AIAA JOURNAL
Vol. 33, No. 1, January 1995

Approach for Dynamic Grids

J. W. Slater* and M. S. Liou†
NASA Lewis Research Center, Cleveland, Ohio 44135
and
R. G. Hindman‡
Iowa State University, Ames, Iowa 50011

An approach is presented for the generation of two-dimensional, structured, dynamic grids utilizing grid speeds computed from the time differentiation of a set of grid equations. In application for the computation of unsteady, inviscid flows, it is shown that the approach is more efficient and accurate than an approach in which the grid speeds are computed using a finite difference of the grid with respect to time.

Introduction

As computational methods become more efficient and computational resources become more powerful, the methods of computational fluid dynamics are being applied to more complex problems. Among these are problems involving the motion of the boundaries of the computational domain and problems involving transient flow conditions with solution adaptive grids. These generally require a dynamic grid. Much work has been done on methods for the generation of static grids with proper qualities. A dynamic grid should maintain these same qualities; however, this topic has not received much attention.

We present an approach for the generation of dynamic grids for which the qualities of grid smoothness, orthogonality, and cell volume adaptation are maintained. The dynamic grids are generated through the time integration of the grid speeds computed from a system of grid speed equations. The grid speed equations are derived from the time differentiation of the grid equations. This approach was first presented by Hindman et al.[1] for an application involving the motion of a shock-fitted boundary. Holcomb and Hindman[2] extended the approach for the adaptation of the grid to the physical solution. The previous cited work involved grid equations based on the Poisson equations and did not account for grid orthogonality. The present approach uses a system of grid equations based on the variational approach of reference.[3]

An alternative approach for computing the grid speeds is to use a time difference of the grid. This approach is also examined.

The motivation for investigating the approach using the grid speed equations has been the desire for greater efficiency and accuracy than the approach using grid speeds obtained from a time difference of the grids.

The objectives of this paper are to present the system of grid and grid speed equations for two-dimensional space, present an explicit dynamically adaptive grid method, and demonstrate the behavior of the method through applications. These applications include a simple problem involving a moving boundary and an unsteady, inviscid flow through a converging-diverging nozzle during startup.

Grid Equations

The grid equations follow the variational approach of Brackbill and Saltzman[3] which derive the grid equations from the Euler-Lagrange equations,

$$\left(\frac{\partial}{\partial\xi}\frac{\partial}{\partial x_\xi}+\frac{\partial}{\partial\eta}\frac{\partial}{\partial x_\eta}-\frac{\partial}{\partial x}\right)L=0 \tag{1}$$

and

$$\left(\frac{\partial}{\partial\xi}\frac{\partial}{\partial y_\xi}+\frac{\partial}{\partial\eta}\frac{\partial}{\partial y_\eta}-\frac{\partial}{\partial y}\right)L=0 \tag{2}$$

The grid equations relate the physical space (x, y) to the parametric, computational space (ξ, η). The subscripts (ξ, η) denote differentiation. The Lagrangian L is defined so as to include measures of smoothness S, orthogonality O, and cell volume adaptation A, and is of the form

$$L = L_S + L_O + L_A \tag{3}$$

The expressions for L_O and L_A presented here are a modification of those presented in Ref. 3 to nondimensionalize the Lagrangians. The Lagrangians are of the form

$$L_S = \lambda_S[(\boldsymbol{r}_\xi \cdot \boldsymbol{r}_\xi + \boldsymbol{r}_\eta \cdot \boldsymbol{r}_\eta)/J] \tag{4}$$

$$L_O = \lambda_O(\boldsymbol{r}_\xi \cdot \boldsymbol{r}_\eta/J)^2 \tag{5}$$

$$L_A = \lambda_A(WJ)^2/K^2 \tag{6}$$

The λ_S, λ_O, and λ_A are specified constants which weigh the importance of each Lagrangian. The cell volume adaptation Lagrangian L_A is essentially a statement of the equidistribution of the product WJ over the grid. The W is a weighting function usually directly related to the gradients in the flow solution. The J is the Jacobian and is a measure of the cell volume. The K is the average value of WJ over the grid.

The system of grid equations is derived by inserting the expressions for the Lagrangians into the Euler-Lagrange equations and is expressed in the form

$$G(\boldsymbol{r}) = A\boldsymbol{r}_{\xi\xi} + B\boldsymbol{r}_{\xi\eta} + C\boldsymbol{r}_{\eta\eta} + D\boldsymbol{r}_\xi + E\boldsymbol{r}_\eta = 0 \tag{7}$$

where $\boldsymbol{r}^T = [x, y]$. For a structured grid, the solution of the grid equations is of the form $\boldsymbol{r}_{i,j}$ where (i, j) denote the indices of a rectangular matrix. The coefficient matrices (A, B, C, D, E) are sums of the form

$$A = A_S + A_O + A_A \tag{8}$$

The B, C, D, and E matrices are of the same form.

Received March 10, 1994; revision received Aug. 2, 1994; accepted for publication Sept. 6, 1994.

*National Research Council Research Associate, Computational Fluid Dynamics Branch, Internal Fluid Mechanics Division, MS 5-11, 21000 Brookpark Road. Member AIAA.

†Senior Scientist, Internal Fluid Mechanics Division, MS 5-11, 21000 Brookpark Road. Member AIAA.

‡Associate Professor, Department of Aerospace Engineering and Engineering Mechanics, 2102 Black Engineering Building. Member AIAA.

The expressions for the smoothness matrices are

$$A_S = \lambda_S \alpha / J^3 [S] \tag{9}$$

$$B_S = -2\lambda_S \beta / J^3 [S] \tag{10}$$

$$C_S = \lambda_S \gamma / J^3 [S] \tag{11}$$

and D_S and E_S are the null matrix [0]. Also,

$$S = \begin{bmatrix} \delta & -\epsilon \\ -\epsilon & \rho \end{bmatrix} \tag{12}$$

The remaining variables are defined as

$$J = x_\xi y_\eta - x_\eta y_\xi \tag{13}$$

$$\alpha = x_\eta^2 + y_\eta^2 \tag{14}$$

$$\beta = x_\xi x_\eta + y_\xi y_\eta \tag{15}$$

$$\gamma = x_\xi^2 + y_\xi^2 \tag{16}$$

$$\delta = y_\xi^2 + y_\eta^2 \tag{17}$$

$$\epsilon = x_\xi y_\xi + x_\eta y_\eta \tag{18}$$

$$\rho = x_\xi^2 + x_\eta^2 \tag{19}$$

The expressions for the orthogonality matrices are

$$A_O = \frac{\lambda_O \alpha}{J^4} \begin{bmatrix} 2\beta y_\eta y_\xi + \alpha y_\xi^2 & \psi \\ \psi & 2\beta x_\eta x_\xi + \alpha x_\xi^2 \end{bmatrix} \tag{20}$$

$$B_O = -\frac{\lambda_O (2\beta^2 + \alpha\gamma)}{J^4} \begin{bmatrix} 2 y_\xi y_\eta & -\phi \\ -\phi & 2 x_\xi x_\eta \end{bmatrix} \tag{21}$$

$$C_O = \frac{\lambda_O \gamma}{J^4} \begin{bmatrix} \gamma y_\eta^2 + 2\beta y_\xi y_\eta & \kappa \\ \kappa & \gamma x_\eta^2 + 2\beta x_\xi x_\eta \end{bmatrix} \tag{22}$$

and D_O and E_O are the null matrix [0]. The ϕ, ψ, and κ are defined as

$$\phi = x_\xi y_\eta + x_\eta y_\xi \tag{23}$$

$$\psi = -(\gamma x_\eta y_\eta + 2\alpha x_\xi y_\xi) \tag{24}$$

and

$$\kappa = -(\alpha x_\xi y_\xi + 2\gamma x_\eta y_\eta) \tag{25}$$

The expressions for the cell volume adaptation matrices are

$$A_A = \frac{\lambda_A W^2}{K^2} \begin{bmatrix} y_\eta^2 & -x_\eta y_\eta \\ -x_\eta y_\eta & x_\eta^2 \end{bmatrix} \tag{26}$$

$$B_A = \frac{\lambda_A W^2}{K^2} \begin{bmatrix} -2 y_\xi y_\eta & \phi \\ \phi & -2 x_\xi x_\eta \end{bmatrix} \tag{27}$$

$$C_A = \frac{\lambda_A W^2}{K^2} \begin{bmatrix} y_\xi^2 & -x_\xi y_\xi \\ -x_\xi y_\xi & x_\xi^2 \end{bmatrix} \tag{28}$$

$$D_A = \frac{\lambda_A J W W_\eta}{K^2} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{29}$$

$$E_A = \frac{\lambda_A J W W_\xi}{K^2} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{30}$$

Grid Speed Equations

The grid speed equations are obtained from the time differentiation of the grid equations. Applying the chain rule of differentiation and collecting terms results in the grid speed equations of the form

$$G_\tau(x, y) = A^* z_{\xi\xi} + B^* z_{\xi\eta} + C^* z_{\eta\eta} + D^* z_\xi + E^* z_\eta + T^* = 0 \tag{31}$$

where the grid speed vector is defined as

$$z^T = [x_\tau, y_\tau] \tag{32}$$

The subscript τ denotes differentiation with respect to time. This involves performing the time differentiations A_τ, B_τ, C_τ, D_τ, and E_τ and factoring out z. The procedure is straightforward; however, the algebra is quite involved. The details are presented in Ref. 4. One observation is that the time differentiation of the matrices only result in first-order time derivatives of the grid and so

$$[A^*] = [A], \qquad [B^*] = [B], \qquad \text{and} \qquad [C^*] = [C] \tag{33}$$

Weighting Function

The weighting function W is defined in the form

$$W = \lambda_0 + \lambda_1 |\nabla f(U)| + \lambda_2 \exp(\lambda_3 |\nabla f(U)|) \tag{34}$$

The $\nabla f(U)$ is the gradient with respect to the computational coordinates (ξ, η) of a function f, such as density, of the physical solution U. The function f may be discontinuous; thus, an elliptic smoothing is used to make f a continuous function. The λ_0, λ_1, λ_2, and λ_3 are specified constants which influence the spatial character of W. A value of $\lambda_0 = 1.0$ assures that W is unity when there are no gradients in the physical solution. The λ_1 emphasizes a linear relationship between the gradient and W. The λ_2 and λ_3 put greater emphasis on higher gradient regions, which is useful for adaptation to viscous boundary layers. The values for λ_1, λ_2, and λ_3 are problem dependent, and the selection of the values for individual problems are discussed in the results section.

The elliptic smoothing tends to reduce peaks in f, and this reduces the magnitude of the gradient and results in a smoother distribution of W. As the number of smoothing iterations is increased, a higher value of λ_1 is needed to maintain the same level of grid clustering (defined by the minimum grid spacing), and the grid clustering is spread over a larger region.

The grid equations require the derivatives W_ξ and W_η. The grid speed equations also require the derivatives W_τ, $W_{\xi\tau}$, and $W_{\eta\tau}$. These are computed by applying the chain rule and using second-order spatial and first-order temporal finite differences.

Solving the Grid and Grid Speed Equations

The grid equations are nonlinear partial differential equations whereas the grid speed equations are linear partial differential equations. This is attractive since linear equations are easier to solve than nonlinear equations.

The smoothness terms in the grid equations form a system of elliptic equations. The orthogonality and cell volume adaptation terms form a system of equations which is not elliptic and can produce nonunique solutions. However, the combined system of equations remains fairly elliptic and iterative methods for solving elliptic equations can be applied with success.

The grid and grid speed equations are discretized using second-order, central differences and solved using the Gauss–Seidel point relaxation method.

Neumann boundary conditions are applied to impose orthogonality of the grid lines at the boundaries. The orthogonality condition is

$$\boldsymbol{r}_\xi \cdot \boldsymbol{r}_\eta = 0 \tag{35}$$

The boundary is parameterized according to the curve length along the boundary. A local linearization is applied, and an expression can be developed for the change in the curve-length coordinate s

needed for orthogonality for the iteration. For the $j = 1$ boundary, this expression is

$$(\Delta s)^m_{i,1} = \left[\frac{(\boldsymbol{r}_s)_{i,1} \cdot [4\boldsymbol{r}_{i,2} - 3\boldsymbol{r}_{i,1} - \boldsymbol{r}_{i,3}]}{3(\boldsymbol{r}_s \cdot \boldsymbol{r}_s)_{i,1}}\right]^m \tag{36}$$

where m is the iteration index. Thus,

$$s^{m+1} = s^m + (\Delta s)^m \tag{37}$$

and the new boundary point grid point is

$$\boldsymbol{r}_B^{m+1} = \boldsymbol{r}_{\text{fit}}(s^{m+1}) \tag{38}$$

where the subscript B denotes a boundary point, and $\boldsymbol{r}_{\text{fit}}$ is the curve fit of the boundary.

The grid speeds at the boundaries are of the mixed type to impose a prescribed boundary motion and to impose the orthogonality condition at the boundary. The boundary condition for imposing orthogonality is derived from the time differentiation of the grid orthogonality condition

$$\frac{\partial}{\partial \tau}(\boldsymbol{r}_\xi \cdot \boldsymbol{r}_\eta) = 0 \tag{39}$$

Using the parameterization of the boundary and a local linearization, an expression for the the time derivative of the curve-length coordinate can be obtained. For the $j = 1$ boundary, the expression is

$$s_\tau = \frac{\boldsymbol{z}_s \cdot (4\boldsymbol{r}_{i,2} - 3\boldsymbol{r}_{i,1} - \boldsymbol{r}_{i,3}) + \boldsymbol{r}_s \cdot (4\boldsymbol{z}_{i,2} - \boldsymbol{z}_{i,3})}{3(\boldsymbol{r}_s \cdot \boldsymbol{r}_s)} \tag{40}$$

Thus, for each iteration, a new s_τ is computed, and

$$s^{m+1} = s^m + \Delta\tau s_\tau \tag{41}$$

and from the curve fit of the boundary,

$$\boldsymbol{r}_B^{m+1} = \boldsymbol{r}_{\text{fit}}(s^{m+1}) \tag{42}$$

The mixed boundary condition is imposed as

$$\boldsymbol{z}_B(t) = \boldsymbol{z}_{B(\text{motion})}(t) + \left(\boldsymbol{r}_B^{m+1} - \boldsymbol{r}_B^m\right)/\Delta\tau \tag{43}$$

Solving the grid speed equations for the grid speeds and then integrating to obtain the grid is expected to generate a grid which satisfies the grid equations. However, small deviations may develop. One way to ensure that the grid continues to satisfy the grid equations is to rewrite the grid speed equations in the form of a first-order homogeneous dynamic system[2] as

$$G_\tau + \lambda_C G = 0 \tag{44}$$

where λ_C is a prescribed damping constant which damps deviations from the grid equations.

Coupling of the Flow and Grid Equations

The flow equations and the grid speeds are integrated in time using an explicit, two-stage Lax-Wendroff method[5] of the form

$$\Phi^*_{i,j} = \Phi^n_{i,j} + \Delta\tau\Psi^n_{i,j} \tag{45}$$

$$\Phi^{**}_{i,j} = \Phi^*_{i,j} + \Delta\tau\Psi^*_{i,j} \tag{46}$$

$$\Phi^{n+1}_{i,j} = \tfrac{1}{2}\left\{\Phi^n_{i,j} + \Phi^{**}_{i,j}\right\} \tag{47}$$

For the grid, $\Phi = \boldsymbol{r}$ and $\Psi = \boldsymbol{z}$. For the flow equations, $\Phi = \hat{\boldsymbol{U}}$ and $\Psi = \hat{\boldsymbol{R}}$. The $\hat{\boldsymbol{U}}$ and $\hat{\boldsymbol{R}}$ denote the algebraic vectors of the generalized conservative variables and fluxes defined from a cell-vertex, finite-volume approximation of the unsteady Navier–Stokes equations. Thus,

$$\hat{\boldsymbol{U}} = \boldsymbol{U}V \tag{48}$$

where V is the area of the planar finite-volume cell and $\boldsymbol{U}$ is the algebraic vector of conservative variables. The $\hat{\boldsymbol{R}}$ is the flux residual defined as

$$\hat{\boldsymbol{R}}_{i,j} = \hat{\boldsymbol{F}}_{i+\frac{1}{2},j} + \hat{\boldsymbol{F}}_{i-\frac{1}{2},j} + \hat{\boldsymbol{F}}_{i,j+\frac{1}{2}} - \hat{\boldsymbol{F}}_{i,j-\frac{1}{2}} \tag{49}$$

where the flux vector for a cell face is defined as

$$\hat{\boldsymbol{F}} = \boldsymbol{H} \cdot \hat{n}\, \mathrm{d}S \tag{50}$$

The $\boldsymbol{H}$ is the flux dyadic, which for a time-varying control volume is

$$\boldsymbol{H} = \boldsymbol{F} - \boldsymbol{z}\boldsymbol{U} \tag{51}$$

The $\boldsymbol{F}$ is the Cartesian flux dyadic for the two-dimensional, unsteady Navier–Stokes equations. The flow equations are complete with Sutherland's formula, the definition of the Prandtl number, and the assumptions of a perfect gas (air) and laminar viscous flow.

The time step is computed using the Courant-Fredrichs-Lewy (CFL) condition. The inviscid fluxes are computed using the Roe flux-difference splitting with a total variation diminishing (TVD) limiter as presented in Ref. 5. The viscous fluxes are computed using differences and averages computed at the cell faces. Characteristic boundary conditions are used to compute the solution points on the boundaries.

The V needed to decode $\hat{\boldsymbol{U}}$ is computed from the grid conservation law as discussed in Refs. 6 and 7. The grid conservation law relates the change in volume of the cell to the motion of the cell faces and is derived from the flow integration equations with the assumption of a uniform solution for the conservative variables. The grid conservation law follows the form of Eqs. (45–47) with $\Phi = V$ and $\Psi = \hat{\boldsymbol{Z}}$ where $\hat{\boldsymbol{Z}}$ is the vector sum of speeds of the cell faces.

The flow and grid equations are coupled through the grid speed terms in the flow equations and the W function in the grid speed equations. The two-stage approach advances the solution and grid in time and space in ideally second-order accuracy. The steps in the explicit, dynamically adaptive grid method using the grid speed equations include the following.

1) Establish an initial grid $\boldsymbol{r}^n$ satisfying the grid equations and a consistent initial solution $\boldsymbol{U}^n$.

2) Solve the grid speed equations for $\boldsymbol{z}^n$.

3) Compute or specify the time step $\Delta\tau$.

4) Compute $\hat{\boldsymbol{U}}^*$ for the first stage,

$$\hat{\boldsymbol{U}}^*_{i,j} = \hat{\boldsymbol{U}}^n_{i,j} - \Delta\tau\hat{\boldsymbol{R}}^n_{i,j} \tag{52}$$

5) Solve the grid conservation law,

$$V^*_{i,j} = V^n_{i,j} - \Delta\tau\hat{\boldsymbol{Z}}^n_{i,j} \tag{53}$$

6) Compute the first-stage flow solution,

$$\boldsymbol{U}^*_{i,j} = \hat{\boldsymbol{U}}^*_{i,j}/V^*_{i,j} \tag{54}$$

7) Compute the flow boundary conditions.

8) Integrate the grid speeds to obtain the grid,

$$\boldsymbol{r}^*_{i,j} = \boldsymbol{r}^n_{i,j} + \Delta\tau\boldsymbol{z}^n_{i,j} \tag{55}$$

9) Recompute the cell volumes and cell-face areas.

10) Solve the grid speed equations for $\boldsymbol{z}^*$.

11) Compute $\hat{\boldsymbol{U}}^{**}$ for the second stage,

$$\hat{\boldsymbol{U}}^{**}_{i,j} = \hat{\boldsymbol{U}}^*_{i,j} - \Delta\tau\hat{\boldsymbol{R}}^*_{i,j} \tag{56}$$

12) Compute $\hat{\boldsymbol{U}}^{n+1}_{i,j}$,

$$\hat{\boldsymbol{U}}^{n+1}_{i,j} = \tfrac{1}{2}\left[\hat{\boldsymbol{U}}^n_{i,j} + \hat{\boldsymbol{U}}^{**}_{i,j}\right] \tag{57}$$

13) Solve the grid conservation law,

$$V^{**}_{i,j} = V^*_{i,j} - \Delta\tau\hat{\boldsymbol{Z}}^*_{i,j} \tag{58}$$

14) Compute the volume at the $n+1$ time level,

$$V_{i,j}^{n+1} = \tfrac{1}{2}\left(V_{i,j}^{n} + V_{i,j}^{**}\right) \tag{59}$$

15) Compute the flow solution,

$$U_{i,j}^{n+1} = \hat{U}_{i,j}^{n+1} / V_{i,j}^{n+1} \tag{60}$$

16) Compute the flow boundary conditions.

17) Integrate the grid speeds to obtain the grid,

$$r_{i,j}^{n+1} = r_{i,j}^{n} + \tfrac{1}{2}\Delta\tau\left(z_{i,j}^{n} + z_{i,j}^{*}\right) \tag{61}$$

18) Recompute the cell volumes and cell-face areas.

19) Solve the grid speed equations for z^{n+1}.

20) Go to step 3 and repeat until the final time.

Computing the grid speeds using a time difference of the grid requires solving the grid equations at each stage prior to the differencing.

Results

Simple Dynamic Boundary

The dynamically adaptive grid method was applied to a uniform grid on a unit square domain in which the lower boundary was prescribed to rotate at a rate of 20 deg/s about the lower left corner. This problem tested the ability of the method to compute the grid speeds and integrate them in time to create a grid which continuously satisfied the grid equations and, accordingly, maintain the desired grid qualities.

The time integrations were performed for 50 time steps with a time step of $\Delta\tau = 0.02$ s to a final time of 1 s. The bottom boundary rotated 20 deg about the lower left corner. The grid parameters were $(\lambda_S,\ \lambda_O,\ \lambda_A) = (1.0,\ 1.0,\ 1.0)$ and $\lambda_C = 30.0$.

For an iteration tolerance of 1.0×10^{-8}, the computation required 17,308 iterations of the grid speed equations. The average error was 0.0396%, and the maximum error was 0.8176%. The errors are the differences in grid point locations between the final grid obtained from the time integration and the grid which satisfied the grid equations at the final time. The errors are expressed as percentages of the length of the uniform grid spacing. Figure 1 shows the grid and grid speed vectors at the final time. The final grid shows that the grid maintained the desired grid qualities, specifically, the orthogonality of the grid lines to the lower boundary.

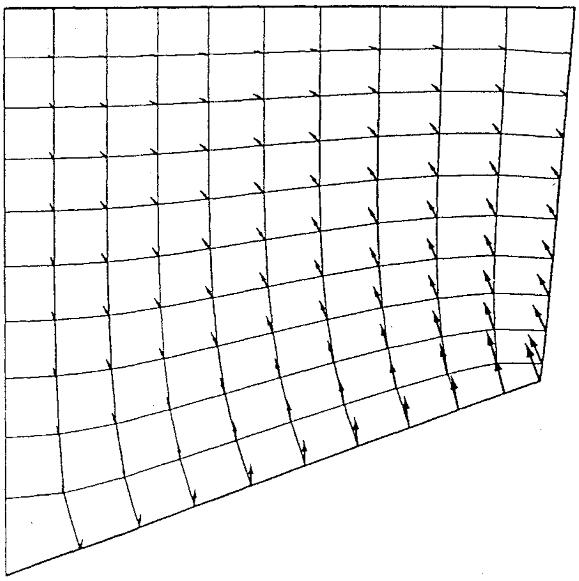

Fig. 1 Grid and grid speed vectors at t = 1.0 s for the rotation of the lower boundary of a unit square.

The effectiveness of the grid control law damping factor λ_C is shown in Fig. 2. The average error was reduced for increased values of λ_C. The fact that the curve bottomed out indicates that one should be able to pick a value of λ_C and obtain a good reduction in error without having to search for an optimum value. However, although the errors decreased, the equations became stiffer, and more iterations of the grid speed equations were required. The effectiveness of λ_C remained consistent with other test problems and, accordingly, these results guided the selection of λ_C for the flow problems.

The pointwise iteration of the grid speed equations demonstrated excellent convergence characteristics. When the iteration tolerance was increased to 1.0×10^{-4}, only 304 iterations were required, and the average error and maximum error only increased to 0.1065% and 2.8635%, respectively. When the number of iterations per call was limited to 5 iterations, the total number of iterations was 332, and the average and maximum errors were only 0.1023% and 2.0723%, respectively. This provided some guidance for setting the number of iterations and tolerances appropriate for the efficient solution of the grid speed equations.

Inviscid Startup Flow in a Nozzle

The dynamically adaptive grid method was applied to the computation of the inviscid startup flow in a converging-diverging (CD) nozzle. Figures 3 and 4 show the shape of the nozzle. The flow was initially stationary at a total pressure of $p_t = 104074.6$ N/m^2 and a total temperature of $T_t = 252.9$ K. The exit pressure was then decreased from the total pressure value to a value of $p_{\text{exit}} = 84{,}000$ N/m^2 over a time interval of 10% of a reference time. The reference time was defined by the ratio L/V_{inflow} where L is the length of the nozzle ($L = 10$ m) and V_{inflow} is the inflow velocity ($V_{\text{inflow}} = 75.92$ m/s). A two-point, cubic spline variation with zero end-slope conditions was used for the time variation of the exit pressure. After the transition time interval, the exit pressure was held fixed. The flow was computed until a final time of 0.3 s. All computations were performed on the NASA Lewis Cray YMP.

The grids were generated with 95 grid points in the streamwise direction and 15 grid points in the transverse direction with grid parameters of $(\lambda_S,\ \lambda_O,\ \lambda_A) = (1.0,\ 1.0,\ 0.5)$. The initial grid, which is the grid at $t = 0.0$ seconds in Fig. 4, was used for the static grid computation, which involved no grid adaptation.

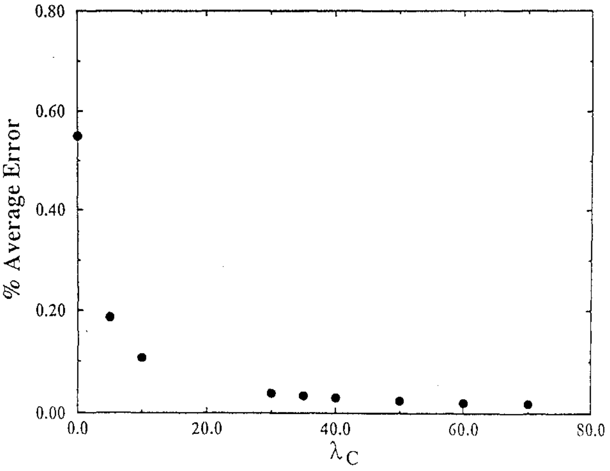

Fig. 2 Effectiveness of the λ_C damping factor.

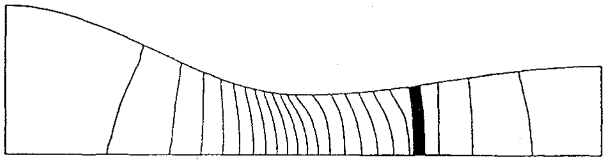

Fig. 3 Density contours of the CD nozzle flow at t = 0.3 s for the static grid computation.

The function f used in defining the weighting function W for the grid adaptation was the flow density. This choice was based on the desire to cluster grid points near the shock. The grid adaptation parameters were $\lambda_1 = 500$, $\lambda_2 = 0$, and $\lambda_3 = 0$ to emphasize a linear relationship between density gradients and W. To smooth the density solution prior to computing the weighting function W and its derivatives 30 smoothing passes were used. These values were determined through a series of computations and consideration of the resulting quality of the adapted grid. Selection of the values of these inputs is problem dependent and rules for selecting the values for general problems can not be stated at this time.

The unsteady flow in the nozzle was computed using a static grid approach (no grid adaptation), a dynamic grid approach in which the grid speeds were computed from a first-order, backwards time difference, and a dynamic grid approach in which the grid speeds were computed from the grid speed equations. The flow computations were performed using a CFL number of $\nu = 0.6$ with a maximum time step of 1.05×10^{-4} s. These limits were imposed due to stability limits determined from numerical experimentation with the dynamic grid computations. They were used for all of the computations so as to evaluate performance in a consistent manner. The grid and grid speed equations were iterated with a tolerance of 1.0×10^{-4} with a maximum of 3 iterations allowed for each stage of the Lax-Wendroff method. At the start of the dynamic grid computations, the grid and grid speed equations required all 3 iteration. As the steady-state flow solution was reached, the grid residual fell below the tolerance, and less iterations were required until only 1 iteration was required per stage. The grid control law damping factor in the solution of the grid speed equations was set to a value of $\lambda_C = 50$. The performance of the computations is summarized in Table 1. The ratio in Table 1 is the CPU/step values normalized to the value from the static grid approach. The dynamic grid approaches required more time steps because the minimum grid spacing was less than for the static grid, and this reduced the maximum allowable time step. The grid spacing at the shock at the final time for the adapted grids was about 47% of the spacing for the static grid.

Figure 3 shows the density contours for the static grid approach at the final time, $t = 0.3$ s. Figure 4 shows the grids and density contours at several times during the dynamic grid computations. Note that the grid adapts to the shock as it forms and moves down the nozzle to its steady-state position. The density contours indicate that the shock is captured in a smaller region.

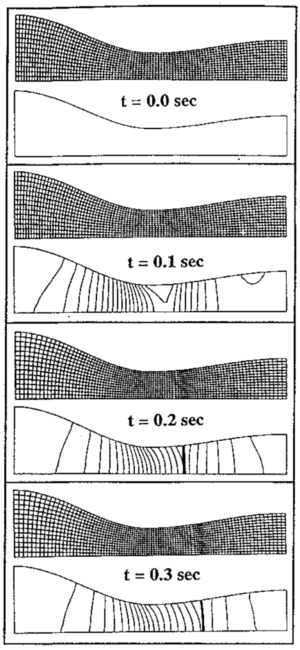

Fig. 4 Grids and density contours of the CD nozzle flow at various times for the dynamically adaptive grid computation.

Table 1 Summary of computational effort for unsteady CD nozzle problem

Grid approach	Time steps	CPU, s	CPU/step	Ratio
Static	2983	115.3	0.0387	1.00
Time differenced	4440	450.5	0.1015	2.62
Grid speeds	4472	351.3	0.0786	2.03

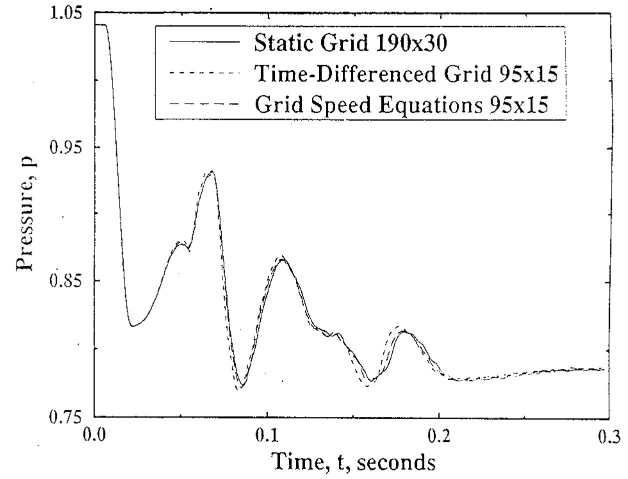

Fig. 5 Time histories of the pressure at the location x = 8 m in the CD nozzle.

In using the grid speeds computed from a time difference, one would expect that integrating the grid speeds prior to solving the grid equations at the next stage would provide a good first guess of the new grid. Computations have shown that for this problem, performing the integrations caused the boundary of the grid to deviate significantly from its true geometry and introduced perturbations into the flowfield. Thus, the grid speeds were not integrated for this approach. This problem is under further study.

The variation of the static pressure in the streamwise direction for all of the computed solutions compared very well with quasi-one-dimensional, inviscid theory. The dynamically adaptive grid solutions show increased resolution of the shock. The TVD limiter eliminated any oscillations near the shock.

The convergence histories were similar for each computation with the dynamic grid approaches being slightly less convergent. The convergence history for the dynamic grid computation which used the grid speed equations showed some occasional sharp spikes which may indicate movement of a grid point across the shock.

The dynamic grid approach which used the grid speed equations was more efficient for this problem because linear equations are solved rather than nonlinear grid equations. Using a time difference required 3% more memory than the static grid computation, whereas using the grid speed equations required 20% more memory.

The existence of grid-motion-induced errors was examined by comparing the time variation of the static pressure at the location $x = 8.0$ m in the nozzle. Figure 5 shows the variations for the dynamic grid approaches as compared to the variation for the static grid approach with a doubling of the grid density. The differences between the static pressure variations on the coarse static grid (95 × 15) and the fine static grid (190 × 30) were negligible. This suggested that the pressure variations on the static grids were not sensitive to the grid, and a proper grid density was used in the static and dynamic grid computations. From Fig. 5 it can be stated that there appears to be less error when the grid speed equations are used for the dynamic grid approach than when a first-order, backwards time difference is used to compute the grid speeds.

Application to Viscous Flows

The application of the dynamically adaptive grid method to viscous flows is discussed in Refs. 4 and 8. In the computations, the grid adapted to gradients in the Mach number to resolve laminar boundary layers along a flat plate for subsonic freestream flow, supersonic freestream flow, and supersonic freestream with a shock/boundary-layer interaction. In summary, it was found that the dynamically adaptive grid approach did adapt to the boundary layer but was unable to provide the proper grid resolution within the boundary layer. Improvements in applying the dynamic grid approach to viscous flows is under continued study.

Conclusions

An explicit, dynamically adaptive grid method has been presented and demonstrated for two-dimensional problems. The dynamic grids maintained the desired grid qualities nonetheless allowing grid motion due to motion of the boundaries and to adaptation of the grid to solution gradients. Using a dynamically adaptive grid required significantly more computational effort than using a static grid; however, the approach involving the grid speed equations was more efficient than the approach using finite differences of the grid to provide the grid speeds. Further work will be performed to increase the efficiency of the dynamic grid approaches. The results suggest that using the grid speed equations to compute the grid speeds reduces the grid-motion-induced errors more than using a backwards time difference of the grid. The grid adaptation improved the shock resolution; however, the resolution of boundary layers needs to be improved.

Acknowledgments

This work was supported through the NASA Graduate Student Researchers Program through the NASA Lewis Research Center, Contract NGT-50441, and was part of the doctoral dissertation of John Slater through Iowa State University.

References

[1] Hindman, R. G., Kutler, P., and Anderson, D. A., "Two-Dimensional Unsteady Euler-Equation Solver for Arbitrarily Shaped Flow Regions," *AIAA Journal*, Vol. 19, No. 4, 1981, pp. 424–431.

[2] Holcomb, J. E., and Hindman, R. G., "Development of a Dynamically Adaptive Grid Method for Multidimensional Problems," AIAA Paper 84-1668, June 1984.

[3] Brackbill, J. U., and Saltzman, J. S., "Adaptive Zoning for Singular Problems in Two Dimensions," *Journal of Computational Physics*, Vol. 46, No. 2, 1982, pp. 342–368.

[4] Slater, J. W., "A Dynamically Adaptive Mesh Method for Internal Flows," Ph.D. Dissertation, Iowa State Univ., Ames, IA, Dec. 1992.

[5] Liou, M.-S., and Hsu, A. T., "A Time-Accurate Finite Volume High Resolution Scheme for Three Dimensional Navier–Stokes Equations," AIAA Paper 89-1994, June 1989.

[6] Thomas, P. D., and Lombard, C. K., "Geometric Conservation Law and Its Application to Flow Computations on Moving Grids," *AIAA Journal*, Vol. 17, No. 10, 1979, pp. 1030–1037.

[7] Hindman, R. G., "Generalized Coordinate Forms of Governing Fluid Equations and Associated Geometrically Induced Errors," *AIAA Journal*, Vol. 20, No. 10, 1982, pp. 1359–1367.

[8] Slater, J. W., Liou, M.-S., and Hindman, R. G., "An Approach for Dynamic Grids," AIAA Paper 94-0319, Jan. 1994.