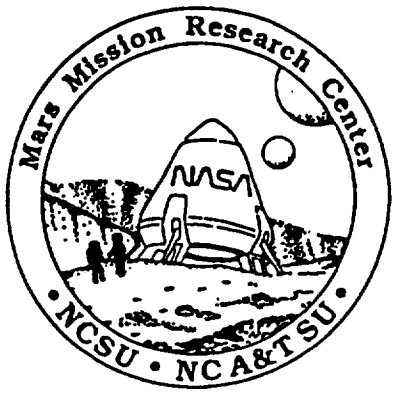
NASA-CR-199428



Final Report

for

NASA Cooperative Agreement NCA2-719
NASA Ames Research Center

Numerical Simulation of Supersonic and Hypersonic Inlet Flow Fields

D. Scott McRae
Dean A. Kontinos
Mars Mission Research Center
North Carolina State University
Raleigh, North Carolina

July 21, 1995

ABSTRACT

This report summarizes the research performed by North Carolina State University and NASA Ames Research Center under Cooperative Agreement NCA2-719, "Numerical Simulation of Supersonic and Hypersonic Inlet Flow Fields". Four distinct rotated upwind schemes were developed and investigated to determine accuracy and practicality. The scheme found to have the best combination of attributes, including reduction to grid alignment with no rotation, was the cell centered non-orthogonal (CCNO) scheme. In 2-D, the CCNO scheme dramatically improved accuracy when used with first order flux interpolation. CCNO also improved rotation when flux interpolation was extended to second order. In 3-D, improvements were less dramatic in all cases, with second order flux interpolation showing the least improvement over grid aligned upwinding. The reduction in improvement is attributed to uncertainly in determining optimum rotation angle and difficulty in performing accurate and efficiently interpolation angle in 3-D. The CCNO rotational technique will prove very useful for increasing accuracy when second order interpolation is not appropriate and will materially improve inlet flow solutions.

INTRODUCTION

The development programs of the High Speed Civil Transport, National Aerospace Plane, and next generation fighter/attack aircraft have identified the need for greater understanding and predictive capability of the complex fluid dynamic processes occurring in high speed inlet configurations. The flow fields in the inlet are replete with intricate fluid phenomena such as crossing shock waves, shock-wave/turbulent-boundary-layer interactions, merging corner flow boundary layers, and vortical flow. Because of limitations in wind tunnel capabilities such as limitations of scale and in non-obtrusive measuring techniques, much of the understanding of inlet flowfields is a result of numerical simulation. These computer solutions rely heavily on state-of-the-art numerical techniques.

A commonly and confidently used numerical procedure is the upwind scheme which has been a major breakthrough in the modeling of fluid flow in the transonic through hypersonic flow regimes. The upwind schemes have provided the ability to capture flow discontinuities within a few grid points without tuning the artificial dissipation. However, it is well known that excess dissipation is generated by upwind schemes when the captured discontinuities are oblique to the computational grid. The excess dissipation smears the captured shock waves, thereby reducing the accuracy of the numerical predictions. High speed inlet configurations are especially prone to this effect because the grid topologies are naturally h-type configurations with oblique crossing shock waves, reflecting shock waves, and shock-wave/boundary-layer interactions. Thus it is the objective of this research to develop a numerical model that improves the prediction and resolution of these dominant flow features in regions where they are captured oblique to the computational mesh and thereby improve the accuracy of the global inlet solution.

A promising approach to reducing the aforementioned smearing effect is employing a rotated upwind scheme. The rotated upwind procedure is one of a class of newly developing numerical procedures that attempt to introduce multi-dimensional dependency into the numerical integration. The basic idea of the rotated upwind scheme is to dynamically align the upwind difference stencil in a direction based on the developing flow field features. This procedure is an alternative to traditional schemes where the dissipation model is affixed by the computational mesh. The realignment of the upwind operator enhances the ability of the computational model to predict more accurately the local flow physics. The rotation is especially beneficial in regions where a dominant flow feature, such as a shock wave, exists and is oblique to the computational grid. As mentioned previously, these features commonly occur in high speed inlet simulations.

A focused investigation was conducted under this cooperative agreement to determine how the rotation could best be performed and to ascertain how rotation improves the computational results. Rather than repeat published explanations and data, a review of the techniques developed and results obtained is presented below and two AIAA papers which include detailed explanations and results are included as appendices. These papers include brief reviews of previous rotation and multidimensional upwind research, including references.

RESEARCH RESULTS

Several rotated upwind schemes have been developed for the Euler equations in two dimensions, all of which show significantly improved shock capturing ability over grid aligned schemes to first-order accuracy when the shock wave is oblique to the grid. It is initially unclear which of the schemes offer the best promise for further development. Therefore an initial study is performed where the essential differences between the previous schemes are distilled and then used as building blocks for competing baseline algorithms. This step of the research resulted in four, first-order accurate, rotated upwind schemes that are presented in AIAA Paper 94-0079 "Rotated Upwind Strategies for Solution of the Euler Equations", a copy of which is attached. The four baseline schemes are the result of a two parameter survey of algorithm characteristics. The parameters are the position in the cell at which the rotation takes place (cell-center vs. cell-edge) and the topological space in which the rotation takes place (physical vs. computational). It is found that the cell-center rotation strategies are more robust and accurate than the cell-edge schemes because of a requisite averaging procedure in the cell-center schemes. The averaging procedure acts as an inherent smoothing agent and eliminates odd-even decoupling which is found to be characteristic of the cell-edge schemes. It is also shown that rotating in computational space vs. a rotation in physical space is simpler to implement and has the advantage that it will revert to a grid aligned formulation. It is also shown that the first-order accurate rotated scheme produces results typical of the second-order accurate grid aligned scheme for the solution of a two-dimensional reflecting shock wave duct configuration.

The comparative study noted above revealed that a baseline first-order accurate rotated upwind scheme where the rotation in computational space is performed at each cell-center and designated as CCNO (Cell Center rotation, Non-Orthogonal fluxes) has the best overall characteristics. The scheme was then extended to second-order accuracy and viscous terms were included to model the Navier-Stokes equations. Furthermore, rotated boundary conditions were also developed. The extended algorithm was applied to various shock reflection and shock-boundary layer interaction flowfields. The results are contained in AIAA Paper 94-2291, "Rotated Upwind Algorithms for Solution of Two- and Three-Dimensional Euler and Navier-Stokes Equations" which is attached. The first test case, that of an inviscid shock wave reflection demonstrates that the CCNO algorithm yields a more accurate result than the grid aligned scheme to both first- and second-order accuracy. Furthermore, the improved accuracy is maintained as the grid density is increased. This result is an improvement over previous rotated schemes which show only marginal improvement to second-order accuracy. The next test case is a shock wave impinging on a turbulent boundary layer. To first-order accuracy, the CCNO scheme is shown to produce results in better agreement with the experimental data as compared to the grid aligned scheme. To second-order accuracy, although the inviscid region of the flowfield is qualitatively improved with the CCNO scheme, no improvement in the wall pressure and skin friction distributions are realized. This result is not unexpected, as the mathematical character of the governing equations changes from essentially hyperbolic outside the viscous layer to essentially parabolic near the surface. Therefore despite the improved prediction in the inviscid region, the wall properties are dominated by the viscous terms prediction which are inherently elliptical in nature and thereby do not benefit from upwinding. Although this is a disappointing result in terms of wall predictions, it is

encouraging in terms of interior flow measurements such as mass flow rates and thrust performance which are integrated quantities across the entire domain.

The two-dimensional CCNO algorithm was extended to three dimensions and is also described in the previously cited paper. The key development made during the extension of the algorithm was the derivation of two sequences of rotation that align a coordinate axis in any given preferred direction. Furthermore, the sequences of coordinate rotation are designed to take advantage of directional symmetry such that the number of possible orientations of the rotated coordinate system with respect to the original coordinate system is minimized. Consequently, the logic required in the interpolation and projection procedures is reduced. The result is an automatic method of aligning efficiently a coordinate axis in any given or computed preferred direction.

The three-dimensional algorithm is shown to be sufficiently robust to compute complex flowfields commonly found in supersonic through hypersonic inlet configurations. An inviscid three-dimensional shock surface reflection test case shows that the CCNO algorithm improves three-dimensional shock wave capturing to both first- and second-order accuracy. This is a significant result in light of other recent multi-dimensional algorithms that show only marginal or no improvement to three-dimensional inviscid calculations. However, it is also shown that the accuracy improvements in the three-dimensional calculation are not as great as what was seen in two dimensions. It is believed that this is a result of uncertainties in selecting a true dominant direction which becomes more problematic in three dimensions as compared to two dimensions. Viscous calculations of an intersecting-wedge/corner-flow geometry and a generic hypersonic inlet configuration (the inlet results are contained in Kontinos, D. A. "Rotated Upwind Algorithms for Solution of the Two- and Three-Dimensional Euler and Navier-Stokes Equations," Ph.D. Dissertation, Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, NC, 1994) show that as in the two-dimensional viscous calculations, the inviscid portion of the flowfield, e.g., the shock waves, are qualitatively improved using the CCNO algorithm. However to first-order accuracy, the predictions of pressure and heat transfer on the wall are not improved using the rotated algorithm. To second-order accuracy, there is some improvement in the wall property properties in regions of a dominant cross flow feature. For instance, in the intersecting wedge case better agreement with the experimental data is achieved with the CCNO algorithm in the cross-flow reattachment region of the multiple shock wave structure in the corner of the geometry. Similar improvement is seen in the generic inlet configuration in the region where the boundary layer is rolled into a corner vortex.

The accuracy improvements gained by the CCNO algorithm come at a considerable cost. Both the two- and three-dimensional algorithms are shown to consume 2-4 times more computer time than their grid aligned counterparts. The increase in computer usage is caused by an increase in the work required per iteration and a reduced stability limit which causes an increase in the number of iterations for convergence. However, regions of potential algorithm improvement are highlighted that may reduce the work load, such as a more accurate temporal linearization and a simplification of the interpolation procedure. It is believed that the CCNO scheme may have application for inviscid flow solutions but the scheme is immature for viscous flowfields. In viscous flowfields, selection of a dominant direction becomes uncertain in subsonic regions such as in boundary layers, especially in three dimensions. Moreover, the fine grid spacing typical of viscous flow field meshes restricts and allowable time step and increases the amount of computer usage. As mentioned previously, a more accurate temporal linearization may overcome this difficulty. Finally, current research into truly multi-dimensional Riemann solvers may reveal an optimum upwind direction in the boundary layer thereby refining the CCNO schemes effectiveness.

CONCLUSIONS

The development and evaluation of rotated upwind procedures revealed that both accuracy and practicality must be considered in the choice of such schemes. The rotation scheme that was chosen, CCNO, was the best compromise between accuracy and ease of implementation. This scheme was demonstrated to improve results in essentially inviscid flow regimes in both 2-D and 3-D. The improvement was most dramatic when a first order accurate rotation was used, producing results that compared well with second-order accurate grid aligned solutions. The second order CCNO scheme also gave improved results over second order grid aligned results. However, the improvement was not as dramatic in 3-D and was achieved at considerable computational cost. Further research is indicated to determine how best to find the optimum upwinding direction in 3-D and how to obtain efficiently interpolated flux quantities in the rotated directions. The lack of improvement due to rotation in the viscous layer is forecast by the mathematical character of the governing equations in this region.

RECOMMENDATIONS

CCNO may be used to increase accuracy to match second order grid aligned results with only first order flux interpolation. This may be particularly useful when the grid is distorted to conform to geometry or flow features.

Further research should be conducted to optimize rotation direction and to improve higher order flux interpolation, particularly in 3-D.

AIAA paper 94-0079
32nd Aerospace Sciences Meeting
Reno, NV

Rotated Upwind Strategies for Solution of the Euler Equations

D.A. Kontinos* and D.S. McRae†
North Carolina State University
Raleigh, North Carolina

Abstract

Four finite volume, fully conservative, rotated upwind strategies for solution of the two-dimensional Euler equations are compared. The four strategies are based on the combinations of two options which are a cell-edge vs. a cell-center rotation, and a rotation in physical space vs. computational space. The four schemes are implemented with maximum commonalty for direct comparison. The solutions are relaxed to steady state by the LU-SGS scheme. Solutions of a Mach 2 channel flow problem are presented and compared in terms of accuracy and robustness. It is shown that the cell-edge strategies create unacceptable oscillations in the solution while the cell-center strategies contain inherent smoothing that allow for accurate solutions with good convergence properties. The first-order rotated upwind results are seen to be comparable to standard second-order upwind results.

1.0 Introduction

The development of upwind schemes has been a major breakthrough in the modeling of fluid flow in the transonic through hypersonic regimes. The upwind schemes have provided the ability to capture flow discontinuities within a few grid points without tuning the artificial dissipation. However, it is well known that excess dissipation is generated when the captured discontinuities are oblique to the grid.

One such upwind algorithm, Roe's[1] scheme, has been used quite successfully by a variety of researchers. The Roe scheme models flow discontinuities as a series of linearized waves. In one dimension, these linearized waves model the actual wave propagation quite accurately. However, in multidimensions, the one dimensional Roe scheme misinterprets the multidimensional flow field and incorrectly models local wave propagation. This manifests itself as excess dissipation and results in the smearing of flow discontinuities. An excellent discussion of this effect is presented in Ref. [2].

In an effort to improve upwind schemes in multidimensional flow, much recent effort has been focused on dissipation models that incorporate multidimensional effects. A comprehensive review by van Leer[3] gives a good overview of such methods. One of the approaches is the use of a rotated upwind scheme. The basic idea is to orient the upwind solver in a preferred direction (or 'rotate' the integration stencil to this preferred direction) such that the one dimensional operator can more accurately model the multidimensional flow. Several rotated upwind schemes have been developed for the Euler equations in two dimensions[4,5,6,7,8], all of which show significant improvement over grid aligned schemes to first-order accuracy and modest improvement to second-order accuracy. The main drawback of these algorithms has been the increase in computer work as compared to the grid aligned algorithms. Moreover, to date a three dimensional algorithm has not been developed and it is unclear whether a rotated formulation will show the same improvements in 3D as it does in 2D.

The objective of this study is to compare four two-dimensional rotated upwind strategies in terms of accuracy and robustness so that a method can be selected for development in three dimensions. Each of the strategies contain elements of previously developed algorithms, although several aspects are new. They are implemented with as much commonalty as possible to promote a good comparison.

This study is organized as follows; first Section 2 presents the governing equations. Section 3 presents a general overview of each of the four strategies. The details of the implementation of the strategies are presented in Section 4. Section 5 will present results of a simple channel flow problem and Section 6 will conclude.

2.0 The Euler Equations

For this study, the governing equations are the two-dimensional Euler equations in integral conservation law form. The set in index notation is as follows:

$$\frac{\partial}{\partial t}\int_{Vol}\rho\, dV + \int_{Surf}\rho\, u_i\, n_i\, dS = 0\ , \qquad \text{mass}$$

$$\frac{\partial}{\partial t}\int_{Vol}\rho\, u_j\, dV + \int_{Surf}\left(\rho u_i u_j + p\delta_{ij}\right)n_i\, dS = 0\ , \qquad \text{mom}$$

---

* Research Assistant, Mechanical and Aerospace Engineering, Student Member AIAA.

† Professor, Mechanical and Aerospace Engineering, Member AIAA.

$$\frac{\partial}{\partial t}\int_{Vol}\rho E_t\, dV + \int_{Surf}(E_t + p)u_i n_i\, dS = 0 \quad . \qquad \text{energy}$$

In the finite volume framework, the conserved variables are taken as averages over the cell volume. Thus, the conserved quantities can be taken out of the volume integrals in the previous equations. Moreover, the surface integrals become summations over the cell sides. Expanding out the momentum equation into its vector components, the equations can be written as,

$$\frac{\partial U}{\partial t} = -\frac{1}{Vol}\sum_{sides}\vec{F}\bullet\vec{n}_s \quad , \tag{2.1}$$

where the conserved variables are,

$$U = \left[\rho \quad \rho u \quad \rho v \quad E_t\right]^T, \tag{2.2}$$

and the fluxes are given by,

$$\vec{F} = \left[\rho\vec{V} \quad \rho u\vec{V} + p\hat{i} \quad \rho v\vec{V} + p\hat{j} \quad (E_t + p)\vec{V}\right]^T, \tag{2.3}$$

where the cell face normal with magnitude equal to the cell face length is denoted by $\vec{n}_s$ and the velocity by $\vec{V}$. The variables p, ρ, u, and v are the pressure, density, and x and y Cartesian velocity components, respectively. Using the perfect gas relation, the total energy E_t is given by,

$$E_t = \frac{p}{(\gamma-1)} + \frac{\rho\left(u^2+v^2\right)}{2} \quad , \tag{2.4}$$

where γ is the ratio of specific heats which has a value of 1.4 for air.

3.0 Rotated Upwind Strategies

This section presents an overview of the four strategies compared in this study. Recall that the main goal of a rotated upwind scheme is to align the one dimensional Riemann solver across flow discontinuities to reduce excess dissipation. In order to achieve this goal, one must compute the flux in a direction independent of the grid. This section will describe four strategies for such a flux computation. Each strategy will base the flux computation at the cell edge or cell center, and compute the preferred direction in physical space or computational space. This section describes the strategies in broad terms while Section 4 presents the details.

3.1 Cell-Edge Orthogonal Flux Strategy (CEO)

The first strategy is a distillation of the scheme developed by Levy, et. al.[5]. Levy's scheme is a cell-edge method wherein a preferred upwind direction is selected at each cell face and two orthogonal fluxes are computed; an upwind flux in the preferred direction, and a central difference flux normal to the preferred direction. This is shown schematically in Fig. 1. Later versions of the algorithm apply Roe's scheme in both directions[9]. The values of the dependent variables are interpolated between cell centers through a linear-quadratic interpolating function. The rotated fluxes are then projected onto the grid face direction to ensure conservation. Levy shows improved shock capturing over grid aligned schemes.

The same strategy is taken by Dadone and Grossman[6]. However, a different tactic is used to compute the numerical flux. An upwind flux (Roe's scheme) is computed in both the primary (preferred) and secondary (orthogonal) direction. Also a simplified interpolation stencil is incorporated that essentially 'grabs' the values of the characteristic variables at the nearest cell center to the rotated stencil. Computations using this scheme also show improved results as compared to grid aligned schemes. Moreover, convergence rate and boundary condition improvements are shown as compared to previous rotated schemes.

The cell-edge, orthogonal flux philosophy is incorporated in this study and is termed CEO. An upwind flux is computed in both the preferred and secondary direction at each cell edge. The interpolation of the primitive variables is linear between adjoining cells.

3.2 Cell-Center Orthogonal Flux Strategy (CCO)

As an alternative to the previous cell-edge strategy, the concept of a cell-center rotation is now proposed (the notion of assigning a rotation angle to a cell center is recognized by Davis[4] although a method is not developed). This second strategy, termed cell-center orthogonal or CCO, is derived from the previous rotated finite difference scheme of Kontinos and McRae[7]. In this previous work, a rotation angle is selected in computational space thereby defining a rotated coordinate system. The flux divergence is then computed in the rotated frame. The method shows improved results over grid aligned schemes for both inviscid and viscous flows. Unfortunately, the grid point residuals are computed independently of each other and the scheme is not conservative as originally implemented.

In order to guarantee conservation, the scheme is converted from finite difference to finite volume to become the CCO strategy. In the CCO scheme, four fluxes are computed in two orthogonal directions originating at the cell center (recall in the cell-edge philosophy, two orthogonal fluxes are computed at each cell face and then projected onto that cell face). These four fluxes are then projected around the cell edges, details of which are presented in the next section. The CCO strategy is represented in Fig. 2 where a single cell is shown schematically along with the rotated fluxes. Because each

cell is rotated independently, non-unique fluxes at the cell edges are created. More precisely, since each cell edge is adjacent to two cell centers, two values of the flux are computed. For example the flux at face $i+1/2$ is computed from the rotation at cell i and $i+1$ (see Fig. 2). To uniquely define the flux, the projected fluxes from each of the adjacent cell centers are averaged.

A consequence of the CCO scheme is that for arbitrary grids, the scheme will not revert to a grid aligned formulation. From Fig. 2 one can see that a set of orthogonal axes will not align with a cell whose sides are not orthogonal.

An important comparison between the CEO and CCO strategies can now be made. In two dimensions, the two schemes require about the same amount of arithmetic operations (work). Let W be the amount of work for n points. The work is mostly composed of the interpolation and the flux computation, and is given by,

$$W_{CE} = \left(\frac{2 \text{ fluxes } + \text{ 4 interp}}{\text{face}}\right) \times 2 \frac{\text{faces}}{\text{point}} \times n \text{ points}$$
$$= 4n \text{ fluxes } + \text{ } 8n \text{ interp} ,$$

$$W_{CC} = \left(\frac{4 \text{ fluxes } + \text{ 4 interp}}{\text{face}}\right) \times n \text{ points}$$
$$= 4n \text{ fluxes } + \text{ } 4n \text{ interp}.$$

However, an advantage of the cell-center scheme over the cell-edge scheme is predicted in three dimensions for an unrestricted rotation. In three dimensions, the cell-edge strategy requires three fluxes computed at each cell face. The cell-center strategy requires six fluxes at each cell center. The work is now given by,

$$W_{CE} = \left(\frac{3 \text{ fluxes } + \text{ 6 interp}}{\text{face}}\right) \times 3 \frac{\text{faces}}{\text{point}} \times n \text{ points}$$
$$= 9n \text{ fluxes } + \text{ } 18n \text{ interp} ,$$

$$W_{CC} = \left(\frac{6 \text{ fluxes } + \text{ 6 interp}}{\text{face}}\right) \times n \text{ points}$$
$$= 6n \text{ fluxes } + \text{ } 6n \text{ interp}$$

So it is seen that the cell-center strategy potentially reduces the work load by at least 1/3.

3.3 Cell-Center Non-Orthogonal Flux Strategy (CCNO)

The desire to have the CCO scheme revert to a grid aligned mode leads to the third method. This again is derived from the previous work of Kontinos and McRae[7]. In this method, the cell center rotation is performed in computational space where for rotation angles of $n\pi/2$, the scheme automatically aligns the difference stencil in the grid contravariant directions. The rotated stencil is shown in Fig. 3a where a set of orthogonal axes are rotated through an angle θ in computational space. Mapped into physical space, the fluxes are computed in **non-orthogonal** directions as shown in Fig. 3b. Thus this strategy is termed cell-center non-orthogonal or CCNO. These fluxes are projected around the cell as in the previous CCO strategy. However, in this case since the fluxes are not orthogonal, the projection must be performed by a coordinate transformation which is described in Section 4.

3.4 Cell-Edge Non-Orthogonal Flux Strategy (CENO)

The ideas of the previous strategy are coupled with the cell-edge method to lead to the cell-edge non-orthogonal (CENO) strategy. As in the CCNO technique, orthogonal axes are rotated in computational space to a preferred direction with the cell edge acting as the origin. This results in two non-orthogonal fluxes being computed in physical space that are projected onto the grid face. The method is shown schematically in Fig. 4.

4.0 Implementation

This section provides the details of the implementation of each of the four strategies. They are implemented with as much commonalty as possible. First, all fluxes are computed using Roe's scheme. Second, linear interpolation of the primitive variables between cell centers is used to obtain the state values for the flux computation. The interpolated values are used to compute both the inviscid flux and the upwind damping term (Riemann solution). It is possible to use higher order interpolations such as a one-dimensional quadratic function, or two-dimensional bilinear, reduced biquadratic, or biquadratic functions. But, most of these functions admit overshoots or undershoots in the interpolating function. Thus, the simplest interpolation is used to compare the four strategies. The order of accuracy is determined by the interpolation stencil which for this study is first-order accurate. Third, the rotation angles are computed in an analogous fashion between strategies. Each cell edge or center is assigned its own rotation angle which is not averaged with surrounding angles. Finally, each strategy is relaxed to steady state using the LU-SGS scheme of Yoon and Jameson[10].

Sections 4.2-4.5 presents the details for the three steps peculiar to the rotated flux computation: the rotation angle calculation, the interpolation, and the flux projection.

4.1 Generalized Projection

In order to guarantee conservation, each of the four methods require the rotated fluxes to be projected back onto the grid faces. Thus, a generalized projection is presented. The development in this section is taken from Ref. [11]. Consider a field point as in Fig. 5. Five vectors are shown: $\bar{n}_\xi$ is a vector in the direction of the cell face

normal with magnitude equal to the cell face length; $\vec{n}^1$ and $\vec{n}^2$ are two arbitrary non-collinear vectors in the direction the flux is to be computed and are designated as the contravariant directions; and $\vec{n}_1$ and $\vec{n}_2$ are the covariant base vectors. The covariant vectors are related to the contravariant vectors by,

$$\vec{n}_1 = \sqrt{g}\left(\vec{n}^2 \times \hat{k}\right), \quad \vec{n}_2 = \sqrt{g}\left(\hat{k} \times \vec{n}^1\right), \tag{4.1}$$

where,

$$\frac{1}{\sqrt{g}} = \left(\vec{n}^1 \times \vec{n}^2\right) \bullet \hat{k} \ . \tag{4.2}$$

The flux vector expressed in the covariant vector base is given by,

$$\vec{F} = \left(\vec{F} \bullet \vec{n}^1\right)\vec{n}_1 + \left(\vec{F} \bullet \vec{n}^2\right)\vec{n}_2 \ . \tag{4.3}$$

The flux vector projected onto the cell face (see Eq. (2.1)) is given by,

$$\vec{F} \bullet \vec{n}_g = \left(\vec{F} \bullet \vec{n}^1\right)\vec{n}_1 \bullet \vec{n}_g + \left(\vec{F} \bullet \vec{n}^2\right)\vec{n}_2 \bullet \vec{n}_g . \tag{4.4}$$

4.2 Cell-Edge Orthogonal Flux Implementation

The details of the computation of the rotation angle, interpolation, and projection will be presented for an *i+1/2* face in the CEO strategy. An analogous computation is performed for the *j+1/2* faces and will be omitted for brevity.

The rotation angle is based on a flow field gradient. Let a flow field quantity be represented by q. The local gradient is computed as,

$$\begin{aligned} q_{\text{right}} &= \frac{1}{4}\left(q_{i+1,j-1} + 2q_{i+1,j} + q_{i+1,j+1}\right), \\ q_{\text{left}} &= \frac{1}{4}\left(q_{i,j-1} + 2q_{i,j} + q_{i,j+1}\right), \\ q_{\text{up}} &= \frac{1}{2}\left(q_{i,j+1} + q_{i+1,j+1}\right), \\ q_{\text{down}} &= \frac{1}{2}\left(q_{i,j-1} + q_{i+1,j-1}\right), \end{aligned} \tag{4.5}$$

and,

$$\begin{aligned} \frac{\partial q}{\partial \xi} &= q_{\text{right}} - q_{\text{left}}, \\ \frac{\partial q}{\partial \eta} &= \frac{1}{2}\left(q_{\text{up}} - q_{\text{down}}\right). \end{aligned} \tag{4.6}$$

Then locally at face *i+1/2* the metrics are given as,

$$\overline{\xi}_x = \xi_{x_{i+1/2,j}}, \quad \overline{\xi}_y = \xi_{y_{i+1/2,j}}, \tag{4.7}$$

$$\begin{aligned} \overline{\eta}_x &= \frac{1}{4}\left(\eta_{x_{i,j+1/2}} + \eta_{x_{i+1,j+1/2}} + \eta_{x_{i,j-1/2}} + \eta_{x_{i+1,j-1/2}}\right), \\ \overline{\eta}_y &= \frac{1}{4}\left(\eta_{y_{i,j+1/2}} + \eta_{y_{i+1,j+1/2}} + \eta_{y_{i,j-1/2}} + \eta_{y_{i+1,j-1/2}}\right), \end{aligned}$$

and the gradient is,

$$\begin{aligned} \frac{\partial q}{\partial x} &= \overline{\xi}_x \frac{\partial q}{\partial \xi} + \overline{\eta}_x \frac{\partial q}{\partial \eta}, \\ \frac{\partial q}{\partial y} &= \overline{\xi}_y \frac{\partial q}{\partial \xi} + \overline{\eta}_y \frac{\partial q}{\partial \eta}. \end{aligned} \tag{4.8}$$

A rotation direction can be computed directly from the gradient through the inverse tangent function. However, in order to eliminate oscillations of the rotation angle in the freestream, the following blending between the computed gradient direction θ_q and the grid face normal θ_g is incorporated,

$$\|\nabla q\| = \sqrt{\left(\frac{\partial q}{\partial x}\right)^2 + \left(\frac{\partial q}{\partial y}\right)^2}, \tag{4.9}$$

$$\omega = e^{\left(\frac{-DMIN}{\|\nabla q\|}\right)}. \tag{4.10}$$

Then the cosine of the rotation angle θ is given as,

$$\cos(\theta) = \omega \cos\left(\theta_q\right) + (1-\omega)\cos\left(\theta_g\right). \tag{4.11}$$

The quantity ω acts as a blending coefficient controlled by the input parameter *DMIN*.

The interpolation of the primitive variables is performed by mapping the local stencil into (ξ,η) space where $-0.5 \le \xi \le 0.5$ and $-1 \le \eta \le 1$. This is shown schematically in Fig. 6. The mapping is based on the averaged metrics of Eq. (4.7). Although holding the metrics of the transformation constant over the interpolation stencil is not strictly accurate in general, it is adequate for grids where the metrics do not change drastically. From Fig. 6 it is seen that four (ξ,η) pairs are needed to compute the interpolation. A point on the interpolating ray is given by,

$$\begin{aligned} \xi' &= \xi_0 + \overline{\xi}_x dx + \overline{\xi}_y dy, \\ \eta' &= \eta_0 + \overline{\eta}_x dx + \overline{\eta}_y dy. \end{aligned} \tag{4.12}$$

Setting the origin to (0,0) and using the rotation angle, Eq. (4.12) becomes,

$$\begin{aligned} \xi' &= \overline{\xi}_x \cos(\theta) + \overline{\xi}_y \sin(\theta), \\ \eta' &= \overline{\eta}_x \cos(\theta) + \overline{\eta}_y \sin(\theta). \end{aligned} \tag{4.13}$$

The value of (ξ',η') must be clipped to lie in the interpolating region. One option would be to have (ξ,η) lie on the ellipse drawn in Fig. 6 and use a bilinear, or linear-

quadratic interpolating function over the local two-dimensional region. The option used in this study is to clip the interpolating ray to lie on the rectangular stencil of the surrounding cell centers and use linear interpolation. The clipping is given by,

$$SW = \begin{cases} 1 & \text{for } 2|\xi'| \geq |\eta'| \\ 0 & \text{for } 2|\xi'| < |\eta'| \end{cases}, \tag{4.14}$$

$$\xi = \text{sign}(\xi')\left(\frac{SW}{2} + (1-SW)\frac{|\xi'|}{(|\eta'| + SW)}\right), \tag{4.15}$$

$$\eta = \text{sign}(\eta')\left(1 - SW + \frac{SW}{2}\frac{|\eta'|}{(|\xi'| + 1 - SW)}\right).$$

The quantity SW is a switching function and is included in the denominators of Eq. (4.15) to avoid a divide by zero. The interpolant value denoted by $\tilde{q}$ is then given by,

$$\tilde{q} = c_0 + \alpha\,\eta\left(-c_0 + c_1 q_{i+1,j1} + c_2 q_{i,j1}\right), \tag{4.16}$$

where,

$$c_1 = \tfrac{1}{2} + \xi, \quad c_2 = \tfrac{1}{2} - \xi, \quad c_0 = c_1 q_{i+1,j} + c_2 q_{i,j}, \tag{4.17}$$

$$\alpha = \begin{cases} 1 & \text{for } \eta \geq 0 \\ -1 & \text{for } \eta < 0 \end{cases}, \tag{4.18}$$

$$j1 = j + \alpha. \tag{4.19}$$

This interpolation function is used for all four (ξ,η) pairs of Fig. 6. The second (ξ,η) pair is given by,

$$\begin{aligned} \xi' &= -\overline{\xi}_x \sin(\theta) + \overline{\xi}_y \cos(\theta), \\ \eta' &= -\overline{\eta}_x \sin(\theta) + \overline{\eta}_y \cos(\theta), \end{aligned} \tag{4.20}$$

then the clipping function of Eq. (4.15) is used. The final two pairs are simply given by $(\xi,\eta)_3 = (-\xi,-\eta)_1$ and $(\xi,\eta)_4 = (-\xi,-\eta)_2$. The interpolated values resulting from the pairs $(\xi,\eta)_1$ and $(\xi,\eta)_3$ are used as left and right states for the computation of one flux and $(\xi,\eta)_2$ and $(\xi,\eta)_4$ are used for the other flux.

The projection stage is computed by assigning the vectors of Eq. (4.4) as,

$$\begin{aligned} \vec{n}^1 &= \cos(\theta)\hat{i} + \sin(\theta)\hat{j}\ , \\ \vec{n}^2 &= -\sin(\theta)\hat{i} + \cos(\theta)\hat{j}\ . \end{aligned} \tag{4.21}$$

In this case, the Jacobian of the transformation is unity and the set of contravariant and covariant vectors are identical. Equation (4.4) becomes,

$$\vec{F} \bullet \vec{n}_g = \left(\vec{F} \bullet \vec{n}_1\right)\cos\left(\theta - \theta_g\right) - \left(\vec{F} \bullet \vec{n}_2\right)\sin\left(\theta - \theta_g\right). \tag{4.22}$$

Although the scheme just developed is based on the work of Levy, it is not to be confused with the algorithms of Ref. [5,6]. That work incorporates different interpolation methods, rotation angle selections, and integration schemes. Moreover, because of the current interpolation scheme, the cell edge formulation will not revert to a grid aligned scheme for arbitrary grids. The current formulation is used solely for simplicity and is probably not the optimal combination for a robust algorithm. These 'ingredients' of the recipe are selected so that maximum commonalty between the strategies is achieved.

4.3 Cell-Center Orthogonal Flux Implementation

The implementation of the CCO strategy follows the same general outline as the previous CEO strategy. However, the rotation is performed at the cell center. First, the gradient at point (i,j) is computed by the following stencil,

$$\begin{aligned} q_{\text{right}} &= \frac{1}{4}\left(q_{i+1,j-1} + 2q_{i+1,j} + q_{i+1,j+1}\right), \\ q_{\text{left}} &= \frac{1}{4}\left(q_{i-1,j-1} + 2q_{i-1,j} + q_{i-1,j+1}\right), \\ q_{\text{up}} &= \frac{1}{4}\left(q_{i-1,j+1} + 2q_{i,j+1} + q_{i+1,j+1}\right), \\ q_{\text{down}} &= \frac{1}{4}\left(q_{i-1,j-1} + 2q_{i,j-1} + q_{i+1,j-1}\right). \end{aligned} \tag{4.23}$$

The derivative in computational space is given by,

$$\begin{aligned} \frac{\partial q}{\partial \xi} &= q_{\text{right}} - q_{\text{left}}, \\ \frac{\partial q}{\partial \eta} &= q_{\text{up}} - q_{\text{down}}. \end{aligned} \tag{4.24}$$

The gradient in physical space is given by Eq. (4.8). The metrics are taken as cell averages. The rotation angle is computed using the same exponential blending function of Eqs. (4.9-4.11). However, a particular grid face must be selected as the default direction (θ_g) which for this work, is the $(i+1/2)$ face.

Four interpolation points are required for the flux computation. The fluxes are computed with the cell center as one state and the appropriate interpolated point as the other state. The interpolation is computed using the same mapping methodology given in the previous section. In this case, however, the mapping is onto (ξ,η) space where $-1 \leq \xi \leq 1$ and $-1 \leq \eta \leq 1$.

Before the fluxes can be projected, the algorithm must properly label each of the four fluxes. This sorting is required because the preferred direction may be oriented in any manner with respect to the computational coordinates (i.e. grid indices). So to project the "correct" fluxes onto the cell faces, the fluxes must be ordered. Flux sorting is accomplished by denoting the first flux counter-clockwise from cell normal ($i+1/2$) as the $\vec{n}^1$ direction (see Fig. 2). The algorithm selects the proper direction by the following criteria,

$$\vec{n}^1 \bullet \vec{n}_{i+1/2,j} > 0, \quad \vec{n}^1 \times \vec{n}_{i+1/2,j} > 0. \tag{4.25}$$

After the $\vec{n}^1$ direction is defined, the other fluxes are incremented in the counter-clockwise direction.

The last stage of the flux computation is the projection of the fluxes onto the grid faces. In the CEO method, the task is straightforward since the rotated fluxes are computed at each cell face and projected onto that face. However for the CCO method, the set of four fluxes must be projected around the entire cell. Such a projection is not unique so two methods that are implemented will be outlined.

In the first method, an orthogonal pair of fluxes is assigned to each cell edge. With the notation $\vec{F}_k = \left(\vec{F} \bullet \vec{n}^k\right)\hat{n}^k$, ($k$=1,4), the fluxes at the cell edge are given as (see Fig. 2),

$$\begin{aligned}
\vec{F}_{i+1/2,j} &= \left(\vec{F}_1 + \vec{F}_4\right) \bullet \vec{n}_{i+1/2,j}, \\
\vec{F}_{i,j+1/2} &= \left(\vec{F}_2 + \vec{F}_1\right) \bullet \vec{n}_{i,j+1/2}, \\
\vec{F}_{i-1/2,j} &= \left(\vec{F}_3 + \vec{F}_2\right) \bullet \vec{n}_{i-1/2,j}, \\
\vec{F}_{i,j-1/2} &= \left(\vec{F}_4 + \vec{F}_3\right) \bullet \vec{n}_{i,j-1/2},
\end{aligned} \tag{4.26}$$

Recall that the fluxes are orthogonal so that the Jacobian is unity and the contravariant and covariant vectors are identical.

The second method is a higher order projection (designated by HOP) where the flux across the grid face is not considered constant. In this case, the projected fluxes are weighted according to the local geometry. For example, consider the cell of Fig. 7. In computing the flux for face ($i+1/2,j$), the previous projection method (Eq. (4.26)) selects the flux pair $\vec{F}_1$ and $\vec{F}_4$. The higher order projection divides the ($i+1/2,j$) face into two regions with the $\vec{n}^1$ ray acting as the demarcation line. Then $\vec{F}_4$ is projected onto the region below $\vec{n}^1$ and $\vec{F}_2$ is projected onto the upper region. The other component of the flux is obtained from the projection of $\vec{F}_1$ over the region above the $\vec{n}^4$ ray and $\vec{F}_3$ below the ray. Another way to view this is that each flux acts on the cell faces or parts of cell faces in its half plane. The face fluxes are then given by,

$$\begin{aligned}
\vec{F}_{i+1/2,j} &= \left(a_1\vec{F}_1 + (1-a_1)\vec{F}_3 + a_2\vec{F}_2 + (1-a_2)\vec{F}_4\right) \bullet \vec{n}_{i+1/2,j}, \\
\vec{F}_{i,j+1/2} &= \left(b_1\vec{F}_1 + (1-b_1)\vec{F}_3 + b_2\vec{F}_2 + (1-b_2)\vec{F}_4\right) \bullet \vec{n}_{i,j+1/2}, \\
\vec{F}_{i-1/2,j} &= \left((1-a_1)\vec{F}_1 + a_1\vec{F}_3 + (1-a_2)\vec{F}_2 + a_2\vec{F}_4\right) \bullet \vec{n}_{i-1/2,j} \\
\vec{F}_{i,j-1/2} &= \left((1-b_1)\vec{F}_1 + b_1\vec{F}_3 + (1-b_2)\vec{F}_2 + b_2\vec{F}_4\right) \bullet \vec{n}_{i,j-1/2}
\end{aligned} \tag{4.27}$$

where a_1, a_2, b_1, and b_2 are the weight coefficients. When computing the weight coefficients, the cell is taken to be a parallelogram because averaging the metrics over the cell, as is done for the interpolation, in effect reduces the actual cell to a parallelogram. Or in other words, the approximation in considering the cell to be a parallelogram in computing the weight coefficients is consistent with previous approximations. The projection is done over the actual cell to maintain conservation (in Eq. 4.27, the length scale is contained in the cell edge normal). The weight coefficients are computed by the following algorithm: first the coordinates of the dividing axes are given by,

$$\begin{aligned}
\xi_1 &= \overline{\xi}_x \cos(\theta) + \overline{\xi}_y \sin(\theta), \\
\eta_1 &= \overline{\eta}_x \cos(\theta) + \overline{\eta}_y \sin(\theta),
\end{aligned} \tag{4.28}$$

$$\begin{aligned}
\xi_2 &= -\overline{\xi}_x \sin(\theta) + \overline{\xi}_y \cos(\theta), \\
\eta_2 &= -\overline{\eta}_x \sin(\theta) + \overline{\eta}_y \cos(\theta).
\end{aligned} \tag{4.29}$$

Then, the switch functions are computed,

$$r_k = \left\{ \frac{\min(\xi_k, \eta_k)}{\max(\xi_k, \eta_k)} \quad (k = 1,2), \right. \tag{4.30}$$

$$P_k = \begin{cases} 1 \text{ for } |\xi_k| \geq |\eta_k| \\ 0 \text{ for } |\xi_k| < |\eta_k| \end{cases} \quad (k = 1,2), \tag{4.31}$$

$$S_1 = \begin{cases} 1 \text{ for } \eta_1 \geq 0 \\ -1 \text{ for } \eta_1 < 0 \end{cases}, \tag{4.32}$$

$$S_2 = \begin{cases} 1 \text{ for } \xi_2 \geq 0 \\ -1 \text{ for } \xi_2 < 0 \end{cases}. \tag{4.33}$$

Finally, the weights are computed by,

$$\begin{aligned}
a_1 &= 1 - P_2 + \tfrac{1}{2} P_2 (1 + r_2), \\
a_2 &= \tfrac{1}{2}\left((1-P_1)(1-S_1) + P_1(1 - S_1 r_1)\right), \\
b_1 &= \tfrac{1}{2}\left(P_1(1-S_2) + (1-P_1)(1-S_2 r_2)\right), \\
b_2 &= P_2 + \tfrac{1}{2}(1-P_2)(1+r_1).
\end{aligned} \tag{4.34}$$

Equations (4.28-4.34) may appear complex but are necessary to compute the weight coefficients without using

IF-THEN logic which inhibits vectorization and nearly doubles the amount of required computer time.

4.4 Cell-Center Non-Orthogonal Flux Implementation

The interpolation and logic required in the CCNO strategy is greatly simplified over CCO because everything is performed in computational space where the orientation is always the same. Also, since upwinding is performed in each direction, a full range of rotation is achieved for $0° \le \theta \le 90°$.

The gradient is computed in computational space using Eqs. (4.23-4.24). The rotation angle is given as,

$$\cos(\theta) = -\text{sign}\left(\frac{\partial q}{\partial \xi}\right)\frac{\frac{\partial q}{\partial \eta}}{\max\left(\|\nabla q\|,\ DMIN\right)}, \qquad (4.35)$$

where $\|\nabla q\|$ is the magnitude of the gradient in computational space. Equation (4.35) is taken from Ref. [7]. Also it should be noted that the values of *DMIN* for the blending function of Eq. (4.35) are quite different from the values of the exponential blending of Eq. (4.10) although the effect is nearly the same. Also, the values of θ given from Eq. (4.35) are reflected into the first quadrant.

The interpolation for the CCNO strategy is much less complex than the previous two strategies because the topology in computational space is always consistent. In the CEO and CCO strategies, logic must be written to determine where to interpolate in physical space (Eqs. 4.12-4.20). For instance, depending on the skewness of a cell, a single interpolant value can lie in one of three possible quadrants in computational space. Conversely, in the CCNO method, a single interpolant value is restricted to only one quadrant. Therefore, the indices of the interpolation support stencil can be hardwired into the code.

The main difference in CCNO as compared to CCO is the flux projection. Recall that the fluxes are orthogonal in computational space but are non-orthogonal in physical space. Consequently, the projection of Eq. (4.4) does not reduce to simple trigonometric identities as in Eq. (4.22) for an arbitrary cell shape. However, this formulation reverts to a grid aligned scheme for arbitrary grids. First the following vectors are defined,

$$\begin{aligned}
\vec{n}^1 &= \left(\xi_{x_{i+1/2,j}}\cos(\theta) + \eta_{x_{i,j+1/2}}\sin(\theta)\right)\hat{i} \\
&\quad + \left(\xi_{y_{i+1/2,j}}\cos(\theta) + \eta_{y_{i,j+1/2}}\sin(\theta)\right)\hat{j} \\
\vec{n}^2 &= \left(\eta_{x_{i,j+1/2}}\cos(\theta) - \xi_{x_{i-1/2,j}}\sin(\theta)\right)\hat{i} \\
&\quad + \left(\eta_{y_{i,j+1/2}}\cos(\theta) - \xi_{y_{i-1/2,j}}\sin(\theta)\right)\hat{j} \qquad (4.36)
\end{aligned}$$

$$\begin{aligned}
\vec{n}^3 &= \left(\xi_{x_{i-1/2,j}}\cos(\theta) + \eta_{x_{i,j-1/2}}\sin(\theta)\right)\hat{i} \\
&\quad + \left(\xi_{y_{i-1/2,j}}\cos(\theta) + \eta_{y_{i,j-1/2}}\sin(\theta)\right)\hat{j} \\
\vec{n}^4 &= \left(\eta_{x_{i,j-1/2}}\cos(\theta) - \xi_{x_{i+1/2,j}}\sin(\theta)\right)\hat{i} \\
&\quad + \left(\eta_{y_{i,j-1/2}}\cos(\theta) - \xi_{y_{i+1/2,j}}\sin(\theta)\right)\hat{j}
\end{aligned}$$

Pairs of these vectors will be selected as the contravariant directions to be used in the projection for the cell faces. As before in the CCO strategy, two methods of projection are employed. The first assigns a flux pair to each face. The assignment follows the same schedule of Eq. (4.26) although now the full transformation must be computed. The second is the higher order projection where the rotated flux is considered to act on the regions of the faces in its half plane. The weight coefficients of the higher order formulation are much simpler to compute in CCNO because of the symmetry in the computational plane. First, let $\{i,j\}$ denote the pair of fluxes in the $\vec{n}^i$ and $\vec{n}^j$ directions which will be used for the projection of Eq. (4.4). The $\{i,j\}$ pair establishes the vector base for the projection. Further, let $\{i,j\}\bullet\vec{n}_g$ denote this projection. Then the cell face fluxes are given by,

$$\begin{aligned}
\bar{F}_{i+1/2,j} &= \left[a_2\left(SW\{1,2\} + (1-SW)\{3,4\}\right)\right. \\
&\quad \left. + a_1\{1,4\}\right]\bullet\vec{n}_{i+1/2,j}\ , \qquad (4.37)\\
\bar{F}_{i,j+1/2} &= \left[a_2\left(SW\{2,3\} + (1-SW)\{1,4\}\right)\right. \\
&\quad \left. + a_1\{1,2\}\right]\bullet\vec{n}_{i,j+1/2}\ , \\
\bar{F}_{i-1/2,j} &= \left[a_2\left(SW\{3,4\} + (1-SW)\{1,2\}\right)\right. \\
&\quad \left. + a_1\{2,3\}\right]\bullet\vec{n}_{i-1/2,j}\ , \\
\bar{F}_{i,j-1/2} &= \left[a_2\left(SW\{1,4\} + (1-SW)\{2,3\}\right)\right. \\
&\quad \left. + a_1\{3,4\}\right]\bullet\vec{n}_{i,j-1/2}\ ,
\end{aligned}$$

where,

$$SW = \begin{cases} 1 & \text{for } \xi \ge \eta \\ 0 & \text{for } \xi < \eta \end{cases} \qquad (4.38)$$

$$\alpha = \min(\xi, \eta) \qquad (4.39)$$

$$a_1 = \tfrac{1}{2}(1+\alpha)\ ,\quad a_2 = \tfrac{1}{2}(1-\alpha)\ . \qquad (4.40)$$

This completes the implementation of the CCNO method.

4.5 Cell-Edge Non-Orthogonal Flux Implementation

The CENO strategy is directly analogous to the previous CCNO method. The rotation angle calculation and interpolation are performed in computational space defined

by the average metrics of Eq. (4.7). The rotation angle blending is given by Eq. (4.35) and the interpolating rays are given by Eq. (4.12) and Eq. (4.20). The projection onto the grid face is based on the vectors,

$$\vec{n}^1 = \left(\overline{\xi}_x \cos(\theta) + \overline{\eta}_x \sin(\theta)\right)\hat{i} + \left(\overline{\xi}_y \cos(\theta) + \overline{\eta}_y \sin(\theta)\right)\hat{j}\,, \tag{4.41}$$

$$\vec{n}^2 = \left(\overline{\eta}_x \cos(\theta) - \overline{\xi}_x \sin(\theta)\right)\hat{i} + \left(\overline{\eta}_y \cos(\theta) - \overline{\xi}_y \sin(\theta)\right)\hat{j}\,.$$

The projection in the CENO method reduces to a simple trigonometric relation because the local metrics are constant. For example, for a $(i+1/2)$ face the projection is,

$$\vec{F}_{i+1/2,j} = \vec{F}_1 \cos(\theta) - \vec{F}_2 \sin(\theta)\ . \tag{4.42}$$

The CENO formulation also suggests the possibility of selecting a preferred direction for the secondary flux as well as the primary flux (as opposed to setting a condition of orthogonality in physical or computational space). For instance, $\vec{n}^1$ could be arranged to upwind across shock waves while $\vec{n}^2$ is triggered to upwind across a shear wave. Such a scheme would require a full transformation of the fluxes and Eq. (4.42) would not apply.

This completes the implementation of the four strategies. The next two subsections will briefly describe the boundary conditions and integration.

4.6 Boundary Conditions

For a supersonic inviscid calculation, three types of boundary conditions are employed. The first is supersonic inflow where the flux is set to freestream on the inflow plane. The second is supersonic outflow where the variables are extrapolated to zeroth-order accuracy. The third is a nonporous wall condition where properties are reflected about the slip wall.

In the cell center strategy, the fluxes on the four sides of the cell are coupled to the rotation angle and hence to each other. Consequently, it is unclear how to define the rotation angle on the first row of cells next to the wall. So for this study, the fluxes on and near the wall are computed in a grid aligned fashion. More precisely, the nonporous wall fluxes are computed based on the boundary condition. The flux on the faces normal and immediately adjacent to the wall are computed using a grid aligned formulation. All other fluxes are computed using the rotated flux functions of each particular strategy. This includes the outflow fluxes where a column of ghost cells is created with the flow quantities extrapolated to zeroth-order from the interior of the domain. This additional column of cells provides support for the interpolation stencil at the last column of interior cells.

4.7 Integration Method

All four methods will be integrated in the same fashion. Thus, the only difference will be in the calculation of the cell residuals (RHS). The solution will be relaxed in time by the LU-SGS scheme of Yoon and Jameson[10]. Using the LU-SGS scheme with a rotated RHS is thought to be the best method to relax the solution to steady-state because of the following reasons: An explicit update scheme such as a multistep Lax-Wendroff type or Runge-Kutta requires multiple calls to the flux computation which should be avoided since the rotation requires extra work. If one were to seek an implicit solution, the temporal linearization of the fluxes would require some spatial approximation based on the local rotation angle since the fluxes are functions of interpolated quantities. This would expand the implicit stencil to a nonadiagonal block matrix for a two dimensional solution. With such a large banded matrix, an approximate factorization or Gauss-Seidel inversion becomes prohibitively expensive. Thus, the LU-SGS scheme is chosen. It is hoped that the diagonal dominance of the system, upon which the LU-SGS scheme depends, is sufficient to mask the incompatibility of the LHS to the RHS.

5.0 Results

The four rotation strategies are tested by computing the flow in a channel with a 15° ramp. The test case is taken from Ref. [5] where a more complete description is given. The 61x21 grid is shown in Fig. 8. The inflow Mach number is 2 and the flow is from left to right. The test case is first computed to first-order accuracy using a grid aligned algorithm. Mach number contours are shown in Fig. 9. The range of the contours for this case and most subsequent cases are from 1.3 to 2.0 in increments of 0.05. The first-order grid aligned result shows a smeared initial shock wave and severely smeared reflected shock waves.

The grid aligned algorithm is then extended to second-order accuracy by MUSCL extrapolation of the primitive variables. The MINMOD limiter is employed to damp oscillations around the shock wave. The resulting Mach contours are shown in Fig. 10. The second-order result shows the shock waves to be much more clearly defined.

The next set of results are solutions of the four strategies where the rotation angle θ is set to 45°. For this channel flow problem, 45° nearly aligns the flux function across all the shock and expansion waves. Thus it presents a good initial test case for the rotated schemes without introducing any complications from a rotation angle calculation. However, a priori knowledge of the rotation angles will not be known in general so the calculation of a preferred direction remains an important issue.

The CEO strategy is presented first. All rotation angles are set to 45° on all faces (other than the particular

boundary faces discussed earlier). The Mach contours are shown in Fig. 11. Note for this case the increment in Fig. 12 is 0.1 for plotting clarity, thus there are half as many contour lines as the other figures. The solution is highly oscillatory and shows signs of odd-even decoupling which appear as alternating shock waves and expansion shock waves. As currently implemented, Roe's scheme admits expansion shock waves as solutions to the Riemann problem in violation of the entropy condition. The case is recomputed with a minimum eigenvalue condition. The new result is shown in Fig. 12 where the increment is returned to 0.05. The entropy correction significantly reduces the amount of odd-even decoupling (in comparing Fig. 12 to 13 keep in mind the number of contour lines). However, unacceptable oscillations still exist in the solution.

The previous results demonstrate the shock capturing ability of the rotated flux that is the goal of the method. In spite of the oscillations, one can see that the shock waves are captured as sharply as the second-order grid aligned solution. However, the oscillations do exist and must be addressed.

The 45° case is now computed using CCO. The first solution uses the projection schedule of Eq.(4.26). The resulting Mach contours are shown in Fig. 13. The results show significantly improved shock capturing ability over the grid aligned results. Furthermore, the solution shows only a mild overshoot at the initial shock wave. The cell-center strategy captures the shock waves as well as the cell-edge technique but without the oscillations. This result is due to the averaging of the fluxes at the cell edge. Recall that each cell is rotated and the fluxes are projected to the cell edge. Therefore, each cell edge flux receives a contribution from each adjacent cell center. The averaging acts as a smoothing agent to the fluxes thereby reducing the amount of oscillations. This effect is a fortuitous by-product of the cell-center strategy. Also apparent from Fig. 13 is an indication of an expansion shock wave. This will be discussed below.

The next solution of Fig. 14 is computed using the CCO strategy coupled with the higher order projection described in Section 4.2. The result is identical to the previous case to plotting accuracy.

The CCNO strategy is now used to compute the solution for a constant rotation angle of 45°. Recall that the rotation in this strategy is performed in computational space. Therefore, the rotation angle in physical space is somewhat different than the 45° in computational space depending on the cell shape and aspect ratio. The result for the standard projection schedule is shown in Fig. 15. The result is most interesting. The scheme captures the shock waves quite accurately. The first reflected shock wave is captured significantly better than the second-order grid aligned result of Fig 10. However, the solution also shows a captured expansion shock wave. Again, this is because an entropy condition is not included in the flux function. The presence of the expansion shock can be viewed as a positive result. This is an indication that the Riemann solver is producing results consistent with the one-dimensional operator independent of the grid. Not shown is the higher order projection result which does not affect the solution to plotting accuracy as in the previous cell-center strategy.

The results of the CENO strategy are shown in Fig. 16. This solution is computed with the entropy fix to reduce the oscillations generated from the cell-edge scheme. Recall, in this strategy the rotation is performed in computational space defined by the local metrics therefore, the fluxes across the cell edge are non-orthogonal. It is seen that the oscillations generated in this strategy are somewhat reduced from the CEO result (Fig. 12). However, the results are not as clean as either of the cell-center strategies.

The convergence histories of the four strategies along with the first- and second-order grid aligned schemes are shown in Fig. 17. All schemes are run at the optimal CFL (the optimal CFL is determined by trial-and-error and is usually the maximum allowable time step but not always). The CFL numbers, and run time performance numbers for the 45° case are shown in Table 1. All solutions are run on a Cray-YMP and convergence is declared when the mass equation L_2-Norm reaches 10^{-11}. The "Time" column in Table 1 represents total run time including overhead. Since the run time of the first-order grid aligned computation is only 2. seconds, the time per grid point per iteration measure is somewhat misleading because the overhead is a significant percentage of the run time.

Table 1
Code Performance Parameters, $\theta = 45°$

	CFL	Iterations	Time-seconds	μsec/pt-iter
O(1) GA	∞	248	2.00	6.72
O(2) GA	∞	705	5.14	6.07
CEO	3.	1276	15.66	10.22
CCO	4.	715	7.60	8.85
CCO-HOP	8	634	7.19	9.45
CCNO	8.	1177	11.41	8.08
CCNO-HOP	8.	1165	12.70	9.08
CENO	6.	1304	13.51	8.63

It is readily apparent that the rotation impedes convergence. This is caused by two effects. The first is the reduced dissipation and possibly increased dispersion from the rotation. The second is that the LHS does not account for the rotation on the RHS. Recall that all strategies employ the same LU-SGS LHS. It is also seen that the cell-center schemes converge faster than the cell-edge schemes. This is not surprising in view of the highly

oscillatory cell-edge results. The fastest converging rotated scheme is shown to be CCO-HOP.

The next set of results is computed using the four strategies with the rotation angles based on the pressure gradient. The angles are computed every time step and then frozen at a particular level of convergence or maximum iteration limit.

The CEO result is shown in Fig. 18. The solution based on pressure gradient does not show the signs of odd-even decoupling that were apparent in the 45° case. However, the solution is highly dispersive. The rotation angles are frozen after 300 iterations with a value of *DMIN*=0.00001. The frozen directions for the vertical faces are shown in Fig. 19 (every other line is plotted in the streamwise direction while every line is plotted in the crossflow direction). While viewing Fig. 19, keep in mind that an upwind flux is computed in both the preferred and normal direction. The horizontal face directions are similar to those presented in Fig. 19. The horizontal vectors next to the wall indicate that a grid aligned flux is computed. The directions are shown to basically align with the flow field shock waves. However, around the initial shock wave, the angles are shown to be oscillatory. The oscillations in the solution feed back to the rotation angle computation and create further oscillations in the solution. This effect can be reduced by filtering the rotation angle data as reported in Ref. [5].

The CCO results are presented next and are shown in Figs. 20 and 21. It is seen that the pressure gradient results are inferior to the 45° case but are still comparable to the second-order grid aligned solution. More precisely, the initial shock wave is captured better but the improvement diminishes as the shock waves are reflected down the channel. The rotation angles are frozen when the density residual reached a value of $5.x10^{-4}$ and are shown in Fig. 21. For this case *DMIN*=0.0005. The directions are shown to align with the flow features and are well behaved. Recall that the same angle computation algorithm is used in this case as in the previous CEO case. This is evidence that the ill-behaved directions of the CEO result are triggered by the cell-edge scheme. Again, no improvement is gained by the more complicated projection.

The next set of results are for the CCNO method. The Mach contours are shown in Fig. 22. It is seen that the results are similar to the CCO strategy although slightly more dissipative. This is best seen in the first reflected shock wave. Also it is observed that the additional dissipation, resulting from upwinding across the computed pressure gradient as opposed to setting the upwind direction to 45°, triggers the spreading of the expansion fan thereby eliminating the expansion shock wave. The rotation angles were frozen at the same convergence level as the previous CCO solution. A value of *DMIN*=0.001 is used in conjunction with Eq. [4.35]. The rotation directions are mapped back into physical space and are shown in Fig. 23. The directions are shown to vary smoothly. Once again there is no improvement using the higher order projection.

The last case computed is the CENO strategy with the rotation based on pressure gradient. The result is shown in Fig. 24. The solution is very similar to the CEO result of Fig. 18 although the non-orthogonal projection is slightly less dispersive. The rotation angles were frozen after 300 iterations and are shown in Fig. 25 (*DMIN*=0.001).

The convergence histories of the rotated solutions based on pressure gradient are plotted in Fig. 26. The CFL numbers and run time data for the pressure gradient cases are shown in Table 2. The following observations are made. First, the cell-edge schemes do not converge until the rotation angles are frozen at 300 iterations. After the freezing, the convergence rate is less than the cell-center schemes. The cell-center schemes converge faster than the second-order grid aligned case and, interestingly enough, take less CPU time to compute a comparable solution. From experience, the cell-center strategies will converge about 3 to 3.5 orders of magnitude before a limit cycle is reached. Freezing the angles allows complete convergence.

Table 2
Code Performance Parameters, θ based on ∇p

	CFL	Iterations	Time-seconds	μsec/pt-iter
O(1) GA	∞	248	2.00	6.72
O(2) GA	∞	705	5.14	6.07
CEO	3.	789	10.42	11.01
CCO	20.	376	4.26	9.44
CCO-HOP	∞	360	4.33	10.02
CCNO	20.	389	4.04	8.65
CCNO-HOP	∞	399	4.58	9.57
CENO	7.	790	8.44	8.90

Throughout this study, the cell-edge schemes are shown to consistently create significant oscillations in the solution of this channel problem. There have been a variety of approaches to reduce these oscillations. Levy computes the 45° rotation case by only rotating the horizontal faces and computing a grid aligned flux on the vertical faces. This eliminates the oscillations but the improvement in shock capturing ability is reduced. In the pressure gradient case, averaging of the rotation angles throughout the domain is employed to reduce the feedback between the flow field oscillations and the rotation angle computation. Another technique is the simplified interpolation scheme of Ref. [6]. The simplified scheme reduces the dependence of the flux on the rotation angle since small changes in the angle do not change the interpolated dependent states. The cell-center strategies, on the other hand, allow greater flexibility in the angle selection and the interpolation. This is because the flux averaging, required in order to define a unique flux at the face, acts as an inherent smoothing agent in the scheme.

6.0 Conclusion

Four rotated upwind strategies have been tested on a simple Mach 2 channel flow problem. It is concluded that the cell-center strategies offer better promise in extending the study to three dimensions than the cell-edge strategies because of the reduction in work, the inherent smoothing in the flux averaging, and acceptable convergence rates. The CCO strategy produces the best results of the four strategies. The results are comparable to the second-order grid aligned scheme. The CCNO strategy also produces good results, is simpler to incorporate than CCO, and has the added feature of reverting to a grid aligned formulation. Finally, it is shown that for this simple test case, the higher order projection does not improve the cell-center results.

Acknowledgments

This research is supported by NASA Grant No. NAGW-1331 to the Mars Mission Research Center. Additional support has been allocated by the NASA-Ames Research Center, Moffet Field, California, under Interchange No. NCA2-719. The authors would also like to thank the North Carolina Supercomputing Center for the use of their facilities.

References

[1] Roe, P.L. (1981), "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," Journal of Computational Physics, Vol. 43, 1983, pp. 357-372.

[2] Rumsey, C.L., "Development of a Grid-Independent Approximate Riemann Solver," Ph.D. Dissertation, Department of Aerospace Engineering, University of Michigan, 1991.

[3] van Leer, B., "Progress in Multi-Dimensional Upwind Differencing," Proceedings of the 13th International Conference on Numerical Methods in Fluid Mechanics, Rome, July 1992, Lecture Notes in Physics, 414, Springer-Verlag, pp.1-26.

[4] Davis, S.F., "A Rotationally Biased Upwind Difference Scheme for the Euler Equations," Journal of Computational Physics, Vol. 56, pp. 65-92.

[5] Levy, D.W., Powell, K.G and van Leer, B., "An Implementation of a Grid-Independent Upwind Scheme for the Euler Equations," AIAA Paper 89-1931-CP, 1989.

[6] Dadone, A. and Grossman, B., "A Rotated Upwind Scheme for the Euler Equations," AIAA Paper 91-0635, January, 1991.

[7] Kontinos, D.A. and McRae, D.S., "An Explicit, Rotated Upwind Algorithm for Solution of the Euler/Navier-Stokes Equations," AIAA Paper 91-1531-CP, 1991.

[8] Leck, C.L., and Tannehill, J.C., "A New Rotated Upwind Difference Scheme for the Euler Equations," AIAA Paper 93-0066, January, 1993.

[9] Levy, D.W., Powell, K.G and van Leer, B., "Use of a Rotated Riemann Solver for the Two-Dimensional Euler Equations," Journal of Computational Physics, Vol. 106, 1993, pp. 201-214.

[10] Yoon, S. and Jameson, A., "An LU-SSOR Scheme for the Euler and Navier-Stokes Equations," AIAA Paper 87-0600, Jan. 1987.

[11] Thompson, J.F., Warsi, Z.U.A, and Mastin, C.W., "Numerical Grid Generation," Elsevier Science Publishing Co., Inc., 1985, Chapter III.

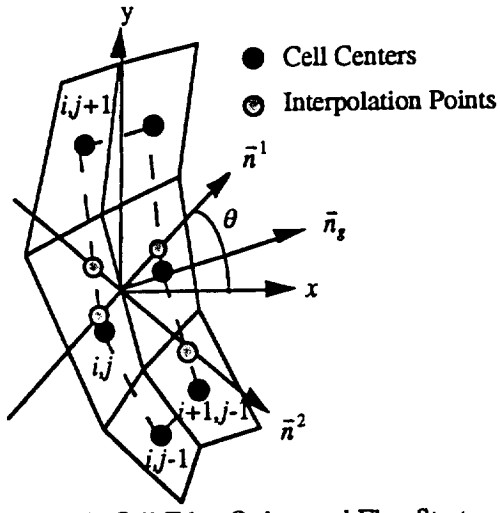


Figure 1.) Cell-Edge Orthogonal Flux Strategy.

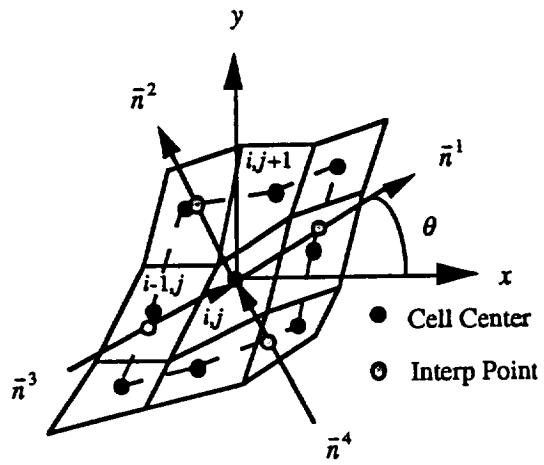
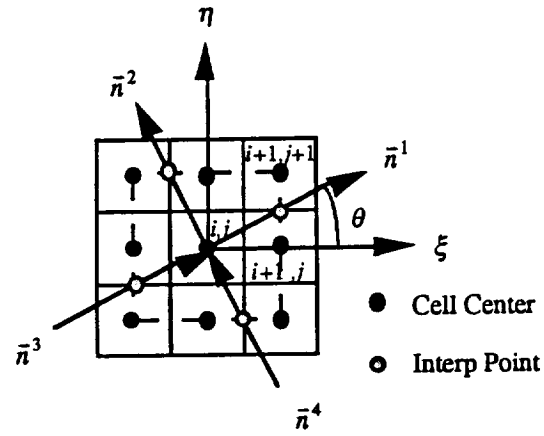
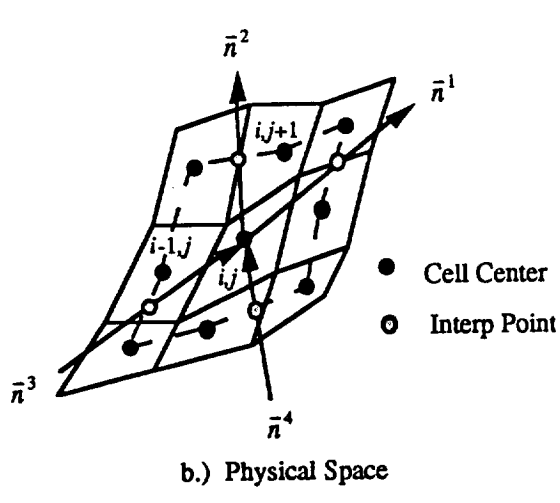


Figure 2.) Cell-Center Orthogonal Flux Strategy.

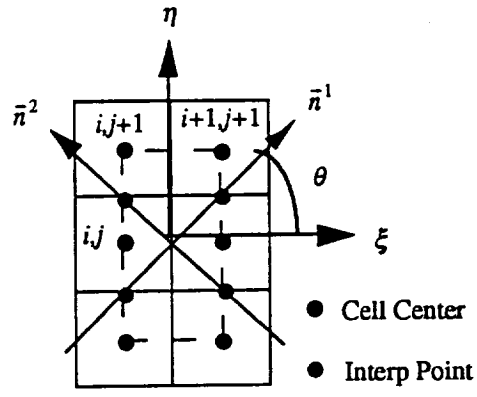


a.) Computational Space

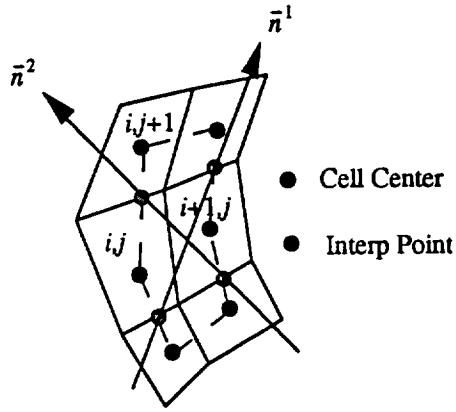


b.) Physical Space

Figure 3.) Cell-Center Non-Orthogonal Flux Strategy.



a.) Computational Space



b.) Physical Space

Figure 4.) Cell-Edge Non-Orthogonal Flux Strategy.

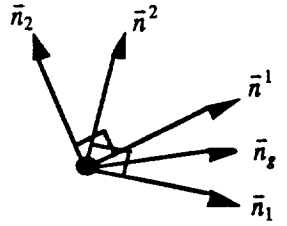


Figure 5.) Projection Vectors.

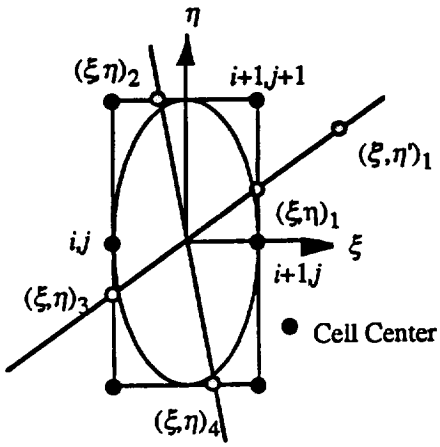


Figure 6.) Cell-Edge Interpolation Stencil.

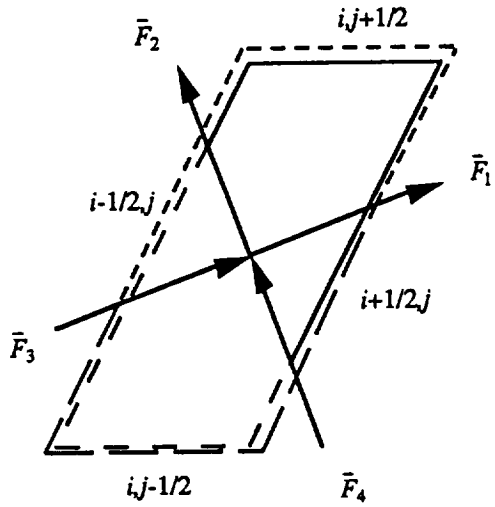


Figure 7.) Flux Projection over Parallelogram

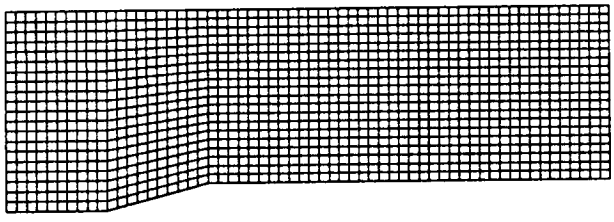


Figure 8.) 61x21 Grid

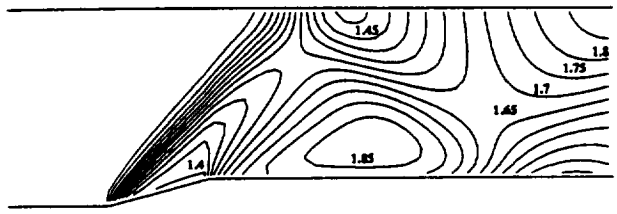


Figure 9.) Mach Contours for First-Order Grid Aligned, Increment = 0.05.

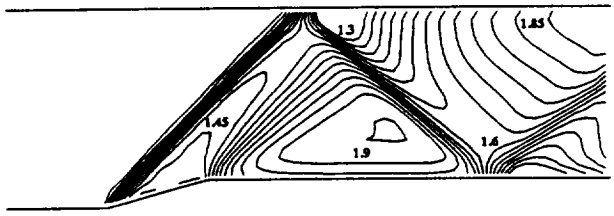


Figure 10.) Mach Contours for Second-Order Grid Aligned, Increment = 0.05.

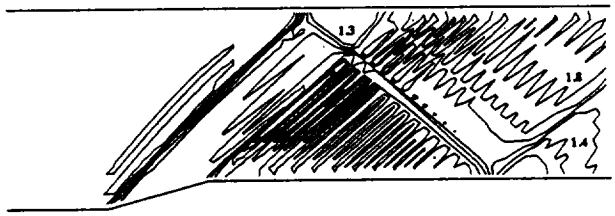


Figure 11.) Mach Contours for CEO, θ=45°, Increment = 0.1.

Figure 12.) Mach Contours for CEO, θ=45°, Entropy Fix, Increment = 0.05.

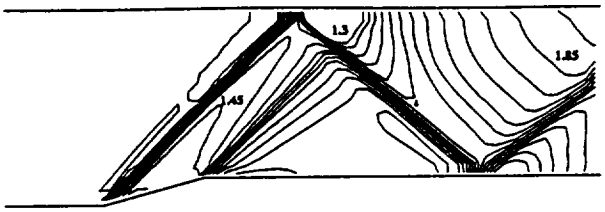


Figure 13.) Mach Contours for CCO, θ=45°, Increment = 0.05.

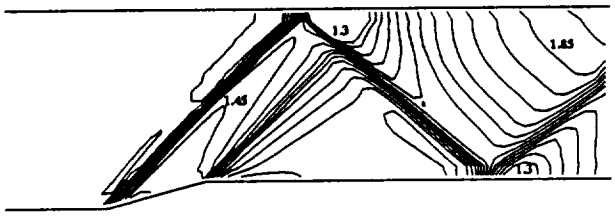


Figure 14.) Mach Contours for CCO, Higher Order Projection, θ=45°, Increment = 0.05.

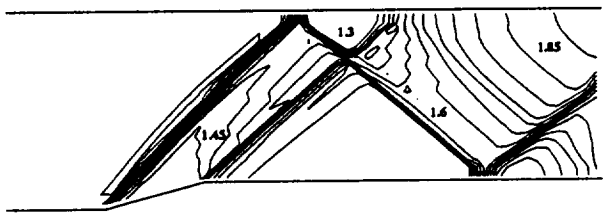


Figure 15.) Mach Contours for CCNO, θ=45°, Increment = 0.05.

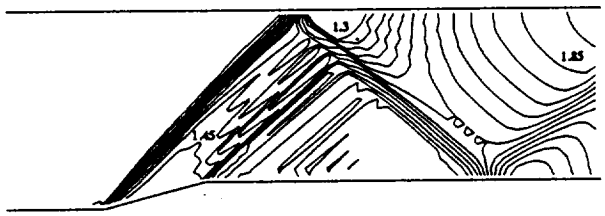


Figure 16.) Mach Contours for CENO, θ=45°, Entropy Fix, Increment = 0.05.

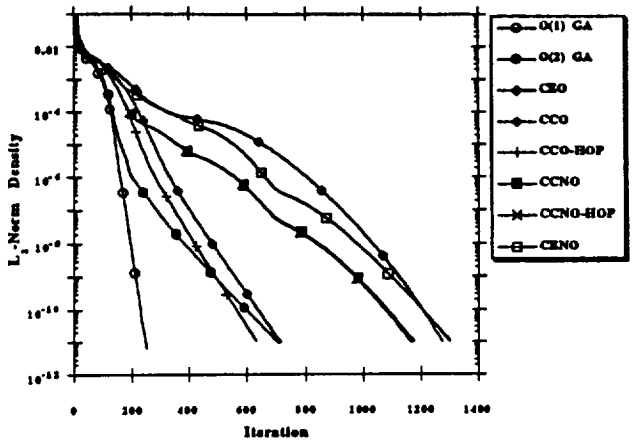


Figure 17.) Convergence Histories for θ=45° Solutions.

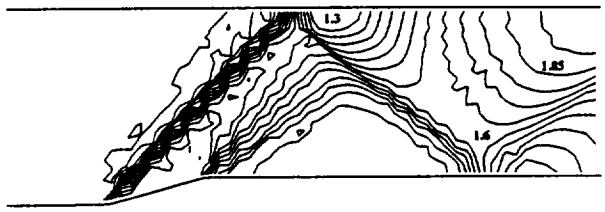


Figure 18.) Mach Contours for CEO, θ based on ∇p, Increment = 0.05.

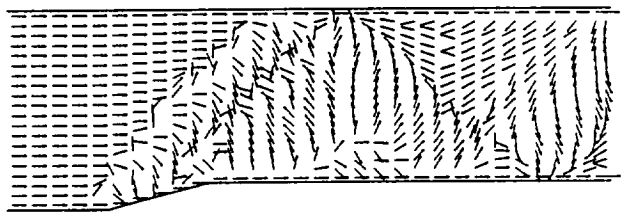


Figure 19.) Frozen Rotation Directions on the vertical faces for CEO, θ based on ∇p.

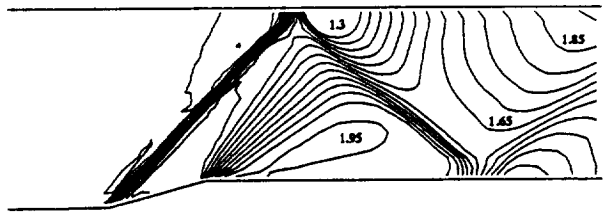


Figure 20.) Mach Contours for CCO, θ based on ∇p, Increment = 0.05.

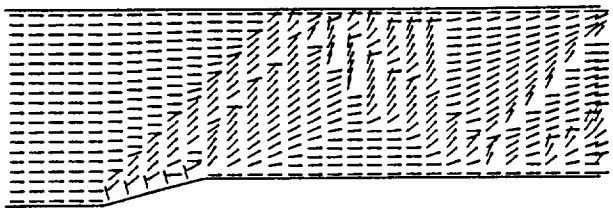


Figure 21.) Frozen Rotation Directions for CCO, θ based on ∇p.

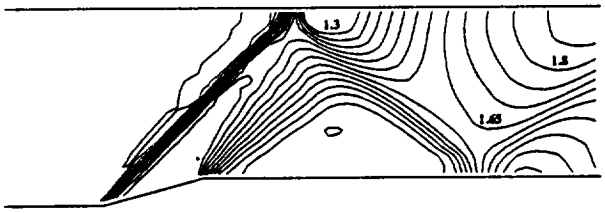


Figure 22.) Mach Contours for CCNO, θ based on ∇p, Increment = 0.05.

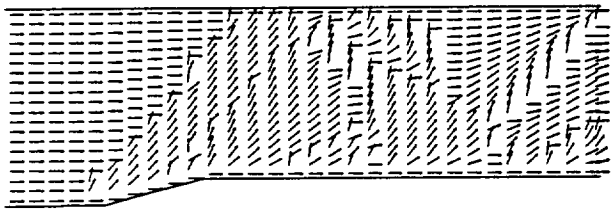


Figure 23.) Frozen Rotation Directions for CCNO, θ based on ∇p.

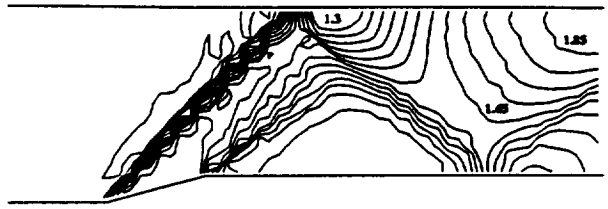


Figure 24.) Mach Contours for CENO, θ based on ∇p, Increment = 0.05.

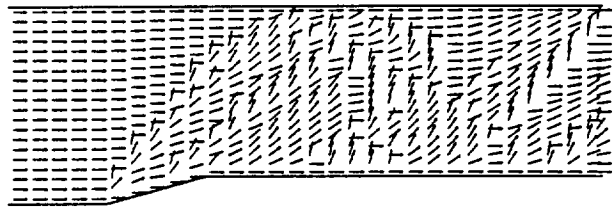


Figure 25.) Frozen Rotation Directions on the vertical faces for CENO, θ based on ∇p.

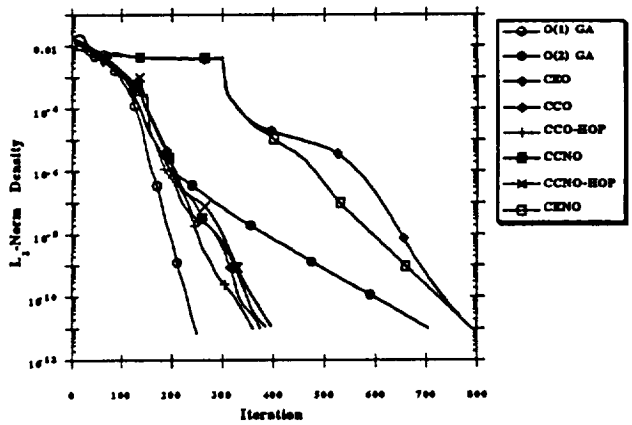


Figure 26.) Convergence Histories for θ based on ∇p solutions.

AIAA 94-2291

Rotated Upwind Algorithms for Solution of the Two- and Three-Dimensional Euler and Navier-Stokes Equations

D.A. Kontinos and D.S. McRae
North Carolina State University
Raleigh, NC

25th AIAA Fluid Dynamics Conference
June 20-23, 1994 / Colorado Springs, CO

Rotated Upwind Algorithms for Solution of the Two- and Three-Dimensional Euler and Navier-Stokes Equations

D.A. Kontinos* and D.S. McRae†
North Carolina State University
Raleigh, North Carolina

Abstract

Rotated upwind algorithms are presented for the numerical solution of the Euler and Navier-Stokes Equations in two and three dimensions. The finite-volume algorithms are designed with the notion of aligning Roe's approximate Reimann solver in a computed preferred direction. The algorithms are applied to shock wave reflection and shock-wave/boundary-layer interaction flowfields typically found in supersonic and hypersonic inlet configurations. Calculation of an inviscid Mach 2 channel flow problem shows that the rotated algorithm produces more accurate results than a traditional grid aligned algorithm to both first- and second-order accuracy. Moreover, the improvements to second-order accuracy are as great as those to first-order accuracy. Viscous solution of a turbulent boundary layer / shock wave impingement show that the rotated scheme improves the shock wave capturing in the inviscid portion of the flowfield to both first- and second-order accuracy. The improvements in the shock wave capturing to first-order accuracy result in improved wall pressure and skin friction distributions. However to second-order accuracy, the wall predictions are not improved. The calculation of an inviscid three-dimensional shock wave surface shows the rotated algorithm to be more accurate than the grid aligned algorithm to both first- and second- order accuracy. The accuracy improvements in three dimensions are not as great as those in two dimensions. Computation of a viscous flowfield in the corner of two intersecting wedges shows that the inviscid portions of the flowfield are qualitatively improved with the rotated algorithm to both first- and second-order accuracy. However, surface pressure predictions are only marginally improved with the rotated algorithm.

1.0 Introduction

Much of the recent algorithm development has been focused on creating artificial dissipation models that include multi-dimensional effects. It is hoped that the great success of the upwind dissipation models can be extended by more accurately modeling wave propagation in multi-dimensional space. A variety of promising and novel approaches are currently being pursued. Since the number of methods has become too numerous to list in this format, the reader is referred to a comprehensive overview of the multi-dimensional methods presented by van Leer[1].

One approach to including multi-dimensional effects in the flux function is the use of a rotated upwind method. The idea, originating with formulations of the potential equations[2,3] and later developed by Davis[4] for the Euler equations, is to rotate the integration stencil to an orientation where the differencing is applied in a direction that more closely models the correct path of information propagation. For the Euler and Navier-Stokes equations, rotation results in applying the upwind formulation across captured discontinuities rather than along grid lines. The orientation of the upwind operator with respect to flow discontinuities significantly impacts the ability of the upwind model to correctly interpret local wave propagation. For example, Roe's[5] scheme is known to model a single shock wave as a sum of acoustic and shear waves if the operator is applied oblique to the shock wave (an excellent discussion of this effect is presented in Ref. [6]). The additional non-physical shear wave generates excess dissipation and smears the captured shock wave. However, the correct wave propagation is predicted when the operator is applied across the shock wave. Thus, we are motivated to calculate an upwind flux across flow discontinuities independent of the orientation of the discontinuity with respect to the grid.

Many rotated upwind schemes for solution of the Euler equations have been developed[7,8,9,10], all of which show increased shock resolution as compared to traditional grid aligned schemes for simple inviscid test problems to first-order accuracy. The improvements to second-order accuracy are shown to be modest but encouraging. Furthermore, none of the previous rotated upwind schemes have been extended to three dimensions. In fact, very few of the newly developed multi-dimensional upwind methods have been tested in three dimensions. Some exceptions are the works of Deconinck et. al.[11] for three-dimensional scalar advection, and Rumsey[6] who solves the Euler equations in three dimensions. Rumsey's results show improvement of three-dimensional shock and shear capturing to first-order accuracy. To second-order accuracy, only improvement to shear capturing is shown. Thus, it has not been demonstrated, nor is it necessarily anticipated, that a rotated upwind scheme that aligns the Roe operator in a single preferred direction in three-dimensional space will improve

---

* Research Assistant, Mechanical and Aerospace Engineering, Student Member AIAA.
† Professor, Mechanical and Aerospace Engineering, Member AIAA.

three-dimensional shock wave capturing. It is the objective of this study to develop a robust, rotated upwind algorithm for the solution of both the Euler and Navier-Stokes equations in two and three dimensions. The necessary characteristics of this scheme are that it is fully conservative, second-order accurate in space, and sufficiently robust to compute a variety of supersonic through hypersonic flowfields.

2.0 The Navier-Stokes Equations

For this study, the governing equations are the set of conservation laws governing three-dimensional viscous fluid flow known collectively as the Navier-Stokes equations. The equation set in integral form and in index notation is given as the following:

$$\frac{\partial}{\partial t}\int_{Volume}\rho dV + \int_{Surface}(\rho u_i n_i)dS = 0 \ , \qquad (2.1)$$

$$\frac{\partial}{\partial t}\int_{Volume}\rho u_j dV + \int_{Surface}\left(\rho u_i u_j + T_{ij}\right)n_i dS = 0 \ , \qquad (2.2)$$

$$\frac{\partial}{\partial t}\int_{Volume}\rho E_t \, dV + \int_{Surface}\left(E_t u_i + T_{ij}u_j + q_i\right)n_i \, dS = 0, \qquad (2.3)$$

where ρ is the density, u_i is the velocity, E_t is the total energy, n_i is the outward normal to the surface bounding the control volume, q_i is the heat flux, dV is the differential element of the control volume, and dS is the differential element of the bounding surface. The term T_{ij} is the stress tensor which contains both the pressure and shear stress terms. The shear terms are modeled assuming a linear stress-strain relation, the heat flux is modeled through Fourier's law, and the dynamic viscosity is calculated using Sutherland's Law. In addition, the perfect gas equation of state is employed.

The governing equations are solved in a finite-volume formulation with the conserved variables taken as averages over the cell volume. The surface integrals of Eq. (2.1 - 2.3) become summations over the cell sides. Expanding the vector momentum equation into its scalar components, the equations for a fixed grid become,

$$\frac{\partial U}{\partial t} = -\frac{1}{Volume}\sum_{sides}\vec{F}\bullet\vec{S} \ , \qquad (2.5)$$

where U is the vector of dependent, conserved variables and $\vec{F}$ contains the fluxes. The cell face normal as expressed in terms of the grid metrics is denoted by $\vec{S}$. The magnitude of $\vec{S}$ is equal to the cell face area. The Euler equation subset is obtained by neglecting shear terms.

3.0 Two-Dimensional Rotated Upwind Algorithm

Traditional upwind schemes solve the Riemann solution or split the flux Jacobian along grid lines. Or, in other words, the upwinding is performed in the grid contravariant directions regardless of the orientation of the flow features with respect to the grid. An alternative to the grid aligned method is the rotated upwind scheme which aligns the upwinding stencil in a direction that is more likely to model the flow physics. This section describes such a rotated upwind algorithm for the solution of the governing equations in two dimensions. This is followed by Section 4.0 which presents the extension of the algorithm to three dimensions.

The two-dimensional algorithm is an extension of the first-order Cell-Centered Non-Orthogonal flux (CCNO) strategy presented by Kontinos and McRae[12]. The approach is to compute a rotation angle in computational space based on the gradient of a flow field scalar such as pressure or Mach number. The difference stencil is then rotated into the preferred orientation as is shown in Fig. 1a. The rotated axes define four rotated contravarient directions which are functions of the cell face metrics and the rotation angle. The primitive variables are then interpolated onto the new stencil through linear interpolation between adjoining cell centers. The four fluxes are computed in the rotated contravarient directions using Roe's scheme with the cell center as one state and the appropriate interpolation point as the second state value. The next step is to project the fluxes back onto the original grid faces. However, the orthogonal axes in computational space become four non-orthogonal directions in physical space as shown in Fig 1b. Therefore, in order to project the fluxes in a conservative manner, each cell face projection must be computed using a full metric tranformation. This results in four metric tranformations being computed at every cell for every iteration. The rotation angles are computed at every cell center every iteration for 2-3 orders of magnitude convergence and are subsequently frozen. Complete details of this algorithm can be found in Ref. [12, 13].

The CCNO algorithm is extended to second-order accuracy using the MUSCL philosophy wherin the values of the state variables are extrapolated to the interface. In a standard grid aligned formulation, the extrapolation is performed along grid lines. For example, the right state of the $(i+1/2, j)$ face uses point $(i+2, j)$ as support for the extrapolation. Similarly, the right state of the $(i, j+1/2)$ face uses point $(i, j+2)$ as support. In the CCNO formulation, the same procedure is implemented, however, the second-order support point is a function of the rotation angle. For example, consider the first quadrant of the rotated stencil in computational space as shown in Fig. 2. The first-order line of interpolation is represented as the dotted line in the figure. Also shown is the next ring of grid points around point (i, j) which provides support for the second-order interpolation. Based on the rotation angle,

one could follow the same liner interpolation procedures as the first-order stencil. However, the second-order support ring consists of six points thus requiring logic to determine the location of the interpolation. Such a procedure significantly increases the amount of work required by the algorithm. An alternative procedure is implemented in this study. The second-order support point is taken to be the closest cell center value to the intersection of the rotated axis with the second-order support ring. Thus, the second-order interpolation is not a continuous function of the rotation angle, but is a step function. Of course, the introduction of a discontinuous interpolating function dictates that the rotation angles must be frozen in order to reach convergence. If the angles are not frozen, it is very probable that the second order interpolation will cycle between two discontinuous values. But since it is seen in Ref. [12] that the rotation angles are frozen in order to converge the first-order algorithm, the discontinuous second-order interpolating function does not introduce any further limitations.

The final component of the flux computation of the CCNO algorithm are rotated boundary conditions. Since the rotation angles are defined at the cell centers, the flux computation of each of the four faces surrounding a particular cell center are essentially coupled. Consequently, reflection and extrapolation boundary procedures commonly employed in grid aligned schemes must be modified to account for the coupling of the boundary faces with adjoining interior faces. The details of the boundary condition procedures are omitted here for brevity but are presented in Ref. [13]. Finally, the viscous fluxes are computed in the standard grid aligned fashion and the solution is relaxed in time using the diagonal form of the LU-SGS scheme of Yoon and Jameson[14].

4.0 Three-Dimensional Rotated Upwind Algorithm

This section will outline the development of the three-dimensional cell-centered non-orthogonal flux rotated upwind scheme. Recall, that the overall strategy of the CCNO scheme is to compute a preferred upwind direction in computational space at each cell center. Then a coordinate rotation is performed such that a computational axis is aligned in the preferred direction. Primitive variables are then interpolated onto the new coordinate system. The inviscid fluxes are computed and transformed back onto the grid faces in a conservative manner. Sections 4.1-4.4 detail the steps of the algorithm. Section 4.1 presents the coordinate rotation that aligns a coordinate axis with any arbitrary direction. The rotation matrix defined in Section 4.1 is then used in Section 4.2 to define the contravariant directions of the rotated fluxes. Section 4.3 presents the interpolation procedure and Section 4.4 describes the projection of the rotated fluxes onto the original grid faces.

4.1 Coordinate Rotation

The primary task of aligning a computational axis with an arbitrary direction can be performed in a variety of ways. For instance, given some direction, three Eulerian angles can be found which can be used to set up a rotation matrix. Application of this rotation matrix will cause one of the original coordinate axes to align with the preferred direction. However, this method of rotation will align the **same** original axis in the preferred direction for every given orientation. In other words, after rotation the original coordinate axis can end up in any of the eight quadrants of the original system. So upon calling the interpolation routine, which is required to compute the fluxes, logic must be included to determine the orientation of the new system with respect to the old. Therefore, the secondary objective is to minimize the number of quadrants in which a particular axis can lie. In a practical sense, the attempt is to define a rotation such that the indices of the interpolating stencil can be hardwired into the code, thereby reducing the amount of logic and computer work. The objectives are achieved by defining two possible rotational transformations. Application of these two transformations is guaranteed to align some computational axis with any given preferred direction. The following discussion describes the two rotational transformations and their implementation into the algorithm. This development is aided by the work of Mayer[15] who presents an excellent discussion of rotation matrices.

The first rotational transformation, Type I, is characterized by a rotation through θ about the ζ axis denoted by $Z(\theta)$ then a rotation about the ξ' axis through ϕ denoted by $\Xi(\phi)$. The rotation is restricted such that $0 \le \theta, \phi \le 90^\circ$. The transformation is then expressed by,

$$(\xi' \quad \eta' \quad \zeta') = (\xi \quad \eta \quad \zeta)\, Z(\theta)\, \Xi(\phi) \,. \qquad (4.1)$$

Figure 3 depicts the Type I rotation. The corners of the cube are cell centers in computational space. The center of the cube is point (i,j,k). For clarity, not all of the cell centers of the interpolating stencil are drawn. Notice in which quadrant of the original system the new axes lie. The $+\eta'$ axis lies in the (-,+,+) quadrant, $-\eta'$:(+,-,-), $+\zeta'$:(+,-,+), $-\zeta'$:(-,+,-). So given a preferred direction in any of these four quadrants, the Type I rotation will align one of the computational axes given a proper definition of the rotation angles. Also note that the $+\xi'$ axis lies on the border of the (+,+,+) quadrant and $-\xi'$:(-,-,-).

As can be expected, the Type II rotation will allow alignment of an axis given a preferred direction in the remaining four quadrants. The Type II rotation is given by a rotation through θ about the ζ axis then a rotation about the η' axis through $-\phi$ denoted by $H(-\phi)$. The negative sign on ϕ is a consequence of the right-handedness of the system. This transformation is expressed by,

$$(\xi' \quad \eta' \quad \zeta') = (\xi \quad \eta \quad \zeta)\, Z(\theta)\, H(-\phi) \ . \qquad (4.2)$$

Figure 4 depicts the Type II rotation. Now it is seen that the $+\xi'$ axis lies in the (+,+,+) quadrant, $-\xi'$:(-,-,-), $+\zeta'$:(+,-,-), $-\zeta'$:(-,+,+). The $+\eta'$ axis lies on the border of the (-,+,+) quadrant and $-\eta'$:(+,-,-). Comparing Type I and Type II rotations it is seen that the ends of ξ' and η' always lie in the same quadrant. Thus, for the ξ' and η' axes the objective of minimizing the number of possible orientations has been achieved. It is only ζ' which requires some logic to determine indices of the interpolating stencil. However, each end of the ζ' axis can only lie in one of two quadrants. Superposition of the two transformations result in the following rotation matrix,

$$R(\alpha,\theta,\phi) = \begin{bmatrix} \cos(\theta)\big(\alpha+\beta\cos(\phi)\big) & \\ -\sin(\theta)\big(\beta+\alpha\cos(\phi)\big) & \\ \sin(\phi)\big(\alpha\sin(\theta)-\beta\cos(\theta)\big) & \end{bmatrix}$$
$$\begin{bmatrix} \sin(\theta)\big(\alpha+\beta\cos(\phi)\big) & \beta\sin(\phi) \\ \cos(\theta)\big(\beta+\alpha\cos(\phi)\big) & \alpha\sin(\phi) \\ -\sin(\phi)\big(\alpha\cos(\theta)+\beta\sin(\theta)\big) & \cos(\phi) \end{bmatrix}, \qquad (4.3)$$

where is $\alpha = 1$ for a Type I rotation, $\alpha = 0$ for a Type II rotation, and $\beta = 1-\alpha$.

To completely define the rotation, the angles and the switch between Type I and Type II rotation must be computed. First let the preferred direction in computational space be given by $\hat{n}_{\text{pref}} = \left\langle p\hat{\xi} + q\hat{\eta} + r\hat{\zeta} \right\rangle$ where the $\hat{}$ denotes some unit normal in computational space. The switch between types of rotations is given as,

$$\alpha = \begin{cases} 1 \text{ for } pq < 0 \\ 0 \text{ for } pq \geq 0 \end{cases} . \qquad (4.4)$$

Before computing the rotation angles, the orientation of the preferred direction is revealed by computing the following switch functions,

$$\delta_{pq} = \begin{cases} 1 \text{ for } pq \geq 0 \\ 0 \text{ for } pq < 0 \end{cases} , \qquad (4.5)$$

$$\delta_{qr} = \begin{cases} 1 \text{ for } qr \geq 0 \\ 0 \text{ for } qr < 0 \end{cases} , \qquad (4.6)$$

$$\delta_{pr} = \begin{cases} 1 \text{ for } pr \geq 0 \\ 0 \text{ for } pr < 0 \end{cases} , \qquad (4.7)$$

$$\delta^* = \alpha\delta_{qr} + \beta\delta_{pr} . \qquad (4.8)$$

Then from geometric considerations the angles are given by,

$$\cos(\theta) = \frac{\delta_{pq}|p| + (1-\delta_{pq})|q|}{\sqrt{p^2+q^2}} , \qquad (4.9)$$

$$\cos(\phi) = \frac{(1-\delta^*)|r| + \delta^*\sqrt{p^2+q^2}}{\sqrt{p^2+q^2+r^2}} . \qquad (4.10)$$

The complete range of rotation is $0 \leq \theta,\phi \leq 90°$ so that the cosine of the angles is sufficient.

4.2 Contravariant Directions

The rotation matrix developed in the previous section is used to define the contravariant directions of the new coordinate system. Recall that the advantage of defining the rotation in computational space is that the scheme will automatically revert to a grid aligned formulation. Moreover, the scheme reverts to a grid aligned mode in the limit of 90° rotation. Conceptually, the coordinate transformation rotates each coordinate axis between adjoining cell face normals which are the contravariant directions of the grid. For example, let J be the metric Jacobian matrix given by,

$$J = \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix} . \qquad (4.11)$$

In a finite-difference formulation, the metrics are defined at the grid point so that the metrics of a new coordinate system, denoted by J' are unambiguously given by,

$$J' = R(\alpha,\theta,\phi)J . \qquad (4.12)$$

In a finite-volume formulation, the metrics are defined at the cell face. Therefore, under a cell centered coordinate rotation, the new metric Jacobian is not clearly defined. However, the rotation matrix does prescribe the proper transition between the cell face normals under a rotation in computational space. Consider the $+\xi'$ rotated contravariant direction as an example. We know that under the given

rotation, the $+\xi'$ axis always lies in the first quadrant of computational space. Furthermore from observing the limits of 0° or 90° rotation of both θ and ϕ, it is apparent that the contravariant direction is some function of the metrics of faces $(i+1/2, j, k)$, $(i, j+1/2, k)$, and $(i, j, k+1/2)$. Therefore the transformation of the $+\xi'$ axis is given by,

$$J' = R(\alpha,\theta,\phi)\begin{bmatrix} \xi_{x_{i+1/2,j,k}} & \xi_{y_{i+1/2,j,k}} & \xi_{z_{i+1/2,j,k}} \\ \eta_{x_{i,j+1/2,k}} & \eta_{y_{i,j+1/2,k}} & \eta_{z_{i,j+1/2,k}} \\ \zeta_{x_{i,j,k+1/2}} & \zeta_{y_{i,j,k+1/2}} & \zeta_{z_{i,j,k+1/2}} \end{bmatrix}. \quad (4.13)$$

It is important to note that this transformation yields the direction of the $+\xi'$ axis only, consequently, only the top row of elements of J' is needed giving,

$$\vec{n}^1 = j'_{11}\hat{i} + j'_{12}\hat{j} + j'_{13}\hat{k}\ , \quad (4.14)$$

where j'_{mn} are the elements of J' and $\vec{n}^1$ is the contravariant direction of the $+\xi'$ coordinate axis.

The other five contravariant directions are defined by developing some simplifying notation. First, let $J(a,b,c)$ be defined as follows,

$$J(a,b,c) = \begin{bmatrix} \xi_{x_{i+a,j,k}} & \xi_{y_{i+a,j,k}} & \xi_{z_{i+a,j,k}} \\ \eta_{x_{i,j+b,k}} & \eta_{y_{i,j+b,k}} & \eta_{z_{i,j+b,k}} \\ \zeta_{x_{i,j,k+c}} & \zeta_{y_{i,j,k+c}} & \zeta_{z_{i,j,k+c}} \end{bmatrix}. \quad (4.15)$$

Here, a, b, and c assume values of $\pm 1/2$ to select the sides of the computational cube at point (i, j, k). Furthermore, let the function $\mathbf{ROW}(m, A)$ give the vector on the mth row of matrix A. Then the $\vec{n}^1$ direction developed previously is denoted as

$$\vec{n}^1 = \mathbf{ROW}\big(1,\ R(\alpha,\theta,\phi)\,J(1/2,1/2,1/2)\big) \quad (4.16)$$

The remaining contravariant directions are developed in a similar manner and are given as,

$$\vec{n}^2 = \mathbf{ROW}\big(2,\ R(\alpha,\theta,\phi)\,J(-1/2,1/2,1/2)\big)\ , \quad (4.17)$$

$$\vec{n}^3 = \mathbf{ROW}\big(1,\ R(\alpha,\theta,\phi)\,J(-1/2,-1/2,-1/2)\big)\ , \quad (4.18)$$

$$\vec{n}^4 = \mathbf{ROW}\big(2,\ R(\alpha,\theta,\phi)\,J(1/2,-1/2,-1/2)\big)\ , \quad (4.19)$$

$$\vec{n}^5 = \mathbf{ROW}\big(3,\ R(\alpha,\theta,\phi)\,J(\alpha-1/2,-1/2,1/2)\big)\ , \quad (4.20)$$

$$\vec{n}^6 = \mathbf{ROW}\big(3,\ R(\alpha,\theta,\phi)\,J(1/2-\alpha,1/2,-1/2)\big)\ , \quad (4.21)$$

where $\vec{n}^2$, $\vec{n}^3$, $\vec{n}^4$, $\vec{n}^5$, and $\vec{n}^6$ are the contravariant directions of the $-\xi'$, $+\eta'$, $-\eta'$, $+\zeta'$, and $-\zeta'$ axes, respectively.

4.3 Interpolation

Once the rotation of Section 4.1 has been defined, the interpolation of the primitive variables can be performed. In fact, one of the advantages of rotating in computational space is that there is symmetry of the interpolation stencil about the cell center. The first-order interpolation is performed by computing coordinates of the interpolating ray and then using a trilinear interpolating function over a quadrant of computational space. For example, consider the (+,+,+) quadrant of Fig. 5 where the $\vec{n}^1$ ray is shown under a Type II rotation. Let the origin of (ξ, η, ζ) be at point (i, j, k). Then over the region of the cube any value $\tilde{q}$ can be given by the trilinear function,

$$\begin{aligned}\tilde{q} = (1-\zeta)\big[&\xi q_{i+1,j,k} + \eta q_{i,j+1,k} - (1-\xi)(1-\eta)q_{i,j,k} \\ &+\xi\eta\big(q_{i+1,j+1,k} - q_{i+1,j,k} - q_{i,j+1,k}\big)\big] \quad (4.22)\\ +\zeta\big[&\xi q_{i+1,j,k+1} + \eta q_{i,j+1,k+1} - (1-\xi)(1-\eta)q_{i,j,k+1} \\ &+\xi\eta\big(q_{i+1,j+1,k+1} - q_{i+1,j,k+1} - q_{i,j+1,k+1}\big)\big].\end{aligned}$$

All that needs to be defined are the coordinates (ξ,η,ζ) in the range $0 \le \xi, \eta, \zeta \le 1$. Of course it is desirable to define the coordinates such that for no rotation, a grid aligned scheme is recovered. One possible option is to set (ξ,η,ζ) such that $\sqrt{\xi^2+\eta^2+\zeta^2} = 1$. This is shown as the shell segment drawn in Fig. 5. This is the easiest option to implement because all that is required is a normalization of the coordinates defining the interpolating ray. The option used in this study is to clip (ξ,η,ζ) where the interpolating ray intersects the outer face of the interpolating stencil. Then the trilinear function of Eq. (4.22) in effect reduces to bilinear interpolation over the outer square region. Moreover, if the interpolating ray happens to intersect the line segment adjoining two grid points, the interpolating function reduces to linear interpolation. This second option is selected so that under/overshoots are minimized through a low order interpolating function. Similar expressions to Eq. (4.22) are written for the other quadrants of the interpolating stencil.

The second-order interpolation of the three-dimensional algorithm is similar to the two-dimensional procedure of

Section 3. The second-order interpolant value is taken as the nearest cell center value to the interpolating ray. Consider the first quadrant of the second-order stencil as shown in Fig. 6. In the first quadrant, there are 19 points in the stencil which are displayed in Fig. 6 as black and gray circles (excluding (i, j, k)). The discontinuous interpolation function must be able to select the nearest cell center value to the interpolating ray. Such a function is costly because there are 19 possible interpolant values. Consequently, the stencil is reduced to be composed of the 7 corner points which are shaded black in Fig. 6. The indices of the nearest cell centers are calculated based on the coordinates of the first-order interpolating ray. Details of this procedure are contained in Ref [13].

4.4 Projection

The inviscid fluxes are computed using Roe's scheme with the cell center point as one state, the appropriate interpolated point for the second state, and the rotated metrics defined in Section 4.2. At this point in the computation, there are six fluxes computed in the rotated contravariant directions. In order to maintain conservation, the fluxes must be projected back onto the original grid faces. Recall that the fluxes are orthogonal in computational space; however, they are non-orthogonal in physical space. Therefore, in order to project the fluxes, a new transformation must be computed for each face based on the contravariant directions as is done in the two-dimensional algorithm. The three-dimensional transformation can be found in Thompson et. al.[16]. Once again for emphasis, the full transformation must be computed at each grid point for each face in order to maintain conservation. This results in six metric transformations being computed at each grid point every iteration. Unfortunately, this requirement causes significant increases in the algorithm's work load.

The projection schedule for each of the six faces of the cell is now given. Three fluxes are required per face and are selected such that in the limit of 90° rotation the contravariant directions are properly oriented in a grid aligned fashion. For example, the $\vec{F}_I \bullet S_{i+1/2,j,k}$ flux (where $\vec{F}_I$ is the inviscid flux and $S_{i+1/2,j,k}$ is the cell face normal of face $(i+1/2, j, k)$) must be a function of the contravariant directions that have a $(i+1/2, j, k)$ metric dependency. From Eq. (4.16) and Eqs. (4.19 - 4.21) it is seen that $\vec{n}^1$, $\vec{n}^4$, $\vec{n}^5$, and $\vec{n}^6$ all have a $(i+1/2, j, k)$ metric dependency. Letting H represent the transformation, the projection dependency for face $(i+1/2, j, k)$ is given by,

$$\vec{F}_I \bullet \vec{S}_{i+1/2,j,k} = H\left(F^1, F^4, \alpha F^5, \beta F^6\right), \qquad (4.23)$$

where $F^i = \vec{F}_I \bullet \vec{n}^i$ and α and β are the switching coefficients between Type I and Type II rotation. The projection schedule for the remaining faces are given as,

$$\vec{F}_I \bullet \vec{S}_{i-1/2,j,k} = H\left(F^2, F^3, \beta F^5, \alpha F^6\right), \qquad (4.24)$$

$$\vec{F}_I \bullet \vec{S}_{i,j+1/2,k} = H\left(F^1, F^2, F^6\right), \qquad (4.25)$$

$$\vec{F}_I \bullet \vec{S}_{i,j-1/2,k} = H\left(F^3, F^4, F^5\right), \qquad (4.26)$$

$$\vec{F}_I \bullet \vec{S}_{i,j,k+1/2} = H\left(F^1, F^2, F^5\right), \qquad (4.27)$$

$$\vec{F}_I \bullet \vec{S}_{i,j,k-1/2} = H\left(F^3, F^4, F^6\right). \qquad (4.28)$$

The two numerical flux values from each adjoining cell center rotation around a particular face are averaged in order to uniquely define the cell face flux. Also, rotated boundary conditions are implemented in three dimensions and are described in Ref. [13].

5.0 Results

5.1 Inviscid, Mach 2, Channel Flow

The two-dimensional CCNO algorithm is tested by computing a Mach 2 flowfield in a channel with a 15° ramp which can be found in Ref. [7]. This geometry has been used quite extensively as a test case for multi-dimensional upwind algorithms. The deceptively simple geometry belies the intricate physics of the flow. In fact, fine grid resolution studies of this case by Parpia and Parikh[17] show that the initial shock wave does not reflect in a regular manner off of the top wall. Instead, a Mach stem develops and a slip surface is generated at a triple point. These flow characteristics are not captured with the 61x21 grid (Fig. 7) used in this study. The case is computed using the standard grid aligned upwind formulation and the resulting Mach contours are shown in Fig. 8. The contours range from 1.3 to 2.0 in increments of 0.05. The case is computed using the CCNO algorithm to first-order accuracy. The preferred upwind direction is selected to be the direction of the pressure gradient. The resulting Mach contours are shown in Fig. 9. The rotation angles are computed for every cell every iteration until the L_2-Norm of the mass equation is reduced by approximately three orders of magnitude whereupon the rotation angles are frozen. The freezing occurs in 260 iterations and the resulting orientations are shown in Fig. 10. The solution converges 12 orders of magnitude in 453 iterations at a CFL of 5 consuming 2.9 seconds of Cray Y-MP/C90 time.

The first-order CCNO method is shown to capture the shock waves with less smearing than the first-order grid aligned result of Fig. 8. The solution is relatively oscillation free except for a leading overshoot ahead of the

ramp shock wave. A pressure survey through the middle of the domain is shown in Fig. 11. The pressure comparison clearly shows that the rotated scheme captures the discontinuities and associated pressure rises more sharply than the grid aligned procedure to first-order accuracy. In order to quantify the error, a fine grid solution of 385x129 points is computed using the second-order grid aligned scheme. This solution is considered to be the "exact" solution of an error norm calculation. Mach number contours of the 385x129 grid solutions are shown in Fig. 12. Based on the fine grid solution, the density L_2-error-norm of the first-order grid aligned solution is calculated to be 4.561×10^{-3} while the first-order CCNO error is 2.521×10^{-3}. Analysis of the error distribution reveals that the major source of error is the smearing of the captured shock waves.

The increase in accuracy of a first-order rotated scheme has been shown by previous researchers e.g. Levy et. al.[7], Dadone and Grossman[8], and Kontinos and McRae[9,12]. However, the improvements of second-order accurate rotated schemes over second-order accurate grid aligned schemes have been shown to be modest at best. Thus, the CCNO algorithm is now compared to grid aligned upwinding to second-order accuracy on the 61x21 grid. The Mach contours of the second-order grid aligned and CCNO solutions are shown in Fig. 13 and 14, respectively. The range and increment of the contours are the same as the previous figures for direct comparison. The frozen rotation angles, based on the pressure gradient, are similar to those in the first-order solution as shown in Fig 10. From comparison to the second-order grid aligned result, it is seen that the rotated solution shows improved shock wave capturing. The improvement in the solution is also indicated in the pressure survey along $j = 10$ shown in Fig. 15. The density error norm of the grid aligned and CCNO solution are calculated to be 2.640×10^{-3} and 1.345×10^{-3}, respectively. Two very important observations can be made from the error calculation. First, the first-order CCNO solution is more accurate than the second-order grid aligned result. Secondly, and most importantly, the ratio of the error norms show that the improvement of the CCNO solution over the grid aligned solution is as great to second-order accuracy than to first-order accuracy. The CCNO solution with CFL=5 converges 12 orders of magnitude in 897 iterations in 7.33 seconds on the Cray Y-MP/C90.

The density error norms are plotted versus the number of points in the x direction on a log-log scale in Fig. 16. Additional data points, generated from 97x33 and 121x41 grids, are also included in Fig. 16. The figure shows that the rotated results are more accurate than the grid aligned results to both first- and second-order accuracy. Furthermore, the improvements to second-order accuracy are as great as to first-order accuracy. It is also seen that the accuracy gains are maintained as the grid is refined. Also shown in Fig. 16 are the slopes of the error lines which indicate the numerical order of accuracy of the schemes for this particular geometry. Of course the grid is not uniform in the y direction so the accuracy calculation based on the x direction spacing is only approximate.

5.2 Turbulent Shock Wave Impingement

The next two-dimensional test case is a shock wave impinging on an equilibrium turbulent boundary layer. The case is studied experimentally by Reda and Murphy[18] and is shown schematically in Fig. 17. A shock generator is inclined 13° to the incoming Mach 2.9 freestream. The Reynolds number is 5.73×10^5 per cm, the temperature is 108° K, and the incoming boundary layer height is 1.694 cm. The wall is measured to have a temperature of 291° K. At these conditions, the impinging shock wave creates an adverse pressure gradient sufficient to separate the turbulent boundary layer. The separation region appears as an increase in the thickness of the boundary layer and a shock wave is formed off of its leading edge. The flow then expands around the separation region and a second shock wave is formed at reattachment. Also shown in Fig. 17 is the location of the theoretical inviscid shock wave impingement location denoted by x_i. The results are presented with x_i as a reference point. This flowfield is measured by Reda and Murphy with two different experimental setups. In the first set-up, designated by G1, the shock generator spans the width of the channel thereby encompassing the sidewall boundary layers. The second setup, designated by G2, includes side plates to remove the effect of the sidewall boundary layers in order to isolate the shock-boundary layer interaction. It is found that including side plates reduces the streamwise scale of the interaction and lowers the pressure level of incipient separation. The flowfields of both setups are inherently three-dimensional in nature. It is found that as the shock generator angle is increased, separation occurs at the corners of the configuration and spreads symmetrically toward the centerline. At a shock generator angle of 13°, the flow is separated across the channel and a nearly uniform pressure distribution is measured. However, the uniformity of the pressure distribution across the span is not a sufficient condition for two-dimensionality of the flowfield. Consequently, comparisons of computed two-dimensional results to the experimental data must be made with caution.

The flowfield is computed using a 101x81 grid. The points are equally spaced in the streamwise direction and are geometrically stretched away from the wall. The minimum wall spacing is near $y^+ = 1$. The turbulence is simulated using the one-equation turbulence model of Edwards and McRae[19]. The first-order grid aligned and CCNO pressure contours are shown in Figs. 18 and 19, respectively. The grid aligned pressure contours show a smeared incident and reflected shock wave. The interaction region does not indicate separation of the boundary layer. The CCNO pressure contours show the shock waves to be captured with less smearing. Furthermore, the pressure contours indicate

an interaction region typical of separated flows i.e. both a separation and reattachment shock wave are visible. A comparison of the skin friction along the wall is plotted in Fig. 20. Also shown are the experimentally measured separation and reattachment locations. The skin friction results verify that the grid aligned solution does not predict a separation and that the CCNO solution does predict a separation. Despite the CCNO scheme's prediction of the separation zone, both solutions produce similar pressure distributions that are not in favorable agreement with the experimental data as shown in Fig. 21. The initiation of the pressures rise is not predicted well and the shape of the distribution is only in qualitative agreement. Although the pressure distributions of the two first-order solutions are very similar, clearly the CCNO scheme more accurately predicts the character of the flowfield. The shock waves are captured with less smearing and the separation zone is reproduced.

The grid aligned solution is run for 5,000 iterations at a CFL of 5,000 consuming 4.04 minutes of CPU on the Cray Y-MP/C90. The CCNO solution is computed with the rotation directions based on the pressure gradient. The rotation angles are damped to zero below the sonic line. It is found that rotating the stencil below the sonic line in the separation zone severely inhibits convergence. The solution is first computed in grid aligned mode for 1,000 iterations at a CFL of 5,000 to allow the incident shock wave to set up. The rotation angles are then computed at a CFL of 50 up to iteration 9,000 upon which the angles are frozen. The CFL is then reduced to 5 and the solution is computed to iteration 16,000. The total CPU time is 15.73 minutes on the Cray Y-MP/C90.

The shock impingement case is computed to second-order accuracy. The grid aligned pressure contours are shown in Fig. 22 while the CCNO pressure contours are shown in Fig. 23. It appears that the CCNO solution is capturing the incident and reflected shock waves with greater clarity. However, examination of the skin friction distribution of Fig. 24 reveals the two solutions predict very similar results. Both second-order solutions over predict the size of the separation zone. The pressure distributions through the interaction zone are shown in Fig. 25. The two schemes produce similar predictions which are in good agreement with the G1 data at the initial pressure rise. The post interaction pressure levels are in good agreement with the experimental data as well. It is difficult to determine which of the two experimental data sets are best used for a two-dimensional comparison. It is the intention of the experiment to reduce the three-dimensionality of the flowfield by cutting off the fully developed turbulent sidewall boundary layers with the side plates. Therefore, one anticipates that the G2 data is best for a two-dimensional comparison. However, adding the side plates generates laminar sidewall boundary layers that are more prone to separation than their turbulent counterparts. Thus, it is unclear which set of data is best to compare against the computational results. Moreover, the effectiveness of the turbulence model on this shock-impingement case is uncertain. Regardless of the comparison to the experimental data, it is important to discover that the CCNO solution does not significantly alter the flowfield predictions to second-order accuracy. The pressure contours suggest that the CCNO algorithm yields a qualitatively better solution in the inviscid region of the flowfield, yet the wall predictions are quite similar.

The second-order grid aligned scheme is run for 19,000 iterations consuming 15.31 minutes of CPU on the Cray Y-MP/C90. The CCNO solution is computed with the same rotation angle calculation as the first-order solution. The computation is run for 16,500 iterations in 19.07 minutes. The rotation angles are frozen at iteration 10,000 and are shown in Fig. 26.

5.3 Inviscid, Mach 2.8, Channel Flow

The first three-dimensional test case is Mach 2.8 inviscid flow in a four wall channel containing a ramp. This test case is taken from Rumsey[6] and the geometry is shown in Fig. 27. The foot of the ramp is at an oblique angle to the oncoming freestream and the inclination of the ramp varies across the domain. The side boundaries are all solid walls. With an inflow Mach number of 2.8, a truly three-dimensional shock surface develops within the channel thus providing a challenging test of the three-dimensional rotated scheme.

The case is run on a coarse grid with 41 points in the streamwise direction and crossflow planes of 17x17. All grid points are evenly spaced along boundary line segments. The interior points are linearly interpolated between opposite faces. One selected plane of the grid in each coordinate direction is shown in Fig. 28. The solution is computed to first-order accuracy with the grid aligned algorithm. The resulting Mach number contours on three selected planes interior to the domain are shown in Fig. 29. The contours for this and all subsequent contour plots range from 1.4 to 2.8 in increments of 0.05. The predominant feature is the ramp shock wave that reflects off of the top wall and exits the domain as shown in the j plane view. The k and i planes show variations across the domain demonstrating the three-dimensional nature of the solution. However, the first-order grid aligned results are highly smeared and secondary flow features are difficult to discern.

The solution is computed using the three-dimensional CCNO algorithm with the pressure gradient selected as the preferred direction. The resulting Mach contours are shown in Fig. 30. Clearly, the shock system is captured with less smearing using the CCNO algorithm. Not only is the primary shock resolved to a higher degree, but secondary shock structures that are smeared out with the grid aligned scheme are now evident. For example, the k plane view in Fig. 30 shows a secondary shock wave emanating from a

triple point midway across the ramp shock wave. This feature is completely smeared out in the grid aligned solution.

The first-order results of the CCNO algorithm are encouraging. Recall that few multi-dimensional upwind schemes have been developed and tested in three dimensions. Although certainly it can be predicted that aligning the Roe solver across a planar shock surface in three dimensions will reduce dissipation, it is not certain that any improvement at all can be expected by aligning a single coordinate axis in a preferred direction when crossing oblique shock waves exist such as in this test case. Thus, the first-order results present promise for the three-dimensional algorithm.

In order to quantify the error, a fine grid solution is computed and considered as the "exact" solution of an error norm calculation. The fine grid mesh spacing is one fourth that of the 41x17x17 grid thus yielding a 161x65x65 grid. The solution is computed to second-order accuracy with the grid aligned scheme. The resulting Mach contours are shown in Fig. 31. Evident in the fine grid solution is the curvature of the shock waves in the i and k planes as a result of the increase in shock strength created by the increase in ramp inclination across the domain. Moreover, the shock wave reflection off of the side walls creates a triple point evident in both the i and k views. A comparison of the first-order solutions to the fine grid solution is provided in Fig. 32 where a pressure survey through the center of the domain in the streamwise direction is shown. The pressure survey demonstrates the improved shock capturing ability of the CCNO algorithm.

The L_2-norm of the mass equation of the first-order grid aligned solution based on the fine grid solution is computed to be 1.889×10^{-3}. The first-order CCNO error is calculated to be 1.1314×10^{-3}. The ratio of the grid aligned to rotated error is 1.670. Recall that in the two-dimensional inviscid channel flow problem the error ratio is nearly 2. This is an indication that the improvement between grid aligned and rotated schemes is greater in two dimensions than in three dimensions to first-order accuracy.

The solution on the coarse grid is computed to second-order accuracy. The grid aligned Mach contours are shown in Fig. 33 while the CCNO Mach contours are shown in Fig. 34. From comparison between these two figures, it is seen that the rotated scheme improves the three-dimensional shock capturing. The CCNO result shows the triple point in the i and k planes while the grid aligned scheme only hints at its existence. Furthermore, a pressure survey through the middle of the domain, as shown in Fig. 35, also indicates improved accuracy. The mass equation error norm (based on the fine grid solution) of the second-order grid aligned scheme is calculated to be 9.566×10^{-4} while the CCNO error is 7.339×10^{-4}. The ratio of the error norms is 1.30. The improvements of the second-order accurate three-dimensional scheme are not as marked as the first-order results. However, this is still an encouraging result. The multi-dimensional wave model of Rumsey[6] showed no improvements to second-order accuracy. Furthermore, keep in mind that only a single preferred direction is defined and that the rotation is complete when a coordinate axis is aligned in that direction. There exists an additional degree of freedom in the plane normal to the preferred direction where an additional rotation can be defined. Then one of the two remaining coordinate axes could be aligned in some secondary direction. In this current formulation, this option is not explored and thus additional improvements may be possible.

The convergence histories of the coarse grid solutions are presented in Fig. 36. The first-order grid aligned solution is run at a CFL of 10,000 and reaches a convergence level of 10^{-11} in 170 iterations and 11.4 seconds on the Cray Y-MP/C90. The second-order grid aligned solution is also run with a CFL of 10,000 and reaches convergence in 369 iterations taking 23.8 seconds. The CCNO solutions are both run at a CFL of 10 with the preferred direction aligned with the pressure gradient. For the first-order CCNO solution the angles are frozen when the residual reaches $9.x10^{-4}$. Convergence is reached in 297 iterations and 41 seconds of CPU. The second-order CCNO rotation angles are frozen after 150 iterations. Convergence is achieved in 593 iterations and 107 seconds.

Consistent with the two-dimensional inviscid results, it is seen that the CCNO algorithm takes longer to converge than the grid aligned scheme. This is a result of the decrease in numerical dissipation and a poor temporal linearizaton of the rotated fluxes. Recall that the implicit side does not account for the rotation of the fluxes. It is also seen that the CCNO algorithm is significantly more computationally expensive than the grid aligned scheme. The grid aligned scheme (both first- and second-order) are running at approximately 6.3 micro-seconds per point per iteration. The first- and second-order CCNO algorithms are running at 13.5 and 17.6 micro-seconds per point per iteration, respectively. A large portion of the increase is due to the second-order interpolation because the computation of the interpolation indices does not optimize well on the Cray Y-MP/C90. The other intensive computation is the three-dimensional transformation of the rotated fluxes. An option to reduce this computation is to store the results of the transformation after the angles have been frozen. The storage requirements are an additional 18 real numbers per point. Furthermore, the indices of the second-order interpolation could also be stored and this could perhaps increase the amount of optimization. Although there is a penalty in the required memory, this option would significantly reduce the CPU usage since a large portion of the calculation is performed with the rotation angles frozen.

5.4 Laminar Intersecting Wedge Corner Flow

The three-dimensional CCNO algorithm is now applied to a viscous flow simulation. The test case is from the experiments of West and Korkegi[20] who study the interaction region in the corner of two intersecting 9.5° wedges in a Mach 3 freestream. At a Reynolds number of 3.9×10^5, the flowfield remains laminar. The interaction of the two wedge shock waves forms an oblique corner shock wave. Originating at the intersection of the wedge and corner shock waves are an embedded shock wave and a shear surface that terminates at the corner. The flowfield is discretized using 41 point in the streamwise direction spaced along 1.45 units in the freestream direction. The crossflow planes are 51x51 with a minimum spacing of 0.0005 units at the wall and then stretched to the outer boundary located at 1.75 units. The freestream temperature is 97.5 Kelvin and the wall temperature is held constant at 2.8 times the freestream temperature.

The solution of the intersecting wedge configuration is computed to first-order accuracy using both the grid aligned and CCNO algorithms. The rotation orientation of the CCNO algorithm is based on the Mach number gradient. Pitot pressure contours on three crossflow planes of the grid aligned and CCNO solutions are shown in Figs. 37 and 38, respectively. The first-order grid aligned solution shows the two wedge shock waves to be joined by a curved shock wave in the corner. The two embedded shock waves are seen to be smeared. The first-order CCNO contours show the wedge shock waves to be captured with less smearing. Furthermore, the union of the two wedge shock waves occurs through a corner shock wave and not a continuous transition as predicted by the grid aligned scheme. Also evident in the CCNO solution is the slip surfaces terminating at the corner. These slip surfaces are not seen in the grid aligned solution at the current contour settings.

As in the two-dimensional calculations, the inviscid portion of the three-dimensional flowfield is captured with greater clarity with the CCNO scheme to first-order accuracy. The previous pitot pressure contours reveal a qualitative improvement of the CCNO scheme as compared to the grid aligned. A quantitative comparison of the two solutions is contained in Fig. 39 where the wall pressure distribution in the transverse direction is compared to the experimental data. The ordinate of the plot is nondimensionalized by the x location of the survey i.e. x = 1.45. Also, the abscissa reference value, p_{wall}, is 1.98 times p_∞. Figure 39 shows that both first-order solutions are in general agreement with the experimental data. Both solutions are seen to under predict the pressure in the corner inside of the imbedded shock wave. This is a result of the coarseness of the grid. Prior to the pressure drop across the embedded shock wave , the CCNO solution is shown to slightly indicate the pressure rise between z/x = 0.2 and z/x = 0.3 whereas the grid aligned solution predicts a level pressure plateau. Both solutions are shown to under predict the position of the embedded shock wave along the z/x axis. Across the embedded shock wave, the smearing of the grid aligned solution is evident. Outboard of the embedded shock wave, the two solutions show a similar prediction which is in general agreement with the experimental data. Overall, the prediction of the wall pressure distributions is shown to be only modestly improved with the CCNO algorithm despite the qualitative improvements in the flowfield. Recall that the first-order two-dimensional results show that the improvements in the inviscid portion of the flowfield yielded more accurate wall predictions. In this three-dimensional test case, this improvement is not realized.

The solution is also computed to second-order accuracy. The resulting pitot pressure contours of the grid aligned and rotated algorithms are shown in Figs. 40 and 41, respectively. Comparing the two second-order solutions, only small differences in the character of the flowfield are detected. The CCNO solution is shown to capture the embedded shock waves with slightly more clarity. This is best seen in the middle of the three crossflow planes. Comparison of the wall pressure distribution is contained in Fig. 42. Evident in the corner region are low amplitude pressure oscillations of the CCNO solution. These oscillations are a result of the rotation in the boundary layer. As mentioned in the discussion of the two-dimensional turbulent shock impingement case, the solution is best computed by damping out the rotation below the sonic line. In this calculation, the angles are not damped and the pressure oscillations develop. The largest difference in the wall pressure distributions is seen immediately outboard of the embedded shock wave. The grid aligned algorithm is shown to under predict the pressure in the crossflow reattachment region. The under prediction of the pressure through this region is characteristic of upwind forumulations as can be seen in the results of Rudy, et. al.[21]. The CCNO solution is shown to be in better agreement with the experimental data in this region. Interestingly enough, the study of Rudy, et. al. shows that only a central difference formulation (MacCormack's scheme) did not predict the pressure dip suffered by the upwind schemes. In the case of the CCNO solution, it appears that realignment of the upwind operator in this interaction region, reproduces results characteristic of a central difference formulation.

The first-order grid aligned solution reaches convergence in 3,707 iterations using 52 minutes of processor time on the Cray Y-MP/C90. The second-order grid aligned solution is run for 9,000 iterations and 126 minutes of computer time reaching a five order of magnitude convergence. The CCNO solutions are computed with the rotation based on the Mach number gradient. The rotation angles are frozen after 3,000 iterations. The first-order solution is run for 10,000 iterations consuming 250 minutes of computer time. The second-order solution is run

for 9,000 iterations which uses 273 minutes of computer time. The two rotated solutions converge four to five orders of magnidue.

6.0 Conclusion

This study has presented the development of a rotated upwind algorithm for the numerical solution of the Euler and Navier-Stokes equations in both two and three dimensions. The two-dimensional algorithm is shown to predict an inviscid channel flow solution with greater accuracy than traditional grid aligned upwinding to both first- and second-order accuracy. The improvements to the second-order accurate results are shown to be as great as the first-order accurate solutions. Futhermore, the accuracy gains are maintained as the grid is refined. The calculation of a turbulent shock impingement flowfield shows that the new algorithm improves the shock wave capturing of the inviscid portion of the flowfield to both first- and second-order accuracy. However, only the first-order accurate scheme shows improved agreement with experimental wall pressure and skin friction distributions.

In three dimensions, a series of rotation matrices is developed that aligns a single coordinate axis in a preferred direction. Furthermore, the rotation matrices are defined such that the orientation of the rotated coordinate axes with respect to the grid axes is always known. The three-dimensional algorithm is applied to an inviscid shock reflection problem. The new algorithm is shown to predict the flowfield with greater accuracy to both first- and second-order accuracy. However, the improvements to the three-dimensional solution are not as great as those occurring in the two-dimensional inviscid solution. The new algorithm is shown to be sufficiently robust to compute the flow in the corner of intersecting wedges. To both first- and second-order accuracy, the inviscid features of the flowfield are shown to be qualitatively improved with the rotated algorithm. However, surface pressure predictions are only marginally improved.

Both the two- and three-dimensional algorithms are shown to consume 2-4 times as much CPU time as the grid aligned solutions. Further possibilities exist for improving the current algorithm. A better temporal linearization of the rotated fluxes may improve the convergence rate. Also, simplified interpolation schemes may produce similar results at less cost.

Acknowledgments

This research is supported by NASA Grant No. NAGW-1331 to the Mars Mission Research Center and by the NASA-Ames Research Center, Moffet Field, CA, under Interchange No. NCA2-719. Computer resources have been kindly granted by the North Carolina Supercomputing Center, NASA-Ames Research Center, and the National Aerodynamic Simulation program.

References

[1] van Leer, B., "Progress in Multi-Dimensional Upwind Differencing," Proceedings of the 13th International Conference on Numerical Methods in Fluid Mechanics, Rome, July 1992, Lecture Notes in Physics, 414, Springer-Verlag, pp.1-26.

[2] Murman, E.M. and Cole, J.D., "Calculation of Plane Steady Transonic Flows," AIAA Journal, Vol. 9, January 1971, pp. 114-121.

[3] Jameson, A., "Iterative Solution of Transonic Flows over Airfoils and Wings, Including Flows at Mach 1," Communications on Pure and Applied Mathematics, Vol. 27, 1974, pp. 283-309.

[4] Davis, S.F., "A Rotationally Biased Upwind Difference Scheme for the Euler Equations," Journal of Computational Physics, Vol. 56, 1984, pp. 65-92.

[5] Roe, P.L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," Journal of Computational Physics, Vol. 43, 1981, pp. 357-372.

[6] Rumsey, C.L., "Development of a Grid-Independent Approximate Riemann Solver," Ph.D. Dissertation, Department of Aerospace Engineering, University of Michigan, 1991.

[7] Levy, D.W., Powell, K.G and van Leer, B., "Use of a Rotated Riemann Solver for the Two-Dimensional Euler Equations," Journal of Computational Physics, Vol. 106, 1993, pp. 201-214.

[8] Dadone, A. and Grossman, B., "Characteristic-Based, Rotated Upwind Scheme for the Euler Equations," AIAA Journal, Vol. 30, No. 9, September, 1992, pp. 2219-2226.

[9] Kontinos, D.A. and McRae, D.S., "An Explicit, Rotated Upwind Algorithm for Solution of the Euler/Navier-Stokes Equations," AIAA Paper 91-1531-CP, 1991.

[10] Leck, C.L., and Tannehill, J.C., "A New Rotated Upwind Difference Scheme for the Euler Equations," AIAA Paper 93-0066, January, 1993.

[11] Deconinck, H., Struijs, R., Bourgois, G., Paillere, H., and Roe, P.L. "Mulitdimensional upwind methods for unstructured grids," AGARD-R-787, May, 1992.

[12] Kontinos, D.A. and McRae, D.S., "Rotated Upwind Strategies for Solution of the Euler Equations," AIAA Paper 94-0079, 1994.

[13] Kontinos, D.A., "Rotated Upwind Algorithms for Solution of the Two- and Three Dimensional Euler and Navier-Stokes Equations," Ph.D. Dissertation, Department of Mechanical and Aerospace Engineering, North Carolina State University, 1994.

[14] Yoon, S. and Jameson, A., "An LU-SSOR Scheme for the Euler and Navier-Stokes Equations," AIAA Paper 87-0600, Jan. 1987.

[15] Mayer, A., "Rotations and Their Algebra," SIAM Review, Vol. 2, No. 2, April, 1960.

[16] Thompson, J.F., Warsi, Z.U.A, and Mastin, C.W., "Numerical Grid Generation," Elsevier Science Publishing Co., Inc., 1985, Chapter III.

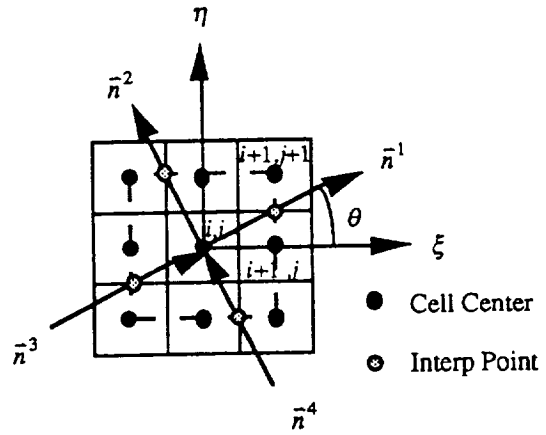
[17] Parpia, I. H. and Parikh, P., "A Solution Adaptive Mesh Generation Method With Cell-Face Orientation Control," AIAA Paper 94-0416, January, 1994.

[18] Reda, D.C. and Murphy, J.D., "Sidewall Boundary-Layer Influence on Shock Wave / Turbulent Boundary Layer Interactions," AIAA Journal, Vol. 11, No. 10, October, 1973, pp. 1367-1368.

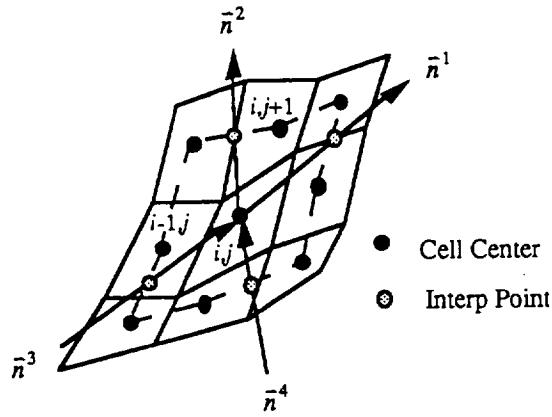
[19] Edwards, J.R. and McRae, D.S., "Nonlinear Relaxation Navier-Stokes Solver for Three-Dimensional, High Speed Internal Flows," AIAA Journal, Vol. 31, No. 7, 1993, pp. 1222-1228.

[20] West, J.E. and Korkegi, R.H, "Supersonic Interaction in the Corner of Intersecting Wedges at High Reynold Numbers," AIAA Journal, Vol. 10, No. 5, May, 1972, pp. 652-656.

[21] Rudy, D.H., Kumar, A., Thomas, J.L., Gnoffo, P.A., and Chakravarthy, S.R., "A Comparative Study and Validation of Upwind and Central-Difference Navier-Stokes Codes for High-Speed Flows," AGARD CP-437, Vol. 1, Dec., 1988, pp. 37-1 - 37-15.



a. Computational Space



b. Physical Space

Figure 1.) Cell-Center Non-Orthogonal Flux Strategy.

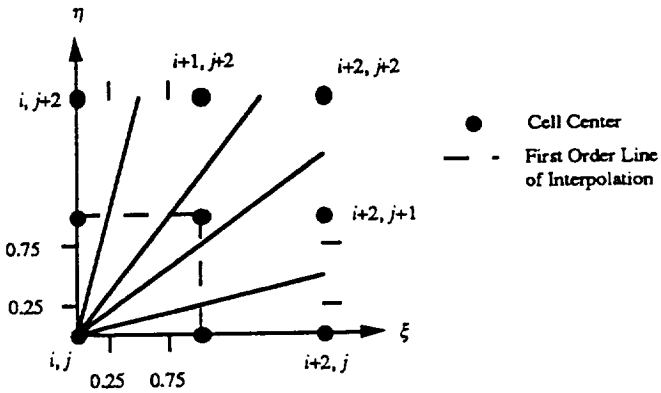


Figure 2.) Second-Order Interpolation Stencil in the First Quadrant of Computational Space for the CCNO Scheme.

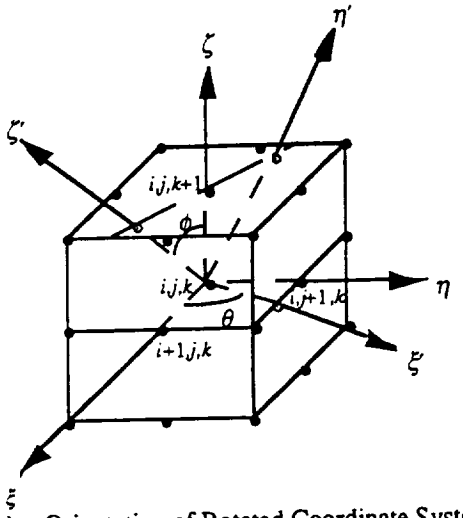


Figure 3.) Orientation of Rotated Coordinate System for Type I Rotation.

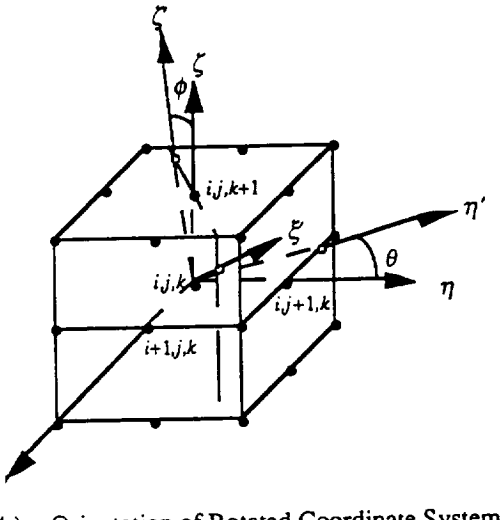


Figure 4.) Orientation of Rotated Coordinate System for Type II Rotation.

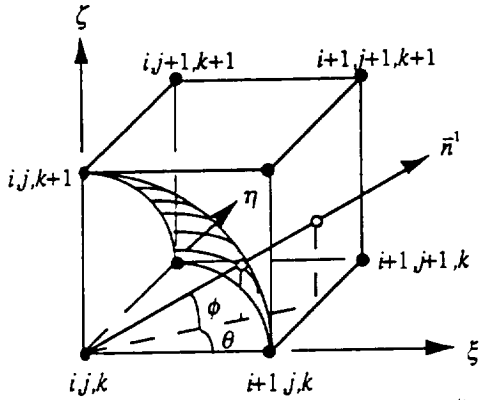


Figure 5.) First-Order Interpolating Stencil of $+\xi'$ Coordinate Ray under Type II Rotation.

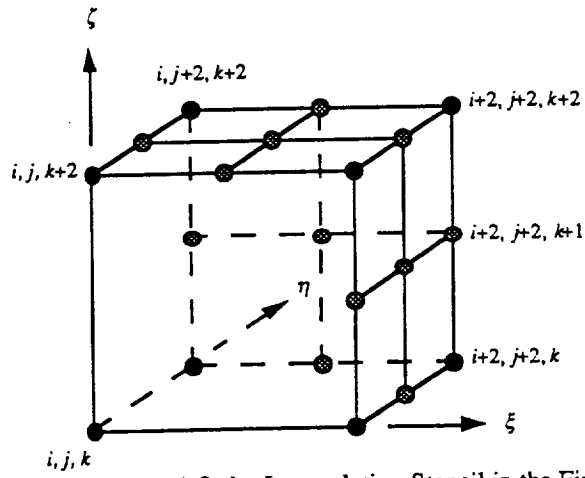


Figure 6.) Second-Order Interpolation Stencil in the First Quadrant of the Three-Dimensional CCNO Scheme.

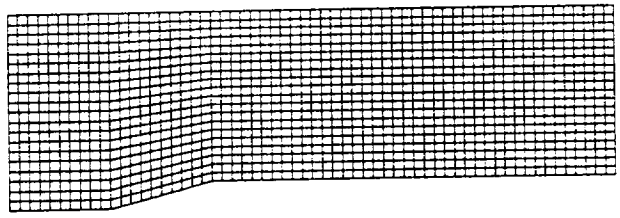


Figure 7.) Inviscid, Mach 2, Channel Flow, 61x21 Grid.

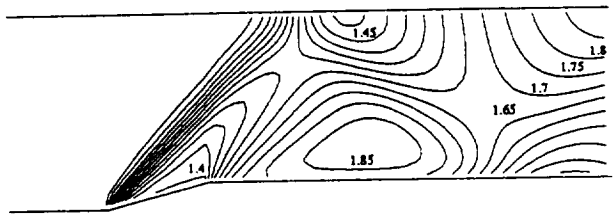


Figure 8.) Inviscid, Mach 2, Channel Flow Mach Contours, First-Order Grid Aligned Scheme, 61x21 Grid, Increment = 0.05.

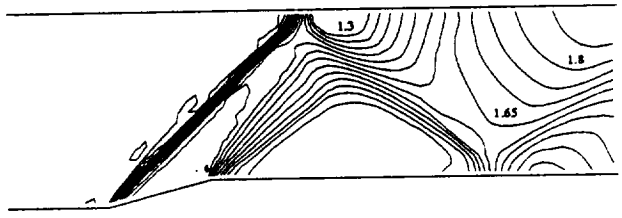


Figure 9.) Inviscid, Mach 2, Channel Flow Mach Contours, First-Order CCNO Scheme, 61x21 grid, Rotation based on ∇p , Increment = 0.05.

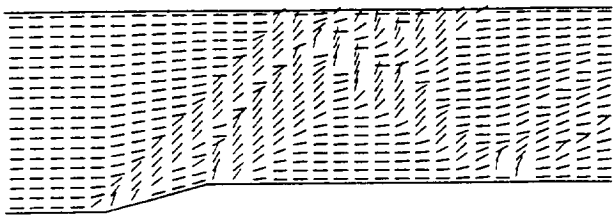


Figure 10.) Inviscid, Mach 2, Channel Flow Steady State Upwind Directions, First-Order CCNO Scheme, 61x21 grid, Rotation based on ∇p .

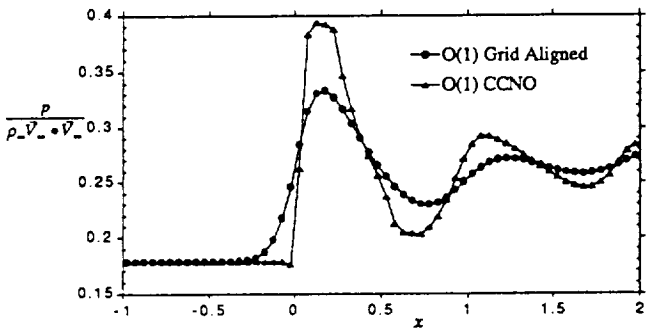


Figure 11.) Inviscid, Mach 2, Channel Flow Pressure Survey along j=10 computational line, First-Order Comparison, 61x21 Grid.

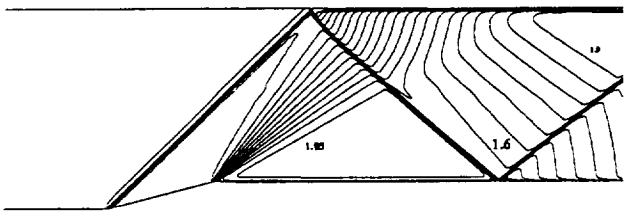


Figure 12.) Inviscid, Mach 2, Channel Flow Mach Contours, Second-Order Grid Aligned Scheme, 385x129 Grid, Increment=0.05.

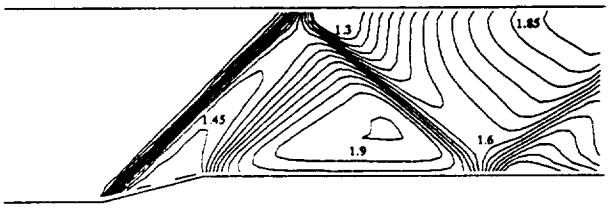


Figure 13.) Inviscid, Mach 2, Channel Flow Mach Contours, Second-Order Grid Aligned Scheme, 61x21 Grid, Increment = 0.05.

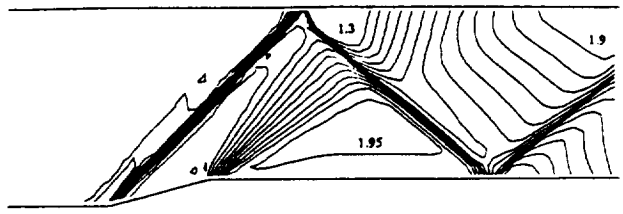


Figure 14.) Inviscid, Mach 2, Channel Flow Mach Contours, Second-Order CCNO Scheme, 61x21 Grid, Rotation based on ∇p , Increment=0.05.

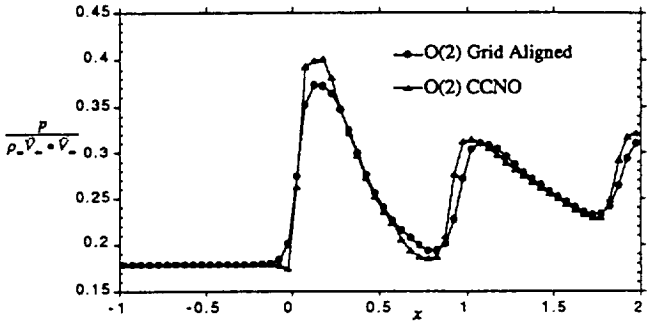


Figure 15.) Inviscid, Mach 2, Channel Flow Pressure Survey along j=10 computational line, Second-Order Comparison, 61x21 Grid.

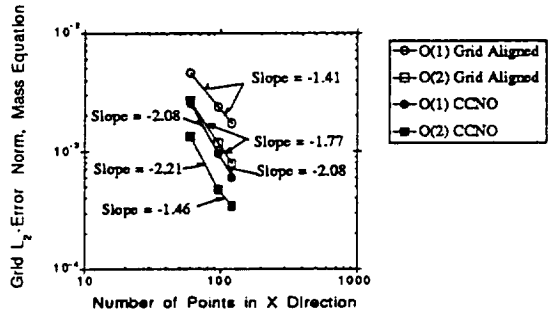


Figure 16.) Inviscid, Mach 2, Channel Flow Error Comparison.

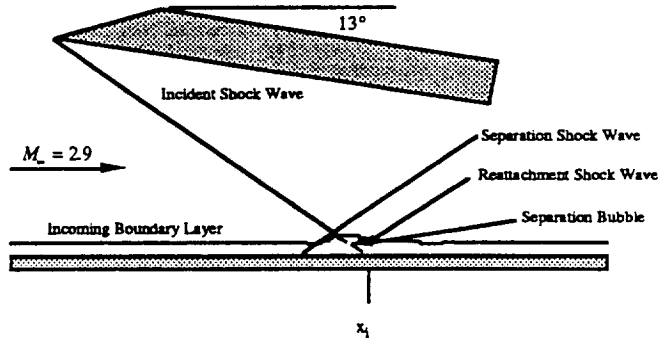


Figure 17.) Turbulent Shock Wave Impingement Geometry and Flowfield Schematic.

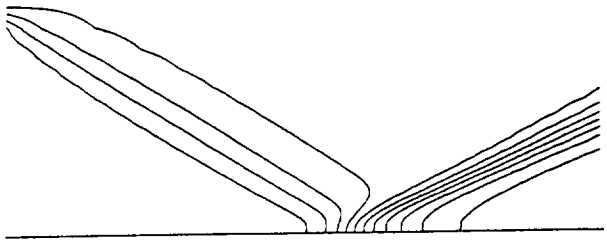


Figure 18.) Turbulent Shock Wave Impingement Pressure Contours, First-Order Grid Aligned Scheme.

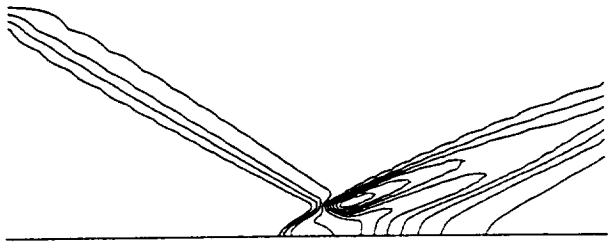


Figure 22.) Turbulent Shock Wave Impingement Pressure Contours, Second-Order Grid Aligned Scheme.

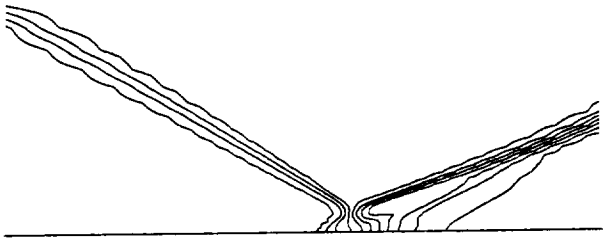


Figure 19.) Turbulent Shock Wave Impingement Pressure Contours, First-Order CCNO Scheme.

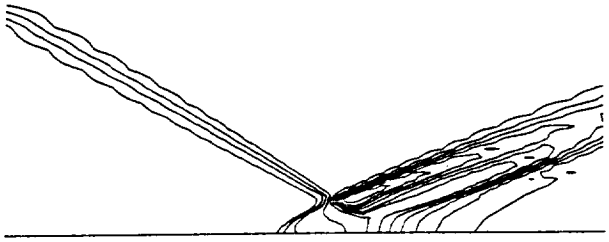


Figure 23.) Turbulent Shock Wave Impingement Pressure Contours, Second-Order CCNO Scheme.

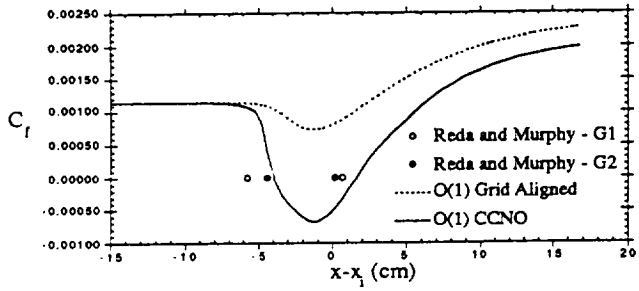


Figure 20.) Turbulent Shock Wave Impingement Wall Skin Friction Distribution, First-Order Comparison.

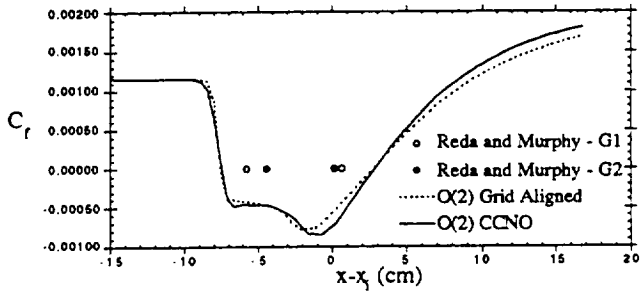


Figure 24.) Turbulent Shock Wave Impingement Wall Skin Friction Distribution, Second-Order Comparison.

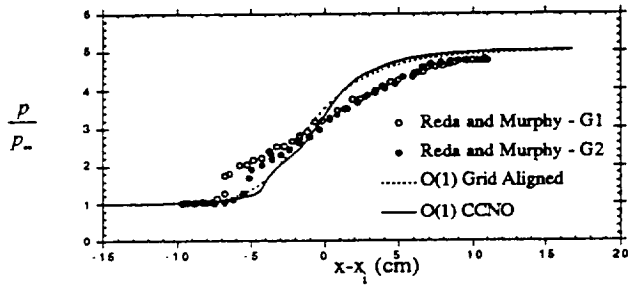


Figure 21.) Turbulent Shock Wave Impingement Wall Pressure Distribution, First-Order Comparison.

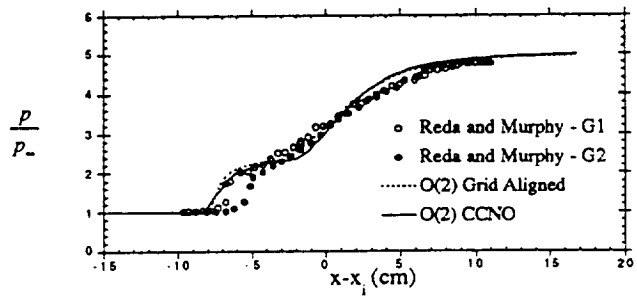


Figure 25.) Turbulent Shock Wave Impingement Wall Pressure Distribution, Second-Order Comparison.

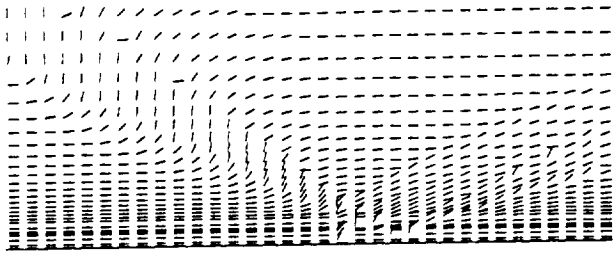


Figure 26.) Turbulent Shock Wave Impingement Steady State Upwind Directions, Second-Order CCNO Scheme.

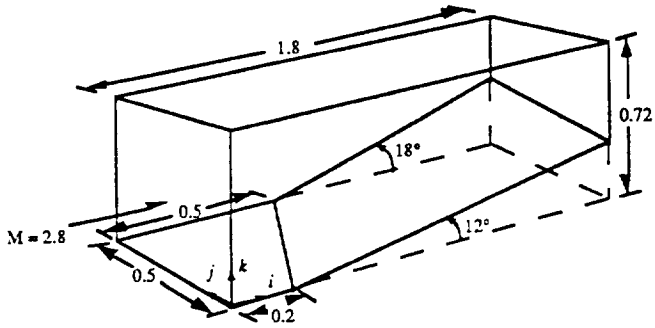
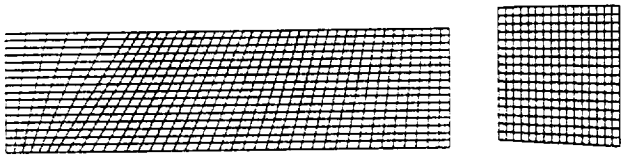
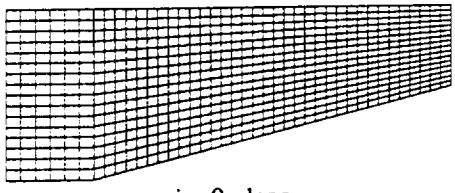


Figure 27.) Inviscid, Mach 2.8, Channel Flow Geometry.

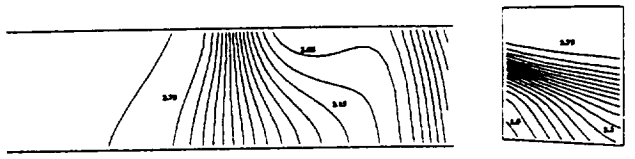


a. k = 9 plane. b. i = 21 plane.

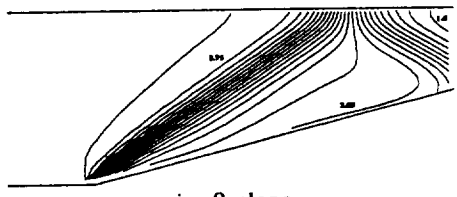


c. j = 9 plane.

Figure 28.) Inviscid, Mach 2.8, Channel Flow, Selected Grid Planes, 41x17x17 Grid.

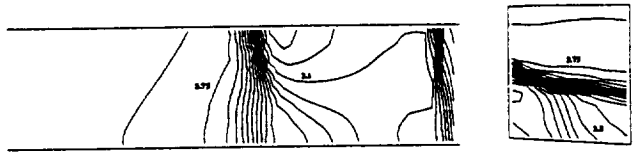


a. k = 8 plane. b. i = 20 plane.

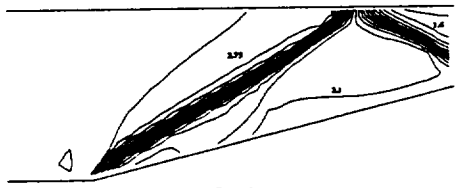


c. j = 8 plane.

Figure 29.) Inviscid, Mach 2.8, Channel Flow Mach Contours, First-Order Grid Aligned Scheme, 41x17x17 Grid.

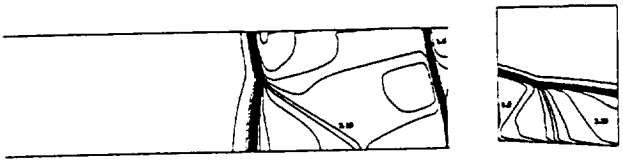


a. k = 8 plane. b. i = 20 plane.

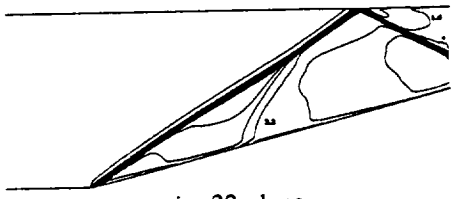


c. j = 8 plane.

Figure 30.) Inviscid, Mach 2.8, Channel Flow Mach Contours, First-Order CCNO Scheme, 41x17x17 Grid.

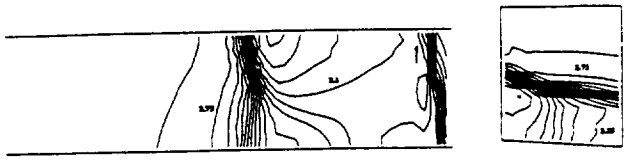


a. k = 32 plane. b. i = 80 plane.

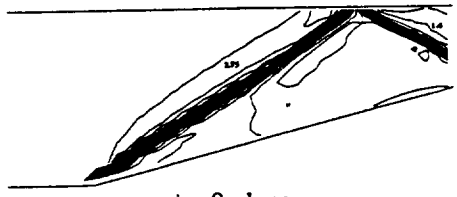


c. j = 32 plane.

Figure 31.) Inviscid, Mach 2.8, Channel Flow Mach Contours, Second-Order Grid Aligned Scheme, 161x65x65 Grid.



a. k = 8 plane. b. i = 20 plane.



c. j = 8 plane.

Figure 34.) Inviscid, Mach 2.8, Channel Flow Mach Contours, Second-Order CCNO Scheme, 41x17x17 Grid.

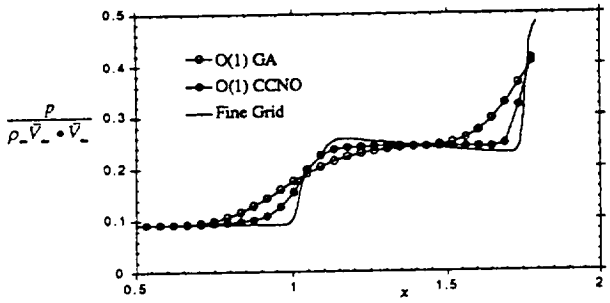


Figure 32.) Inviscid, Mach 2.8, Channel Flow Pressure Survery along j = 8, k = 8 computational line, First-Order Comparison, 41x17x17 Grid.

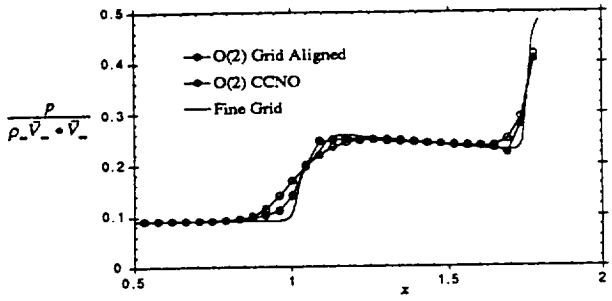
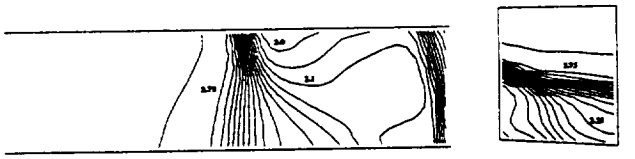
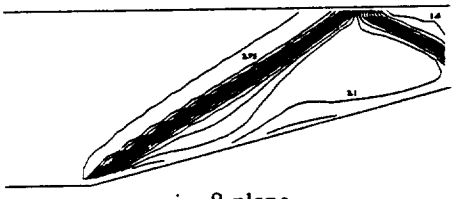


Figure 35.) Inviscid, Mach 2.8, Channel Flow Pressure Survery along j = 8, k = 8 computational line, Second-Order Comparison, 41x17x17 Grid.



a. k = 8. b. i = 20 plane.



c. j = 8 plane.

Figure 33.) Inviscid, Mach 2.8, Channel Flow Mach Contours, Second-Order Grid Aligned Scheme, 41x17x17 Grid.

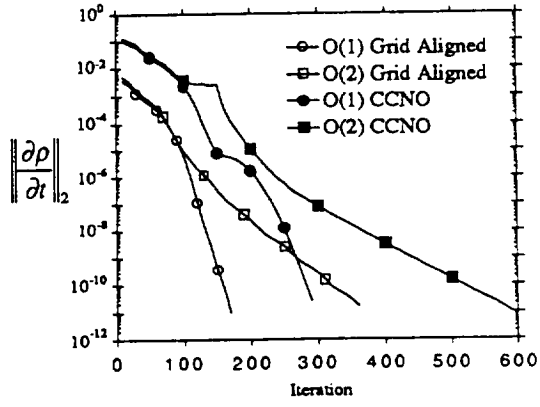


Figure 36.) Inviscid, Mach 2.8, Channel Flow Convergence History, 41x17x17 Grid.

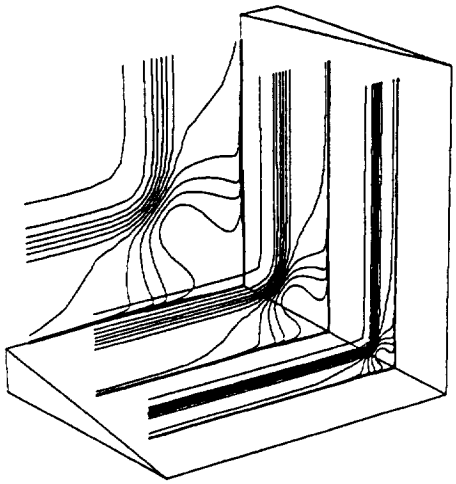


Figure 37.) Laminar Intersecting Wedge Corner Flow Pitot Pressure Contours, First-Order Grid Aligned Scheme.

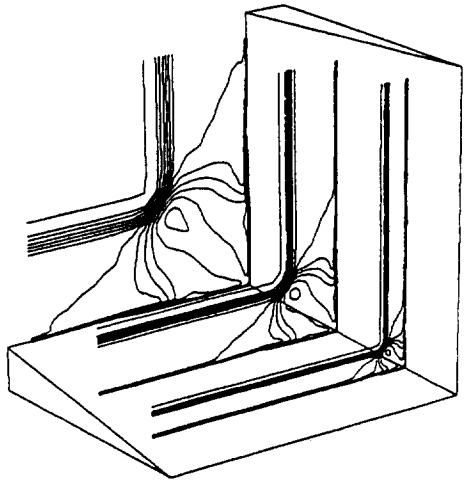


Figure 40.) Laminar Intersecting Wedge Corner Flow Pitot Pressure Contours, Second-Order Grid Aligned Scheme.

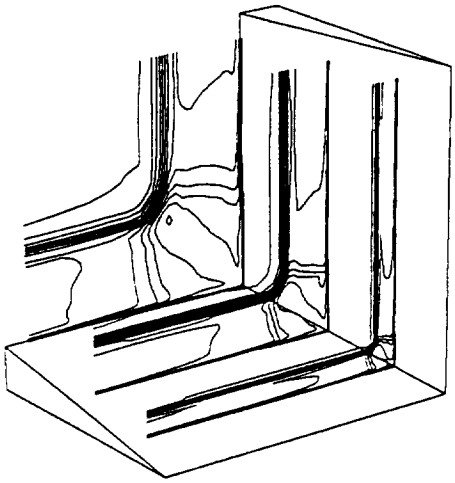


Figure 38.) Laminar Intersecting Wedge Corner Flow Pitot Pressure Contours, First-Order CCNO Scheme.

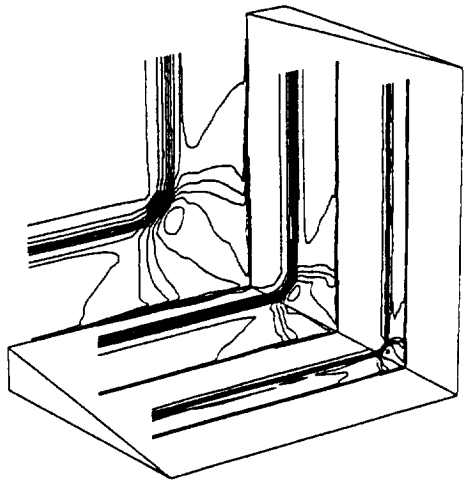


Figure 41.) Laminar Intersecting Wedge Corner Flow Pitot Pressure Contours, Second-Order CCNO Scheme.

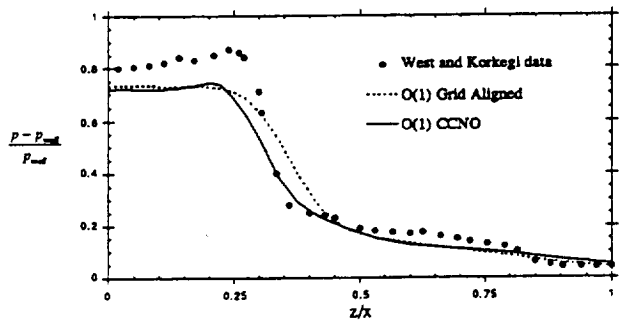


Figure 39.) Laminar Intersecting Wedge Corner Flow Wall Pressure Distribution, First-Order Comparison.

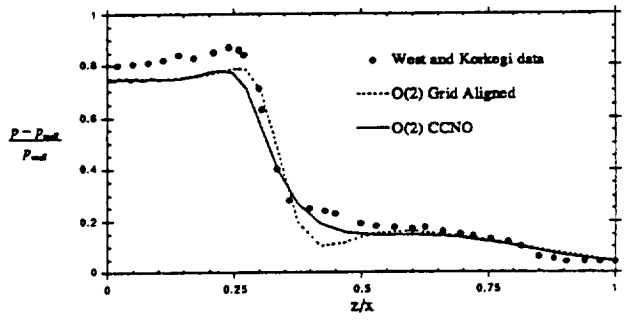


Figure 42.) Laminar Intersecting Wedge Corner Flow Wall Pressure Distribution, Second-Order Comparison.

Technical Publications Resulting from
This Cooperative Research

Kontinos, D. A. and McRae, D. S., "Rotated Upwind Strategies for Solution of the Euler Equations", AIAA 94-0079, 32nd Aerospace Sciences Meeting, Reno, NV, January, 1994.

Kontinos, D. A. and McRae, D. S., "Rotated Upwind Algorithms for Solution of the Two- and Three-Dimensional Euler and Navier-Stokes Equations", AIAA 94-2291, 25th AIAA Fluid Dynamics Conference, Colorado Springs, CO, June 20-23, 1994.

Kontinos, D. A., "Rotated Upwind Algorithms for Solution of the Two-and Three-Dimensional Euler and Navier-Stokes Equations", Ph.D. Dissertation, N.C. State University, Spring, 1994.