AIAA'89

AIAA-89-0283

**A Multigrid Finite Volume Method for
Solving the Euler and Navier-Stokes
Equations for High Speed Flows**

M.J. Siclari and P. DelGuidice,
Grumman Corporation, Bethpage, NY;
and A. Jameson, Princeton University,
Princeton, New Jersey

27th Aerospace Sciences Meeting

January 9-12, 1989/Reno, Nevada

A Multigrid Finite Volume Method for Solving the Euler and Navier-Stokes Equations for High Speed Flows

by

M.J. Siclari*

and

P. DelGuidice**

Grumman Corporate Research Center
Bethpage, New York

and

A. Jameson†

Princeton University
Princeton, New Jersey

Abstract

A node centered, finite volume, central difference scheme is used to solve both the unsteady Euler and Navier-Stokes equations for high speed flows in a spherical coordinate system. The steady state solution is obtained using multi-stage modified Runge-Kutta integration with local time stepping and residual smoothing to accelerate convergence. The basic scheme is augmented by a multigrid method to further enhance convergence. The method is applied to a variety of inviscid and viscous conical flows to determine the advantages of multigrid in conjunction with the basic scheme. Various numerical dissipation models used for viscous flows are also studied to determine their accuracy and effect on convergence.

Introduction

The formulation of the node centered, finite volume method used in the present study has been described in Ref. 1; it has been used extensively in various Euler computations for both conical and three-dimensional supersonic and hypersonic flows over realistic aircraft geometries. The finite volume scheme is applied only to the crossflow plane. For three-dimensional flows, a hybrid scheme is used in which the marching terms are approximated using upwind finite differences in computational space. This was found to be most suitable for the computation of high speed flows, since upwind differences realistically model the physical behavior of the hyperbolic axial flow, and do not require added dissipation in the marching direction. The basic multi-stage scheme, although very efficient for Euler flows, needs improvement for accurate and practical Navier-Stokes computations. Supersonic Euler computations over complete aircraft configurations (see Ref. 2) can take anywhere from 5 to 20 min on a Cray-XMP with 150,000 to 200,000 points, depending on the geometry and freestream condition. This makes the Euler code a useful tool for engineering computations. If viscous effects are included, say in a simple parabolized Navier-Stokes computation, preliminary tests have shown that the code can take as much as five times longer to converge, primarily due to the extreme grid stretching necessary to

adequately resolve the boundary layer. Actually, the inclusion of viscous terms has a minor effect on the convergence rate in comparison with the effects of the grid stretching needed to resolve the viscous effects. Euler computations carried out on viscous grids exhibit a similar degradation in convergence rate. In some computations, viscous effects can actually help convergence by eliminating the existence of strong shocks. Strong shocks also slow the rate of convergence, since implicit residual smoothing is less effective across discontinuities.

In this paper, a multigrid method is adapted to both the Euler and Navier-Stokes equations for high speed conical flows. Our intent is to apply a multigrid technique just to the crossflow plane terms to determine the technique's overall effectiveness, with the future goal of applying this method to three-dimensional flows. This is unique in that existing methods (see Refs. 3,4) using multigrid methods are attuned to computing three-dimensional subsonic/transonic flows by applying multigrid in three dimensions. For the case of very high speed flows, the effectiveness of applying a multigrid algorithm to all three dimensions has yet to be established. A reduction in the computational time by a factor of two or three, even for Euler aircraft computations, would greatly enhance the usefulness of the method as a design tool. For viscous computations, the basic Euler scheme is modified to include viscous effects by simply adding the viscous fluxes to the existing inviscid finite volume method. The velocity derivatives are computed at node points using central finite difference formulae in a computational space. Alternative Navier-Stokes discretization schemes could be devised. The primary focus of the present study, however, is to test the efficiency of the multigrid method. Various numerical dissipation models are also studied to determine their impact on the accuracy of boundary layer computations and their behavior in the presence of highly stretched viscous grids.

Governing Equations

The conical flow equations for viscous and inviscid flow represent a subset of the general Navier-Stokes equations. The unsteady Navier-Stokes equations in conservation form and cartesian coordinates can be written as

$$\frac{\partial Q}{\partial t} + \frac{\partial (F - F_v)}{\partial x} + \frac{\partial (G - G_v)}{\partial y} + \frac{\partial (H - H_v)}{\partial z} = 0 \quad (1)$$

* Senior Staff Scientist, Associate Fellow AIAA

** Staff Scientist, Associate Fellow AIAA

† Professor, Department of Mechanical and Aerospace Engineering, Member AIAA

where the vectors F , G , and H are the inviscid terms and F_v , G_v , and H_v represent the viscous shear stress and heat flux terms. The conservation variables and flux vectors are defined as follows:

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix} \quad F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho uh \end{bmatrix} \quad G = \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ \rho vh \end{bmatrix} \quad H = \begin{bmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ \rho wh \end{bmatrix}$$

$$F_v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ F_h \end{bmatrix} \quad G_v = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{yz} \\ G_h \end{bmatrix} \quad H_v = \begin{bmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ H_h \end{bmatrix}$$

where u , v , and w are the cartesian velocity components in the x , y , z directions, respectively, ρ is the density, p the pressure, and e the total energy. The pressure and total enthalpy are related to the flow variables by

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

$$h = \frac{e + p}{\rho}$$

where γ is the ratio of specific heats.

The Cartesian equations can be transformed to a spherical or conical coordinate system by the following transformation:

$$\bar{x} = x/r, \bar{y} = y/z, \text{ and } R^2 = x^2 + y^2 + z^2 \quad (3)$$

If all lengths are scaled by L , density and velocities by the freestream density (ρ_∞) and speed of sound (a_∞), energy and pressure by ($\rho_\infty a_\infty^2$), and time by (L/a_∞), the Navier-Stokes equations can be rewritten in spherical or conical coordinates as

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial \bar{x}} \left(\bar{F} - \frac{1}{R_e} \bar{F}_v \right) + \frac{\partial}{\partial \bar{y}} \left(\bar{G} - \frac{1}{R_e} \bar{G}_v \right) + 2 \left(\bar{I} - \frac{1}{R_e} \bar{I}_v \right) + \frac{1}{\Omega} \frac{\partial}{\partial R} \left(\bar{H} - \frac{1}{R_e} \bar{H}_v \right) = 0 \quad (4)$$

where the flux vectors ($\bar{F}$, $\bar{G}$, $\bar{H}$, $\bar{I}$) and ($\bar{F}_v$, $\bar{G}_v$, $\bar{H}_v$, $\bar{I}_v$) are defined as

$$\bar{F} = \begin{bmatrix} \rho \bar{U} \\ \rho \bar{U} + p \\ \rho v \bar{U} \\ \rho w \bar{U} - \bar{x} p \\ \rho h \bar{U} \end{bmatrix} \quad \bar{G} = \begin{bmatrix} \rho \bar{V} \\ \rho u \bar{V} \\ \rho v \bar{V} + p \\ \rho w \bar{V} - \bar{y} p \\ \rho h \bar{V} \end{bmatrix}$$

$$\bar{H} = \begin{bmatrix} \rho \bar{W} \\ \rho u \bar{W} + \bar{x} p \\ \rho v \bar{W} + \bar{y} p \\ \rho w \bar{W} + p \\ \rho h \bar{W} \end{bmatrix} \quad \bar{I} = \begin{bmatrix} \rho w \\ \rho u w \\ \rho v w \\ \rho w^2 + p \\ \rho h w \end{bmatrix}$$

$$\bar{F}_v = \begin{bmatrix} 0 \\ \tau_{xx} - \bar{x} \tau_{xz} \\ \tau_{xy} - \bar{x} \tau_{yz} \\ \tau_{xz} - \bar{x} \tau_{zz} \\ F_h - \bar{x} H_h \end{bmatrix} \quad \bar{G}_v = \begin{bmatrix} 0 \\ \tau_{xy} - \bar{y} \tau_{xz} \\ \tau_{yy} - \bar{y} \tau_{yz} \\ \tau_{yz} - \bar{y} \tau_{zz} \\ G_h - \bar{y} H_h \end{bmatrix}$$

$$\bar{H}_v = \begin{bmatrix} 0 \\ \bar{x} \tau_{xx} + \bar{y} \tau_{xy} + \tau_{xz} \\ \bar{x} \tau_{xy} + \bar{y} \tau_{yy} + \tau_{xz} \\ \bar{x} \tau_{xz} + \bar{y} \tau_{yz} + \tau_{zz} \\ \bar{x} F_h + \bar{y} G_h + H_h \end{bmatrix} \quad \bar{I}_v = \begin{bmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ H_h \end{bmatrix}$$

and

$$\bar{U} = \bar{u} - \bar{x} w$$

$$\bar{V} = \bar{v} - \bar{y} w$$

$$\bar{W} = \bar{u} \bar{x} + \bar{v} \bar{y} + w$$

$$\Omega = (1 + \bar{x}^2 + \bar{y}^2)^{1/2}$$

$$(R_e)^{-1} = M_\infty / R_{e_\infty}$$

The shear stress and heat flux terms become, assuming Stokes hypothesis,

$$\begin{aligned} \tau_{xx} &= 2\bar{\mu} \bar{u}_x - \sigma & \tau_{xy} &= \bar{\mu} (\bar{u}_y + \bar{v}_x) \\ \tau_{yy} &= 2\bar{\mu} \bar{v}_y - \sigma & \tau_{xz} &= \bar{\mu} (\bar{u}_z + \bar{w}_x) \\ \tau_{zz} &= 2\bar{\mu} \bar{w}_z - \sigma & \tau_{yz} &= \bar{\mu} (\bar{v}_z + \bar{w}_y) \end{aligned}$$

$$F_h = \bar{u} \tau_{xx} + \bar{v} \tau_{xy} + \bar{w} \tau_{xz} + A \bar{\mu} a_x^2$$

$$G_h = \bar{u} \tau_{xy} + \bar{v} \tau_{yy} + \bar{w} \tau_{yz} + A \bar{\mu} a_y^2$$

$$H_h = \bar{u} \tau_{xz} + \bar{v} \tau_{yz} + \bar{w} \tau_{zz} + A \bar{\mu} a_z^2$$

where

$$\sigma = \frac{2}{3} \bar{\mu} (\bar{u}_x + \bar{v}_y + \bar{w}_z), \quad A = \{P_r (\gamma - 1)\}^{-1}$$

and Sutherland's Law has been assumed for the viscosity: $\mu = a^3 \left\{ \frac{1+S}{2+S} \right\}$ and S is a constant.

Crossflow Plane Finite Volume Scheme

The crossflow terms in Eq. (4), assuming that the R derivatives vanish for the problem of conical flow, can be written in integral form as

$$\frac{\partial}{\partial t} \iint_S Q d\bar{x} d\bar{y} + \int (\hat{F} d\bar{y} - \hat{G} d\bar{x}) + 2 \iint \hat{I} d\bar{x} d\bar{y} = 0 \quad (5)$$

where

$$\hat{F} = \bar{F} - \frac{1}{R_e} \bar{F}_v$$

$$\hat{G} = \bar{G} - \frac{1}{R_e} \bar{G}_v$$

$$\hat{I} = \bar{I} - \frac{1}{R_e} \bar{I}_v$$

$$\hat{t} = \frac{\partial}{R} t$$

The coordinate coefficient and metric Ω/R have been absorbed into the unsteady term since only steady state solutions are sought in this paper.

A node centered finite volume scheme is applied to a discretized version of Eq. (5) in the crossflow plane. Figure 1 illustrates the basic scheme. Both the flow variables Q and the residuals are stored at node points. The cell centered fluxes are first computed over each cell of the grid by summing the fluxes across the four individual sides of the cell in a similar fashion to a cell centered scheme. Unlike the cell centered scheme, the flow variables are stored at the endpoints of each side comprising the boundaries of a cell. This has the potential of yielding a more accurate estimate for the flux across the sides of a highly skewed or irregular mesh than that provided by a cell centered scheme. The residual at each node point is then computed by summing the cell centered fluxes of the four cells surrounding that node point. In discretized form, the conical terms of Eq. (5) can be written at a node point i,j as

$$\Delta S \frac{\Delta Q}{\Delta t} + \sum_{\text{cell}=1}^4 \sum_{\text{side}=1}^4 (\hat{F} \Delta \bar{y} - \hat{G} \Delta \bar{x}) + 2 \hat{I} \Delta S = 0 \quad (6)$$

- + Cell center
- Node
- Node residual

Fig. 1 Schematic of Node Centered Finite Volume Scheme

where $\hat{F}$ and $\hat{G}$ are the total side fluxes which include viscous terms, and I represents a forcing function evaluated at the node points. The area ΔS is defined as

$$\Delta S = \sum_{\text{cell}=1}^4 \Delta A$$

and is the sum of the areas of the four individual cells surrounding a node point. The node residuals thus correspond to flux balances on interlaced control volumes, each consisting of a group of four cells.

To include the viscous shear stress and heat flux terms, the velocity derivatives are estimated at the node points using central difference formulas in a computational space. The mesh transformation derivatives needed for the velocity derivatives are computed numerically. At the boundary, one-sided differences are used to estimate the velocity derivatives.

Artificial Dissipation Models and Local Time Stepping

A blend of second and fourth order differences is used for added artificial dissipation. The fourth order differences are added as background dissipation to prevent odd and even point decoupling. The second order differences are added primarily to smooth out oscillations in regions of severe pressure gradients associated with shock waves.

A detailed account of the form of the dissipation can be found in Ref. 1-4. The basic form of the dissipative terms is

$$D_2 Q = \epsilon_2 \left\{ \delta_{\bar{x}} (S_{\bar{x}} \bar{P}_{\bar{x}\bar{x}} \delta_{\bar{x}} Q) + \delta_{\bar{y}} (S_{\bar{y}} \bar{P}_{\bar{y}\bar{y}} \delta_{\bar{y}} Q) \right\} \quad (7)$$

$$D_4 Q = \epsilon_4 \left\{ \delta_{\bar{x}} (S_{\bar{y}} \delta_{\bar{x}}^3 Q) + \delta_{\bar{y}} (S_{\bar{x}} \delta_{\bar{y}}^3 Q) \right\}$$

where $\bar{P}_{\bar{x}\bar{x}}$ and $\bar{P}_{\bar{y}\bar{y}}$ are pressure gradient switches that appear only in the second order dissipation and ϵ_2 and ϵ_4 are dissipation constants. The

magnitude of the dissipation is governed by the scaling coefficients $S_{\bar{x}}$ and $S_{\bar{y}}$. The local time step limit Δt_c , corresponding to a CFL number equal to unity, is evaluated for each cell, and the nodal time step is taken to be the average of the four surrounding local time steps. The magnitude of the dissipation and the behavior of the local time stepping scheme is directly linked to the manner in which the scaling coefficients and local time step are evaluated. For Euler flows, these quantities can be evaluated as described below.

Model A:

$$S_{\bar{x}} = S_{\bar{y}} = \Delta x_i [a + \|Q_N\|]_i + \Delta x_j [a + \|Q_N\|]_j \quad (8)$$

$$\Delta t_c^* = \frac{\Delta A}{\Delta x_i [a + \|Q_N\|]_i + \Delta x_j [a + \|Q_N\|]_j}$$

where a is the speed of sound, Δx is the side length, and Q_N is the normal velocity through the side. Then S_x , S_y and Δt^* at a node are taken to be the average of the four surrounding cells. Hence, the dissipation coefficients are proportional to $\Delta A/\Delta t^*$, and thus to the sum of the maximum wave speeds $\bar{S}$ in the i and j directions multiplied by the mesh intervals. Since the dissipation is scaled according to Δt^* , the steady state solution is independent of the CFL number actually used in the time stepping scheme. This scaling was found to be adequate for the computation of Euler flows and is referred to as an isotropic dissipation when the side lengths of the cells remain the same order of magnitude. For viscous flows, with boundary layer stretching normal to the body, the average side lengths in the two coordinate directions can vary by orders of magnitude. For the dissipation to vanish using Model A, the mesh must be refined in both directions, which is not practical for viscous flows. It would be more desirable to be able to keep the cell size in the i direction, tangential to the body fixed, and refine the cell size normal to the body and through the boundary layer. In the case of viscous flows, a more appropriate anisotropic model for the local time step and dissipation coefficients can be formulated as follows.

Model B: Dissipation scaling,

$$\begin{aligned} S_x &= \Delta x_i [a + \|Q_N\|]_i \\ S_y &= \Delta x_j [a + \|Q_N\|]_j \end{aligned} \quad (9)$$

and the same estimate as before for Δt_C^* . Thus, a separate local length scale is used in each coordinate direction in order to avoid unnecessary dissipation normal to the body and through the boundary layer.

Model B, at first glance, appears to be the anisotropic model used in Refs. 5 and 6 but is really quite dissimilar. The anisotropic models used in Refs. 5 and 6 make the dissipation in the i direction scale with the length of the cell in the j direction unlike the model of Eq. (9). With an appropriate choice of ϵ_2 , the anisotropic model used in Refs. 5 and 6 corresponds to an upwind biasing of the difference formulas. The scaling used in Refs. 5 and 6 could actually result in more dissipation through the boundary layer.

Both of these models could be further modified for viscous flows by an additional scaling that varies with the total Mach number through the boundary layer to further reduce the artificial dissipation in the boundary layer. This procedure relies on the physical dissipation in the boundary layer for stability and was not used in the present study.

In practice, Model B above may lead to instability or too slow a convergence rate. Hence, Models A and B can be combined into a single blended dissipation model using the following functions. If

$$\begin{aligned} \hat{S}_x &= \Delta x_i [a + \|Q_N\|]_i \\ \hat{S}_y &= \Delta x_j [a + \|Q_N\|]_j \end{aligned}$$

then

$$\begin{aligned} S_x &= \hat{S}_x \left[1 + \left(\frac{\hat{S}_y}{\hat{S}_x} \right)^{\alpha_d} \right] \\ S_y &= \hat{S}_y \left[1 + \left(\frac{\hat{S}_x}{\hat{S}_y} \right)^{\alpha_d} \right] \end{aligned} \quad (10)$$

and α_d is a constant in the range from 0 to 1. If $\alpha_d = 1$, Eq. (10) reduces identically to the isotropic Eq. (8) or model A. If $\alpha_d = 0$, Eq. (10) corresponds to Eq. (9) or Model B.

Multi-Stage Integration Scheme

The set of unsteady governing equations can be represented as

$$\frac{\partial Q}{\partial t} + \frac{1}{\Delta S} R_{xy}(Q) + \frac{1}{\Delta S} D_{xy}(Q) = 0 \quad (11)$$

where the operator R_{xy} represents the crossflow finite volume spatial approximation to the residual and D_{xy} represents the added artificial dissipation. A modified fourth order Runge-Kutta scheme is used to integrate the set of ordinary differential equations defined by Eq. (10) in the following manner:

$$Q^{(\ell)} = Q^{(0)} - \alpha_{\ell} \frac{\Delta t}{\Delta S} (R_{xy}^{(\ell-1)} + D_{xy}^{(0)}) \quad (12)$$

where $\alpha_{\ell} = (\frac{1}{4}, \frac{1}{3}, \frac{1}{2}, 1)$ $\ell = 1, 4$

The bracketed superscript refers to the stages of the Runge-Kutta scheme. The dissipative terms are frozen throughout the multi-stage integration scheme. The local time step is taken to be

$$\Delta t = \text{CFL } \Delta t^*$$

In addition, implicit residual smoothing is utilized in the crossflow plane to accelerate convergence. The residual smoothing is applied to alternate stages, or the second and fourth stage, of the Runge-Kutta scheme.

Multi-Grid Scheme

A multigrid method is used to accelerate the convergence of the basic multi-stage time stepping scheme. The idea behind the multigrid scheme is to speed up the evolution process by using a series of coarser grids which introduce larger scales and larger time steps and also require less costly computations. The present multigrid scheme follows the work of Jameson (Refs. 3, 4).

The method begins by first computing one or more time steps on a fine grid denoted as the first grid $m=1$. Successively coarser grids are then generated by eliminating alternate points in each coordinate direction. The updated values obtained for the solution vector after a time step on grid m is then transferred to a coarser grid $m+1$ by simply collecting updated values on coincident points, or

$$Q_{m+1}^{(0)} = \sum_{m,q}^{m+1} C_{mq} Q_m \quad (13)$$

On coarse grids $m > 1$, the multi-stage time stepping scheme is modified by a forcing function R_F so that the solution on the successively coarser grids $m+1$ are driven by the residuals computed on the finer grids m . If Eq. (11) is rewritten in a simplified form as

$$\frac{\partial Q}{\partial t} + \hat{R}(Q) = 0 \quad (14)$$

where $\hat{R}(Q) = \frac{1}{\Delta S} [R_{xy}(Q) + D_{xy}(Q)]$

then the original multi-stage scheme of Eq. (12) is modified on coarser grids $m+1$ as

$$Q_{m+1}^{(2)} = Q_{m+1}^{(0)} - \alpha_{\Delta} \Delta t_{m+1} (\hat{R}_{m+1}^{(1)} + \hat{R}_{F,m+1}) \quad (15)$$

The forcing function $\hat{R}_F$ is obtained by first transferring the residuals from grid m to the next coarser grid $m+1$. This is accomplished by a collection process where the residual on a coarser grid is generated by a weighted sum of its coincident residual and its neighboring eight residuals on a finer grid m . If C_R represents the residual collection process from a finer grid m to a coarser grid $m+1$, then the forcing function R_F can then be represented as

$$\hat{R}_{F,m+1} = C_R^{m+1} \hat{R}_m - \hat{R}_{m+1}^{(0)} \quad (16)$$

where $\hat{R}_{m+1}^{(0)}$ is the corresponding residual calculated from the difference equations on the coarse grid. Thus, in the first stage, the forcing function simply cancels the coarse grid residual and replaces it by the weighted sum of the fine grid residuals. Initially, a simple weighted average of the fine grid residuals was used to

collect residuals in the forcing function as indicated in Fig. 2a. The actual weighting used seemed to be fairly insensitive to the multigrid algorithm. This was true for both Euler and viscous computations for circular cones. When viscous elliptic cone cases were computed, the residual collection had to be modified to include a weighted average of the nodal areas, or

$$C_R^{m+1} \hat{R}_m = \frac{9}{1} (\Delta S) \hat{R}_m / \frac{9}{1} (\Delta S) \quad (17)$$

The metric variations in both directions, tangential and normal to the body, are quite severe for the elliptic cone grid. The multigrid algorithm did not work well for the viscous elliptic cone grid until Eq. (17) was adopted.

After one or more time steps are carried out on a given grid, the process is repeated until the coarsest grid is reached. The accumulated corrections are now transferred upward from each grid to the next finer grid by an interpolation process. The correction transferred up from a given grid includes both the correction calculated on that grid and the sum of the corrections interpolated from coarser grids in the sequence. If Q_{m+1} is the final value on grid $m+1$ resulting from both the upward interpolation from a coarser grid and the time stepping scheme applied on grid $m+1$, then the correction on grid m can be represented as

$$\hat{Q}_m = \bar{Q}_m + \sum_{m+1}^m (I_{m+1} (\hat{Q}_{m+1} - Q_{m+1}^{(0)})) \quad (18)$$

where $\bar{Q}_m$ is the value obtained after the time step on grid m , and I is an interpolation operator.

This multigrid scheme results in a simple saw tooth cycle that is illustrated schematically in Fig. 2. The multigrid algorithm was tested on a

Fig. 2 Schematic of Multigrid Scheme

a variety of test problems for both Euler and Navier-Stokes. The cases selected, both circular and elliptic cone cases at high incidence, represent fairly difficult solutions that will serve as a test bed for the multigrid algorithm.

Euler Computations

Circular Cones at High Incidence

Figure 3 shows the Euler multigrid convergence history for a 20° circular cone at $M_\infty = 2.0$, $\alpha = 25^\circ$ on an 81x50 grid in comparison to the basic multi-stage Runge-Kutta scheme with local time stepping and residual smoothing. Plotted are the decay of the maximum residual versus both Cray-XMP CPU seconds and iterations. Figure 4 shows the (81x50) grid and the resulting isobar solution. A weak crossflow shock occurs on the lee side of the cone with attached flow. The multigrid scheme using three coarse grids converges in five times fewer iterations, while the computational time is less than half the time of the basic scheme. The reduction in the computational time is less than the reduction in the number of iterations because each multigrid cycle requires about twice the work of a Runge-Kutta cycle. This is partly due to the need for an additional residual evaluation on the fine grid, which is about 0.4 the work of a complete Runge-Kutta cycle. The additional residual evaluation takes 0.4 instead of 0.25 the work of a complete cycle because it requires the evaluation of both the time step and the dissipation, and these are held frozen in the four-stage scheme. The total work on the coarser grids is about 0.45 the work of the original Runge Kutta cycle. Some additional overhead is paid for the collection and interpolation processes and shuffling in memory between the main fine grid arrays and coarse grid auxiliary arrays. The multigrid solution in this case is converged (i.e., 5 to 6 orders of magnitude) in 15 sec on a Cray-XMP.

Figure 5 shows the Euler multigrid convergence history for a 10° circular cone at $M_\infty = 2.0$, $\alpha = 25^\circ$. Figure 6 shows the grid (81x50) and the resulting isobar solution. This is a more difficult solution in that a strong crossflow shock occurs on the lee side, resulting in shock induced separation. The multigrid scheme for this solution was essentially converged in about 15 sec CPU time. The reduction in the computational time is about a factor of two.

Figure 7 shows the convergence history for the 10° cone on an 81x50 grid at a hypersonic Mach number of 7.95. The convergence gain with multigrid is similar to the lower supersonic Mach number in reducing computational time by a factor of two. The hypersonic flow case required about twice the computational time in comparison to the supersonic solution. Convergence was achieved in less than 45 sec CPU time. Figure 8a shows the mesh used for this case and 8b shows the computed isobar solution.

Elliptic Cone/Squire Wing

Figure 9 shows the convergence history for an elliptic cone (20° by 1.5°) with a center body on an 81x50 grid at $M_\infty = 2.5$, $\alpha = 10^\circ$. The grid and computed results are shown in a subsequent section where they are compared to the viscous computations and the experimental data of Squire (Ref. 9).

Using the simple weighted average collection of fine mesh residuals, this case could only be successfully computed if the forcing residuals and interpolated values of the variables were slightly underrelaxed. Figure 9 shows the result when the residuals were collected using a nodal area weighted average. This removed the necessity for underrelaxation. Even though the number of iterations were reduced by more than a factor of two, the savings in computational time was only about 20%. The basic multi-stage scheme is very efficient for this case, and with multigrid a converged solution is achieved in less than 15 sec CPU time.

Navier-Stokes Computations

Viscous Grid Generation

A viscous grid was generated by embedding a boundary layer mesh within the existing outer inviscid grid. The inner layer grid is generated using a geometric progression (Ref. 7) as is typically used for turbulent boundary layers, and has the following form in terms of a sheared coordinate Y .

$$\text{Inner Viscous Grid: } Y_{BL} = \Delta Y_{BL_0} \left(\frac{\lambda^{J-2} - 1}{\lambda - 1} \right) \quad (19)$$

where ΔY_{BL_0} is the height of the boundary layer grid cell at the wall, and λ is a constant geometric growth factor for the boundary layer cells. The existing inviscid grid employed in the Euler computations uses conformal mapping when necessary as for the elliptic cone and exponential clustering normal to the body. To embed the viscous grid as a sublayer to the inviscid grid without altering the outer inviscid grid, the following procedure is used. A circumferential inviscid grid line ($J=J_{inv}$) is specified. This specifies the outer normal sheared coordinate (Y_{inv}) that must be matched. In general, the actual value of Y may vary circumferentially. The inviscid grid also yields the cell size ΔY_{inv} that must be matched by the inner viscous grid to obtain a smoothly varying grid normal to the body. This specifies two conditions that must be matched by Eq. 19 in the form,

$$\begin{aligned} \Delta Y_{BL} &= \frac{Y_{INV} (\lambda - 1)}{J_{BL} - 1} \\ \lambda &= \left(\frac{\Delta Y_{INV}}{\Delta Y_{BL}} \right)^{\frac{1}{J_{BL} - 2}} \end{aligned} \quad (20)$$

If the number of points ($J=J_{BL}$) in the inner boundary layer grid is further specified, the solution of these equations yields the constant boundary layer grid growth factor λ and the cell size at the wall (ΔY_{BL}).

A drawback to this procedure is the lack of direct control on the first cell size in the boundary layer. The first cell size can be controlled by the choice of the inviscid grid line J_{inv} in combination with the specified number of points J_{BL} in the inner boundary layer grid. The procedure has the advantage, however, that identical outer inviscid grids can be used for both

Fig. 3 Euler Multigrid Convergence History for a 20° Circular Cone at $M_{\infty} = 2.0$, $\alpha = 25^{\circ}$

Fig. 4 Euler Grid & Computed Isobars for a 20° Circular Cone at $M_{\infty} = 2.0$, $\alpha = 25^{\circ}$

Fig. 6 Euler Grid & Computed Isobars for a 10° Circular Cone at $M_{\infty} = 2.0$, $\alpha = 25^{\circ}$

Fig. 5 Euler Multigrid Convergence History for a 10° Circular Cone at $M_{\infty} = 2.0$, $\alpha = 25^{\circ}$

Fig. 7 Euler Multigrid Convergence History for a 10° Circular Cone at $M_\infty = 7.95$, $\alpha = 12.0^\circ$

Fig. 8 Euler Grid & Computed Isobars for a 10° Circular Cone at $M_\infty = 7.95$, $\alpha = 12.0^\circ$

Euler and Navier-Stokes computations, thus eliminating variations in the basic inviscid grid as a source of differences in comparisons of Euler with Navier-Stokes computations.

Figure 10 shows a fairly extreme example of the grid embedding technique for a simple grid about a 10° cone at $M_\infty = 2.0$, and $\alpha = 0^\circ$. This grid is used in a subsequent section to test the effect of dissipation as the grid is refined in the boundary layer. Figure 10a shows the overall grid comprised of 64 circumferential by 65 normal points. Figure 10b shows an enlargement of the grid near the wall. The first four inviscid grid lines are replaced by 20 grid lines in the vicinity of the wall. This results in an extremely stretched grid with an aspect ratio of about 760 at the first cell. The dashed lines represent the two inviscid grid lines replaced by the embedded viscous grid. The outer inviscid grid remains unchanged.

Fig. 9 Euler Multigrid Convergence History for a (20° x 1.5°) Elliptic Cone at $M_\infty = 2.5$, $\alpha = 10^\circ$

Fig. 10 Example of Viscous Grid Embedding

It should be noted that the above conditions result in a smoothly varying grid but do not necessarily assure continuous derivatives across the grid interface. The effect of this grid discontinuity on the convergence rate is uncertain.

Effects of Artificial Dissipation

Before examining the efficiency of Navier-Stokes solutions with multigrid, it is necessary to examine the effect of the two dissipation models reflected by Eq. 8 and 9 and combined in Eq. 10. For this purpose, the flow past a 10° circular cone at $M=2.0$ and $\alpha=0^\circ$ was calculated on a (65×64) grid using both models. At $R_e = 1 \times 10^5$, the computation shows about 18 to 20 points in the laminar boundary layer on the basic grid. Figure 11 shows results on the same grid at $R_e = \infty$. In this limit, the solution should approach an Euler result with a slip line at the wall. The results at $R_e = \infty$ were achieved by inserting a zero coefficient for the factor R_e^{-1} in the viscous terms. Figure 11 shows the two sets of results using the artificial dissipation Model A (Eq. 8) and the Model B (Eq. 9) for $R_e = \infty$ in comparison to the boundary layer computed on the same grid at $R_e = 1 \times 10^5$.

The left side of Fig. 11 plots the Mach number profile as a function of the point index, and the right hand side plots the profile as a function of the physical distance from the wall nondimensionalized by the scale length of the body. Model A introduces larger numerical dissipation than Model B, smearing the wall layer to about 9 or 10 points. Model B, on the other hand, smears the layer by only 2 or 3 points but leads to a larger overshoot in the Mach number profile. It should be mentioned that both computations were run with values of the dissipation constants ϵ_2, ϵ_4 equal to $-0.50, 1/64$, respectively. The rate of convergence to a steady state exhibited by Model B suffers due to the lack of both artificial and physical dissipation arising in the overshoot in the profile. Hence, the artificial dissipation Model B exhibits higher accuracy, but the convergence rate of the method suffers considerably.

To illustrate further the accuracy of the two dissipation models, for the test case of a 10° cone at $R_e = 1 \times 10^5$ and $M_\infty = 2.0$, $\alpha = 0^\circ$, a variety of grids was used to determine the values of the normal velocity derivative (dQ/dn) at the wall. dQ/dn was chosen to be a parameter indicative of skin friction. The total number of grid points was kept constant as was the circumferential grid. By altering the embedded viscous grid, the cell aspect ratio at the wall could be changed by almost two orders of magnitude from 12 to 760. The cell height at the wall varied from 7×10^{-4} to 1×10^{-8} . The highest aspect ratio grid resulted in twice as many points in the boundary layer. Figure 12 summarizes the results for these two extreme cases. Figure 12a shows the percent change in dQ/dn at the wall for the crudest viscous grid with a wall cell aspect ratio of only 12, plotted against various values of the fourth order dissipation coefficient ϵ_4 and using both the isotropic dissipation model A and the directionally scaled model B. Figure 12b shows the same type of plot but for the highest wall aspect ratio grid used of 760. The dissipation coefficient ϵ_4 was varied from $1/32$ to $1/256$. The percent change in dQ/dn is based on an arbitrary number assumed to be near the final converged value. The isotropic Model A shows a maximum error of about 3.5% for the aspect ratio 12 grid. Model B shows less than 1% error for all values of the coefficient ϵ_4 . Both models converge to the same answer as ϵ_4 is made smaller. Interestingly, the Model A shows larger errors of about 7% on the higher aspect ratio grid even though this grid yields more points in the boundary layer. Model B also shows a slightly increased maximum error of about 2%. The cell growth rate was about 1.1 for the low aspect ratio grid and 1.35 for the high aspect ratio grid. The larger errors on the higher aspect ratio are introduced by the more severe metric scale length variations included in the conservation form of the dissipation derivatives of Eq. (7).

On both grids, both dissipation models converged to within 1% for the value of dQ/dn at the wall. In summary, both dissipation models were capable of computing dQ/dn at the wall, but Model A consistently exhibited larger errors and showed higher sensitivity to the dissipation coefficient ϵ_4 . The two models agreed better on the less highly stretched grid which resulted in only about 10 points in the boundary layer. Better accuracy with the present form of dissipation model may perhaps be more easily achieved on less highly stretched grids using more points than on very severely stretched grids which inherently introduce more errors. Figure 13 shows a comparison of the boundary layer profiles using Model B for the two dissimilar grids of Figure 12. Good agreement is achieved on the two grids.

In all the computations to follow except for the hypersonic cone, $\alpha_d = 0.50$ was used. This effects convergence only slightly but also insures that the boundary layer will become resolved as the grid becomes finer normal to the wall.

It should also be mentioned that numerical tests indicate that the basic scheme exhibits a degradation in convergence when the spacing normal to the wall $\Delta z_{wall} R_e \leq 1$ independent of the cell aspect ratio or grid stretching.

Fig. 11 Effect of Artificial Dissipation Models on Boundary Layer Mach No. Profiles for a 10° Circular Cone at $M_\infty = 2.0$, $\alpha = 0^\circ$, $Re = \infty$

Fig. 12 Sensitivity of DQ/Dn at the Wall to Dissipation Model, Grid Aspect Ratio

Fig. 13 Comparison of Computed Boundary Layer Mach No. Profiles for the $AR = 12$ & $AR = 700$ Grids

Circular Cones at High Incidence

Figures 14 and 16 show the multigrid convergence history in comparison with the basic scheme for laminar Navier-Stokes solutions at $Re = 1.0 \times 10^5$ for the two previous Euler cases, the 20° and 10° circular cones, respectively, at $M_\infty = 2.0$, $\alpha = 25^\circ$. For these solutions, the outer inviscid grid remained unchanged with respect to the Euler solutions, and 16 points were added within a given inviscid grid line $J = \text{constant}$ according to the boundary layer stretching Eq. (19), resulting in a finer (81x66) grid.

For the 20° cone, Fig. 14 shows a reduction in computational time by more than a factor of two. Convergence is achieved in about 50 sec with the multigrid algorithm. Figure 15 shows three of the computed Mach number boundary layer profiles on the windward, side, and leeward body surface. The viscous grids were designed to yield 15 to 20 boundary layer points. The spacing at the wall for

this case varied from a maximum of 7×10^{-5} to a minimum of 3×10^{-5} resulting in a maximum cell aspect ratio of 408. Figure 16 shows better than a factor of three reduction in computational time for the 10° cone with convergence also in about 50 sec CPU time. Figure 17 shows the computed boundary layer profiles for this case. The spacing at the wall for the 10° cone varied from a maximum of 8×10^{-5} to a minimum of 2×10^{-5} resulting in a maximum cell aspect ratio of 310. A large separated region occurs in the leeward plane for this case. Figure 18 shows an overall comparison for both cones of the crossflow velocity vectors for both Euler and Navier-Stokes computations. For the 20° cone, the Euler solution shows attached flow. The laminar Navier-Stokes solution shows a small separation region. The Euler solution for the 10° cone shows a shock vorticity induced separation. The Navier-Stokes solution shows a more complex separated flow pattern including primary, secondary, and tertiary vortices.

Fig. 14 Navier-Stokes Multigrid Convergence History for a 20° Circular Cone at $M_\infty = 2.0$, $\alpha = 25^\circ$, $Re = 1.0 \times 10^5$

Fig. 15 Computed Mach No. Boundary Layer Profiles for the 20° Circular Cone at $M_\infty = 2.0$, $\alpha = 2.0^\circ$, $Re = 1.0 \times 10^5$

Fig. 16 Navier-Stokes Multigrid Convergence History for a 10° Circular Cone
at $M_\infty = 2.0$, $\alpha = 2.5^\circ$, $Re = 1.0 \times 10^5$

Fig. 17 Computed Mach No. Boundary Layer Profiles for the 10° Circular Cone
at $M_\infty = 2.0$, $\alpha = 2.5^\circ$, $Re = 1.0 \times 10^5$

Figure 19 shows the convergence history for a laminar viscous hypersonic flow solution for a 10° cone on an (81x64) grid at $M_\infty = 7.95$, $\alpha = 12^\circ$. Unlike all previous cases which were adiabatic, this solution was computed using a fixed wall temperature. The hypersonic viscous flow solution takes two to three times more computational time in comparison to the lower Mach number case for this cone. The multigrid algorithm reduces computational time by about a factor of two in a similar fashion to the lower Mach number solutions. This computation, like all previous cases, was started from freestream conditions on a single grid and by bringing the wall to rest. Figure 19 was generated using the dissipation Model A because any use of Model B caused divergence near the strong bow shock. A composite dissipation model was then adopted which implemented the more accurate Model B near the wall for the first 20 points. A transition layer was used which linearly transgressed from the scaling coefficients of Model

B to Model A from point 20 to 40. Model A was then implemented from points 40 to the outer boundary. The composite dissipation scaling implementation allows for accuracy within the boundary layer and enough dissipation in the outer layer to be able to capture the bow shock. The implementation of the composite dissipation model causes some degradation in convergence, but it was deemed necessary to be able to accurately predict the wall heat transfer. Figure 20 shows a sampling of the temperature profiles computed on the wind, side, and lee surfaces of this cone. The cell height at the wall varied from 3×10^{-4} to 7×10^{-5} . From the shape of the temperature profile, it is immediately obvious that difficulty in predicting heat transfer will result if accurate grid resolution is not used, particularly on the windward side of the cone. A maximum or peak temperature occurs at less than 1% of the body scale. Figure 21a shows the computed isobars and 21b shows the computed entropy contours. The

Fig. 18 Comparison of Euler & Navier-Stokes Crossflow Velocity Vectors for 10° & 20° Circular Cones at $M_\infty = 2.0$, $\alpha = 25^\circ$

leeside boundary layer separates at this incidence as indicated by the entropy contours. Figure 22 shows a comparison of the computed distribution of circumferential heat transfer ratio with the experimental data of Tracy (Ref. 8). The agreement is good considering that the computed results are conical and the real boundary layer on a cone is three-dimensional. The slight discrepancy in the windward plane needs further study.

Squire Wing/Elliptic Cone

Figure 23 shows a comparison of the multigrid convergence with the basic multi-stage scheme for the viscous laminar flow at $M_\infty = 2.5$, $\alpha = 10^\circ$ and $Re = 2.5 \times 10^5$. In this case, the multigrid solution took almost nine times fewer iterations to converge in comparison to the basic scheme, resulting in a reduction in computational time by better than a factor of three. Convergence was

achieved in about 90 sec with the multigrid algorithm. Figure 24 shows three of the computed boundary layer Mach number profiles with 15 to 20 grid points being maintained in the boundary layer region. The cell height at the wall varied from 6×10^{-4} to 3×10^{-5} with a maximum cell aspect ratio of 68. At the leading edge, the boundary layer is less than 1% of the body scale. Weighting the collected residuals with the nodal areas was essential for this case.

Figure 25 shows a comparison of the Euler (81x50) and Navier-Stokes (81x64) grids used for the Squire Wing of Ref. 9, which consists of an elliptic cone ($20^\circ/1.5^\circ$) with a centerbody. Figure 26 shows both the computed Euler and laminar Navier-Stokes isobars at $M_\infty = 2.5$ and $\alpha = 10^\circ$. The Euler solution has a strong crossflow shock on the lee side of the wing leading to a small region of shock induced separated flow. The Navier-Stokes

Fig. 19 Navier-Stokes Multigrid Convergence History for a 10° Circular Cone at $M_\infty = 7.95$, $\alpha = 12^\circ$, $Re = 3.6 \times 10^5$

Fig. 20 Computed Temperature Profiles for the 10° Circular Cone at $M_\infty = 7.95$, $\alpha = 12^\circ$, $Re = 3.6 \times 10^5$

Fig. 21 Computed Navier-Stokes Isobars & Entropy Contours for a 10° Circular Cone at $M_\infty = 7.95$, $\alpha = 12^\circ$, $Re = 3.6 \times 10^5$

Fig. 22 Comparison of Computed Circumferential Heat Transfer with Experimental Data at $M_\infty = 7.95$, $\alpha = 12^\circ$, $Re = 3.6 \times 10^5$ for the 10° Circular Cone

Fig. 23 Navier-Stokes Multigrid Convergence History for the Squire Wing/Elliptic Cone ($20^\circ \times 1.5^\circ$) at $M_\infty = 2.5$, $\alpha = 10^\circ$, $Re = 1.0 \times 10^5$ for the Circular Cone

Fig. 24 Computed Mach No. Boundary Layer Profiles for the Squire Wing/Elliptic Cone ($20^\circ \times 1.5^\circ$) at $M_\infty = 2.5$, $\alpha = 10^\circ$, $Re = 1.0 \times 10^5$ for the Circular Cone

Fig. 25 Comparison of Euler & Navier-Stokes Grids for the Squire Wing/Elliptic Cone ($20^\circ \times 1.5^\circ$) at $M_\infty = 2.5$, $\alpha = 10^\circ$

Fig. 26 Comparison of Euler & Navier-Stokes Isobar Solutions for the Squire Wing/Elliptic Cone ($20^\circ \times 1.5^\circ$) at $M_\infty = 2.5$, $\alpha = 10^\circ$

Fig. 28 Comparison of Euler & Navier-Stokes Surface Pressure Distributions to the Experimental Data of Squire Wing at $M_\infty = 2.5$, $\alpha = 10^\circ$

solution shows a large separated region on the lee surface. Figure 27 further shows the computed crossflow streamline patterns for both solutions. The Navier-Stokes solution also has a secondary separation region. Finally, Fig. 28 shows a comparison of Euler and Navier-Stokes surface pressures with the experimental data of Squire (Ref. 9). Very good agreement is exhibited between the Navier-Stokes solution and the experimental data.

Conclusions

A multigrid algorithm has been applied successfully to the computation of both inviscid and viscous supersonic/hypersonic conical flows. With the exception of one case, the multigrid algorithm reduced the computational time required

by at least a factor of two. It actually performed somewhat better for viscous flows where the basic multi-stage Runge-Kutta scheme slows down due to viscous grid stretching.

Two dissipation models were studied for accuracy. The isotropic dissipation model yielded higher errors in comparison to the anisotropic model, which scaled the dissipation with its respective mesh scale lengths. The anisotropic model is less robust and requires more computational time for convergence. It was found that with both models larger errors were introduced on severely stretched grids than on mildly stretched grids. Good agreement with experimental data was obtained both for a 10° circular cone at $M = 7.95$, $\alpha = 10^\circ$, and for the Squire wing at $M_\infty = 2.50$, $\alpha = 10^\circ$.

Fig. 27 Comparison of Euler & Navier-Stokes Crossflow Streamline Patterns for the Squire Wing/Elliptic Cone ($20^\circ \times 1.5^\circ$) at $M_\infty = 2.5$, $\alpha = 10^\circ$

References

- 1) Siclari, M.J. and DelGuidice, P., "A Hybrid Finite Volume Approach to Euler Solutions for Supersonic Flows," AIAA Paper No. 88-0225, presented at AIAA 26th Aerospace Sciences Meeting, Reno, NV, Jan. 1988.
- 2) Siclari, M.J., "Three-Dimensional Hybrid Finite Volume Solutions to the Euler Equations for Supersonic/Hypersonic Aircraft," AIAA Paper No. 89-0281, presented at the AIAA 27th Aerospace Sciences Meeting, Reno, NV, Jan. 1989.
- 3) Jameson, A., "A Vertex Based Multigrid Algorithm for Three-Dimensional Compressible Flow Calculations," presented at the ASME Symp on Numerical Methods for Compressible Flow, Anaheim, CA, Dec. 1986.
- 4) Volpe, G., Siclari, M.J., and Jameson, A., "A New Multigrid Euler Method for Fighter-Type Configurations," AIAA Paper No. 87-1160, presented at the AIAA 8th Computational Fluid Dynamics Conference, Honolulu, HI, June 1987.
- 5) Martinelli, L., Jameson, A., and Grasso, F., "A Multigrid Method for the Navier-Stokes Equations," AIAA Paper 86-0208, presented at the AIAA 24th Aerospace Sciences Meeting, Reno, NV, Jan. 1986.
- 6) Swanson, R.C., and Turkel, E., "Artificial Dissipation and Central Difference Schemes for the Euler and Navier-Stokes Equations," AIAA Paper No. 87-1107-CP, presented at the AIAA 8th Computational Fluid Dynamics Conference, Honolulu, HI, June 1987.
- 7) Cebeci, T. and Bradshaw, P., "Momentum Transfer in Boundary Layers," McGraw-Hill, 1977.
- 8) Tracy, R.R., "Hypersonic Flow Over a Yawed Cone," California Institute of Technology, Memorandum No. 69, Aug. 1963.
- 9) Squire, L.C., "Leading-Edge Separation and Crossflow Shocks on Delta Wings," AIAA Journal, Vol. 23, No. 9, March 1985, pp. 321-325.