Copyright © 1998, by the International Council of the Aeronautical Sciences (ICAS)
and the American Institute of Aeronautics and Astronautics, Inc.

A98-31515 ICAS-98-2,7,2

A FAST AND ACCURATE METHOD FOR SOLVING THE NAVIER-STOKES EQUATIONS

Robert W. MacCormack
Department of Aeronautics and Astronautics
Stanford University
Stanford, CA 94305–4035
USA

Abstract

An implicit procedure for minimizing the error introduced by approximate factorization is presented for solving the unsteady Euler or Navier-Stokes equations. Numerical viscosity is added directly to the flow equations by augmenting the natural viscosity, similar in concept to that of an eddy viscosity. This approach preserves the frame independence of the of the governing equations even when solved in arbitrary curvilinear coordinate systems. Convergence rates of approximately 0.8 are achieved upon application for high Reynold's number supersonic flow past a blunt body. Converged solutions, to at least engineering accuracy, are obtained in about 100 time steps on adapted grids.

Introduction

Implicit numerical methods for solving the equations of compressible viscous flow have been developed and used widely during the past quarter century. These methods first approximate the governing partial differential flow equations with a matrix equation and then replace, by approximate factorization, the original matrix with a sequence of efficiently invertible matrix factors. In addition, these methods often add artificial dissipation to control numerical instability. The difference between the original equations to be solved and those actually solved numerically represents computation error. This error is introduced through

1 discretization of derivatives by finite difference quotients,

2 approximate factorization of the original matrix equation, and

3 the addition of artificial dissipation.

These methods typically operate at time steps far smaller than are required by time accuracy considerations and often require thousands of time steps for convergence to even engineering accuracy. Errors introduced by approximate factorization or decomposition are principally responsible for slow convergence. Relatively small time steps are required to contain these errors introduced during the initial transients of the flow and then many time steps must be taken to rid the solution of the garbage introduced. This causes slow convergence. On the other hand, discretization error and artificial dissipation are principally responsible for degrading the accuracy of the numerical solution.

The strategy for obtaining fast and accurate solutions to the equations governing compressible inviscid or viscous flow is to first strip away, as much as possible, sources of numerical error and then reintroduce only that which is necessary to control instability without destroying the physics of the governing equations. The strategy is given as follows.

1 Discretize the derivatives by finite difference approximations of high order accuracy. — In the calculations to be described, the invisid spacial derivatives were approximated by third order accurate upwind biased difference quotients in subsonic flow regions and second order accuarate upwind spacial differences in regions of supersonic flow. Central second order accurate approximations were used for the viscous terms.

2 Minimize the error of approximate factorization. — The numerical procedure for minimizing approximate factorization error is described by MacCormack[1]. It consists of two parts: an approximate factorization, used earlier by Bardina and Lombard[2], with decomposition error much smaller than that used in the Briley and McDonald[3] or Beam and Warming[4] methods, and an additional procedure to eliminate further the approximate factorization error that is introduced.

3 Add frame independent artificial viscous stress and dissipation of minimal strength. — An approach to control numerical instability, similar in concept to that of an eddy viscosity used to include the effects of turbulent mixing in the Reynolds Averaged Navier-Stokes equations, is used to add numerical viscosity directly to the flow equations by augmenting the natural viscosity. The Navier-Stokes stress tensor is itself frame independent and hence this approach preserves frame independence, believed to be important for a physical balance of the governing equations when solved numerically in arbitrary curvilinear coordinate systems.

Copyright © 1998 by ICAS and AIAA. All rights reserved

Copyright © 1998, by the International Council of the Aeronautical Sciences (ICAS)
and the American Institute of Aeronautics and Astronautics, Inc.

The procedure is both fast and accurate because of the minimization of numerical error. This procedure has been applied to solve two difficult benchmark flow problems. One presented as a challenge by Pulliam[5] for Euler solvers and the other proposed by Blottner[6] as a test for Navier-Stokes solvers.

Implicit Matrix Equations

Consider the unsteady equations of compressible viscous flow in two dimensions represented in general coordinates ξ and η as follows.

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} = 0 \tag{1}$$

A finite difference/volume equation, written in *delta* law form, approximating the above equation, is given by Eq. (2) below (2)

$$\left\{ I + \Delta t \left(\frac{D_{-}\cdot}{\Delta \xi} \bar{A}^n{}_{i+1/2,j} + \frac{D_{-}\cdot}{\Delta \eta} \bar{B}^n{}_{i,j+1/2} \right) \right\} \delta U^{n+1}_{i,j}$$
$$= -\Delta t \left(\frac{F^n_{i+1/2,j} - F^n_{i-1/2,j}}{\Delta \xi} + \frac{G^n_{i,j+1/2} - G^n_{i,j-1/2}}{\Delta \eta} \right)$$

where $\bar{A}$ and $\bar{B}$ are the matrix jacobians of the flux vectors F and G with respect to state vector U, respectively, and the dots, $\cdot$, appearing in the equation indicate that the difference operators operate on all factors to the right. Upon application of the difference operators an equation of the following form is usually obtained.

$$\mathbf{B_{i,j}}\delta U^{n+1}_{i,j+1} + \mathbf{A_{i,j}}\delta U^{n+1}_{i,j} + \mathbf{C_{i,j}}\delta U^{n+1}_{i,j-1} + \mathbf{D_{i,j}}\delta U^{n+1}_{i+1,j} + \mathbf{E_{i,j}}\delta U^{n+1}_{i-1,j} = \Delta U^n_{i,j} \tag{3}$$

where $\mathbf{A_{i,j}}$, $\mathbf{B_{i,j}}$, $\mathbf{C_{i,j}}$, $\mathbf{D_{i,j}}$ and $\mathbf{E_{i,j}}$ are block matrix elements and

$$\Delta U^n_{i,j} = -\Delta t \left(\frac{F^n_{i+1/2,j} - F^n_{i-1/2,j}}{\Delta \xi} + \frac{G^n_{i,j+1/2} - G^n_{i,j-1/2}}{\Delta \eta} \right)$$

Eq. (3) can also be expressed in full matrix form.

$$\mathbf{M} \cdot [\delta \mathbf{U}] = [\mathbf{\Delta U}]$$

where $\mathbf{M} =$

$$\begin{bmatrix}
x & x & \cdot & \cdot & x & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
x & x & x & \cdot & \cdot & x & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & x & x & x & \cdot & \cdot & x & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & x & x & x & \cdot & \cdot & x & \cdot & \cdot & \cdot & \cdot & \cdot \\
x & \cdot & \cdot & x & x & x & \cdot & \cdot & x & \cdot & \cdot & \cdot & \cdot \\
\cdot & x & \cdot & \cdot & x & x & x & \cdot & \cdot & x & \cdot & \cdot & \cdot \\
\cdot & \cdot & \bar{\mathbf{D}} & \cdot & \cdot & \bar{\mathbf{B}} & \bar{\mathbf{A}} & \bar{\mathbf{C}} & \cdot & \cdot & \bar{\mathbf{E}} & \cdot & \cdot \\
\cdot & \cdot & \cdot & x & \cdot & \cdot & x & x & x & \cdot & \cdot & x & \cdot \\
\cdot & \cdot & \cdot & \cdot & x & \cdot & \cdot & x & x & x & \cdot & \cdot & x \\
\cdot & \cdot & \cdot & \cdot & \cdot & x & \cdot & \cdot & x & x & x & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & x & \cdot & \cdot & x & x & x & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & x & \cdot & \cdot & x & x & x \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & x & \cdot & \cdot & x & x
\end{bmatrix},$$

And the solution change vectors ΔU, calculated explicitly, and δU, calculated implicitly, are given by

$$[\delta \mathbf{U}] = \begin{bmatrix} \cdot \\ \cdot \\ \delta U_{i-1,j} \\ \cdot \\ \cdot \\ \delta U_{i,j-1} \\ \delta U_{i,j} \\ \delta U_{i,j+1} \\ \cdot \\ \cdot \\ \delta U_{i+1,j} \\ \cdot \\ \cdot \end{bmatrix} \quad \text{and} \quad [\mathbf{\Delta U}] = \begin{bmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \Delta U_{i,j} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

Matrix $\mathbf{M}$ is of size $[(n\ x\ n)x(n\ x\ n)]$ and vectors $[\delta \mathbf{U}]$ and $[\mathbf{\Delta U}]$ are of length $(n\ x\ n)$ on an $(n\ x\ n)$ mesh.

The block element matrices $\bar{\mathbf{A}}_{\mathbf{i,j}}$, $\bar{\mathbf{B}}_{\mathbf{i,j}}$ $\bar{\mathbf{C}}_{\mathbf{i,j}}$, $\bar{\mathbf{D}}_{\mathbf{i,j}}$, and $\bar{\mathbf{E}}_{\mathbf{i,j}}$ are defined below with $\alpha = 1.0$.

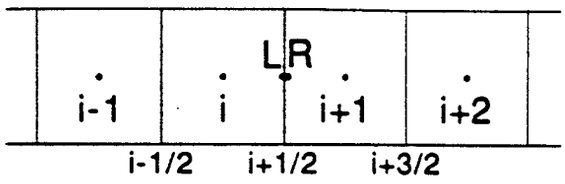
$$\begin{aligned}
\bar{\mathbf{A}}_{\mathbf{i,j}} &= I + \alpha \frac{\Delta t}{\Delta x} \left(\bar{A}^n_{+i+1/2,j} - \bar{A}^n_{-i-1/2,j} \right) + \alpha \frac{\Delta t}{\Delta y} \left(\bar{B}^n_{+i,j+1/2} - \bar{B}^n_{-i,j-1/2} \right) \\
\bar{\mathbf{B}}_{\mathbf{i,j}} &= +\alpha \frac{\Delta t}{\Delta y} \bar{B}^n_{-i,j+1/2} \\
\bar{\mathbf{C}}_{\mathbf{i,j}} &= -\alpha \frac{\Delta t}{\Delta y} \bar{B}^n_{+i,j-1/2} \\
\bar{\mathbf{D}}_{\mathbf{i,j}} &= +\alpha \frac{\Delta t}{\Delta x} \bar{A}^n_{-i+1/2,j} \\
\bar{\mathbf{E}}_{\mathbf{i,j}} &= -\alpha \frac{\Delta t}{\Delta x} \bar{A}^n_{+i-1/2,j}
\end{aligned} \tag{4}$$

The *bars* appearing on the matrix elements indicate that the Modified-Steger-Warming method is being used to approximate these jacobian matrices using arithmetically averaged data from both sides of the flux surface. The Roe method, using geometrically averaged data, could be used as well. The subscripts appearing on the jacobian matrices, for example $\bar{A}^n_{+i-1/2,j}$, indicate the sign of the split matrix and the flux surface location. For this example, the split jacobian matrix contains only non-negative eigenvalues and the flux surface lies midway between mesh points (i,j) and $(i+1,j)$.

Copyright © 1998, by the International Council of the Aeronautical Sciences (ICAS)
and the American Institute of Aeronautics and Astronautics, Inc.

Modified-Steger-Warming Flux Splitting

The split fluxes can be written in terms of U_L and U_R defined at, for example, coordinate surface $\xi_{i+1/2}$.



(*i*) The Steger-Warming split vector flux is

$$F^{(S-W)}_{i+1/2} = A_{+L}U_L + A_{-R}U_R,$$

(*ii*) The Modified-Steger-Warming split vector flux is

$$F^{(M-S-W)}_{i+1/2} = \bar{A}_{+i+1/2}U_L + \bar{A}_{-i+1/2}U_R$$

(*iii*) The Roe split difference vector flux is

$$F^{(Roe)}_{i+1/2} = \frac{F_L + F_R}{2} - \frac{1}{2}\left|\hat{A}_{i+1/2}\right|(U_R - U_L)$$

The split jacobians are evaluated also using the state vectors U_L and U_R. For the Modified-Steger-Warming method

$$\bar{A}_{i+1/2} = \bar{A}(\bar{U}_{i+1/2}), \text{ where } \bar{U}_{i+1/2} = \frac{U_L + U_R}{2}$$

Similarly, the Roe method uses a geometric averaging of the values U_L and U_R to form $\hat{A}_{i+1/2}$. Three appoximations for U_L and U_R follow.

(1) First order upwind:

$$U_L = U_i$$
$$U_R = U_{i+1}$$

(2) Second order upwind:

$$U_L = \frac{1}{2}(3U_i - U_{i-1})$$
$$U_R = \frac{1}{2}(3U_{i+1} - U_{i+2})$$

(3) Third order upwind biased:

$$U_L = \frac{1}{8}(3U_{i+1} + 6U_i - U_{i-1})$$
$$U_R = \frac{1}{8}(3U_i + 6U_{i+1} - U_{i+2})$$

In the present computations, second order upwind approximation was used in supersonic regions, third order upwind biased approximation was used in subsonic regions and first order upwind approximation was used, as will be described, just ahead of shock waves.

Approximate Factorization (AF)

The standard approach to solve Eq.(2) is approximate factorization, which when applied yields

$$\left\{I + \Delta t\frac{D_-\cdot}{\Delta\xi}\bar{A}^n{}_{i+1/2,j}\right\}\left\{I + \frac{D_-\cdot}{\Delta\eta}\bar{B}^n{}_{i,j+1/2}\right\}\delta U^{n+1}_{i,j} = -\Delta U_{i,j}$$

The matrix M is factored by

$$\mathbf{M} \simeq \mathbf{M}_\xi \cdot \mathbf{M}_\eta \qquad (5)$$

These matrix factors are of form

$\mathbf{M}_\xi =$

$$\begin{bmatrix} x & \cdot & x & \cdot & \cdot & \cdot & \cdot \\ \cdot & x & \cdot & x & \cdot & \cdot & \cdot \\ x & \cdot & x & \cdot & x & \cdot & \cdot \\ \cdot & \bar{\mathbf{D}} & \cdot & \bar{\mathbf{A}}_\xi & \cdot & \bar{\mathbf{E}} & \cdot \\ \cdot & \cdot & x & \cdot & x & \cdot & x \\ \cdot & \cdot & \cdot & x & \cdot & x & \cdot \\ \cdot & \cdot & \cdot & \cdot & x & \cdot & x \end{bmatrix}$$

and $\mathbf{M}_\eta =$

$$\begin{bmatrix} x & x & \cdot & \cdot & \cdot & \cdot & \cdot \\ x & x & x & \cdot & \cdot & \cdot & \cdot \\ \cdot & x & x & x & \cdot & \cdot & \cdot \\ \cdot & \cdot & \bar{\mathbf{B}} & \bar{\mathbf{A}}_\eta & \bar{\mathbf{C}} & \cdot & \cdot \\ \cdot & \cdot & \cdot & x & x & x & \cdot \\ \cdot & \cdot & \cdot & \cdot & x & x & x \\ \cdot & \cdot & \cdot & \cdot & \cdot & x & x \end{bmatrix}$$

Each factor represent a block tridiagonal matrix. During matrix inversion, each need to be of only dimension (n x n) when solved *line by line* through each mesh direction. The matrix elements $\bar{\mathbf{B}}_{i,j}$,$\bar{\mathbf{C}}_{i,j}$, $\bar{\mathbf{D}}_{i,j}$ and $\bar{\mathbf{E}}_{i,j}$ are defined as before in Eq.(4) and

$$\bar{\mathbf{A}}_{\xi i,j} = I + \frac{\Delta t}{\Delta x}\left(\bar{A}^n_{+i+1/2,j} - \bar{A}^n_{-i-1/2,j}\right)$$
$$\bar{\mathbf{A}}_{\eta i,j} = I + \frac{\Delta t}{\Delta y}\left(\bar{B}^n_{+i,j+1/2} - \bar{B}^n_{-i,j-1/2}\right)$$

Unfortunately, upon matrix multiplication of the two factors in Eq.(5), none of the original matrix elements are returned exactly and some *zero* elements of M become significantly *nonzero*. This factorization error reduces accuracy and slows algorithm convergence.

Modified Approximate Factorization (MAF)

The approximate factorization procedure can be modified to significantly reduce the adverse effects of decomposition error. The modified procedure has the property of Stone's Strongly Implicit Method (SIP)[7] for matrix decomposition of returning exactly the original *nonzero* elements of the matrix M upon factor multiplication, although some originally *zero* elements become *nonzero*. This modification was used by Bardina and Lombard[2] in 1987, which they called their Diagonally Dominant Alternating Direction Implicit

Copyright © 1998, by the International Council of the Aeronautical Sciences (ICAS)
and the American Institute of Aeronautics and Astronautics, Inc.

(DDADI) procedure. The basic idea of the diagonally dominant procedure also appears much earlier in the literature[8,9]. It is apparently rediscovered every decade or so and then, unfortunately, forgotten by the computational community. When applied to the matrix M

$$\mathbf{M} \simeq \mathbf{M}'_\xi \cdot [\mathbf{D}]^{-1} \cdot \mathbf{M}'_\eta \qquad (6)$$

with $\mathbf{M}'_\xi =$

$$\begin{bmatrix} x & \cdot & x & \cdot & \cdot & \cdot & \cdot \\ \cdot & x & \cdot & x & \cdot & \cdot & \cdot \\ x & \cdot & x & \cdot & x & \cdot & \cdot \\ \cdot & \bar{\mathbf{D}} & \cdot & \bar{\mathbf{A}} & \cdot & \bar{\mathbf{E}} & \cdot \\ \cdot & \cdot & x & \cdot & x & \cdot & x \\ \cdot & \cdot & \cdot & x & \cdot & x & \cdot \\ \cdot & \cdot & \cdot & \cdot & x & \cdot & x \end{bmatrix}$$

$\mathbf{M}'_\eta =$

$$\begin{bmatrix} x & x & \cdot & \cdot & \cdot & \cdot & \cdot \\ x & x & x & \cdot & \cdot & \cdot & \cdot \\ \cdot & x & x & x & \cdot & \cdot & \cdot \\ \cdot & \cdot & \bar{\mathbf{B}} & \bar{\mathbf{A}} & \bar{\mathbf{C}} & \cdot & \cdot \\ \cdot & \cdot & \cdot & x & x & x & \cdot \\ \cdot & \cdot & \cdot & \cdot & x & x & x \\ \cdot & \cdot & \cdot & \cdot & \cdot & x & x \end{bmatrix}$$

and diagonal matrix

$\mathbf{D} =$

$$\begin{bmatrix} x & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & x & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & x & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \bar{\mathbf{A}} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & x & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & x & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & x \end{bmatrix}$$

The block matrix elements appearing above are again defined by Eq.(4) with the exception that the parameter α has now been set to 2.

In three dimensions M can be decompositioned by

$$\mathbf{M} \simeq \mathbf{M}'_\xi \cdot [\mathbf{D}]^{-1} \cdot \mathbf{M}'_\eta \cdot [\mathbf{D}]^{-1} \cdot \mathbf{M}'_\zeta$$

Although the original *nonzero* elements of matrix *M* are returned exactly by the above MAF decomposition procedure, other formerly *zero* elements are disturbed and can contribute to both error and reduced convergence speed. The following section describes an iterative procedure for eliminating or further reducing this remaining decomposition error.

Removal of Decomposition Error

Eq.(6) can be expanded as follows.

$$\mathbf{M} \simeq \mathbf{M}'_\xi \cdot [\mathbf{D}]^{-1} \cdot \mathbf{M}'_\eta = \mathbf{M} + \mathbf{P}$$

The equation actually solved, instead of Eq.(3), is

$$\mathbf{M} \cdot [\delta \mathbf{U}] + \mathbf{P} \cdot [\delta \mathbf{U}] = [\Delta \mathbf{U}] \qquad (7)$$

The difference between Eq.(3) and Eq.(7) is the decomposition error term $\mathbf{P} \cdot [\delta \mathbf{U}]$. We now present an iterative procedure for removing this decomposition error.

The decomposition error term is fed back into the matrix equation on the right hand side for self cancellation. A k-step iterative modified approximate factorization algorithm[1], MAF(k), is defined as follows.

MAF(k) Algorithm

$$\mathbf{M} \cdot \left[\delta \mathbf{U}^{(\mathbf{k})}\right] + \mathbf{P} \cdot \left[\delta \mathbf{U}^{(\mathbf{k})}\right] = [\Delta \mathbf{U}] + \mathbf{P} \cdot \left[\delta \mathbf{U}^{(\mathbf{k-1})}\right]$$

$$\text{where} \qquad \left[\delta \mathbf{U}^{(\mathbf{0})}\right] = [0] \qquad k = 1, 2, 3, \cdots$$

MAF(1) is the same as Eq. (7) above. For this iterative procedure to work, each MAF iteration must be numerically stable and the sequence must converge. For it to be efficient, the number of iterations must be kept small, ideally at two. A stability analysis and examples showing sequence convergence have been given[1]. The optimum value for the maximum number of k subiterations per time step was shown to be two for flows going to steady states.

Added Numerical Dissipation

It is possible to distinguish *artificial* dissipation introduced as additional terms to the difference equations from *genuine* dissipation introduced by choices made, for example upwinding, in the discretization of terms actually appearing in the original set of differential equations. The former can be considered to be of a lower form in the sense that human intervention adds it at will or can dial up its strength until subjectively pleasing results are obtained. On the other hand, dissipative first order accuarate upwind differencing is often blended with higher order derivative approximations, through the use of flux limiters, in regions containing steep gradients, for example at shock waves, to control numerical oscillation. Both forms, *artificial* or *genuine* dissipation, can introduce serious side effects adversely affecting the numerical solution. It is introduced anisotropically, usually in the direction of strong gradients, and can destroy the physical balance of the governing equations solved in general curvilinear coordinate systems typically used for flows about aerodynamically shaped bodies. However, numerical dissipation is the fundamental requirement for numerical stability and obtaining solutions. It has to be present. This section presents a rational way to formulate and implement it. We first observe the sensi-

Copyright © 1998, by the International Council of the Aeronautical Sciences (ICAS)
and the American Institute of Aeronautics and Astronautics, Inc.

tive balance displayed by the Navier-Stokes stress tensor by comparing its form in cartesian and cylidrical coordinate systems.

The Viscous Stress Tensor

The Navier-Stokes momentum equations in three dimensional cartesian coordinates are as follows.

$$\frac{\partial \rho u}{\partial t}+\frac{\partial \rho u^2+p}{\partial x}+\frac{\partial \rho uv}{\partial y}+\frac{\partial \rho uw}{\partial z}=-\frac{\partial \tau_{xx}}{\partial x}-\frac{\partial \tau_{xy}}{\partial y}-\frac{\partial \tau_{xz}}{\partial z}$$
$$\frac{\partial \rho v}{\partial t}+\frac{\partial \rho vu}{\partial x}+\frac{\partial \rho v^2+p}{\partial y}+\frac{\partial \rho vw}{\partial z}=-\frac{\partial \tau_{yx}}{\partial x}-\frac{\partial \tau_{yy}}{\partial y}-\frac{\partial \tau_{yz}}{\partial z}$$
$$\frac{\partial \rho w}{\partial t}+\frac{\partial \rho wu}{\partial x}+\frac{\partial \rho wv}{\partial y}+\frac{\partial \rho w^2+p}{\partial z}=-\frac{\partial \tau_{zx}}{\partial x}-\frac{\partial \tau_{zy}}{\partial y}-\frac{\partial \tau_{zz}}{\partial z}$$

$$\tau_{xx}=-\lambda div\cdot\vec{q}-2\mu\frac{\partial u}{\partial x},\quad \tau_{xy}=\tau_{yx}=-\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)$$
$$\tau_{yy}=-\lambda div\cdot\vec{q}-2\mu\frac{\partial v}{\partial y},\quad \tau_{yz}=\tau_{zy}=-\mu\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right)$$
$$\tau_{zz}=-\lambda div\cdot\vec{q}-2\mu\frac{\partial w}{\partial z},\quad \tau_{zx}=\tau_{xz}=-\mu\left(\frac{\partial w}{\partial x}+\frac{\partial u}{\partial z}\right)$$
$$\text{and}\quad div\cdot\vec{q}=\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}$$

The Navier-Stokes momentum equations in cylindrical coordinates are as follows.

$$\frac{\partial \rho q_r}{\partial t}+\frac{\partial (\rho {q_r}^2+p)r}{r\partial r}+\frac{\partial \rho q_r q_\theta}{r\partial\theta}+\frac{\partial \rho q_r q_z}{\partial z}-\frac{\rho {q_\theta}^2}{r}$$
$$=-\frac{\partial r\tau_{rr}}{r\partial r}-\frac{\partial \tau_{r\theta}}{r\partial\theta}-\frac{\partial \tau_{rz}}{\partial z}+\frac{p+\tau_{\theta\theta}}{r}$$
$$\frac{\partial \rho q_\theta}{\partial t}+\frac{\partial \rho q_\theta q_r r}{r\partial r}+\frac{\partial \rho {q_\theta}^2+p}{r\partial\theta}+\frac{\partial \rho q_\theta q_z}{\partial z}+\frac{\rho q_r q_\theta}{r}$$
$$=-\frac{\partial r\tau_{\theta r}}{r\partial r}-\frac{\partial \tau_{\theta\theta}}{r\partial\theta}-\frac{\partial \tau_{\theta z}}{\partial z}-\frac{\tau_{r\theta}}{r}$$
$$\frac{\partial \rho q_z}{\partial t}+\frac{\partial \rho q_z q_r r}{r\partial r}+\frac{\partial \rho q_z q_\theta}{r\partial\theta}+\frac{\partial \rho {q_z}^2+p}{\partial z}$$
$$=-\frac{\partial r\tau_{zr}}{r\partial r}-\frac{\partial \tau_{z\theta}}{r\partial\theta}-\frac{\partial \tau_{zz}}{\partial z}$$

$$\tau_{rr}=-\lambda div\cdot\vec{q}-2\mu\frac{\partial q_r}{\partial r},$$
$$\tau_{\theta\theta}=-\lambda div\cdot\vec{q}-2\mu\left(\frac{\partial q_\theta}{r\partial\theta}+\frac{q_r}{r}\right),$$
$$\tau_{zz}=-\lambda div\cdot\vec{q}-2\mu\frac{\partial q_z}{\partial z},$$
$$\tau_{r\theta}=\tau_{\theta r}=-\mu\left(\frac{\partial q_r}{r\partial\theta}+\frac{\partial q_\theta}{\partial r}-\frac{q_\theta}{r}\right)$$
$$\tau_{\theta z}=\tau_{z\theta}=-\mu\left(\frac{\partial q_\theta}{\partial z}+\frac{\partial q_z}{r\partial\theta}\right)$$
$$\tau_{zr}=\tau_{rz}=-\mu\left(\frac{\partial q_z}{\partial r}+\frac{\partial q_r}{\partial z}\right)$$
$$\text{and}\quad div\cdot\vec{q}=\frac{\partial rq_r}{r\partial r}+\frac{\partial q_\theta}{r\partial\theta}+\frac{\partial q_z}{\partial z}$$

On comparing the two descriptions above we can find similarities and explain differences. For example, consider the two additional terms on the left hand side of the momentum equations in cylindrical coordinates, $-\rho {q_\theta}^2/r$ and $+\rho q_r q_\theta/r$. The first is the centrifugal force term always adding momentum in the radial direction if the fluid has angular momentum. The second represents the "ice skater" term where angular momentum is increased if radial fluid motion is toward the axis and reduced otherwise. The physics of the flow requires these two terms to balance the system and they naturally arise in transforming the cartesian form of the momentum equations into the cylindrical system. The viscous terms given on the right hand side of the equations also show additional terms in the cylindrical system and are required to balance the flow physics.

The equations of fluid dynamics are often solved numerically in generalized coordinates, some not even orthogonal, and care is taken to transform the cartesian equations, either directly or through finite volume formulations, into the computational coordinate system so that the frame independent flow physics is preserved. However, when it comes time to add dissipation, either directly or through the choice of a lower order accurate difference approximation, little thought is given to the preservation of physical balance. The desired effect of artificial viscosity is to damp numerical oscillation, which often leads to instability. However, an unanticipated side effect is the destruction of the physical balance of the overall system. It would be similar to adding a significant new term to the governing cylindrical equations that directly changes radial momentum and causes an unwanted significant "ice skater-like" reaction in angular momentum. It is hopeless to add artificial viscosity in multidimensional curvilinear coordinate systems and simultaneously preserve physical balance - with one exception.

The exception is to use the already frame indepentent Navier-Stokes viscous stress tensor itself as a vehicle to add numerical dissipation. Similar to the concept of an eddy viscosity used to include the effects of turbulent mixing in the Reynolds Averaged Navier-Stokes equations, the addition of numerical viscosity by augmenting the natural viscosity can be used in a frame independent manner to control numerical instability.

$$\mu \leftarrow \mu_{physical}+\mu_{numerical} \tag{8}$$

The Choice of Added Numerical Viscosity

Assuming the numerical procedure has been stripped as bare as is reasonably possible of sources of numerical

Copyright © 1998, by the International Council of the Aeronautical Sciences (ICAS)
and the American Institute of Aeronautics and Astronautics, Inc.

error and that the procedure requires added dissipation to control stability, we will illustrate the choice of numerical viscosity for two flow situations, at shock waves and subsonic convection. Numerical results will be given for each case.

Numerical Viscosity at Shock Waves

Shock waves are probably the most prominent feature of compressible flow and they have received the most attention in computational fluid dynamics. Yet, they continue to be a major source of difficulty in multidimensional flow. Poor representations of them will occur if they are strong, above a normal Mach number of two, and are allowed to cut indiscriminantly across a computational grid. Disturbing irregularities are often found near the axis of symmetry for blunt body calculations[10]. Mesh adaptation is not sufficent in itself to represent shock waves accurately. Numerical dissipation is required. The following choice for added numerical viscosity was used to obtain the results for high Mach number flow past blunt bodies to be presented later.

First, the locations of all shock points in each computational coordinate direction are found by determing if the normal velocity component changes from supersonic to subsonic flow across a computational coordinate surface with an accompanying increase in pressure. Consider a ξ, η and ζ computational coordinate system with points 1 and 2 adjacent to each other in coordinate direction ξ and $\xi_2 > \xi_1$. A shock point is present if

$$q_{n1} > c_1, \qquad q_{n2} < c_2 \qquad \text{and} \qquad p_2 > p_1$$

or

$$-q_{n1} < c_1, \qquad -q_{n2} > c_2 \qquad \text{and} \qquad p_1 > p_2$$

where q_n is the velocity normal to surface $\xi = constant$, located midway between points ξ_1 and ξ_2, c is the sound speed and p is the pressure.

If either of these conditions is met

$$\mu_{numerical_i} = 1/2 \cdot \Delta h_{\xi_i} \cdot \rho_i c_i \qquad (9)$$

where Δh_ξ represents the grid spacing distance in the ξ direction and ρ is the density. At both points $i = 1$ and 2 the viscosity is augmented as shown above in Eq.(8). Similarly, the other coordinate directions, η and ζ, are checked for the presense of shock points.

The above procedure augments the natural viscosity along a narrow band, two points wide, aligned with the shock wave. This in turn can influence the calculation along a band four points wide. However, supersonic points adjacent to points 1 or 2 should be shielded from being affected by numerical viscosty. They don't need it and the upstream propagation of information within a supersonic flow through numerical viscosity is not physical.

Further Comment on Shock Waves

An efficient algorithm is not enough, in general, to insure rapid convergence to accurate numerical solutions. The importance of the grid is fundamental and has a strong influence on both convergence and accuracy. For the strong bow shock cases to be presented herein mesh adaptation[10] was used. A grid line of the mesh was adjusted during the computation, into alignment with the Mach one contour location within the shock wave of the numerical solution. In addition, mesh refinement near the shock Mach one location was used for some cases.

Also an attempt to preserve the integity of the freestream just ahead of the shock wave was made by modifying the flux approximation. This region, very vulnerble to changes at the shock because of its relatively low density, pressure and internal energy, was shielded as follows. Again, consider a ξ, η and ζ computational coordinate system with points 1 and 2 adjacent to each other in, for example, the ξ coordinate direction. Assume that ξ_1 and ξ_2 straddle the Mach one contour location. Let ξ_3 be an adjacent point located upstream of the Mach one contour. Point ξ_3 is the nearest point to the shock wave surrounded completely by supersonic neighboring points. It should be unaffected by the strong gradients at the shock. For simplicity, assume points 1,2 and 3 correspond to mesh points i+1, i and i-1. The flux leaving grid point ξ_3 uses a simple first order upwind approximation,

$$F_{i-1/2} = F_{i-1}$$

Using a higher order approximation requires extrapolation or interpolation of data from more than one mesh point and can cause an unwanted disturbance to the flow near shock waves. In addition, the supersonic point located at the shock, ξ_2, also requires adjustment. The flux leaving it is a blended value of first and second order upwind approximations, as follows.

$$F_{i+1/2} = (1-\beta) * F_{i+1/2}^{1st\ order} + \beta * F_{i+1/2}^{2nd\ order}$$

where

$$\beta = \frac{1-M_1}{M_2 - M_1} \text{ and } F_{i+1/2}^{1st\ order} = F_i$$

As M_1, the Mach number at point 1, approaches one, the flux at leaving point 2 becomes similar to that at point 3 and as M_2 approaches one the flux becomes

Copyright © 1998, by the International Council of the Aeronautical Sciences (ICAS)
and the American Institute of Aeronautics and Astronautics, Inc.

fully second order accurate. The blended flux represents a smooth transition from supersonic to subsonic flow across the shock. All other mesh points away from the shock are approximated to second order accuracy in supersonc flow regions or to third order in subsonic regions as given in an earlier section.

Numerical Viscosity for Subsonic Convection

Numerical instabilities often begin with oscillations in the convection velocity within subsonic regions where they can be reinforced through self stimulation. The following choice for added numerical viscosity was used to obtain the results for Mach 0.2 flow past an ellipse presented in the next section.

Again, consider two points 1 and 2 adjacent to each other and check for subsonic flow. If

$$|q_{n1}| < c_1, \qquad \text{and} \qquad |q_{n2}| < c_2$$

then

$$\begin{aligned}\mu_{numerical_i} = & 0.05 \cdot min\{\Delta h_\xi, \Delta h_\eta, \Delta h_\zeta\} \\ & \cdot (\rho_1|q_{n1}| + \rho_2|q_{n2}|) \\ & \cdot \frac{|q_{n3} - 3(q_{n1} - q_{n2}) - q_{n4}|}{|q_{n3}| + 3(|q_{n1}| + |q_{n2}|) + |q_{n4}| + \epsilon}\end{aligned} \tag{10}$$

where points 3 and 4 are adjacent to 1 and 2 and are ordered as $\xi_3 < \xi_1 < \xi_2 < \xi_4$ and $\epsilon \simeq 10^{-9}$ is added to prevent division by zero in motionless regions. The quotient factor is bounded by one and is proportinal to

$$\frac{1}{8q_n} \cdot \Delta\xi^3 \cdot |\frac{\partial^3 q_n}{\partial \xi^3}|$$

The above value of numerical viscosity is added, as in Eq.(8), at points 1 and 2.

Computational Results

The method for solving the equations of compressible viscous flow[1], with added numerical viscosity as just described, was applied to solve two benchmark calculations with known numerical difficulties. The first, inviscid Mach 0.2 flow past an ellipse with 6 to 1 axis ratio at 5 degrees angle of attack, was suggested by Pulliam[5], and the second, Mach 5 viscous laminar flow past a sphere was suggested by Blottner[6]. In addition, solutions for real gas equilibrium flow past a sphere at Mach 18.5 and perfect gas flow past a sphere-cone body at Mach 7.97 at 7 degrees angle of attack are shown.

Mach 0.2 Inviscid Flow Past an Ellipse

In 1990 T.H. Pulliam[5] presented the outstanding result that Euler flow codes were incapable of calculating low Mach number flow past the simple ellipse. "The basic result obtained here is a lifting solution for any combination of grid and/or angle of attack which is nonsymmetric." He then challenged the computational fluid dynamics community to "explain this unusual behavior" and he defined a benchmark flow problem, an ellipse with axis ratio 6:1 at 5 degrees angle of attack in Mach 0.2 inviscid flow, which exhibited "disturbing" results. The flow should produce zero lift and drag but all known numerical Euler solutions written in terms of varaibles ρ, u, v, and energy reproduced neither. Pulliam's results converged to a significant lift coefficient of 1.545. This problem has since remained the numerical equivalent to d'Alembert's paradox.

Highly accurate solutions for this problem have been obtained by Hafez and Brucker[11] for the *steady* incompressible and compressible form of the Euler equations. Also, Winterstein and Hafez[12] present excellent solutions for this problem with the Euler equations solved on triangular meshes. In the latter study, they fixed the rear stagnation point by varying its location until reaching the condition of zero lift. In the present study, the ***unsteady*** Euler equations were solved without any conditions on the location of the stagnation points. There is much discussion concerning multiple solutions for this inviscid flow problem, which has no sharp trailing edge and no specification of a Kutta condition. However, if the flow is initialized with no vorticity, and none is generated during the course of the calculation, the symmetric non-lifting solution should occur.

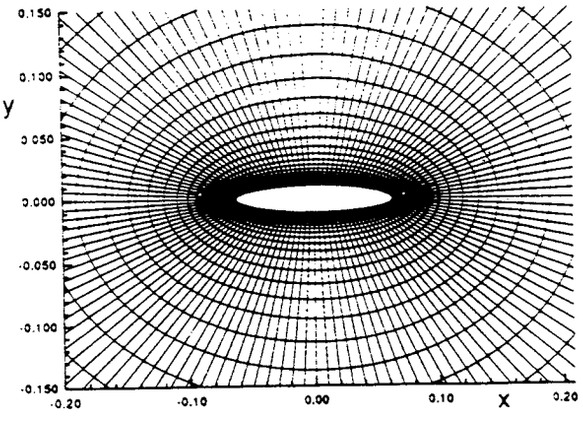


Figure 1. Mesh about an ellipse.

Fig. 1 shows a portion of a 130x66 point mesh for this benchmark problem. The outer boundaries of the stretched mesh were approximately 5000 major axis radii away from the body. The natural viscosity was set to zero and slip boundary conditions were implemented at the body. The far field boundaries were held to constant Mach 0.2 flow. The calculation was started impulsively by suddenly placing the body in a

Copyright © 1998, by the International Council of the Aeronautical Sciences (ICAS)
and the American Institute of Aeronautics and Astronautics, Inc.

uniform Mach 0.2 flow. The initial CFL number was 3.0×10^3, was increased to 2.5×10^9 after 80 steps and thereafter was held fixed. The calculation was run to 2000 time steps to test convergence, but the solution was fairly steady much earlier.

Figs. 2 and 3 show fairly symmetric patterns for pressure contours and streamlines. If significant lift were produced the rear stagnation point would be expected to move from the top of the trailing edge around toward the lower surface, as shown by Pulliam[5]. Fig. 4 shows the coefficient of pressure along the surface of the ellipse. Again, it appears fairly, though not perfectly, symmetric, as required for small lift and drag. Fig. 5 shows lift and drag versus time step. The final values are 4.00×10^{-3} and 1.66×10^{-3}, respectively. The lift fell to 7.40×10^{-2} after 100 time steps. The risidual versus time step is shown in Fig. 6.

A reasonably accurate Euler solution has been obtained for "Pulliam's paradox" problem. This time again, as with d'Alembert's paradox, the answer appears to be viscosity, in this case too much or frame dependent numerical viscosity, which can destroy the sensitive physical balance of the Euler equations.

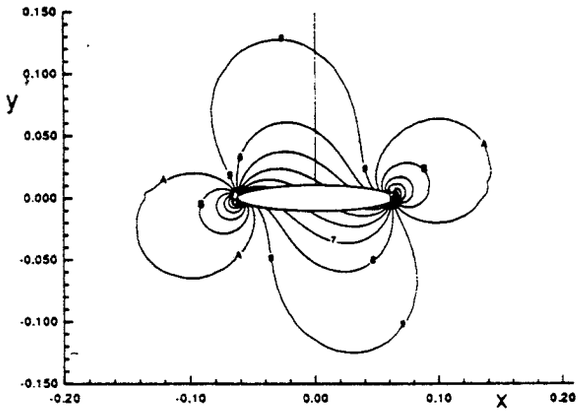


Figure 2. Pressure contours for subsonic flow about an ellipse.

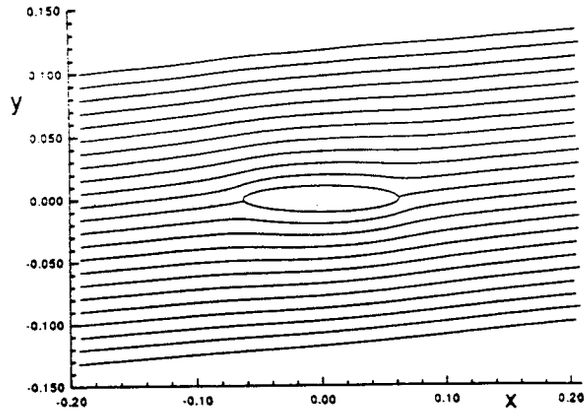


Figure 3. Streamlines for flow about an ellipse.

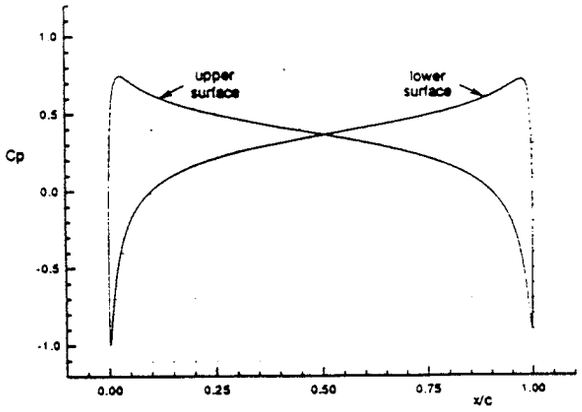


Figure 4. Coefficient of pressure versus x/c.

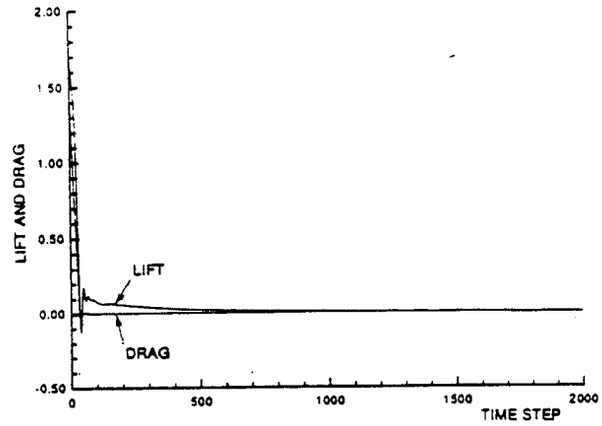


Figure 5. Lift and drag versus time step.

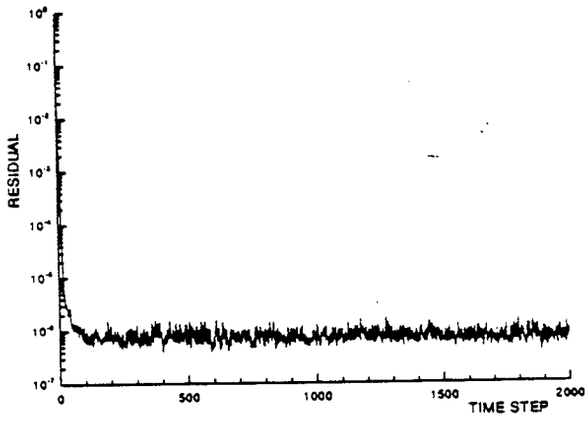


Figure 6. Residual versus time step.

Mach 5 Viscous Flow Past a Sphere

The Mach 5 laminar flow past a sphere benchmark problem was originally chosen by Blottner[6] in 1990 to study numerical difficulties encountered near the intersection of the axis of flow symmetry with the bow shock wave. This region is very sensitive because of the coordinate singularity caused as $r \rightarrow 0$ in terms

Copyright © 1998, by the International Council of the Aeronautical Sciences (ICAS)
and the American Institute of Aeronautics and Astronautics, Inc.

containing factors $1/r$. See, for example, the Navier-Stokes momentum equations given earlier in cylindrical coordinates. Many papers have observed numerical difficulties in this region. Blottner did an excellent job of discussing the nature of this numerical problem and presented excellent computational results using a thin layer Navier-Stokes code. The bow shock wave was used as an outer boundary of the flow. In the present paper, a full Navier-Stokes calculation is presented and the shock wave was captured as an internal feature of the flow on a grid adapted to the bow shock wave, as discussed earlier.

The flow was solved on three different grids, a course mesh consisting of 22x26 points, a medium mesh of 42x50 points, and a fine mesh of 82x98 points, each

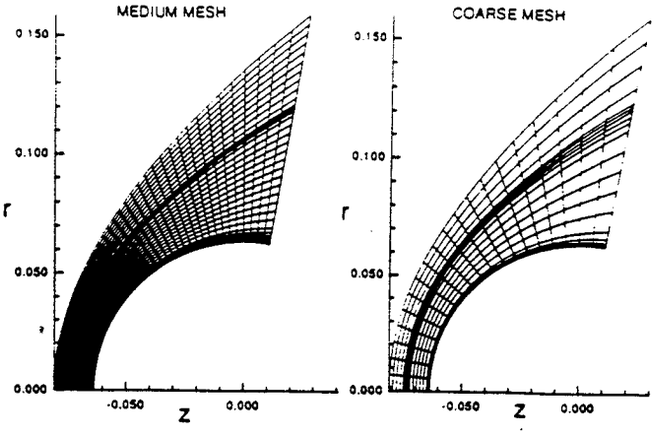
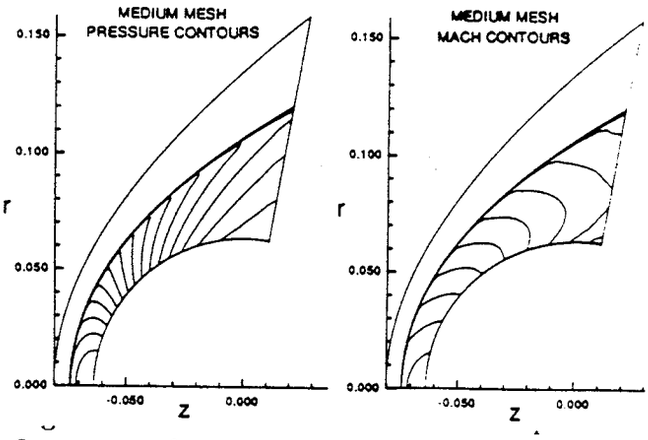


Figure 7. Medium and coarse meshes about a sphere.

spanning the same flow volume. The computational difficulty increases as points are placed closer to the axis of symmetry. Fig. 7 shows the medium and coarse meshes. An initial bow shock wave location was chosen to be aligned with a grid line of the mesh. The Rankine-Hugoniot shock-jump relations were used to initialize the flow behind the shock wave with the normal-to-the-body-surface velocity component reducing to zero as the body surface was approached. The surface temperature was held at 98.89 degrees Kelvin. Sutherland's formula was used to determine viscosity. The Reynolds number, based on sphere diameter, was 1.89×10^6.

The calculation was started with a CFL number, the ratio of the time step size actually used to the maximum allowed by an explicit method, of about 6×10^3 The CFL number was then increased to 2.5×10^7 during the first 52 time steps and thereafter held fixed. This medium mesh case was run for 256 time steps. The mesh was also dynamically kept aligned with the bow shock wave and refined in the neighborhood of the shock.



flow past a sphere.

Pressure and Mach contours for the medium mesh are shown in Fig. 8 and surface pressure, skin friction and heat transfer results for all three meshes are shown in Fig. 9. Note that, as expected, skin friction increases linearly with distance from the axis of symmetry, surface pressure and heat transfer have bell shaped curves with their maxima at the axis, and that the three solutions converge. The stagnation pressure ratios, p_{stag}/p_∞, on the three meshes, from coarse to fine, are 32.5800, 32.6228 and 32.6667, which compare well to the Richardson extrapolated value of 32.6558 given by Blottner[6]. The corresponding heat transfer values, expressed in kW/m^2, 103.920, 104.628 and 105.613 also compare favorably with the value 107.360 given by Blottner.

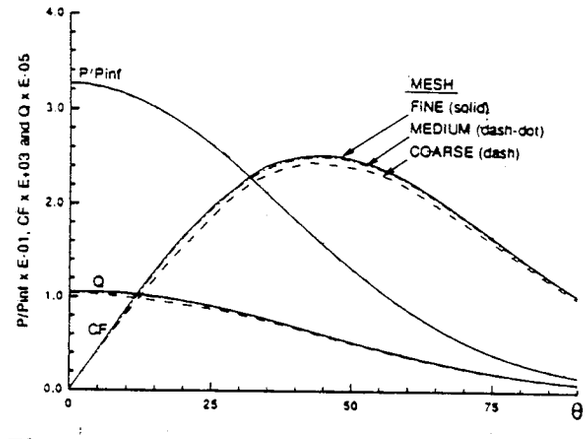


Figure 9. Surface pressure, skin friction and heat transfer for Mach 5 flow past a sphere (pressure is normalized by the freestream pressure, skin friction was devided by the dynamic freestream pressure and Q, the heat transfer, is given in Watts/m^2).

Fig. 10 shows the residual versus time step. During the first 64 steps the mesh tracked the shock by relaxing the position of a chosen grid line toward the Mach

Copyright © 1998, by the International Council of the Aeronautical Sciences (ICAS)
and the American Institute of Aeronautics and Astronautics, Inc.

1 contour[10]. During this time the average residual reduction rate was approximately 0.86, a value larger than the rate of 0.80[1] on a fixed grid for this same problem. From step 64 to 96 the grid was refined in the vicinity of the shock as well as tracked and thereafter the grid only tracked the shock. These results were obtained on a 32 bit per word workstation and residual reductions are limited to about five orders of magnitude.

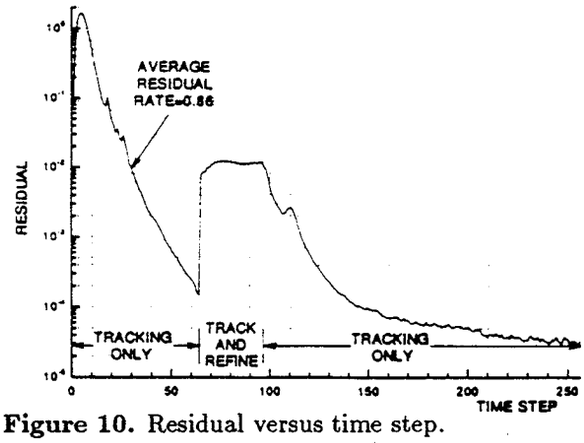


Figure 10. Residual versus time step.

Mach 18 Real Gas Flow Past a Sphere

The approach discussed in the last section was extended to real gas equilibrium flow using the curve fits of Tannehill and Mugge[13]. The axisymmetric flow for Mach 18.5 was solved on a mesh of 24x40 points, which adapted without refinement to the bow shock wave, and converged in approximately 100 time steps.

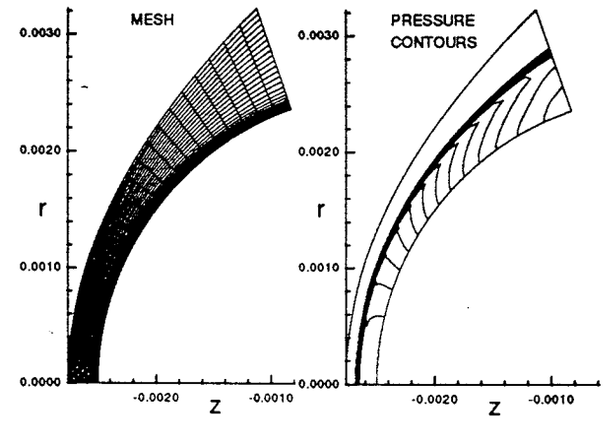


Figure 11. Adapted mesh and pressure contours for Mach 18 real gas flow.

The adapted mesh and pressure contours are shown in Fig. 11. The bow shock, which at the axis of symmetry increased pressure by a factor of 461, is captured cleanly because of the mesh adaptation and the numerical viscosity introduced along the shock jump. The local increase in numerical viscosity at the shock is used the smooth the flow tangentially to the shock as well as removing under and over shoots across the shock. The former is important to preserve the freestream direction of the flow approaching the shock. Small deflections can cause significant errors in surface pressure, skin friction and heat transfer distributions in the vicinity of the stagnation point.

Three Dimensional Flow Past a Sphere-Cone Body

Mach 7.97 perfect gas flow about a sphere-cone body was solved on a mesh of 24x20x20, which adapted without refinement to the bow shock wave. The Reynolds number based on the body nose diameter was 70 million and the sphere-cone body, with cone half angle 10.5 degrees, was placed at 7 degrees angle of attack. The flow converged in approximately 200 steps. Fig. 12 shows a side view of the mesh and Mach contours and Fig. 13 shows a cross sectional view of the mesh and pressure contours along the cone section of the body.

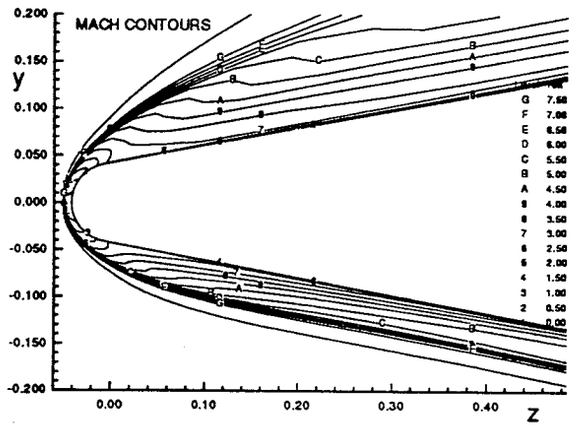


Figure 12. Mach contours in side view.

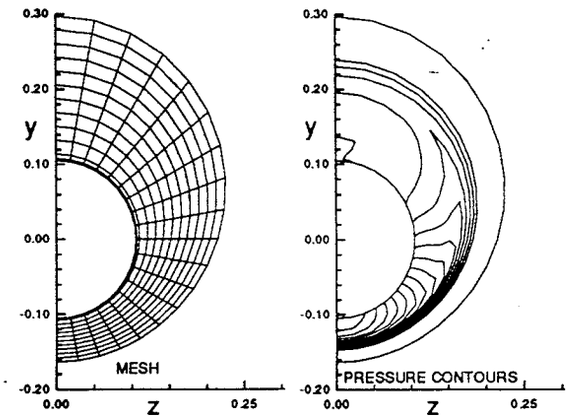


Figure 13. Mesh and pressure contours in cross sectional view

Conclusion

A new procedure has been developed that is both fast and accurate. It contains two approaches toward

Copyright © 1998, by the International Council of the Aeronautical Sciences (ICAS)
and the American Institute of Aeronautics and Astronautics, Inc.

reducing computational error. First, factorization error is minimized by using a diagonally dominant matrix decomposition and an iterative procedure to self cancel remaining error. Second, numerical instability is controlled using an approach similar in concept to that of an eddy viscosity. Numerical viscosity is added directly to the flow equations by augmenting the natural viscosity. This approach preserves frame independence, believed to be important for a physical balance of the governing equations when solved in arbitrary curvilinear coordinate systems.

This procedure has been applied solve two difficult benchmark flow problems. The computed results were in good agreement with accepted computational and theoretical results. The procedure has been extended to real gas equilibrium flow and three dimensional flow.

References

[1] MacCormack, R.W., "A New Implicit Algorithm for Fluid Flow," *AIAA Paper No. 97-2100*, 1997.

[2] Bardina, J. and C.K. Lombard, "Three Dimensional Hypersonic Flow Simulations with the CSCM Implicit Upwind Navier-Stokes Method," *AIAA Paper No. 87-1114*, 1987.

[3] Briley, W.R. and H. McDonald, "Solution of the Multidimensional compressible Navier-Stokes Equations by a Generalized Implicit Method," *Journal of Computational Physics*, Vol. 24, 1977, pp. 372–397.

[4] Beam, R. and R.F. Warming, "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," *AIAA Journal*, Vol. 16, 1978, pp. 293–402.

[5] Pulliam, T.H., "Computational Challenge: Euler solulion for Ellipses," *AIAA Journal*, Vol.28, No.10, 1990, pp. 1703–1704.

[6] Blottner, F.G., "Accurate Navier-Stokes Results for the Hypersonic Flow over a Spherical Nosetip," *Journal of Spacecraft*, Vol.27, No.2, 1990, pp. 113–122.

[7] Stone, H.L., "Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations," *Siam J. Numer. Anal.*, Vol. 5, No. 3, Sept., 1968, pp. 530–558.

[8] Varga, R.S., "Matrix Iterative Analysis," Englewood Cliffs, N.J., Prentice Hall, 1962.

[9] Wachspress, E.L., "Iterative Solution of Elliptic Systems and Applications to the Neutron Diffusion Equations of Reactor Physics," Englewood Cliffs, N.J., Prentice Hall, 1966.

[10] MacCormack, R.W., "Considerations for Fast Navier-Stokes Solvers," presented at the Symposium *Advances of Flow Simulation Techniques*, Davis, CA, May 2-4, 1997, proceedings to be published.

[11] Hafez, M. and D. Brucker, "The Effect of Artificial Vorticity on the Discrete Solution of Euler Equations," *AIAA Paper No. 91-1553-CP*, 1991.

[12] Winterstein, R. and M. Hafez, "Euler Solutions for Blunt Bodies Using Triangular Meshes: Artificial Viscosity Forms and Numerical Boundary Conditions," *AIAA Paper No. 93-3333-CP*, 1993.

[13] Tannehill, J.C.,, and P.H. Mugge, "Improved Curve Fits for the Thermodynamic Properties of Equilibrium Air Suitable for Numerical Computations Using Time-Dependent or Shock Capturing Methods," NASA CR-2740, 1974.