Solution-Adaptive Grid Procedure for High-Speed Parabolic Flow Solvers

Albert D. Harvey III* and Sumanta Acharya†
Louisiana State University, Baton Rouge, Louisiana 70803
Scott L. Lawrence‡
NASA Ames Research Center, Moffett Field, California 94035
and
Samson Cheung‡
MCAT Institute, Moffett Field, California 94035

A solution-adaptive grid procedure based on an error equidistribution scheme is developed and applied to a parabolized Navier-Stokes solver. An improved method of selecting weighting functions is introduced that involves normalizing a combination of flowfield gradients and curvature of a number of dependent variables and then selecting the largest at each point. The scheme redistributes grid points line by line with grid point motion controlled by forces analogous to tensional and torsional spring forces with the spring constants set equal to the weighting functions. Torsional terms are functions of the grid point positions along neighboring grid lines and provide grid smoothness and stability. A grid-fitting scheme is introduced for external flows in which the number of grid points in the freestream are reduced to a minimum. Results for several problems are presented to demonstrate the improvements obtainable with the solution-adaptive grid procedure.

Introduction

THE search for more accurate and computer efficient solutions to complex problems in fluid flow and heat transfer has led to the utilization of improved discretization methods. In many such problems, there occur regions in the physical domain where the dependent variables exhibit large changes in gradient and/or curvature. For supersonic flowfields, these regions could include shocks, expansion fans, contact surfaces, as well as boundary layers, which, when present simultaneously, can produce physically complex interactions difficult to simulate numerically without strategic grid point placement. In most cases, the locations of these regions are not known a priori, and, hence, the initial distribution of grid points is unsatisfactory. What is needed is a method for redistributing the grid points as the solution evolves. This redistribution should sufficiently cluster points in high-gradient regions while not completely devoiding other regions of grid points.

Numerous studies on adaptive grid techniques are presently available. Thompson[1] and Hawken[2] independently provide complete surveys of the most widely accepted methods. As noted by Thompson,[1] most adaptive grid procedures attempt to equally distribute some measure of the solution error; however, each differs in its individual approach.

The most popular approach has been to maintain a constant product of a weighting function, which is proportional to an error measure, and the grid interval throughout the solution domain. Dwyer et al.[3] used a linear combination of the gradients and curvature of a dependent flow variable as the error measure. Rai and Anderson[4] and Greenberg[5] used an attraction/repulsion method where grid points possessing weighting functions larger than some average value attract each other while those points with values less than the average repel each other. Gnoffo[6] introduced a method analogous to a system of tension springs set between grid points whose spring constants are functions of an error measure. Nakahashi and Deiwert[7-9] extended the method to include torsion spring forces that relate grid point positions along neighboring lines and thus provide for grid smoothness. This method has also been successfully used by Djomehri and Deiwert[10] and Davies and Venkatapathy.[11]

Variational methods have also been used as a method of grid adaptation. This approach involves the minimization of an integral whose integrand is a function of some error estimate of the evolving solution. Thompson[1,12] has shown that this minimization process is equivalent to the solution of an elliptic set of partial differential equations (Poisson's equations) with the nonhomogeneous terms proportional to the error measure. The method of Gnoffo[6] can be described in terms of variational methods as the minimization of the energy in a system of springs set between grid points.

The objective of this study is to apply an efficient solution-adaptive grid scheme to a parabolic space-marching flow solver. Solutions to spatially parabolic equations are obtained by marching in space rather than time and, therefore, are obtained much more efficiently than solutions to the time-dependent Navier-Stokes equations. The present flow solver, developed by Lawrence,[13,14] applies upwinding to the parabolized Navier-Stokes equations and is based on Roe's scheme.[15]

The present paper describes the development of a solution-adaptive grid procedure for a parabolized Navier-Stokes solver. Results obtained using the adaptive grid method are presented along with results obtained on a fixed grid to allow for evaluation of the efficiency and accuracy of the new scheme.

Flow Solver

As stated earlier, the present flow solver involves the integration of the parabolized Navier-Stokes (PNS) equations, in which the spatial propagation of flowfield information is locally modeled using a steady version of Roe's scheme. The PNS equations are obtained from the steady-state Navier-Stokes equations by neglecting streamwise viscous derivatives and by extracting the portion of the streamwise pressure gra-

Received July 17, 1990; revision received Nov. 28, 1990; accepted for publication Dec. 3, 1990.

*Research Assistant, Mechanical Engineering.
†Professor, Mechanical Engineering. Member AIAA.
‡Research Scientist. Member AIAA.

dient term that is responsible for introducing ellipticity into the equations. For a two-dimensional flow, they can be written with respect to a generalized coordinate system as

$$\frac{\partial \boldsymbol{E}^*}{\partial \xi} + \frac{\partial \boldsymbol{E}^p}{\partial \xi} + \frac{\partial (\boldsymbol{F}_i - \boldsymbol{F}_v)}{\partial \eta} = 0 \tag{1}$$

where $\boldsymbol{F}_i$ and $\boldsymbol{F}_v$ represent the inviscid and viscous crossflow numerical fluxes, respectively. The ξ coordinate is defined as the streamwise direction and η is the crossflow direction. The terms $\boldsymbol{E}^*$ and $\boldsymbol{E}^p$ are the result of employing the Vigneron technique[16] to split the streamwise flux vector. They are defined by

$$\boldsymbol{E}^* = \begin{bmatrix} \rho\hat{U} \\ \rho u\hat{U} + \left(\frac{\xi_x}{J}\right)\omega p \\ \rho v\hat{U} + \left(\frac{\xi_y}{J}\right)\omega p \\ (E_t + p)\hat{U} \end{bmatrix}$$

$$\boldsymbol{E}^p = \begin{bmatrix} 0 \\ \left(\frac{\xi_x}{J}\right)(1 - \omega)p \\ \left(\frac{\xi_y}{J}\right)(1 - \omega)p \\ 0 \end{bmatrix}$$

with

$$\hat{U} = \left(\frac{\xi_x}{J}\right) u + \left(\frac{\xi_y}{J}\right) v$$

The vector $\boldsymbol{E}^*$ is the resultant streamwise flux, and $\boldsymbol{E}^p$ is the portion of the original streamwise flux responsible for introducing ellipticity into the equations through the subsonic boundary layer. It is shown in Ref. 16 that Eq. (1) is hyperbolic parabolic with respect to the dependent vector $\boldsymbol{E}^*$, provided that ω satisfies the relation

$$\omega = \min\left[1, \frac{\sigma\gamma M_\xi^2}{1 + (\gamma - 1)M_\xi^2}\right]$$

where M_ξ is the Mach number in the ξ-coordinate direction and σ is a safety factor.

Equation (1) is integrated using the methods outlined in Refs. 13 and 14 by employing a space-marching procedure in which the solution at each successive ξ plane is solved using the upstream solution as the initial condition.

Grid Adaptation Procedure

Numerical Method

Since the algorithm described in the previous section marches in space rather than time, each streamwise location of the flowfield is solved with the upstream condition as the corresponding initial condition. An appropriate adaptive grid scheme would construct a grid plane at the streamwise location of interest, solve for the flowfield variables at this location based on the upstream conditions, and then recluster the grid points based on the gradients and/or curvature of the computed flow. An improved solution can then be obtained on the resulting refined grid plane. A new grid plane can then be constructed (with grid points proportioned to that of the previous plane) at the next streamwise location and then the procedure can be continued.

The basic philosophy adopted in this paper to recluster grid points is somewhat similar to that of Gnoffo[6] and is analogous to minimizing the energy in a system of tension springs set between grid points. The spring constants represent functions of the gradient and/or curvature of a selected dependent variable. Nakahashi and Deiwert[7] later refined Gnoffo's method by introducing the concept of a torsion spring to reduce grid skewness between grid lines or planes. The method is based on variational principles; however, the problem is posed by Nakahashi and Deiwert in an algebraic unidirectional manner by applying tension and torsion spring forces proportional to flow gradients at each grid point along a fixed coordinate line and solving for the equilibrium position of the resulting grid system. The resulting system of equations is tridiagonal, which can be solved efficiently for the final positions of the grid points.

The grid adaptation technique in this paper is based on the error equidistribution method and involves the redistribution of grid points such that an error measure represented by a positive weighting function w_i is equally distributed over a coordinate line,

$$w_i \Delta s_i = K \tag{2}$$

where w_i is the weighting function based on flow properties and, in the terminology of Refs. 9 and 11, represents the spring constant with K as the resultant force. The grid interval Δs_i is defined as the distance between adjacent grid points along a line of constant computational coordinate. Figure 1 illustrates the grid topology surrounding the grid point (i, j).

The weighting function w_i is defined as a function of a selected normalized flow property $\bar{f}$ such that

$$w_i = 1 + A\bar{f}_i^B \tag{3}$$

where A and B are constants related to the desired minimum and maximum grid spacings; their expressions are given in Ref. 11. The choice of the normalized flow property $\bar{f}$ is an important issue and is described in the next section.

Summing both sides of Eq. (2), solving for K, and then substituting the resulting expression for K back into Eq. (2) yields

$$\Delta s_i = s_{\max} \Big/ \left(w_i \sum_{l=1}^{n} \frac{1}{w_l}\right) \tag{4}$$

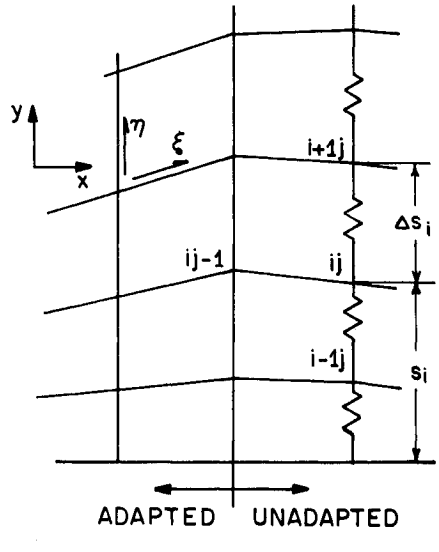

Fig. 1 Adaptive mesh geometry.

where s_{max} is the total length of the coordinate line. Equation (4) is an expression for the new grid point locations based on local flow properties. Thus, where the error measure w_i is large, Δs_i will turn out to be small and, therefore, the local truncation error at a point, whose leading term is proportional to $w_i(\Delta s_i)^n$, will tend to be equidistributed over the complete domain.

If Eq. (4) is used alone to determine the new grid spacing along each grid line, the grid would quickly become skewed due to the lack of dependence of the adapted grid points on the previous grid line. To ensure a smoother grid, there needs to be a relationship between the current grid adaptation line (where the grid adaptation is to be performed) and the previously adapted grid line. Torsional terms are introduced to accomplish this task. Figure 2 graphically illustrates the addition of a torsion spring to the system. This procedure, developed by Nakahashi and Deiwert,[7] relates grid points along adjacent grid lines, thus providing for a smoother grid. Performing a force balance on the system of tension and torsion springs acting on the point (i, j) of Fig. 2, the following equation is obtained:

$$w_i(s_{i+1} - s_i) - w_{i-1}(s_i - s_{i-1}) - C_t\theta_{i,j-1} = 0$$

where $-C_t\theta_{i,j-1}$ denotes the force due to the torsion spring. The torsion angle θ can be approximated by $\theta_i \approx (s_i - s_i')/|DA'|$, where DA' is the length of the line segment from the point $(i, j-1)$ to the point of intersection of the torsional reference line with the current adaptation line. The quantity s_i' is the point where the streamwise torsional reference line crosses the current adaptation line and, physically, corresponds to the point A' of Fig. 2. Substituting this expression for the last term in the previous equation and rearranging yields

$$w_{i-1}s_{i-1} - (w_{i-1} + w_i + \tau_i)s_i + w_is_{i+1} = -\tau_i s_i' \quad (5)$$

which is tri-diagonal and can be solved easily for s_i, the final grid point positions. Equation (5) is solved by employing an iterative procedure, interpolating the weighting functions at each iteration until the grid becomes stationary. The constant $\tau_i = C_t/|DA'|$ is the magnitude of the torsional influence from the streamwise direction. The choice of C was observed to greatly affect consistency in grid smoothness and continuity. Nakahashi and Deiwert[8] set C proportional to the average of the tension spring constants along a line. Davies and Venkatapathy[11] assumed that C is proportional to the maximum w_i and the local aspect ratio of the grid cell. This method is adopted in the present study with the modification that C is also assumed proportional to the length of the torsion spring $|DA'|$. As a result, τ_i is expressed in the following manner:

$$\tau_i = \frac{\lambda w_{max}(s_{i+1,j-1} - s_{i-1,j-1})}{2|DA'|}$$

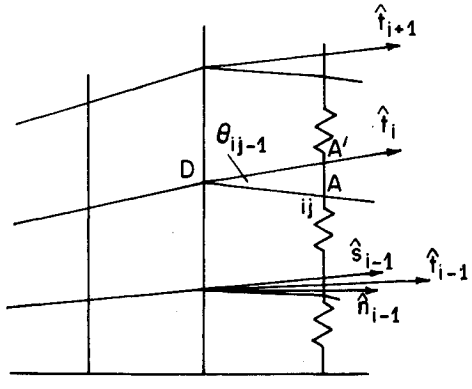

Fig. 2 Tensional and torsional spring system.

The constant λ is used as a user input to control the amount of torsion or rigidity the system possesses and is usually on the order of 10^{-1}–10^{-4}. A value of zero for λ completely removes torsional dependence and allows grid points to move without influence from adjacent grid lines. The torsional reference line DA' is constructed in the direction of the torsional vector $\hat{t}_i$, which in this paper is taken to be proportional to the orthogonal and straightness vectors (see Fig. 2). The orthogonal vector $\hat{n}_i$ is defined as a vector in the positive x direction and represents the direction that the grid must take in order to maintain a modest degree of overall orthogonality. The straightness vector $\hat{s}_i$ at the point (i, j) is the average of two vectors. The first of these vectors is established from the point $(i, j-2)$ to the point $(i, j-1)$ and the second is taken from the point $(i, j-1)$ to the point (i, j). The torsional vector $\hat{t}_i$ is defined as

$$\hat{t}_i = C_t\hat{s}_i + (1 - C_t)\hat{n}_i$$

where C_t is a user input. A value of C_t close to zero leads to a more orthogonal grid, whereas C_t closer to unity adds more straightness to the grid. The torsional reference line DA' acts in the direction of the torsional vector $\hat{t}_i$.

As noted earlier (see Ref. 11), the constants A and B used to compute the weighting functions defined by Eq. (3) were derived with the torsionless equations (Eq. 4). With the addition of the torsion terms, the computed minimum and maximum grid spacing will be somewhat altered and may not correspond to the requested Δs_{min} and Δs_{max}. Using the terminology found in Ref. 11 to help alleviate this problem and provide for improved convergence, a weighting of the form

$$w_i = (1 + A\bar{f}_i^B)w_i$$

is used instead of Eq. (3). When Eq. (5) is solved, each Δs_i is checked to see if it lies within the user requested limits of Δs_{min} and Δs_{max}. If so, the value of w_i is set equal to unity. If Δs_i is not within these limits, the conditional operator

$$w_i = \begin{cases} \frac{1}{2}\left(\frac{\Delta s_i}{\Delta s_{min}} + 1\right) & \text{for} \quad \Delta s_i < \Delta s_{min} \\ \frac{1}{2}\left(\frac{\Delta s_i}{\Delta s_{max}} + 1\right) & \text{for} \quad \Delta s_i > \Delta s_{max} \end{cases} \quad (6)$$

is used to define w_i. Although Eq. (6) does not guarantee Δs_i will lie within the user requested maximum and minimum grid spacings, their use is observed to enhance the stability of the algorithm.

Special treatment of weighting functions is also required at the boundaries of the adaptation domain, which can include all, or only a subset, of the entire calculation domain. Either technique requires the adapted mesh near the boundary to be smooth and continuous. This task is accomplished by adjusting the edge weighting function to correspond with the desired edge spacing at each iteration of Eq. (5). If wall spacing exceeds a specified value Δs_{edge} by a prescribed tolerance, the weighting function at the wall is then adjusted in the following manner:

$$w_{edge} = \frac{w_1\Delta s_1}{\Delta s_{edge}}$$

Surrounding weighting functions are then smoothed to provide continuous spacing outward from the boundary of the adaptation domain.

Selection of Weighting Functions

The selection of weighting functions and the normalized flow variable f used to drive the grid adaptation is considered next. Ideally, it is desired to minimize the truncation error

and distribute the error uniformly over the computational domain. Although the truncation error cannot be estimated accurately, it is generally proportional to $(\Delta s)^n \partial^n \phi / \partial s^n$, where n denotes the order of accuracy.

As in many of the studies on solution-adaptive gridding, a linear combination of gradients and curvature of selected dependent and/or physical variables are used here as weighting functions. Finding a flowfield variable that will consistently represent solution error in all regions of the solution domain is difficult, if not impossible. Some variables change very rapidly in certain regions, but remain fairly smooth in other regions where another variable could be undergoing severe change. In the present study, an algorithm is developed so that it automatically chooses which variables are to represent the weighting function at each grid point location. The scheme computes gradients and/or the curvature of each user specified variable and then, after normalizing each of these, chooses the largest in magnitude to represent the weighting function at each point. This method has proven superior to conventional weighting function selection processes, especially where large changes in different dependent variables exist in different regions of the flow.

The normalized flow variable $\bar{f}_{i,j}$ at a cross-stream location i and a streamwise location j is computed as

$$\bar{f}_{i,j} = \frac{f_{i,j} - f_{\min}}{f_{\max} - f_{\min}}$$

where $f_{i,j}$ is a linear combination of the gradients and curvature of the dependent flow variables ϕ_k

$$f_{i,j} = \alpha\left(\frac{\partial \phi_i}{\partial s}\right) + \beta\left(\frac{\partial^2 \phi_i}{\partial s^2}\right) \tag{7}$$

where

$$\frac{\partial \phi_i}{\partial s} = \max_k \left|\frac{\partial \phi_i}{\partial s}\right|_k, \qquad \frac{\partial^2 \phi_i}{\partial s^2} = \max_k \left|\frac{\partial^2 \phi_i}{\partial s^2}\right|_k$$

for all specified ϕ_k (i.e., Mach number, pressure, density, etc.). Hence, the gradients of all selected flow variables are computed, then the maximum is chosen to represent $f_{i,j}$ at each point along the current adaptation line. A similar procedure is done for the curvatures if β is not equal to zero. The constants α and β are specified by the user. For all test cases studied, α was given a value of unity, whereas β was given a value between 0.1 and 0. Nonzero β increases grid point density in regions where the gradient is changing rapidly. The net effect is to thicken slightly the clustering regions. Increased grid clustering is also observed at the edge of boundary layers when nonzero values of β are used.

The solution-adaptive grid procedure is done in a two-step manner. Once a solution has been obtained at a certain streamwise location, the first step in obtaining the solution at the next streamwise location is to project the just obtained adapted grid downstream. An initial solution is then obtained on this grid. Based on this solution, the new grid is adapted and, in the second step, a refined solution is recomputed. Once this is done, the calculation proceeds to the next streamwise location and the two-step procedure is repeated.

A summary of user required parameters necessary in implementing this solution-adaptive grid algorithm is given in the following: maximum allowable grid spacing $\Delta s_{\max}$; minimum allowable grid spacing $\Delta s_{\min}$; the amount of torsional dependence λ; degree of grid straightness or orthogonality desired C_t; and adaptation variables ϕ_k (selected dependent flow variables). In later sections, numerical values for these parameters are presented that were used in obtaining solution-adaptive results for a variety of two-dimensional problems.

Grid-Fitting Algorithm

The grid-fitting algorithm is developed for high Reynolds number external flow situations with freestream conditions enforced at the far field or outer boundary. At each marching station, to successfully implement these boundary conditions, a grid plane must first be constructed that extends a sufficient distance into the freestream. It would obviously be desirable to minimize the number of grid points placed in the freestream, keeping a maximum number of points inside the important region of the flowfield.

The PNS solver used in this study incorporates a shock-capturing scheme in which a computational grid is stretched a sufficient distance into the freestream and shocks and other discontinuities evolve from the governing equations without any special treatment. Conventional shock-fitting schemes use the bow shock as a flowfield boundary. Normal shock relations can then relate freestream conditions to those downstream of the shock. These shock-fitting schemes can effectively reduce the calculation domain by using the evolving bow shock as the flowfield boundary, provided an initial shock position can be reasonably estimated. The grid-fitting scheme introduced in this section incorporates the reduced calculation domain inherent in shock-fitting schemes with the relatively simple far-field boundary condition of the shock-capturing method. The present grid-fitting scheme locates the innermost bound of the freestream conditions at the current marching station by scanning outward along the marching station, comparing flowfield conditions at each point to freestream conditions. The outermost grid line is then adjusted to a prescribed distance above this newly defined freestream boundary, minimizing the cross-stream calculation domain. The interior grid points are then redistributed or adapted inside this reduced domain.

This procedure can be summarized as follows. A preliminary solution is first obtained on an initial grid at the current marching station. A search of flowfield variables is then performed outward from the solid boundary to find the point where freestream conditions are first encountered. The outermost grid point is then placed a small distance above this position (which is a prescribed percentage of the total length of the grid line). It is in this manner that the geometrical boundary of the solution domain is altered during the course of the calculations, reducing the number of unnecessary grid points present in the freestream and moving them to more important regions of the flowfield. Next, the interior grid points are redistributed inside this reduced domain using the adaptive technique described in previous sections. A new, improved solution is then computed on the refined mesh. The refined mesh (including the total height) is then projected downstream to the next marching station location and the procedure is repeated. At each marching station, the outermost grid boundary is monitored, maintaining a minimum number of grid points outside the shock wave. It is important to point out that use of the grid-fitting algorithm described in this section eliminates the need for an initial grid. All that is needed is a surface grid and a grid point distribution at the starting plane (initial marching station).

Test Results

Hypersonic Compression Corner

The first test case involves hypersonic laminar flow over a 15-deg compression corner. This geometry has been examined by other investigators. Lawrence et al.[14] have computed numerical results for this case. This case has also been studied experimentally by Holden and Moselle,[17] and the resulting pressure and heat transfer are used here for comparison with present results. The flow conditions are the following: $M_\infty = 14.1$; $\tilde{l} = 0.439$ m; $\tilde{T}_\infty = 72.2$ K; $\tilde{T}_w = 297$ K; $Re_{\tilde{l}} = 1.04 \times 10^5$; $Pr = 0.72$; and $\gamma = 1.4$. Re_t is the freestream Reynolds number based on the distance from the leading edge to the beginning of the ramp. The flowfield contains multiple shock

waves, which require a high degree of grid resolution to be captured clearly. Mach number and pressure were selected as the ϕ_k of Eq. (7) ($k = 2$) and α and β set equal to 1.0 and 0.1, respectively. An initial plane was generated at $x = 0.015$ m by exponentially stretching 45 grid points in the normal direction. The solution process was started at this location from freestream conditions using a step size of 10^{-4} m. Grid adaptation was initiated at $x = 0.05$ m, where the marching step size was increased to 10^{-3} m. The maximum and minimum allowable grid spacings Δs_{max} and Δs_{min} were chosen to be 4.5 and 0.3 times the average upstream grid spacing, respectively. The torsional parameter λ was set at 5×10^{-5} and the orthogonality parameter C_t was set to ½. Figures 3 show the geometry and the resulting grids. For clarity, in each part of Figs. 3, every other grid line is printed in the normal direction, whereas every tenth grid line is shown in the streamwise direction. Figure 3*a* shows the grid structure without adaptive gridding. In Fig. 3*b*, grid adaptation is performed, but without any grid fitting. The outermost grid line of Fig. 3*c* was positioned using the grid-fitting algorithm described earlier. Gradients of density and streamwise velocity component were used to locate the innermost bound for the freestream conditions. Figures 4 illustrate the region surrounding the bow-ramp-shock interaction. All normal lines are shown. For the adapted cases, point density has increased in the appropriate regions to resolve the shock waves. The grid-fitted adapted case shows increased grid point density in virtually all regions of the flow. For this case, increased grid clustering is evident around the expansion fan emanating from the point of intersection of the two shock waves. In both adapted cases, wall spacing was allowed to decrease if necessary, but not allowed to increase beyond 10^{-4} m.

Figures 5 show contours of constant pressure in the region of interest around the shock wave interaction. The adapted case shows increased shock wave resolution over the unadapted case, and even over the unadapted case employing twice as many grid points. The pressure contour oscillations evident in the ramp shock region of Fig. 5a are not present in Figs. 5b and 5d due to the ability of the grid adaptation scheme to identify these high-gradient regions and locally increase grid point density. Another reason for the absence of this waviness in the adapted cases is the improved alignment of the grid with the shock waves. This alignment of the shocks with the computational grid has a stabilizing effect on the PNS solver. The pressure contours of Figs. 5 clearly illustrate the intersection of the bow shock with the ramp shock. These features are less clearly resolved for the unadapted cases. The figure corresponding to the grid-fitted adapted mesh algorithm developed in this paper appears to be the most optimal one in terms of clearly resolving shock waves.

Pressure profiles at $x = 0.7$ m are shown in Fig. 6. The leading edge shock is marked by a slight compression, which is followed by a sharp compression due to the ramp shock. Each grid point is marked by a symbol to show the migration of points into the high-gradient regions. Because of the increased point density in these regions for the adapted case, the corresponding pressure profiles show improvements in shock definition over the fixed grid cases, even for the fixed grid employing twice the number of grid points. The adapted, not grid-fitted, case would show similar point distribution. The grid-fitted case also shows improvements over the adapted case without grid fitting in the leading-edge shock region.

In Figs. 7 and 8, wall pressure and heat transfer coefficient distributions for adapted and unadapted cases are compared with the experimental results obtained by Holden and Moselle.[17] The pressure coefficients of Fig. 7 were computed in the following manner

$$C_p = \frac{p_w}{\rho_\infty V_\infty^2}$$

Heat transfer coefficients are defined by

$$C_h = \frac{\mu_w}{PrRe_\infty} \frac{\sec\theta}{\frac{1}{2}(\gamma - 1)M_\infty^2 + 1 - T_w} \frac{\partial T}{\partial y}$$

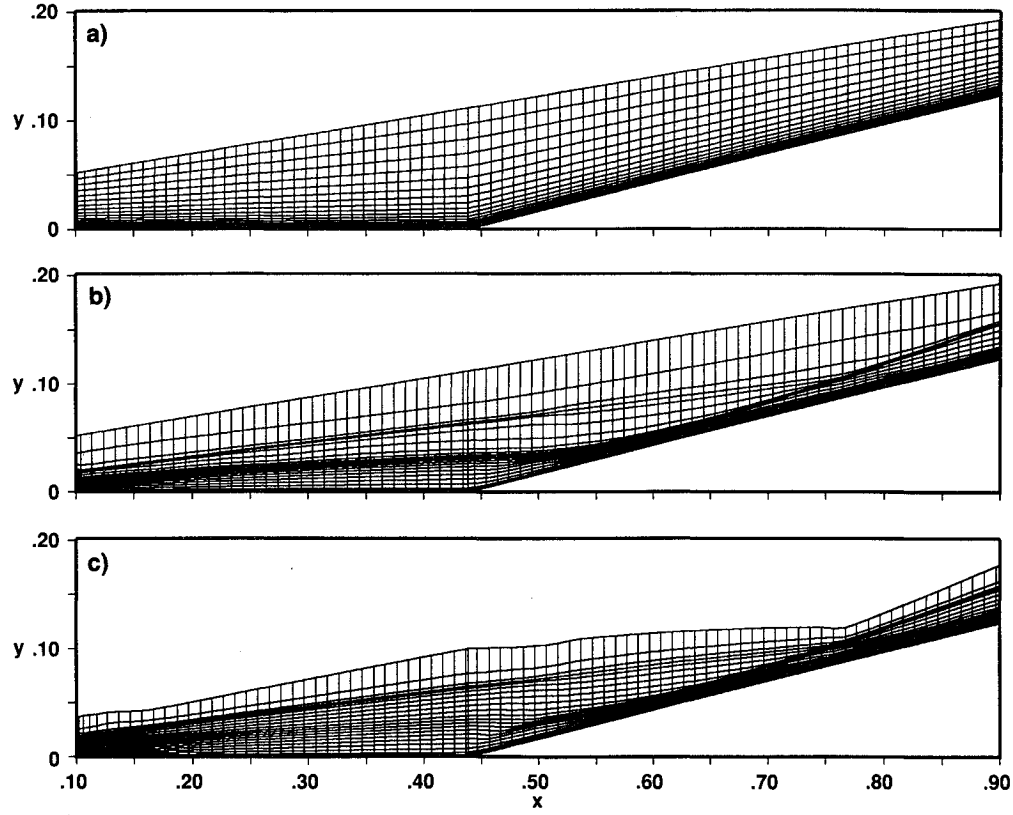

Fig. 3 Computational grids for compression corner: a) not adapted; b) adapted; c) adapted and grid fitted.

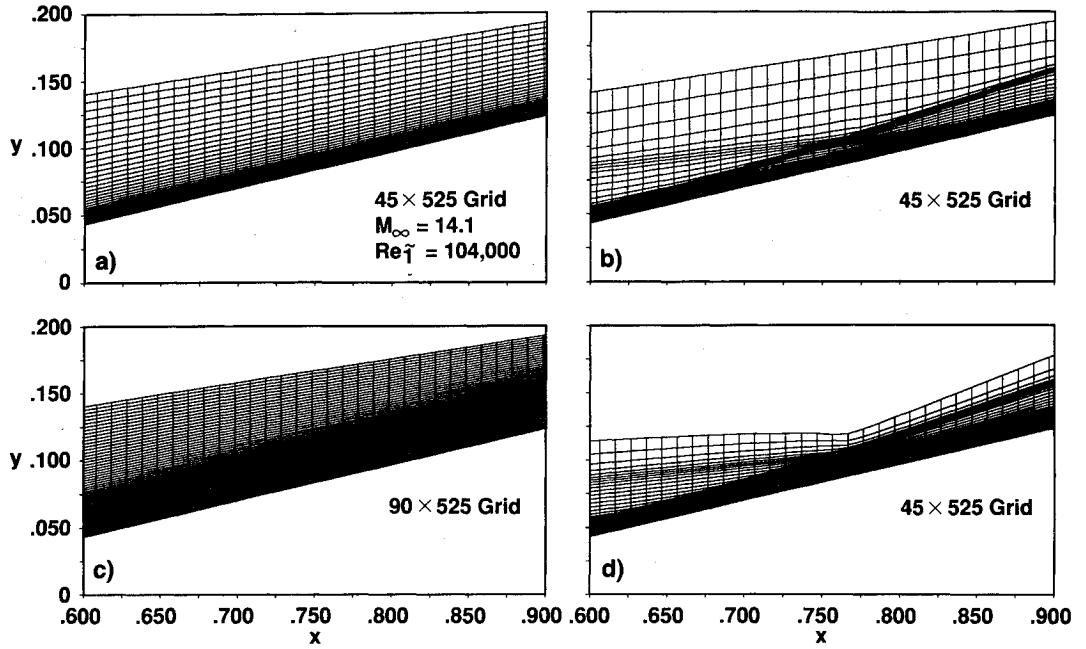

Fig. 4 Computational grids in the vicinity of the shock interaction region: a) without grid adaptation; b) with grid adaptation; c) without grid adaptation; d) with grid adaptation and grid fitting (all dimensions are in meters).

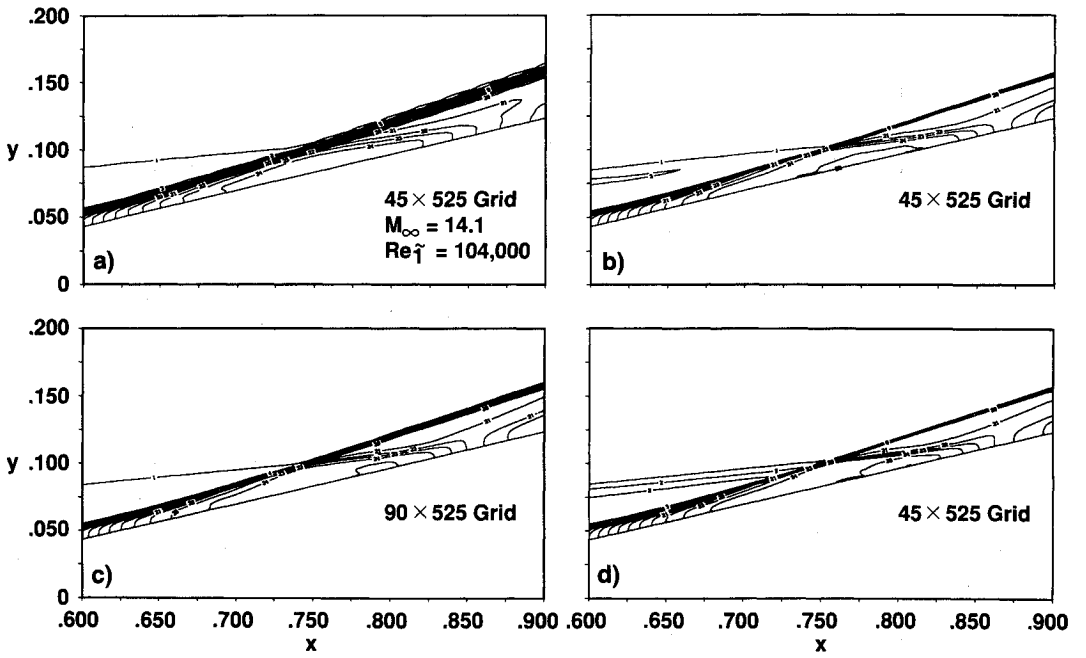

Fig. 5 Pressure contours in the shock interaction region: a) without grid adaptation; b) with grid adaptation; c) without grid adaptation; d) with grid adaptation and grid fitting.

where subscript w denotes values at the wall and θ is the angle between the y axis and a line normal to the wall. The present results show reasonable agreement with the experimental results, although both pressure and heat transfer coefficient distributions are slightly overestimated, more so for the heat transfer (Fig. 8). A slight improvement in pressure coefficient is noticed in the region directly above the base of the ramp. This can possibly be attributed to the slight reduction in wall spacing that occurs in the adapted cases. As expected, both pressure and heat transfer coefficient distributions for the unadapted case employing 90 grid points lie between the adapted cases and the unadapted case that employs only 45 grid points.

Computational effort was measured on a Cray Y-MP/832 computer. The unadapted case employing 45 grid points in the normal direction required 17.0 s of CPU time, whereas the adapted cases required 36.4 s. The unadapted case employing 90 grid points used 34.9 s of CPU time. The use of the grid-fitting algorithm added no significant effort to the computations of the solution-adapted case. The unadapted case employing 90 grid points and the adapted case employing 45 points used approximately the same CPU time; however, shock wave resolution is significantly better for the adapted cases.

Hypersonic Inlet

A planar, cowl-type inlet geometry was studied to further examine performance of the grid adaptation procedure. This geometry (Fig. 9) is composed of two compression surfaces

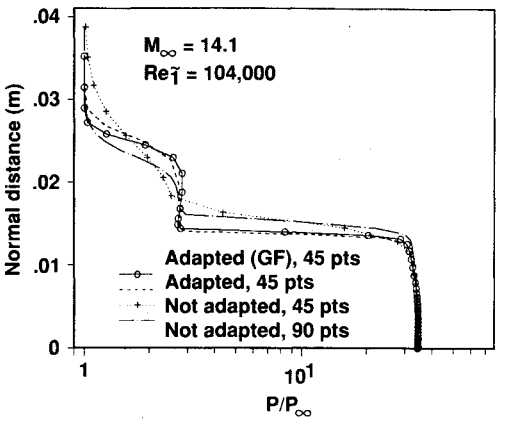

Fig. 6 Comparison of pressure profiles (x = 0.7 m).

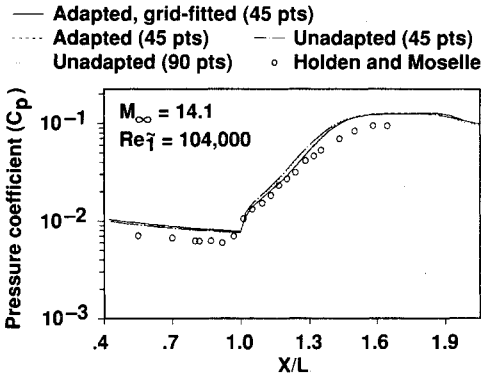

Fig. 7 Comparison of wall pressure coefficients.

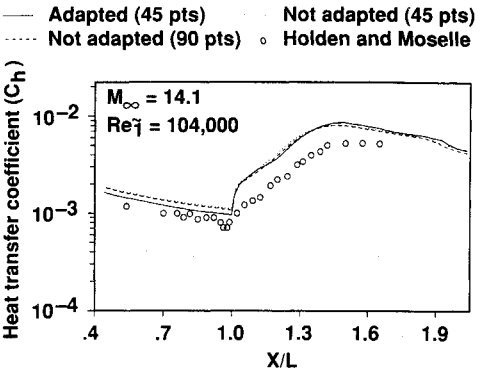

Fig. 8 Comparison of heat transfer coefficients.

followed by a narrow inlet channel. The cowl surface begins at x = 0.9 m (grid points in the freestream, upstream of this x location in Fig. 9, have been omitted). Because of the internal nature of this configuration, the outer grid-line adjustment technique is not appropriate and was not employed here. This case was chosen to illustrate the ability of the present adaptation method to resolve shock wave and shock-boundary-layer interactions. The freestream conditions are M_∞ = 6.5; $\tilde{l}$ = 1.0 m; $\tilde{T}_\infty$ = 702 K; $\tilde{T}_y$ = 1216 K; $Re_{\tilde{l}}$ = 2.0×10^5; Pr = 0.72; and γ = 1.4. $\tilde{l}$ is the distance to the downstream end of the second compression surface. No-slip boundary conditions were applied along the entire upper wall, whereas at the lower boundary, freestream conditions were applied on the forebody (i.e., upstream of the inlet plane located at x = 0.9 m). At the inlet face, no-slip conditions were introduced. The grid consists of 80 grid points clustered at both upper and lower flowfield boundaries using a hyperbolic tangent stretching function.[12] In order to provide grid continuity at the inlet plane, grid stretching was extended the entire length of the geometry; to clarify the inlet plane lo-

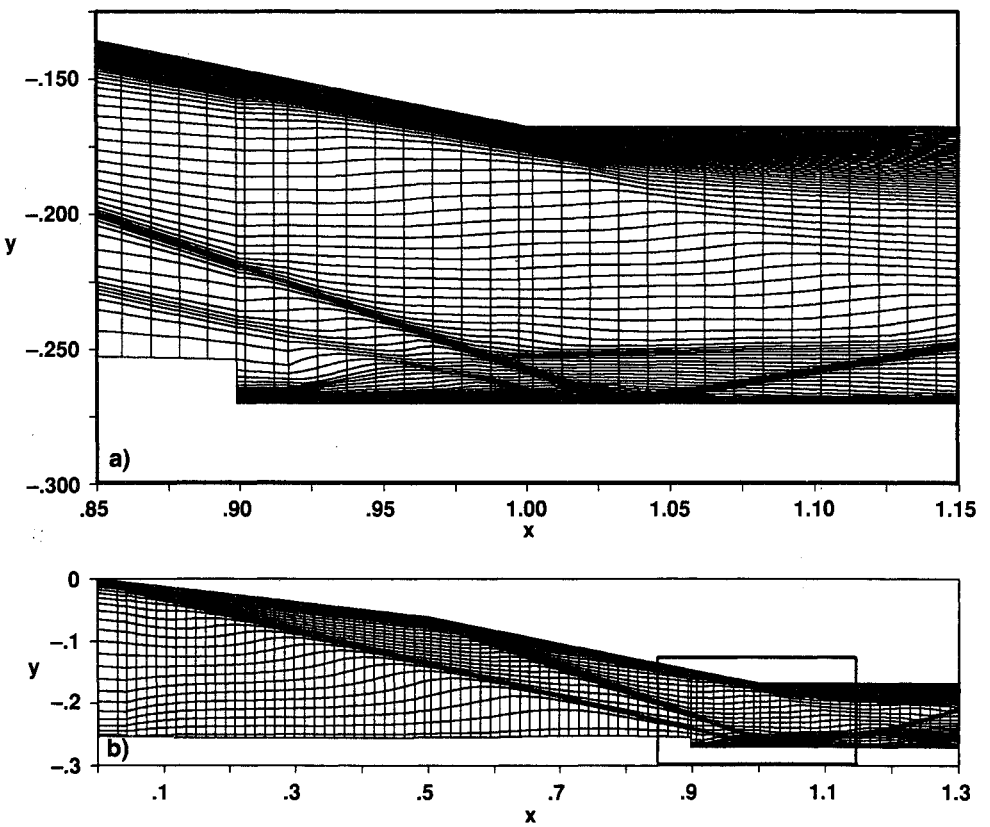

Fig. 9 Computational grid for planar inlet: a) geometry, b) inlet plane region (dimensions are in meters).

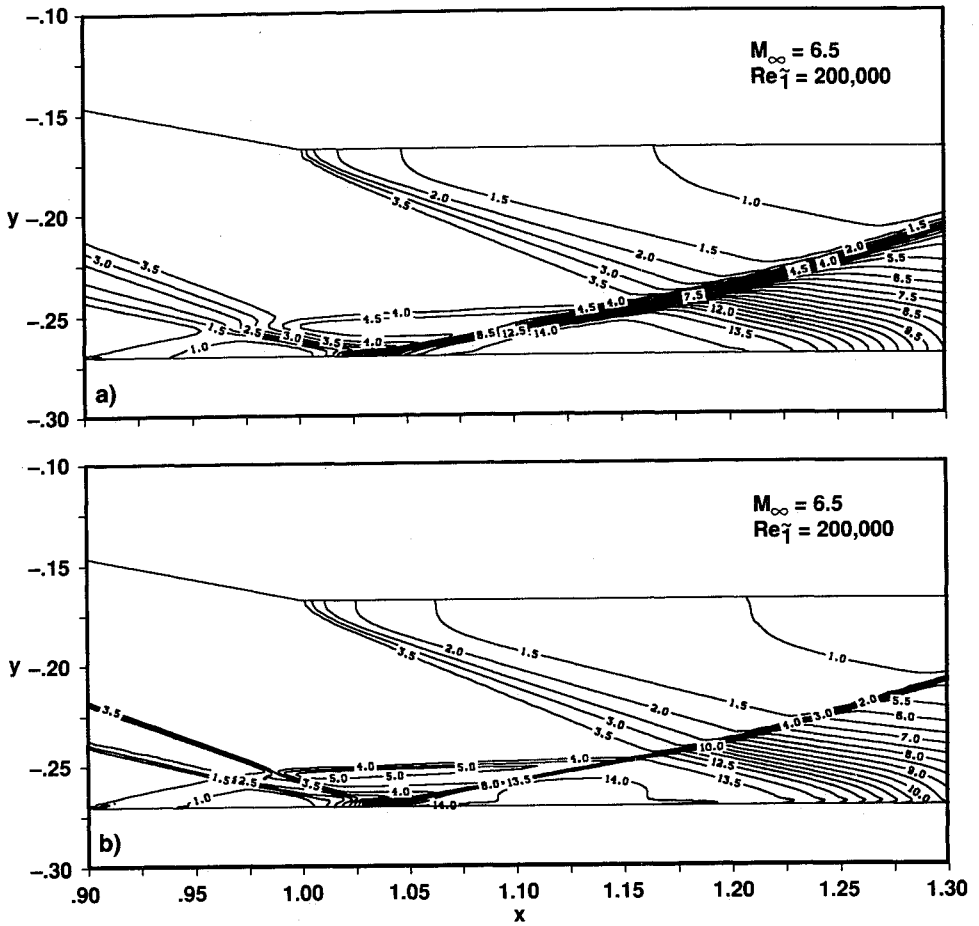

Fig. 10 Comparison of computed pressure contours for planar inlet: a) not adapted; b) adapted (dimensions are in meters).

cation, only 60 grid points are shown on the forebody in Figs. 9.

Solution-adaptive gridding was initiated at $x = 0.04$ m. The maximum and minimum grid spacings were set to 4.5 and 0.2 times the average grid spacings, respectively. The torsion parameter λ was set to 2×10^{-4} and α and β of Eq. (7) were given values of unity and zero, respectively. On the forebody, the adaptation domain consisted of the 60 points nearest the upper wall. At the inlet face location ($x = 0.9$ m), no-slip boundary conditions were introduced at the cowl lip and the adaptation domain was extended to include the entire calculation domain. This procedure maintained a sufficient grid clustering at the lower boundary to resolve the viscous effects present at the cowl lip. The enlarged region of Fig. 9 shows the grid surrounding the shock wave interaction at $x \approx 1.0$ m.

The pressure field in the region directly downstream of the inlet plane is shown in Fig. 10 for both unadapted and adapted cases. The two shocks can be clearly seen impinging on the boundary layer that is developing on the cowl surface. These two shock waves penetrate the lip shock, deflecting it downward slightly, and then meet at the wall where they reflect as a single, stronger shock. The resulting reflected shock travels downstream, interacting with the lip shock again at $x = 1.15$ m, and then is deflected by the expansion fan formed by the top of the compression surface. These characteristics are much more clearly defined in the adapted case (Fig. 10b).

Static pressure profiles at $x = 0.9$ m are shown in Fig. 11. Both adapted and fixed-grid solutions are compared with inviscid, oblique-shock theory. The two shock waves that cross this x location are difficult to distinguish from one another for the fixed-grid case. The shape of the pressure profile for the adapted case agrees well with the shape of the inviscid-theory curve in that both shock waves are clearly defined. To verify that the upward shift in profiles for the computational

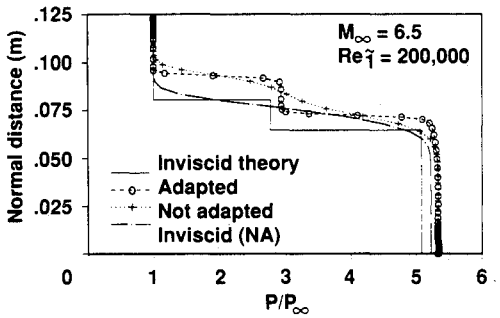

Fig. 11 Comparison of pressure profiles ($x = 0.9$ m).

results is attributed to viscous effects, an inviscid fixed-grid solution was computed. A comparison of these results clearly shows that the discrepancy between the viscous numerical solution and that of inviscid theory is due primarily to viscous effects and is not the result of numerical error.

Cone Cylinder

The final test case involves supersonic flow over a cone-cylinder arrangement. This test case is related to the study of sonic boom predictions. In sonic boom predictions, it is desirable to predict the far-field pressure impulse caused by a moving shock wave. It is often difficult to obtain a satisfactory grid for this type of problem because the primary region of interest is a great distance from the body, making the solution domain very large. Conventional grid point clustering in the far field is often not possible without overpopulating other less important regions of the flowfield, resulting in wasted time and effort. Thus, the use of an adaptive grid scheme for

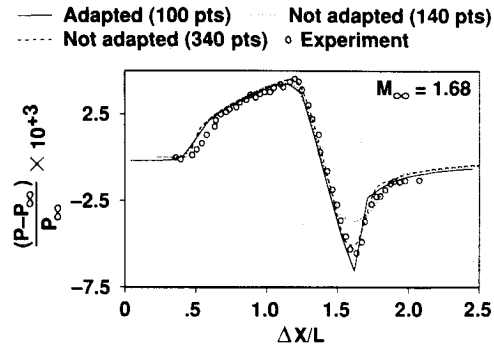

Fig. 12 Comparison of pressure impulses.

Table 1 Comparison of computational effort

Case	Grid	Number of grid points	CPU, s
Compression corner	Fixed	90	35
	Fixed	45	17
	Adapted	45	36
Inlet	Fixed	80	170
	Adapted	80	345
Cone cylinder	Fixed	340	1320
	Adapted	100	880

this case is virtually imperative for obtaining an accurate solution. For the present study, the geometry consists of a 3.24-deg half-angle cone joined by a slender cylinder. The length of the cone is 4.3 in. The flow is axisymmetric and calculations were performed using the PNS solver, neglecting viscosity, at a freestream Mach number of 1.68. For the solution-adaptive case, 100 points were used normal to the body. A conical step-back procedure was started from freestream conditions at $x = 0.01$ in. until a satisfactory conical starting solution was obtained. The maximum and minimum allowable grid spacings Δs_{max} and Δs_{min} were initially set to 3.0 and 0.2, respectively. The torsional parameter λ was set to 1×10^{-3} and α and β of Eq. (7) were given values of unity and zero, respectively. The solution proceeded from the starting plane to a distance of 130 in. downstream. This x distance corresponds to the distance where the bow shock is at an altitude of 10 cone lengths above the body axis.

In addition to the bow shock, the flowfield features include a strong overexpansion at the base of the cone followed by a recompression wave. Figure 12 compares experimental pressure measurements at an altitude of 10 cone lengths with computed results using both adapted and fixed grids. The adapted case using 100 points is seen to be superior to even the finest unadapted case. This clearly illustrates the benefit of an adapted grid scheme for use with sonic boom calculations where the far-field flow structure is of major importance.

The fixed-grid case using 340 grid points required 22.5 min of CPU, whereas the adapted case employing 100 grid points required 15 min. A summary of computational effort for each test case is provided in Table 1. All computations presented here where performed on a Cray Y-MP/832 and CPU times are measured in seconds.

Concluding Remarks

A solution-adaptive grid procedure has been developed and used with an upwind space-marching scheme. This line-by-line method of redistributing grid points is found to be compatible with the plane-by-plane solution process of the marching scheme. The adaptive grid algorithm provides improved shock resolving characteristics over the conventional flow algorithm. In all test cases presented in this paper, the adaptive grid scheme was able to accurately align the computational grid with existing shock waves. This increased grid alignment was observed to enable the PNS solver to remain stable for a larger marching step size than that used with fixed grids. The increased computational effort required for the solution-adaptive process lies principally in the need to recompute each ξ-plane on the refined grid. The effort required for actual grid point redistribution is negligible.

Acknowledgments

This work was supported by the Applied Computational Fluids Branch at NASA Ames Research Center under Joint Interchange NCA2-326 and under Branch Chief Terry L. Holst.

References

[1]Thompson, J. F., "A Survey of Dynamically-Adaptive Grids in the Numerical Solution of Partial Differential Equations," *Applied Numerical Mathematics*, Vol. 1, 1985, pp. 3–27.

[2]Hawken, D. F., "Review of Adaptive-Grid Techniques for Solution of Partial Differential Equations," University of Toronto, Toronto, Ontario, Canada, IAS Review No. 46, Dec. 1985.

[3]Dwyer, H. A., Kee, R. J., and Sanders, B. R., "Adaptive Grid Method for Problems in Fluid Mechanics and Heat Transfer," *AIAA Journal*, Vol. 18, No. 10, 1980, pp. 1205–1212.

[4]Rai, M. M., and Anderson, D. A., "Application of Adaptive Grids to Fluid-Flow Problems with Asymptotic Solutions," *AIAA Journal*, Vol. 20, No. 4, 1982, pp. 496–502.

[5]Greenberg, J. B., "A New Self-Adaptive Grid Method," AIAA Paper 83-1934, Jan. 1983.

[6]Gnoffo, P. A., "A Vectorized, Finite-Volume, Adaptive Grid Algorithm Applied to Planetary Entry Problems," AIAA Paper 82-1018, June 1982.

[7]Nakahashi, K., and Deiwert, G. S., "A Practical Adaptive-Grid Method for Complex Fluid-Flow Problems," NASA TM-85989, June 1984.

[8]Nakahashi, K., and Deiwert, G. S., "A Self-Adaptive Grid Method with Application to Airfoil Flow," AIAA Paper 85-1525, July 1985.

[9]Nakahashi, K., and Deiwert, G. S., "Three Dimensional Adaptive Grid Method," *AIAA Journal*, Vol. 24, No. 6, 1986, pp. 948–954.

[10]Djomehri, M. J., and Deiwert, G. S., "Three-Dimensional Self-Adaptive Grid Method for Complex Flows," NASA TM 101027, Nov. 1988.

[11]Davies, C. B., and Venkatapathy, E. W., "A Simplified Self-Adaptive Grid Method, SAGE," NASA TM-102198, Oct. 1989.

[12]Thompson, J. F., Warsi, Z. U. A., and Mastin, C. W., *Numerical Grid Generation, Foundations and Applications*, North-Holland, Amsterdam, The Netherlands, 1985.

[13]Lawrence, S. L., "Application of an Upwind Algorithm to the Parabolized Navier-Stokes Equations," Ph.D. Dissertation, Iowa State Univ., Ames, IA, May 1987.

[14]Lawrence, S. L., Tannehill, J. C., and Chaussee, D. S., "An Upwind Algorithm for the Parabolized Navier-Stokes Equations," *AIAA Journal*, Vol. 22, No. 12, 1984, pp. 1755–1763.

[15]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, Oct. 1983, pp. 357–372.

[16]Vigneron, Y. C., Rakich, J. C., and Tannehill, J. C., "Calculation of Supersonic Viscous Flow over Delta Wings with Sharp Subsonic Leading Edges," AIAA Paper 78-1137, July 1978.

[17]Holden, M. S., and Moselle, J. R., "Theoretical and Experimental Studies of the Shock Wave-Boundary Layer Interaction on Compression Surfaces in Hypersonic Flow," CALSPAN, Buffalo, NY, Rept. AF-2410-A-1, Oct. 1969.