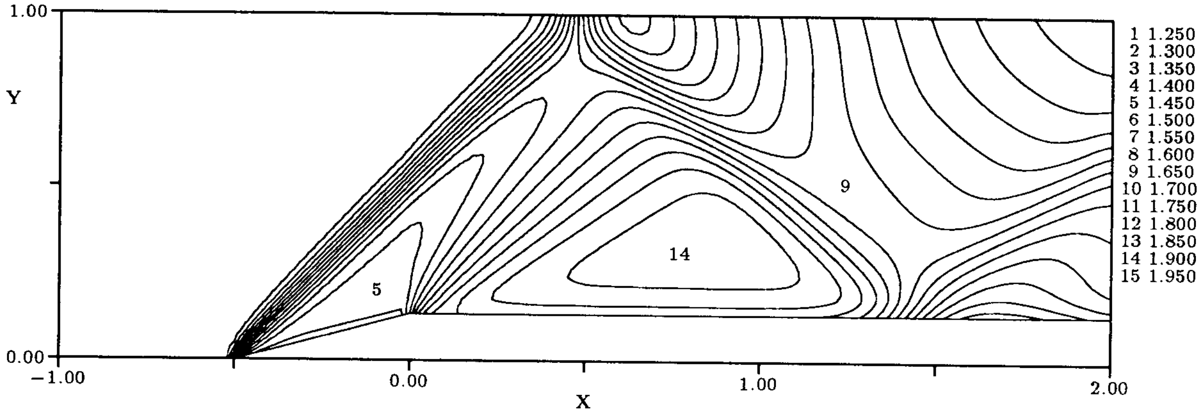

Figure 8: Mach Contours for Grid-Aligned Upwinding, Increment = 0.05, Fine Grid (120x40)

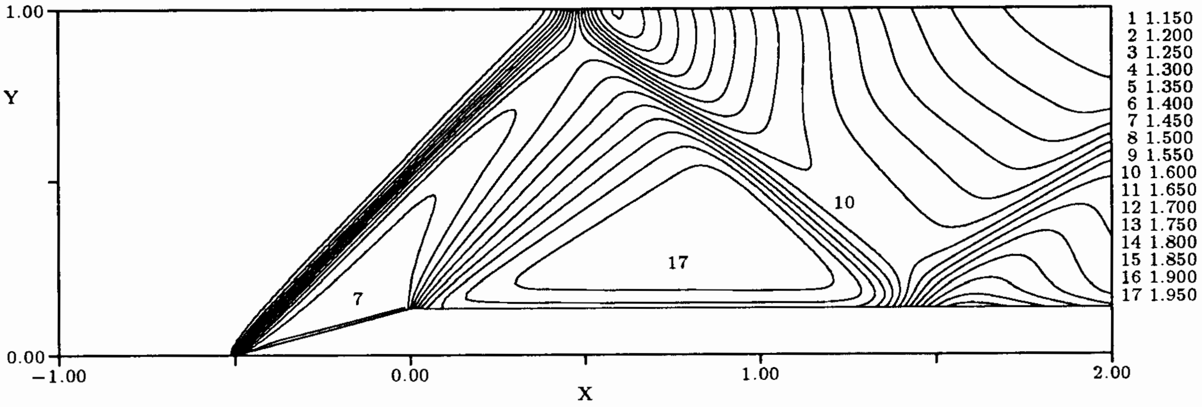

Figure 9: Mach Contours for Grid-Aligned Upwinding, Increment = 0.05, Extra Fine Grid (180x60)

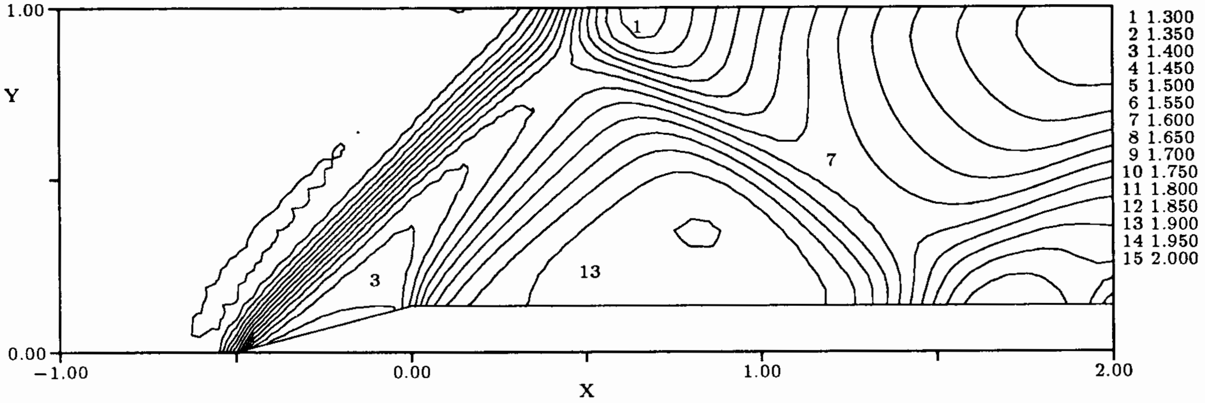

Figure 10: Mach Contours for Fixed-θ Upwinding, Increment = 0.05, Coarse Grid, $\theta = 45°$

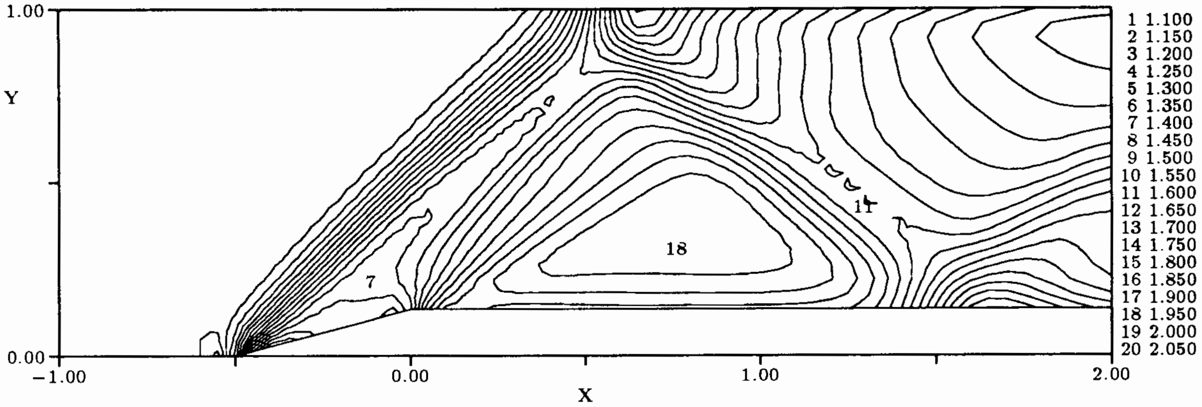

Figure 11: Mach Contours for Streamwise Upwinding, Increment = 0.05, Coarse Grid

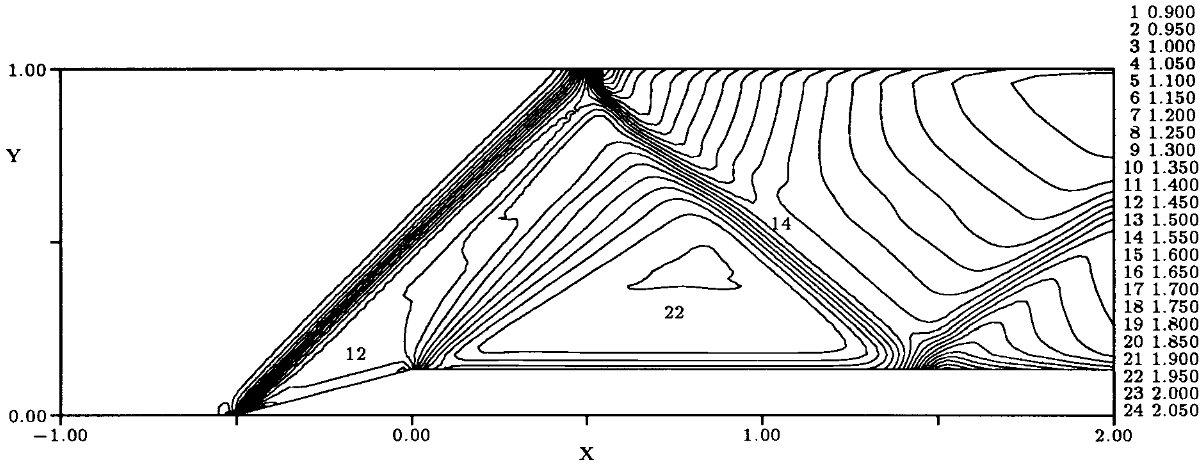

Figure 12: Mach Contours for Streamwise Upwinding, Increment = 0.05, Fine Grid

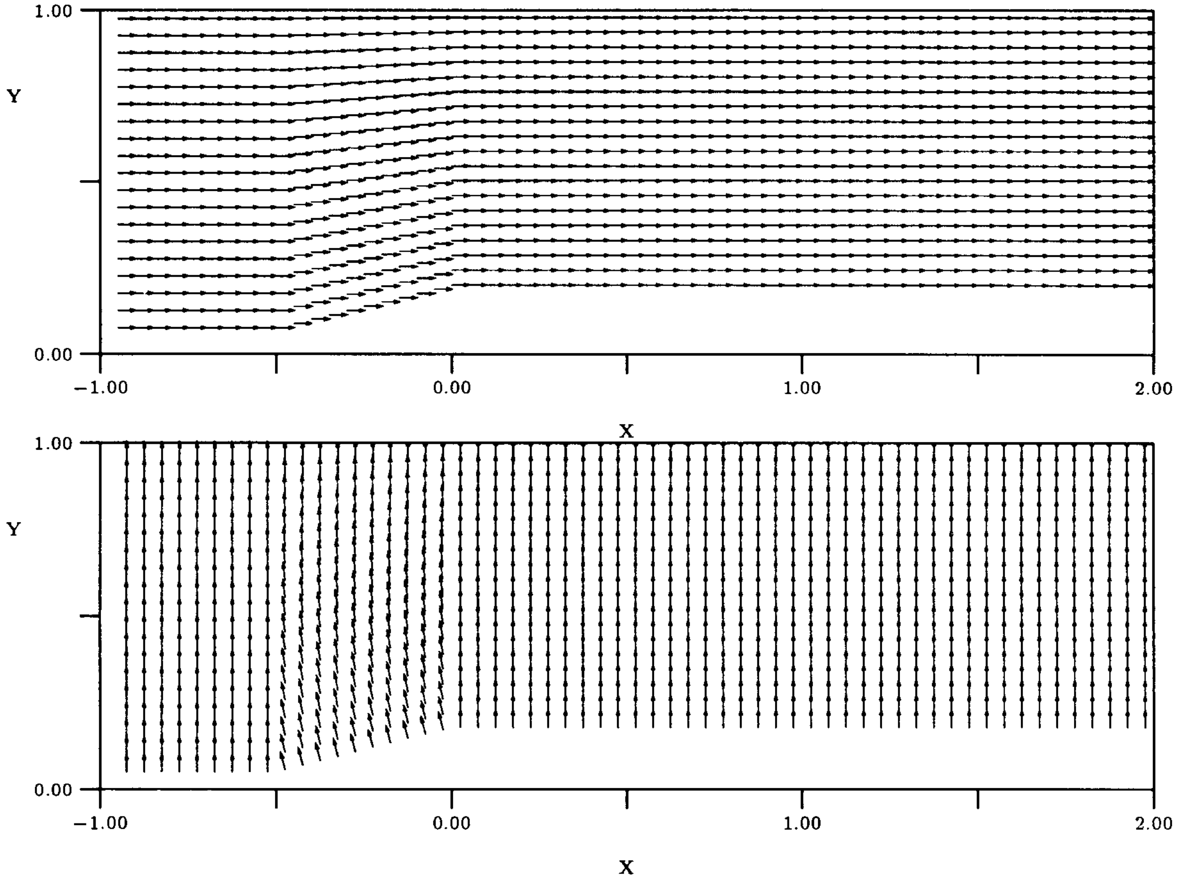

Figure 13: Upwinding Angles for Grid-Aligned Case, Coarse Grid

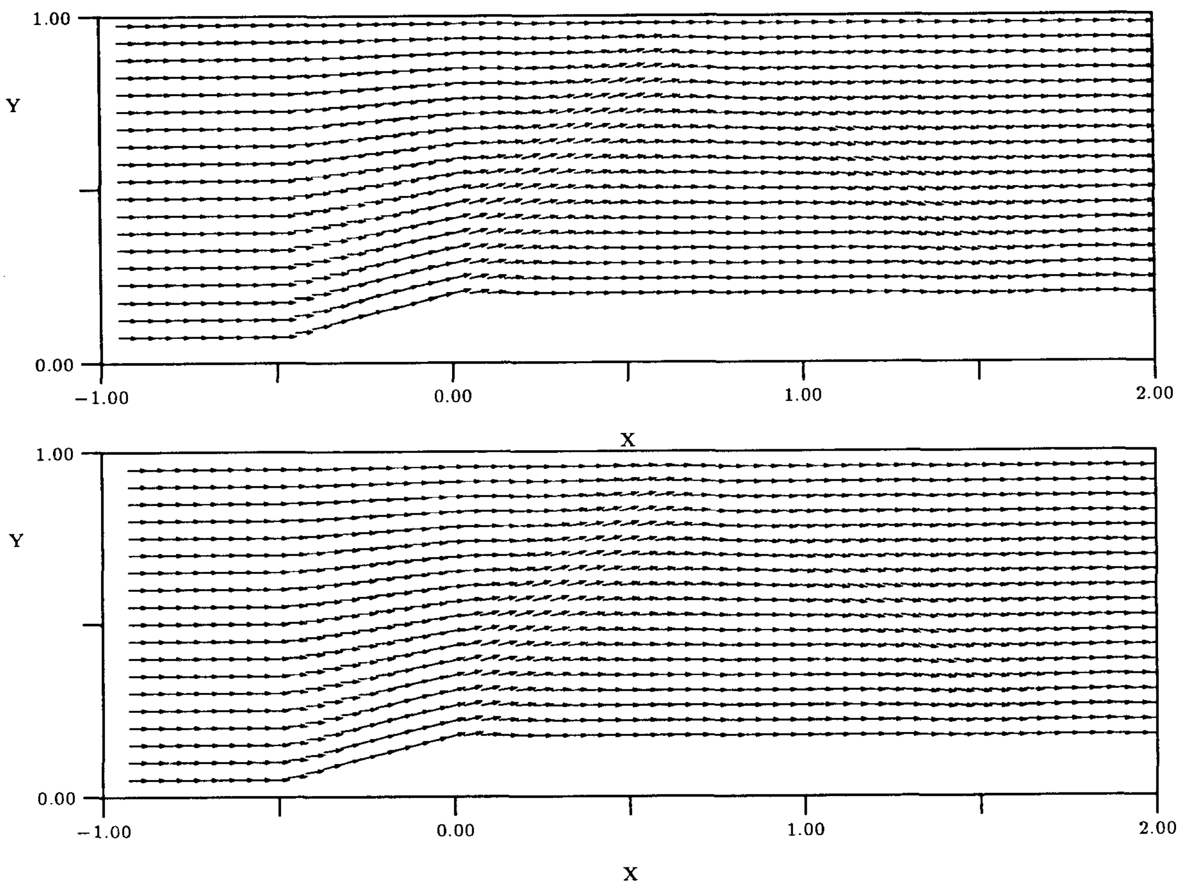

Figure 14: Upwinding Angles for Streamwise Case, Coarse Grid

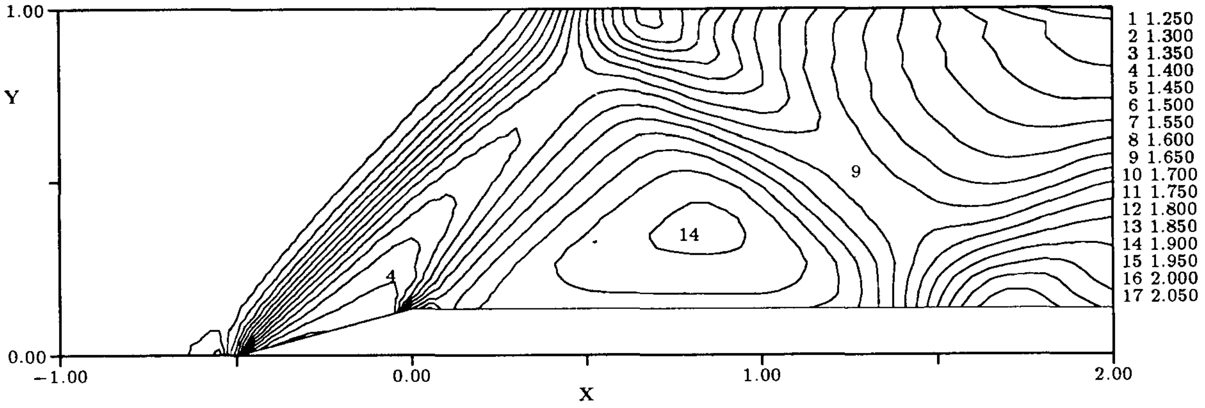

Figure 15: Mach Contours for Pressure-Gradient Upwinding, Increment = 0.05, Coarse Grid

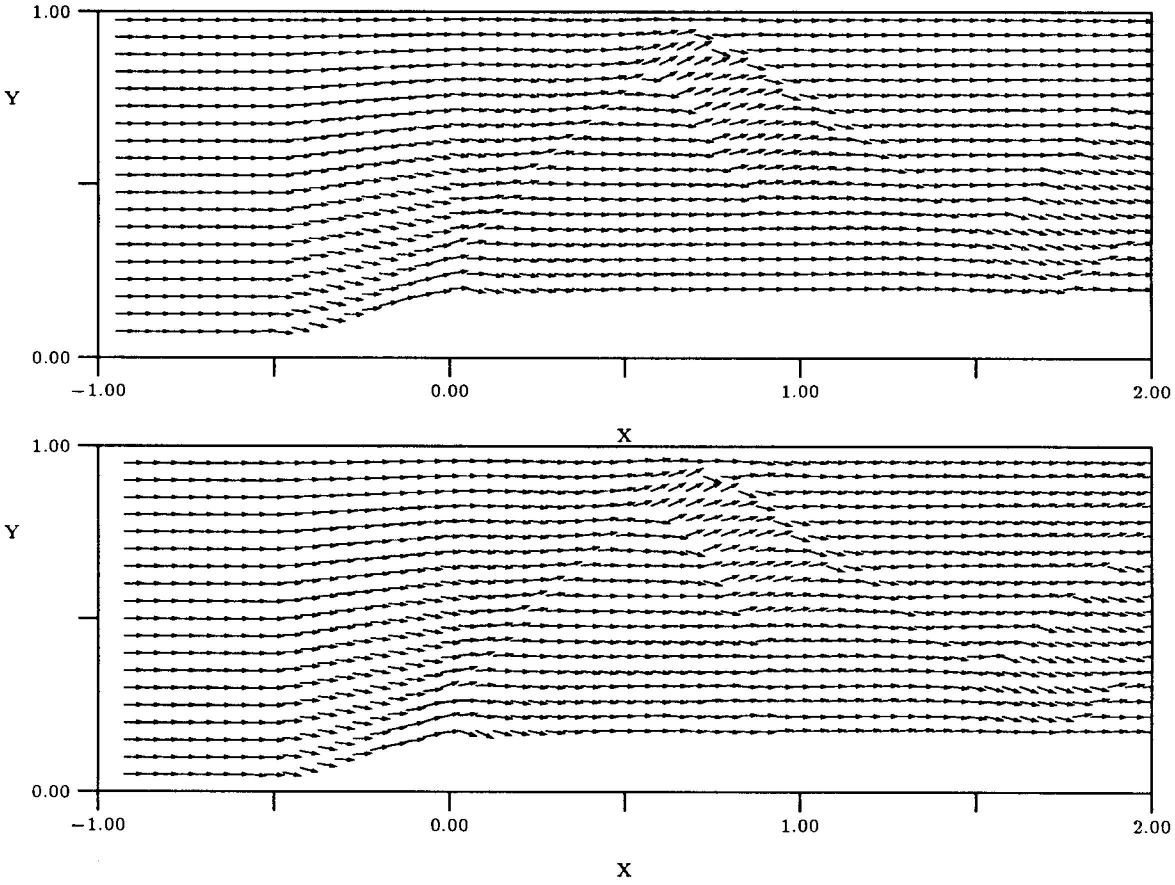

Figure 16: Upwinding Angles for Pressure-Gradient Case, Coarse Grid

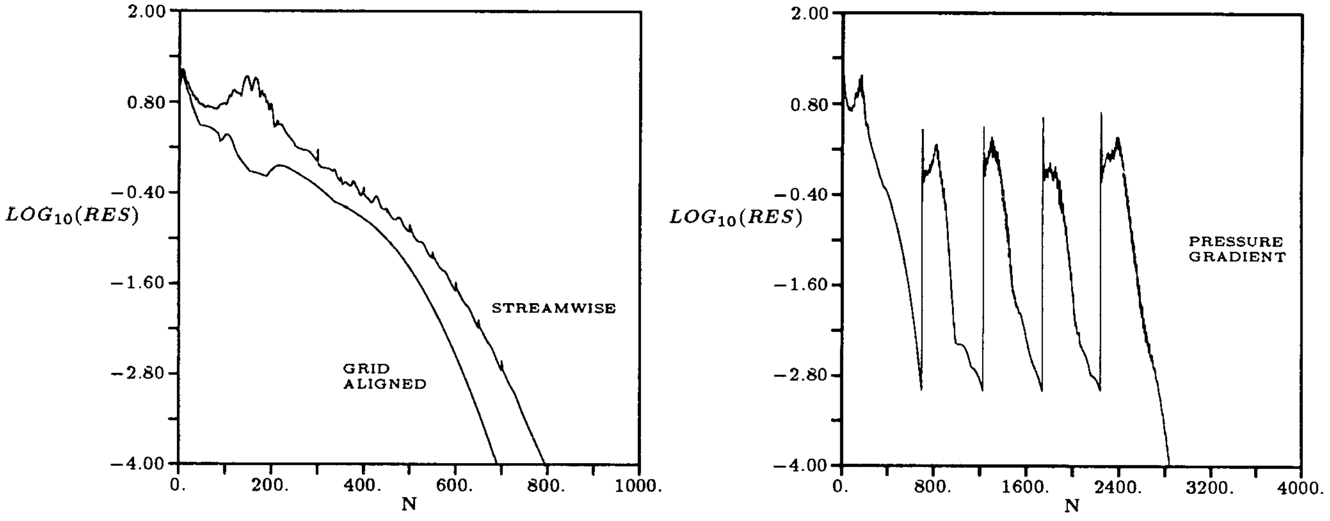

Figure 17: Convergence characteristics for Grid-Aligned, Streamwise, and Pressure-Gradient Upwinding

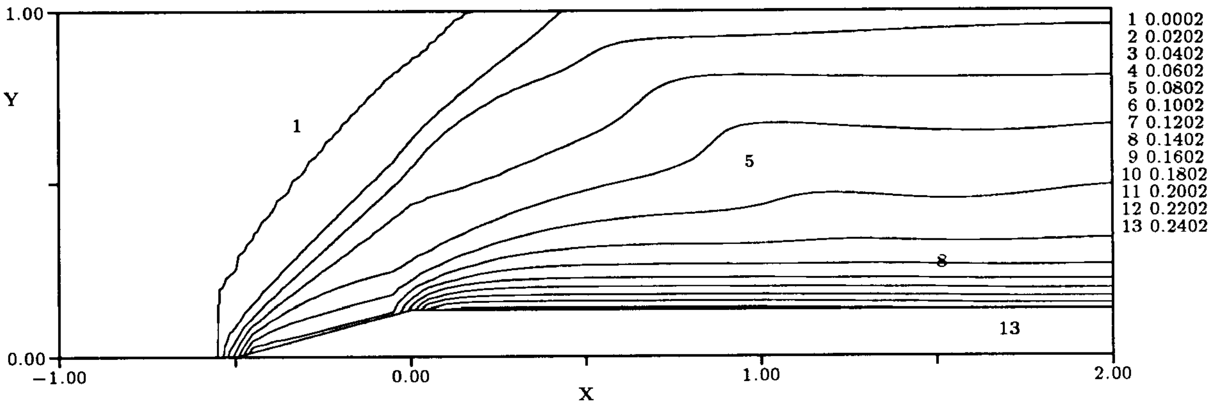

Figure 18: Stagnation Pressure Loss, $1 - p_s/p_{s_\infty}$, Contours for Grid-Aligned Upwinding, Increment = 0.02, Coarse Grid

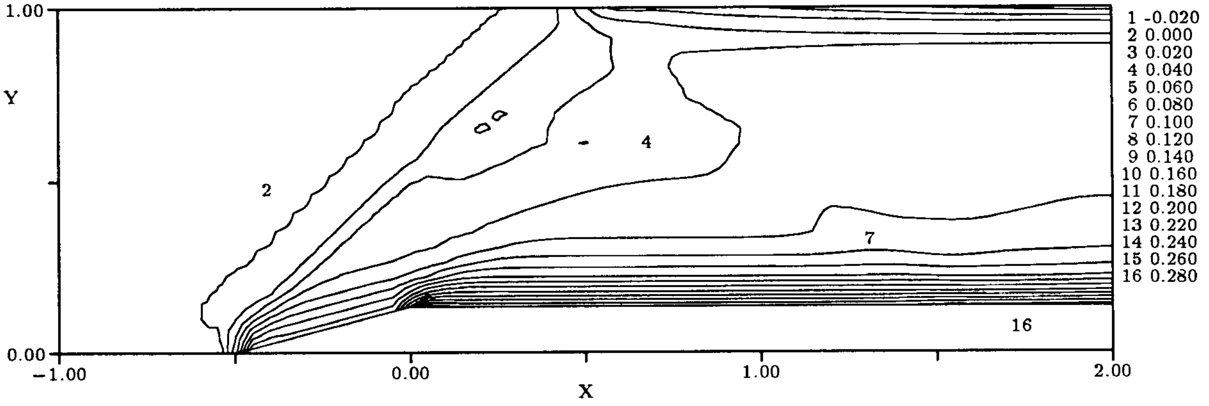

Figure 19: Stagnation Pressure Loss, $1 - p_s/p_{s_\infty}$, Contours for Streamwise Upwinding, Increment = 0.02, Coarse Grid

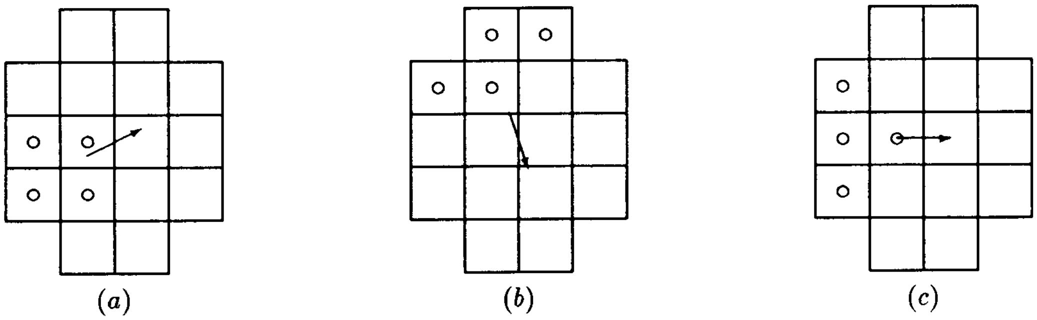

Figure 20: Cells Used in the Interpolation Method for the Second-Order Scheme

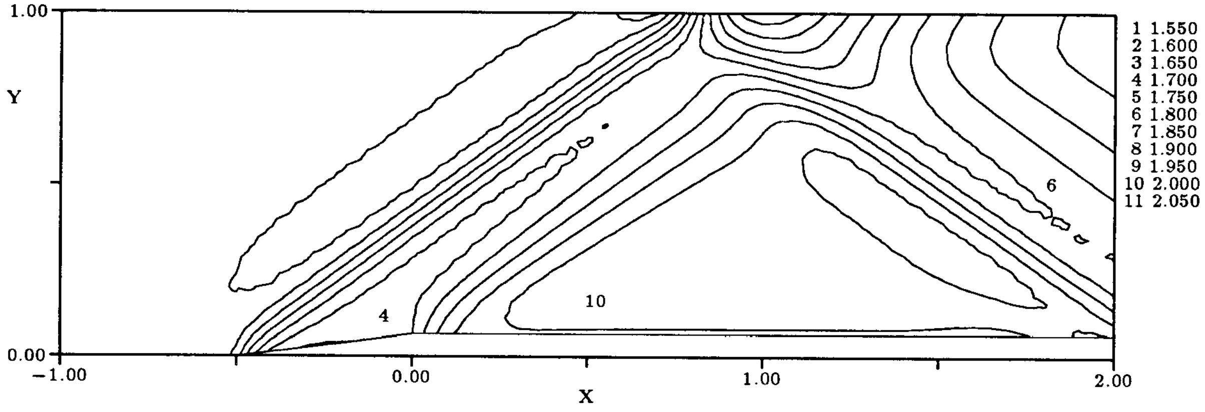

Figure 21: Mach Contours for Second-Order, Grid-Aligned Upwinding, Increment = 0.05, Coarse Grid

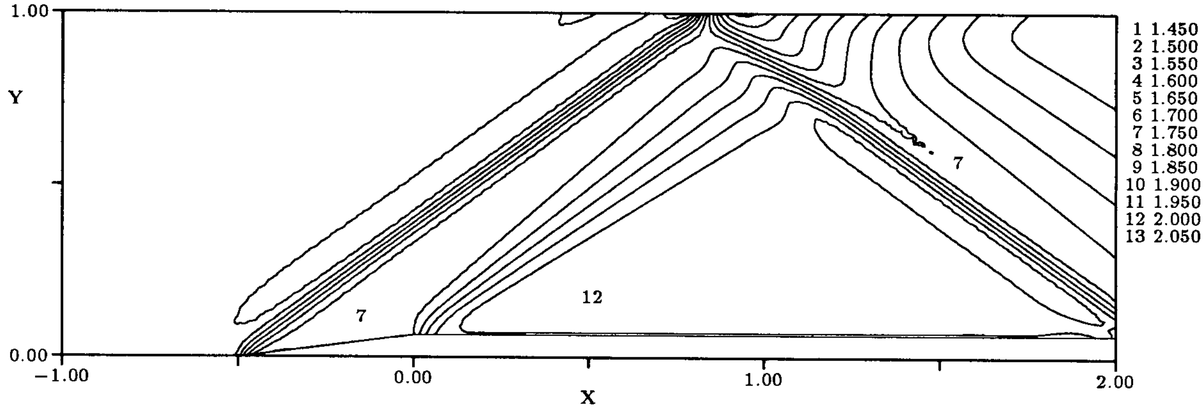

Figure 22: Mach Contours for Second-Order, Grid-Aligned Upwinding, Increment = 0.05, Fine Grid

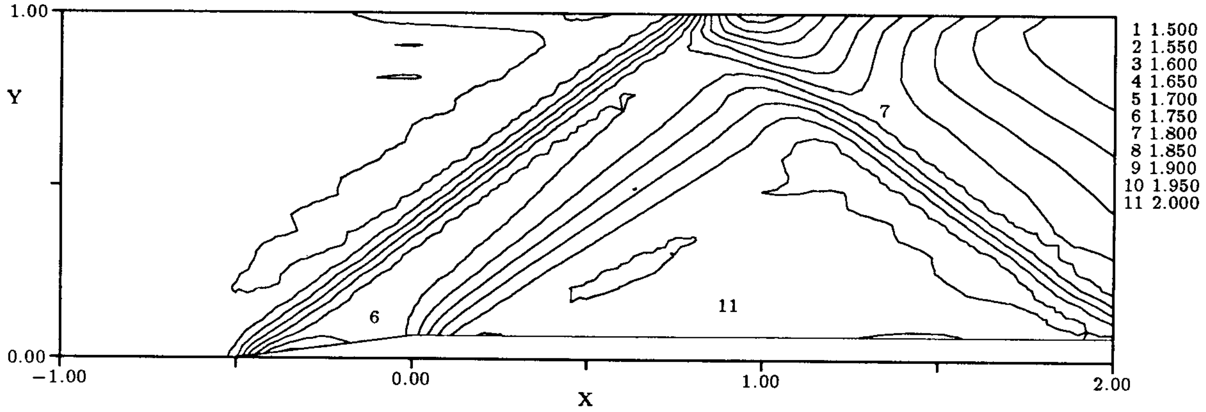

Figure 23: Mach Contours for Second-Order, Streamwise Upwinding, Increment = 0.05, Coarse Grid

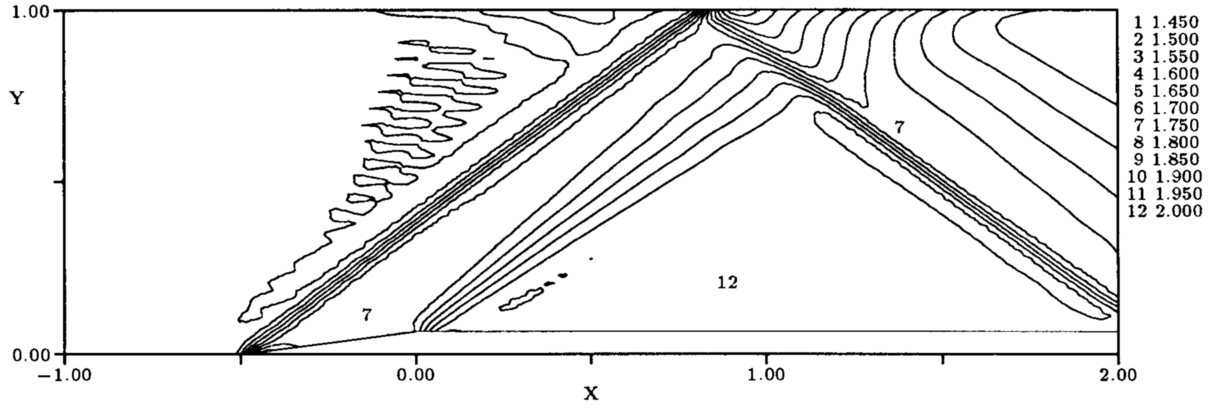

Figure 24: Mach Contours for Second-Order, Streamwise Upwinding, Increment = 0.05, Fine Grid