Adaptation Methods for a New Navier-Stokes Algorithm

Yannis G. Kallinderis* and Judson R. Baron†
Massachusetts Institute of Technology, Cambridge, Massachusetts

Various adaptation techniques for the computation of two-dimensional laminar viscous flows are presented. An initially coarse grid is automatically embedded locally via a feature detection algorithm to provide accurate predictions of boundary-layer regions. The need for resolution in a specific direction is used to limit embedding to that direction. Last, the Navier-Stokes equations are solved within appreciably viscous regions, whereas for the remaining areas the description is reduced to the Euler equations. The current procedure combines three adaptation techniques—viscous, directional, and equation—and attains equivalent accuracy with more than an order-of-magnitude decrease in computing time over nonadaptive methods. The basic algorithm uses a new finite-volume scheme that has been developed for the discretization of the viscous terms and has the conservation property that is of some importance when shocks are present. Example flowfields that are considered include circular arc cascades in both subsonic and supersonic flow. Comparisons are made with previous results.

Introduction

In recent years, considerable progress has been made in the development of numerical methods for the solution of the Navier-Stokes equations. Most of those methods, however, are impractical for the calculation of complicated flows in a design environment. The primary reason is that the efficiency of the current algorithms is poor and makes it difficult to obtain accurate results. Very fine resolution is needed, which results in long computation times even with the use of available supercomputers. An approach often adopted is to use a simplified set of equations to describe the flowfield (e.g., potential flow or Euler equations). Such approximations have proven successful for specific kinds of flowfields but cannot cope with complicated flows as are usually of engineering interest.

The classical way of achieving the needed resolution is to cluster grid points in regions with high flow gradients. This makes grid generation more difficult, especially in the third dimension. More important is the fact that the specific clustering procedure also may create problems that are related to accuracy and stability due to the resulting grid stretching and skewness. Even for simple geometries, clustering frequently results in unnecessary resolution in some regions of the domain.

A promising approach is to locally embed a fine grid in an initially coarse grid in regions with large flow gradients (e.g., boundary layers, shocks, wakes, etc.). In order to accomplish this, the algorithm must sense high-gradient regions and automatically must divide grid cells in such regions. This approach has been used for the resolution of shocks in flowfields described by Euler equations.[1-6]

Flowfields involving multiple-scale phenomena are of primary interest here, such as represented by the simultaneous presence of boundary layers and shocks. Special adaptation techniques, as well as a discretization scheme for the viscous terms in the Navier-Stokes system, have been developed in order to predict such flows accurately and efficiently.

Presented as Paper 87-1167 at the AIAA 8th Computational Fluid Dynamics Conference, Honolulu, HI, June 9-11, 1987; received June 26, 1987; revision received Feb. 10, 1988.

*Research Assistant, Computational Fluid Dynamics Laboratory, Department of Aeronautics and Astronautics. Student Member AIAA.

†Professor, Computational Fluid Dynamics Laboratory, Department of Aeronautics and Astronautics. Associate Fellow AIAA.

The aforementioned inviscid adaptation method has been extended to include laminar viscous regions (viscous adaptation). However, unnecessary and therefore inefficient embedding is avoided by locally refining cells only in the direction of flow gradients (directional adaptation). Similarly, viscous terms are evaluated only in those regions where viscous stresses are appreciable. Elsewhere, the description is reduced to the Euler equations (equation adaptation).

A conservative discretization scheme is required in order to correctly model shocks. The present approach is an extension of an existing finite-volume, Lax-Wendroff-type scheme,[7] with viscous terms added and suitably discretized.

The numerical scheme is presented first and validated by comparison with existing solutions. Then, the adaptive techniques are described and computed results are presented to demonstrate their accuracy. Finally, the efficiency of these techniques is discussed.

Governing Equations

The Navier-Stokes equations written in Cartesian two-dimensional conservation form are

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}=\frac{\partial R}{\partial x}+\frac{\partial S}{\partial y} \tag{1}$$

where

$$U=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ E\end{bmatrix},\quad F=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ (E+p)u\end{bmatrix},\quad G=\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ (E+p)v\end{bmatrix} \tag{2}$$

are the state and convective flux vectors in the x and y directions, respectively. The viscous flux vectors are

$$R=\begin{bmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ u\tau_{xx}+v\tau_{xy}+q_x\end{bmatrix},\quad S=\begin{bmatrix}0\\ \tau_{xy}\\ \tau_{yy}\\ u\tau_{xy}+v\tau_{yy}+q_y\end{bmatrix} \tag{3}$$

where τ_{xx}, τ_{yy}, τ_{xy} are the viscous stresses. For a perfect gas, the pressure is related to the specific internal energy E by $p=(\gamma-1)[e-(\rho/2)(u^2+v^2)]$.

After we nondimensionalize the foregoing equations, Mach and Reynolds numbers appear as parameters. Sutherland's law was used for the viscosity coefficient for all computations.

Numerical Scheme

An explicit, finite-volume numerical scheme that was developed previously by Ni[7] for the Euler equations was used for discretization of the convective terms. The method uses Lax-Wendroff-type marching in time and consists of two basic operations. The first evaluates the first-order temporal terms at the grid nodes (e.g., E in Fig. 1). Such terms involve first-order spatial derivatives that are evaluated by averaging the surrounding primary control volume center values (e.g., at a through d). The second operation determines the second-order temporal terms at the grid nodes. These contributions involve second-order spatial derivatives that are calculated using a spatially translated control volume (e.g., *abcd* in Fig. 1). This operation involves a piecewise integration around this secondary control volume.

Viscous Terms

The Euler scheme just presented has been extended by including the viscous terms of the Navier-Stokes equations. Other extensions of Euler schemes without adaptation are found in Ref. 8 (for the Ni scheme) and Ref. 9 (for a Runge-Kutta scheme).

The first-order temporal viscous terms involve second-order derivatives, and second-order terms involve fourth-order derivatives. Since it is very important to minimize the number of nodes that contribute to updating properties at each node, only those viscous terms that provide first-order temporal accuracy are kept, in view of our interest in the steady-state solution. The spatially translated cell *abcd* is used to compute those viscous terms.

We illustrate the discretization by consideration of the viscous term u_{xx} at node E. Using Green's theorem for the volume *abcd*, we have

$$u_{xx} = (1/S_{abcd}) \oint_{abcd} (u_x)\, dy$$

$$= (1/S_{abcd})[(u_x)_{cd}\Delta y_{cd} + (u_x)_{bc}\Delta y_{bc} + (u_x)_{ab}\Delta y_{ab} + (u_x)_{da}\Delta y_{da}]$$

where S_{abcd} is the area of the cell *abcd*, $\Delta y_{cd} = y_c - y_d$, etc. The first-order derivative at the face *cd* of the control volume *abcd* is evaluated employing the area *EcFd*. Similar volumes are used for the other face derivatives. Thus,

$$(u_x)_{cd} = (1/S_{cd})(u_F\Delta y_F + u_c\Delta y_c + u_E\Delta y_E + u_d\Delta y_d)$$

where S_{cd} is the area of *EcFd*, and

$$\Delta y_F = (\Delta y_{IF} + \Delta y_{FC})/2$$

$$\Delta y_c = (y_H + y_E)/2 - (y_I + y_F)/2$$

$$u_c = (u_E + u_H + u_I + u_F)/4$$

etc. The foregoing discretization is conservative. The discretization of the convective terms allows odd-even modes to appear in both directions, but the discretization of the viscous terms does not allow those modes to appear.

In order to accelerate convergence to the steady state, a multiple-grid method[7] that acts only on the convective terms was used. Its function is to accelerate the propagation of waves by using coarser grids than the basic ones.

Odd-even modes were suppressed in the essentially inviscid portion of the flow with the aid of a nine-point Laplacian

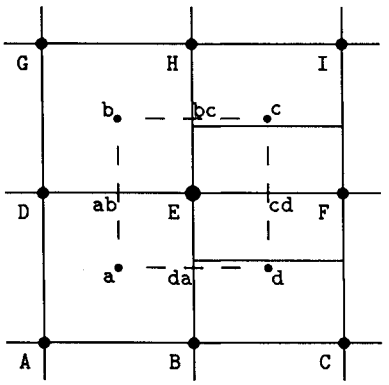


Fig. 1 Integration cell.

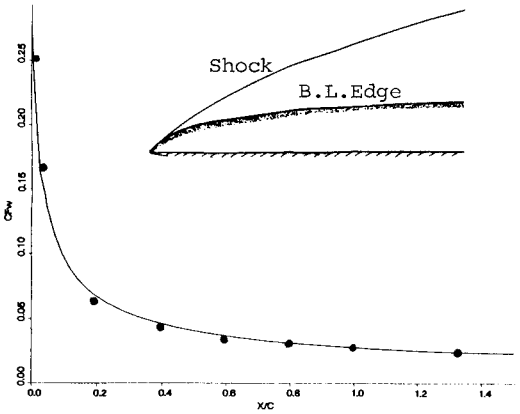


Fig. 2 Comparison of skin-friction distributions for a flat plate ($M_\infty = 3$, $Re = 10^3$); — present work, • Ref. 10.

smoothing operator of the form

$$\mu(U_a + U_b + U_c + U_d - 4U_E)$$

For shock capturing, the following smoothing operator in conservative form was used:

$$\sigma(|\Delta_x p|\Delta_x U + |\Delta_y p|\Delta_y U)$$

where $\Delta_x(.)$, $\Delta_y(.)$ denote differences in the x,y directions. The pressure terms are necessary in order to switch smoothing on and off at and away from the shock regions. Typical values for the smoothing coefficients were $\mu = 0.002$ and $\sigma = 0.01$. These values are sufficiently small such that the solution within the viscous region is not contaminated by the smoothing terms.

An important property of the aforementioned scheme is that all operations can be performed in a piecewise sense within each cell without the need for any external information. This is very useful in dealing with unstructured grids, as will be seen in the following section.

Solver Validation

A test of the Navier-Stokes numerical scheme has been carried out for two problems that permitted comparisons with previous results. The first was a flat plate in a supersonic flow at $M_\infty = 3$ and $Re = 10^3$ (Fig. 2) with a 65×65 grid.

Four quantities (P_{stag}, T_{stag}, v/u, P_∞) were specified at the inlet plane, and a no-slip condition, temperature, and extrapolated pressure were applied at the solid surface boundary. At the upper boundary of the domain, a tangency condition was assumed. All state variables were extrapolated at supersonic portions of the exit plane, and for subsonic portions the pressure was matched to the immediately adjacent supersonic

flow. In Fig. 2, we compare the skin friction distribution at the wall with that from Ref. 10.

A second test was a 10% circular arc cascade in a subsonic flow at $M_\infty = 0.5$ and $Re = 8 \times 10^3$ (Fig. 3) with a 65×33 grid. The use of the multiple-grid accelerator led to a CPU time-savings factor of 5. The wall-pressure and C_f curves for this example are compared with those in Refs. 8, 11, and 12. The agreement is excellent for both cases.

Adaptive Techniques

Local Grid Refinement

Accuracy is achieved with minimal additional computational effort by embedding several levels of finer grids only in those regions of the domain where important features exist. This can be accomplished simply by subdividing cells of the initial coarse grid in both cell directions (Fig. 4). Proceeding in that way the embedded and initial grids would be topologically similar, and if the initial grid is uniform and orthogonal, these desirable properties characterize the embedded meshes as well. However, depending on cell and feature orientation, there are situations in which resolution is needed primarily in one direction in the vicinity of the feature. In that case, it is advantageous to divide the cell only in that direction and thus avoid the creation of unnecessary cells (Fig. 4) (directional refinement).

Equation Adaptation

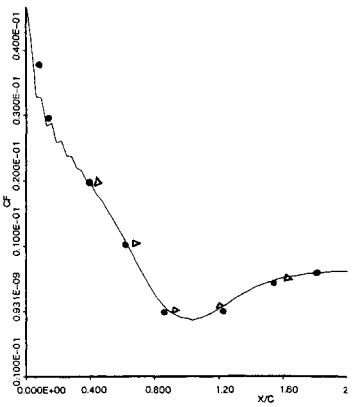
The magnitude of viscous stresses generally decreases very rapidly away from solid boundaries. On the other hand, the evaluation of viscous terms is quite expensive. Thus, the approach here has been to introduce a criterion to monitor the need for solution of the full Navier-Stokes equations and to do so only when required by the presence of shear. The Euler equations then are applicable everywhere else. This is a relatively easy procedure with the present explicit algorithm because each cell is integrated independently at each time step. Essentially different integrators can be used for cells in which different physics dominate.

Feature Detection

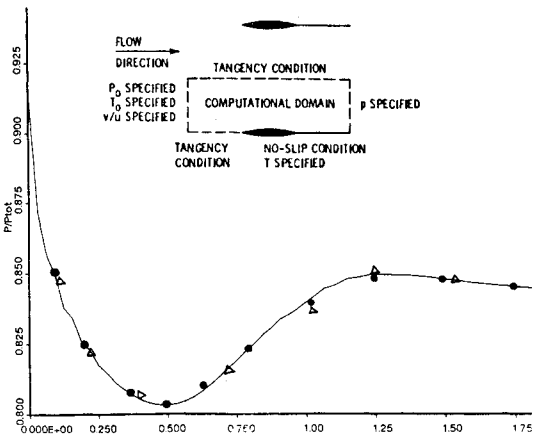
It is an essential part of the adaptation technique that the algorithm be able to sense the existence and track the evolution of special features. The dominant features of present interest are shear layers and shocks.

There is a choice of flow parameters that can be used to detect such features, including velocity, pressure, density, and Mach number distributions. Variations that have been explored include undivided and divided differences for those parameters. Shear layers and shocks are very different flow phenomena with completely different scales, and the use of only one of the foregoing parameters is inadequate. A combination of suitable parameters proves to be necessary.

Figures 5 and 6 illustrate the influence of various criteria on the detection of a boundary layer and a shock separately. Figure 5 illustrates a subsonic flowfield for which a boundary layer is the main feature, whereas Fig. 6 shows the same field



a) Comparison of skin-friction distributions: — present work, • Ref. 8, Δ Ref. 11



b) Comparison of wall-pressure distributions: — present work, • Ref. 8, Δ Ref. 11

Fig. 3 Ten percent circular arc cascade ($M_\infty = 0.5$, $Re = 8 \times 10^3$).

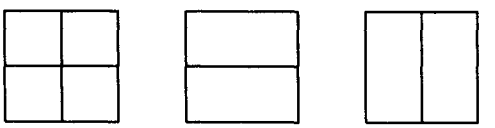
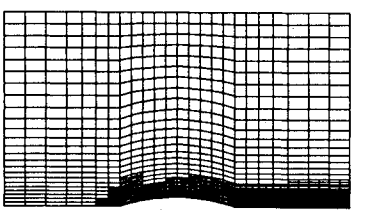
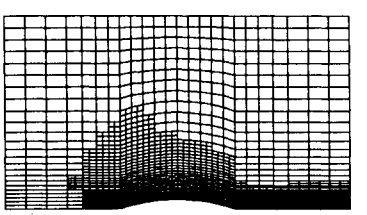


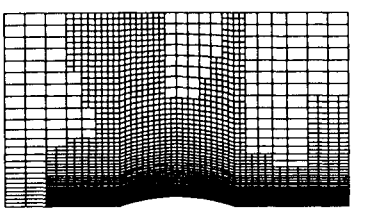
Fig. 4 Types of embedding subdivisions.



a) Velocity gradients $[(\delta_l u/\delta l), (\delta_m u/\delta m), (\delta_l v/\delta l), (\delta_m v/\delta m)]$



b) Velocity differences $(\delta_l u, \delta_m u, \delta_l v, \delta_m v)$



c) Density differences $(\delta_l \rho, \delta_m \rho)$

Fig. 5 Refinement parameter influence on grid for boundary-layer detection, 10% circular arc cascade (subsonic).

in supersonic flow. In the latter case, a shock is formed at the leading edge and is reflected at the upper boundary. Both flowfields are described in more detail in the next sections. All of the criteria have been applied in a directional sense (l,m being the cell directions), and the boundary-layer cells are generally much smaller than those in the inviscid region. As a consequence, an undivided difference of a detection parameter at an inviscid cell can be of the same magnitude as those at a viscous cell. Therefore, divided differences are required for the correct capture of shear layers. On the other hand, the use of parameter gradients when detecting shock regions would lead to increasing gradients after each adaptation and a decision to continue adapting. Thus, it proves worthwhile to use both divided differences for shear layers and undivided differences for shocks. It is evident (Fig. 5) that the use of density leads to an excessive number of embedded cells. More important is the inappropriateness of density for incompressible flow. Mach number is a poor parameter for shock detection in the case of weak discontinuities. However, pressure and velocity differences perform quite well in detecting shocks. It is suggested that velocity gradients be used to detect viscous regions and velocity differences in order to detect shocks.

In addition to parameter selection, a choice must be made of threshold levels for the detection parameters. The threshold is set by using the average and standard deviation values of the parameters. More specifically, we use threshold = $\Phi_{ave} + \alpha\sigma_\phi$, where Φ_{ave} and σ_ϕ are the average and standard deviation values of the detection parameter Φ, and α is a weighting factor that is chosen empirically but has been found to be applicable for a variety of flowfields and conditions.

Data Structure

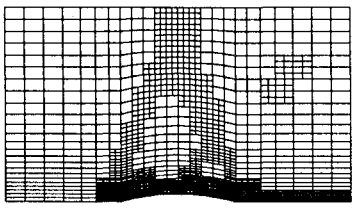
Very important considerations in the adaptive scheme relate to the storage and availability of information that is necessary for the various calculations. A special data structure is required to service the unstructured grid arising from adaptation. Since each cell is computed independently of its neighbors, the grid structure has no impact on the solver that remains the same for any grid topology. In particular, the pointer system was based on cells rather than blocks of cells and thus provided greater flexibility. A system of cellwise pointers, quite similar to finite-element connectivity arrays, keeps track of the information required by the solver. Figure 7 illustrates the basic principle of storing information at the cells of an unstructured grid. The four nodes corresponding to each one of the cells are clear (e.g., nodes 2, 5, 6, 3 belong to cell *B;* node 9 is appointed to both cells *D* and *E*).

Interface Treatment

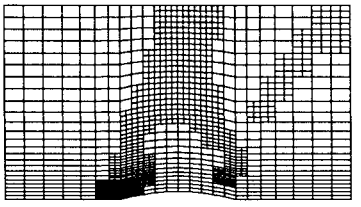
The existence of embedded regions within the interior of the domain introduces internal boundaries (Fig. 8) that must be treated carefully. Stability and accuracy are the two important considerations of the numerical treatment at such an interface. The use of directional refinement introduces several additional types of interfaces and this imposes another requirement on the choice of interface treatment. The treatment preferably should be simple and easily extended to the third dimension.

An interface poses problems for the basic solver for two reasons. The first is that there are cells containing five nodes, whereas the scheme is designed for cells with nodes only at the four corners. The second is that at the interface, there is an abrupt change in the size of the cells. This poses accuracy problems for a Navier-Stokes solver.

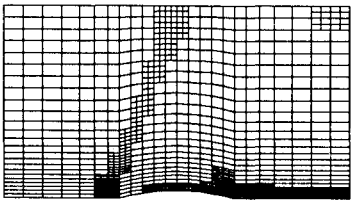
One way to approach interface problems is to perform a special integration for those cells that involve more than four nodes. This is accomplished by modifying the scheme in such a way as to include the additional interface nodes in the integration procedure. It follows that a different integration then would be required for different types of interfaces. This would pose a number of problems when combined with directional embedding, and especially so for three-dimensional fields, because of the many different types of interfaces that appear.



a) Velocity differences ($\delta_l u$, $\delta_m u$, $\delta_l v$, $\delta_m v$)



b) Pressure differences ($\delta_l p$, $\delta_m p$)



c) Mach number differences ($\delta_l M$, $\delta_m M$)

Fig. 6 Refinement parameter influence on grid for shock detection, 8% circular arc cascade (supersonic).

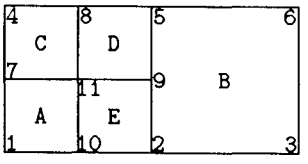


Fig. 7 Basic data structure.

Our approach has been to treat all possible interface configurations in a unique way. This is accomplished by disregarding "problematic" interface cells and using other existing cells instead in order to perform the interface calculations. Specifically, let us examine one type of interface in Fig. 8. The nodes a,b are integrated using the parent cell B instead of cells C and D and ignoring the center node c. Thereafter, the values at node c follow by interpolating from nodes a,b. A similar procedure is employed for each kind of interface. This treatment is not conservative locally. However, it has proven to be accurate and robust in the cases considered so far, even when a shock intersects an interface. It is also easily extended to the third dimension.

Adaptive Solution Procedure

The solution algorithm that was used consists of the following steps.

1) Initialize the field with a uniform, orthogonal coarse grid on which the Navier-Stokes description is applied.

2) Monitor the residual until it decreases to less than a prespecified value; detect the main flow features; refine the grid locally.

3) Continue computation on the updated grid using Euler/Navier-Stokes solvers for the inviscid/viscous cells.

4) Repeat steps 2 and 3 for a specified number of cycles.

5) March the solution to steady state.

At each embedding level, the error wavelengths corresponding to the cell sizes of that level are smoothed. In that sense, adaptation plays the role of a multigrid method.

Coupling of Grids

Two distinct types of cells are involved during the solution procedure. There are the locally finest cells that are used by the basic solver to integrate the equations and the multiple-grid coarser cells that are used by the multiple-grid algorithm after being created from the local fine cells by deleting every other grid line (Fig. 9). The locally finest cells may belong to either an unembedded or an embedded region. In Fig. 9, the indicated level numbers appear at the center of their respective cells.

The solution procedure starts from a sweep of the locally finest (0-level) cells. Then, the multiple-grid accelerator is used within the embedded regions (multiple-grid level 1). Finally, the multiple-grid operator is used throughout the entire domain (level 2). Notice that the interfaces are "invisible" after completing the multigrid levels within the embedded regions (level 1). The multiple-grid accelerator was used for all numerical calculations reported here.

Results

Model problems with flow past a circular arc cascade in both subsonic and supersonic flow have been used to evaluate the accuracy and efficiency of the suggested adaptation techniques.

Subsonic Flow

The flow at $M_\infty = 0.5$ and $Re = 8 \times 10^3$ was calculated using an initial 25×25 mesh with uniform spacing across the boundary layer and up to three levels of embedding allowed in both directions. The final grid is shown in Fig. 10. The area near the trailing edge proves to need less embedding because the boundary layer separates and the fluid is virtually stagnant with negligible stresses. The three peaks in the convergence history (Fig. 10) mark the upset introduced by the adaptive grid embedding process. It is to be noted that adaptation does not alter the overall slope of the residual decay curve.

Indeed, the presence of interfaces in the flowfield does not affect the solution (Fig. 11). In Fig. 11a, we see the separated velocity profile at the trailing-edge station with three marks

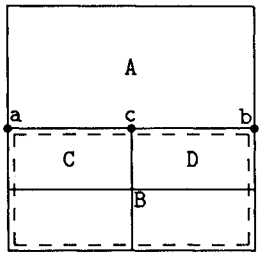


Fig. 8 Interface treatment.

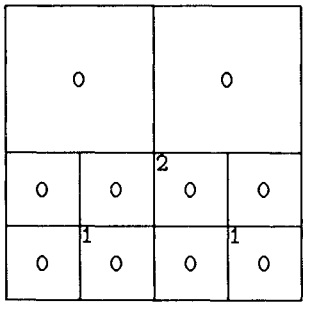
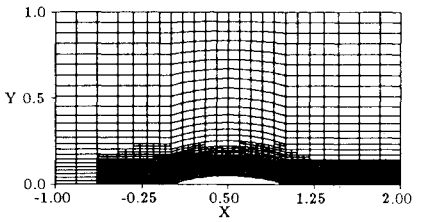
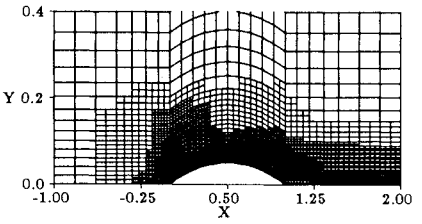


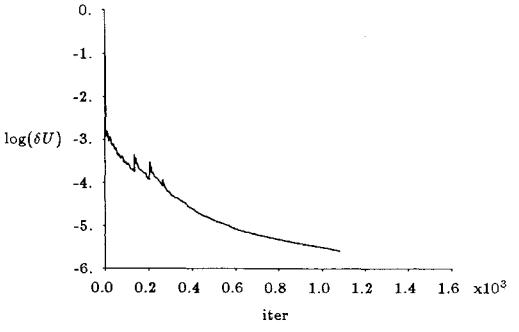
Fig. 9 Coupling of grids.



a) Entire domain

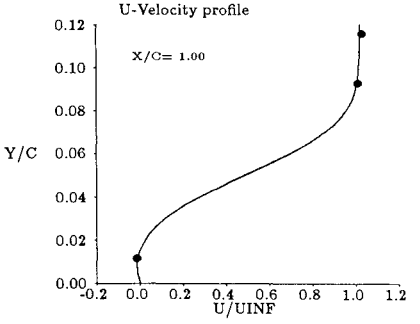


b) Enlarged vertical scale



c) Convergence history (rms residual in ρu)

Fig. 10 Embedded grids for three levels of adaptation, 10% circular arc cascade ($M_\infty = 0.5$).



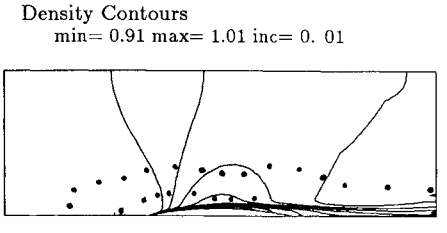


Fig. 11 Effect of interfaces on solution, 10% circular arc cascade (three-level adaptation); a) and b), • interface position.

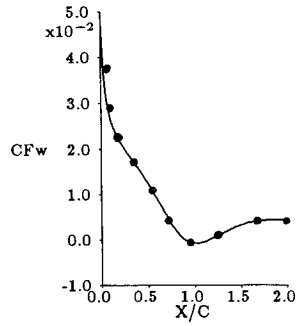
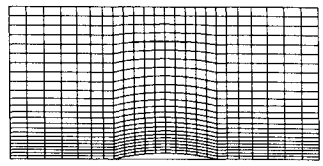
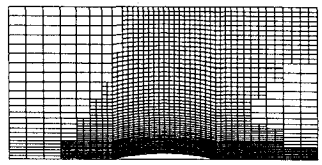


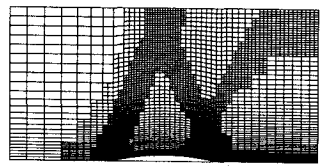
Fig. 12 Comparison of wall-shear distributions for directional and nondirectional grid adaptation, 10% circular arc cascade; — embedding in both directions at all levels, • directional embedding at the third level.



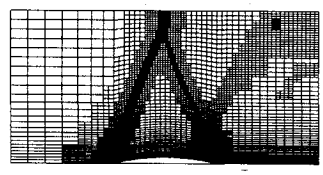
a) Initial grid (25 × 25)



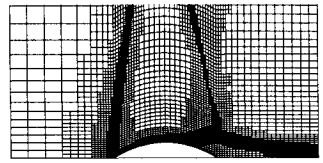
b) First embedding level



c) Second embedding level



d) Third embedding level



e) Third embedding level (vertical scale enlarged)

Fig. 13 Grid evolution, 8% circular arc cascade (supersonic).

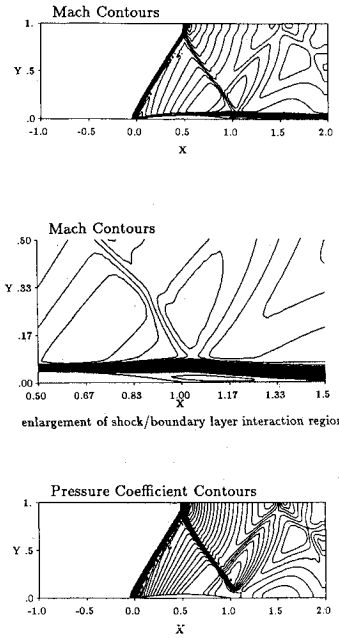
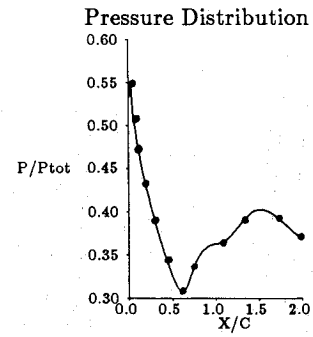


Fig. 14 Mach number and pressure contours for supersonic (M_∞ = 1.4) flow past an 8% circular arc cascade.



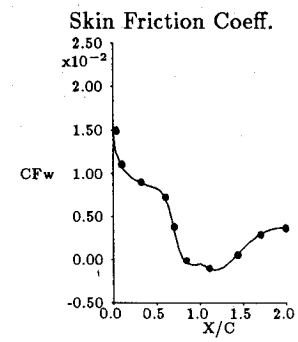


Fig. 15 Comparison of globally refined and adaptive embedding results, 8% circular arc cascade (M_∞ = 1.4); — two-level adaptive embedding, • globally fine grid (87 × 97).

indicating the interface positions. In Fig. 11b, the density contour plots are shown, and in neither figure are kinks observed due to the presence of interfaces. Of course, greater resolution is needed across the boundary layer than in the streamwise direction. Applying directional embedding across the boundary layer at the third level results in significant savings in the number of cells, with no apparent change in the results on comparing wall-shear distributions obtained using adaptation in either both directions or directionally (Fig. 12).

Supersonic Flow

An 8% circular arc cascade, with $M_\infty = 1.4$, $Re = 23 \times 10^3$, and again with a 25 × 25 initial mesh, was used with uniform spacing across the boundary layer and an allowance for three levels of embedding in both cell directions. An oblique shock forms at the leading edge as expected (Fig. 13) and is reflected at the upper-symmetry boundary. The reflected shock then interacts with the boundary layer at the trailing-edge region. The grid evolution is illustrated in Fig. 13 and demonstrates

how the embedding follows the detailed physics of the flow. Figure 13e provides an enlarged detail of the grid near the surface. The boundary layer is essentially "lifted" by the adverse pressure gradient that is induced by the reflected shock. Simultaneously, because of the effective corner created by the boundary layer, compression waves are formed upstream of the interaction region and coalesce into a weaker shock that impinges on the upper boundary. Note that the passage of the shock through interfaces does not induce any stability problems. Further flow details are shown in Mach number and pressure-coefficient contour plots (Figs. 14a–14c). Note that the slip line due to the Mach reflection at the upper boundary is clearly observed in the Mach number contour plot. Notice also the deflection of the reflected shock at the shock/boundary-layer interaction region and the separated recirculating region at the trailing edge. Expansion fans are formed at the interaction region because of the effective wedge formed by the separated boundary layer. The local embedding procedure appears to be effective in capturing the detailed physics of a flowfield in the presence of rather complicated multiple-scale phenomena.

In order to evaluate the accuracy in this case, a globally fine grid (97×97) corresponding to two-level embedding was employed. Comparison was made with results obtained using two levels of local, adaptive embedding. Figures 15a and 15b indicate excellent agreement for the surface-pressure and wall-shear distributions.

Adaptation Efficiency

The fact that the entire Navier-Stokes system is solved only within the viscous regions leads to a 20% time savings due solely to the equation adaptation scheme.

The directional embedding that was applied in the subsonic case at the third level led to a 40% decrease in cells compared to the number formed if embedding in both directions had been applied. This is not surprising in view of the fact that a cell divided in both directions adds three additional cells, whereas directional division adds only one. This results in quite significant savings in the number of cells, especially at higher adaptation levels when many new cells are created. Directional embedding should be even more beneficial for turbulent boundary layers where at least two nodes are needed within the viscous sublayer.

The example subsonic case on an equivalent globally fine mesh would require 140 h of MicroVax computer time, making the conservative assumption that the same number of iterations would be required as for the embedded case. With adaptation in both cell directions, only 11 h are required, and this reduces to 6 h with an allowance for directional embedding at the third level. Finally, 5 h are required if equation adaptation is applied as well. Overall, the adaptive techniques lead to a 28-times reduction in CPU time. The gain would be appreciably larger if more than three adaptation levels were permitted.

In the supersonic case, the reduction factor is 22. The slightly smaller advantage follows from the fact that there are more flow features to resolve, and these result in a larger number of embedded cells.

Conclusions

A conservative finite-volume discretization of the viscous terms of the Navier-Stokes equations has been developed using the same stencil as for the convective terms.

Grid and equation adaptation have been carried out for multiple and overlapping flow features.

The combined use of viscous, directional, and equation adaptation has indicated CPU time reduction factors of approximately 25 for subsonic and supersonic cascade examples for equivalent accuracies.

Acknowledgments

This work was supported by the Air Force Office of Scientific Research under Grant AFOSR-82-0136, Dr. J. D. Wilson technical monitor. The authors would also like to thank Prof. M. Giles for his comments.

References

[1]Berger, M. and Jameson, A., "Automatic Adaptive Grid Refinement for the Euler Equations," *AIAA Journal,* Vol. 23, April 1985, pp. 561–568.

[2]Dannenhoffer, J. F., III and Baron, J. R., "Adaptive Procedure for Steady State Solution of Hyperbolic Equations," AIAA Paper 84-0005, Jan. 1984.

[3]Dannenhoffer, J. F., III and Baron, J. R., "Grid Adaptation for the 2-D Euler Equations," AIAA Paper 85-0484, Jan. 1985.

[4]Dannenhoffer, J. F., III and Baron, J. R., "Robust Grid Adaptation for Complex Transonic Flows," AIAA Paper 86-0495, Jan. 1986.

[5]Lohner, R., Morgan, K., Peraire, J., and Zienkiewicz, O. C., "Finite Element Methods for High Speed Flows," AIAA Paper 85-1531, June 1985.

[6]Shapiro, R. A. and Murman, E. M., "Cartesian Grid Finite Element Solutions to the Euler Equations," AIAA Paper 87-0559, Jan. 1987.

[7]Ni, R.-H., "A Multiple Grid Scheme for Solving the Euler Equations," *AIAA Journal,* Vol. 20, Nov. 1982, pp. 1565–1571.

[8]Davis, R. L., Ni, R.-H., and Carter, J. E., "Cascade Viscous Flow Analysis Using the Navier-Stokes Equations," AIAA Paper 86-0033, Jan. 1986.

[9]Swanson, R. C. and Turkel, E., "A Multistage Time-Stepping Scheme for the Navier-Stokes Equations," AIAA Paper 85-0035, Jan. 1985.

[10]Carter, J. E., "Numerical Solutions of the Navier-Stokes Equations for the Supersonic Laminar Flow Over a Two-Dimensional Compression Corner," NASA TR R-385, 1972.

[11]Chima, R. V. and Johnson, G. M., "Efficient Solution of the Euler and Navier-Stokes Equations with a Vectorized Multiple-Grid Algorithm," *AIAA Journal,* Vol. 23, Jan. 1985, pp. 23–32.

[12]Rhie, C. M., "A Pressure Based Navier-Stokes Solver Using the Multigrid Method," AIAA Paper 86-0207, Jan. 1986.