SIAM J. SCI. COMPUT.
Vol. 20, No. 3, pp. 811–825

AN ADAPTIVE GRID METHOD AND ITS APPLICATION TO STEADY EULER FLOW CALCULATIONS*

FENG LIU†, SHANHONG JI†, AND GUOJUN LIAO‡

Abstract. An adaptive remeshing procedure based on a cell volume deformation method is presented. Starting with an initial grid, this method offers direct cell volume control through the specification of the transformation Jacobian. Grid points are moved with appropriate grid velocities so that the specified cell volume distribution can be achieved at the end of the grid movement without adding or removing grid points. The grid velocities are determined by solving a scalar Poisson equation. This method is applied to solving the compressible Euler equations. Computational test cases of transonic flow over an airfoil are presented and demonstrate the desired control of grid sizes across shock waves.

Key words. Euler flows, adaptive grid, deformation methods, shocks

AMS subject classifications. 65N50, 76-08

PII. S1064827596305738

1. Introduction.

Generation of a suitable computational grid is the prerequisite for solving partial differential equations (PDEs) with either a finite volume or a finite difference method. Fixed grids have been widely used. With a fixed grid approach grid points are distributed in the physical domain before the solution is known. As a result, the grid may not be the best suited for the particular problem. The idea of an adaptive grid is to add, delete, or have the grid points move as the physical solution develops so that grid points may be concentrated in regions of large variation to enhance accuracy, and they may also be coarsened where the solution has little variation in order to achieve better efficiency.

For the general theory of grid generation, we refer the readers to the books [1, 2, 27]. Many structured or unstructured grid adaptation methods use local refining or coarsening by adding or removing grid points [3, 4, 5, 6, 7]. Reference [8] combined a grid moving algorithm (based on an algebraic method) and local grid refinement to solve time dependent PDEs. While such methods are very flexible in the sense that they are minimally constrained by the geometry of the problem to be solved, a distinct drawback is that the data structure of the grid and consequently that of the flow solver have to be adjusted every time the grid is adapted. Grid adaptation through moving grid point locations without adding or removing local grid points offers the advantage of not having to adapt the data structure of the flow solver. One may perceive this approach as one that optimizes grid distribution with a given total number of grid points. As such, more accurate solutions can be achieved with minimum change to an existing flow solver. This paper describes a deformation method for generating adaptive grids of this kind.

*Received by the editors June 24, 1996; accepted for publication (in revised form) June 16, 1997; published electronically October 7, 1998. This work was supported by NSF grant CTS-9410800.

http://www.siam.org/journals/sisc/20-3/30573.html

†Department of Mechanical and Aerospace Engineering, University of California, Irvine, CA 92717 (feng@jedi.eng.uci.edu). The work of these authors was supported by NSF grant CTS-9410800.

‡Department of Mathematics, University of Texas, Arlington, TX 76019-0408 (liao@math.uta.edu). The work of this author was supported by NSF grant DMS9106942.

To adapt computational grids to the evolution of flow solutions one has to have an effective grid generator that offers good control of the grid points. Strategies for grid generation include algebraic and PDE schemes. Algebraic methods use various sophisticated interpolation procedures. While some success has been attained with algebraic methods such as transfinite interpolations, the lack of flexibility in treating different geometry is a major disadvantage. A prominent PDE grid generation technique evolved from the initial work by Winslow [9] where the idea of using Laplace's equation to compute solutions for grid point locations was introduced. Thompson, Thames, and Mastin [10] introduced source terms to control mesh point locations in 1974. Since then grid generation based on elliptic PDEs has become very popular and successful.

In a grid adaptation method for structured grids without adding or removing grid points, adaptation is achieved through moving the grid points toward the desired locations. Changes in the mesh point locations can be controlled by two methods. In the first method, the arc elements forming the sides of a control volume are directly related to specified functions. For a three-dimensional problem, this implies that three arc elements need to be given. In the second method, the cell volume may be altered by specifying that the volume of each element change according to a specific rule. To control the cell size, only one relationship must be specified that relates the volume to the quantity responsible for changes in the mesh. The specification of only one control function is an advantage in simplicity but may be less flexible than independently controlling arc lengths.

Anderson and Steinbrenner [11] and Anderson [12] used Thompson's elliptic method as an adaptive grid generator. They derived explicit relations between arc length and the source terms of the Poisson equations. Grid control functions were modified to produce grids that approximately obey arc equidistribution laws. A weight function was required for each direction. In an attempt to reduce the number of controls or weight functions Anderson [13] showed that the diffusion parameter in Winslow's grid generation equations [9] was approximately proportional to local cell volume or area. The grid control functions in the standard Poisson grid generator could then be related to the diffusion parameter. Therefore, local cell volumes of the grid could be approximately enforced and related to the solution of the physical problem. As the physical solution evolves, the grid can be recomputed, providing grid point placement with prescribed cell volumes. Although this method offers minimum modification to their existing flow solver and elliptic grid generator, it has to solve a system of nonlinear elliptic equations every time the grid is adapted. Possible crossover of grids may also occur for certain source term distributions. Moreover, the cell volume size is only approximately satisfied.

In 1992, Liao and Anderson [14] proposed a new grid generation method based on studies by Moser [15], Banyaga [16], and Dacorogna and Moser [17] on volume elements on a Riemannian manifold. In this new approach a grid can be constructed by moving the grid points such that specified cell volumes can be achieved exactly. The grid points are moved according to a velocity field that can be determined by solving one linear Poisson equation, instead of a system of nonlinear elliptic equations needed in the approach by Anderson [13]. Furthermore, the mathematical principles behind this method guarantee that grid lines of the same grid family will not cross each other. This method appears to be very attractive for generating adaptive grids. In Liao and Anderson [14] the transformation Jacobian, and consequently the cell volumes, were specified on the old grid before adaptation. For use in a grid adap-

tation method, it is desirable to specify the cell volumes as functions of the physical coordinates of the new adapted grid locations. Liao, Pan, and Su [20] modified the original results of Dacorogna and Moser [17] so that cell volumes can be specified as functions of the new grid after adaptation. The method was further extended into a real time moving grid method by Liao and coworkers. In particular, Semper and Liao [19] used the moving grid method for one-dimensional unsteady problems. Simple examples of two-dimensional rectangular domains such as the pillbox problem and an artificially designed solution discontinuity resembling a shock were given by Liao, Pan, and Su [20] and Bochev, Liao, and de la Pena [21]. In this paper we will extend the use of this method to general two-dimensional domains with a numerical algorithm for solving the grid generation equations. The effectiveness of this method will be demonstrated for practical transonic flows by coupling the grid generation method with a finite volume flow solver for the steady compressible Euler equations.

In the following sections, the mathematical basis for the deformation approach of adaptive grid generation is presented first. We will then describe the numerical algorithms for solving the deformation equations in general domains. The flow solver for the Euler equations will be outlined along with the grid adaptation procedure. Examples including transonic flows over an airfoil will be given to illustrate the application of the method in two dimensions.

2. Mathematical formulation of the deformation method. A deformation scheme was first introduced by Moser in [15] in his study of volume elements of a compact Riemannian manifold. His method was modified in [20] for grid generation. In particular, the following problem was solved by a constructive method.

Problem: Let $\Omega \subset R^n$ be a bounded region with smooth boundary $\partial\Omega$, let $f(\underline{x})$ be a smooth function in Ω such that $f(\underline{x}) > 0 : \Omega \to R$ satisfies

$$\int_\Omega \left(\frac{1}{f} - 1\right) d\underline{x} = 0. \tag{1}$$

Find a one-to-one and onto transformation $\underline{\phi} : \Omega \to \Omega$ satisfying

$$\begin{aligned} \det \nabla \underline{\phi}(\underline{x}) &= f(\underline{\phi}(\underline{x})), \qquad \underline{x} \in \Omega, \\ \underline{\phi}(\underline{x}) &= \underline{x}, \qquad \underline{x} \in \partial\Omega. \end{aligned} \tag{2}$$

Notice that in the above transformation x can be regarded as the coordinates of an initial grid and $\underline{\phi}$ the coordinates of the new adapted grid. The function $f(\underline{\phi}(\underline{x}))$ is the transformation Jacobian, or in other words the cell size of the new grid. Hence, the solution of the problem allows us to specify the cell size distribution according to the location $\underline{\phi}(\underline{x})$ of the new grid cells.

$\underline{\phi}$ can be constructed by the following steps.

Step 1. Construct $\underline{v} : \Omega \to R^n$ such that $\underline{v}$ solves

$$\begin{aligned} \nabla \cdot \underline{v}(\underline{x}) &= h(\underline{x}) - 1, \qquad \underline{x} \in \Omega, \\ \underline{v}(\underline{x}) &= 0, \qquad \underline{x} \in \partial\Omega, \end{aligned} \tag{3}$$

where $h(\underline{x}) = \frac{1}{f(\underline{x})}$.

Step 2. Solve, at a fixed $\underline{x} \in \Omega$, the deformation equation

$$\begin{aligned} \frac{d}{dt}\underline{\varphi}(\underline{x}, t) &= \underline{\eta}(\underline{\varphi}(\underline{x}, t), t), \qquad 0 < t < 1, \\ \underline{\varphi}(\underline{x}, 0) &= \underline{x}, \end{aligned} \tag{4}$$

where t is an artificial time parameter and the deformation vector field $\underline{\eta}$ is defined by

$$\underline{\eta}(\underline{y},t) = \frac{-\underline{v}(\underline{y})}{(1-t)+th(\underline{y})}. \tag{5}$$

Step 3. Set

$$\underline{\phi}(\underline{x}) = \underline{\varphi}(\underline{x},1). \tag{6}$$

Then $\underline{\phi}$ satisfies (2). In fact, Appendix A of [20] shows by direct computation that H, defined by

$$H(t,\underline{x}) = \det \nabla \underline{\varphi}(\underline{x},t)[(1-t)+th(\underline{\varphi}(\underline{x},t))],$$

is independent of the parameter t. Consequently, we have

$$H(0,\underline{x}) = H(1,\underline{x}),$$

which is equivalent to

$$h(\underline{\varphi}(\underline{x},1)) \det \nabla \underline{\varphi}(\underline{x},1)) = \det \nabla \underline{\varphi}(\underline{x},0),$$

i.e.

$$h(\underline{\phi}(\underline{x})) \det \nabla \underline{\phi}(\underline{x}) = 1.$$

Therefore,

$$\det \nabla \underline{\phi}(\underline{x}) = f(\underline{\phi}(\underline{x})), \qquad \underline{x} \in \Omega.$$

Notice that this transformation then yields exactly the prescribed Jacobian, or in other words, the specified cell volume. In our adaptive grid method for fluid flow problems, monitors for the flow field are designed to yield a desired cell volume distribution and then the above method is used to move the grid points for better resolution of the flow field. An Euler flow solver based on a finite volume method for steady flows with multigrid [22] is coupled with this grid adaptation scheme. The flow solver will be described in section 4.

3. Numerical solution of the deformation equations. From the previous section, the transformation $\underline{\phi}(\underline{x})$ can be constructed in three steps. In the first step, equation (3) can be solved by expressing $\underline{v}$ as the sum of the gradient of a single valued potential function w and the curl of another real valued function b, so that we have

$$\underline{v} = \nabla w + curl^* b, \tag{7}$$

where the $curl^* b$ term is defined as

$$curl^* b = \left(\frac{\partial b}{\partial x_2}, -\frac{\partial b}{\partial x_1} \right)$$

in two dimensions and is the standard curl in three dimensions. This term was added to insure that boundary points remain fixed in [17]. However, in many applications of

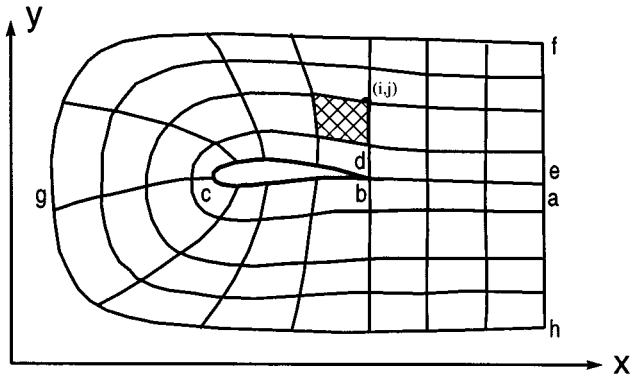

FIG. 1. *General curvilinear grid on physical plane.*

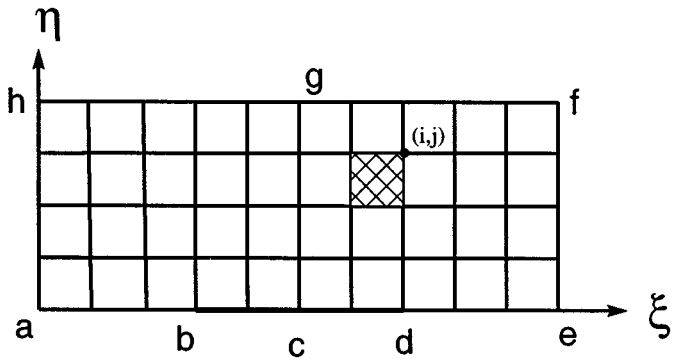

FIG. 2. *Uniform grid on transformed computational plane.*

adaptive gridding strategies, the boundary points are left free to move. And, in fact, the general case would seem to need point motion along boundaries with the exception that some special points such as corner points may be fixed. In the computation of this paper, the *curl*b* term is neglected, and the boundary points are allowed to move along the boundary line with corner points fixed. Since *curl*b* is divergence-free, the problem may then be written in terms of the potential as

$$\nabla^2 w = h(\underline{x}) - 1, \qquad \underline{x} \in \Omega \tag{8}$$

subject to the Neumann boundary conditions

$$\frac{\partial w}{\partial n} = 0, \qquad \underline{x} \in \partial\Omega.$$

For simple geometry such as a square box or an annulus, the above Poisson equation with the Neumann boundary conditions can be solved analytically as was done by Liao and Anderson [14]. For complex geometries, however, equation (8) must be solved numerically on curvilinear grids. In this paper, we use a finite difference method with central differencing and successive line over relaxation (SLOR). Figure 1 shows a general curvilinear grid around an airfoil. In order to solve (8), the physical plane (x, y) in Figure 1 is transformed to a computational plane (ξ, η) shown in

Figure 2 through the relations

$$\begin{cases} x = x(\xi, \eta), \\ y = y(\xi, \eta). \end{cases} \tag{9}$$

Equation (8) can be written in standard tensor notation on the curvilinear system (ξ, η) as

$$\begin{aligned} \nabla^2 w &= \frac{1}{\sqrt{g}} \left[\frac{\partial}{\partial \xi} \sqrt{g} \underline{g}^{\xi} \cdot \left(\underline{g}^{\xi} \frac{\partial w}{\partial \xi} + \underline{g}^{\eta} \frac{\partial w}{\partial \eta} \right) + \frac{\partial}{\partial \eta} \sqrt{g} \underline{g}^{\eta} \cdot \left(\underline{g}^{\xi} \frac{\partial w}{\partial \xi} + \underline{g}^{\eta} \frac{\partial w}{\partial \eta} \right) \right] \\ &= h(\xi, \eta) - 1, \end{aligned} \tag{10}$$

where $\underline{g}^{\xi}$ and $\underline{g}^{\eta}$ are the contravariant basis vectors

$$\underline{g}^{\xi} = (\xi_x, \xi_y)^T, \qquad \underline{g}^{\eta} = (\eta_x, \eta_y)^T,$$

and $\sqrt{g}$ is the transformation Jacobian

$$\sqrt{g} = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{vmatrix}.$$

Equation (10) can be discretized on the computational plane with a uniform grid by the finite difference scheme

$$\begin{aligned} (\nabla_h^2 w)_{ij} = \tfrac{1}{\sqrt{(g)_{ij}}} \Big\{ & \sqrt{(g)_{i+\frac{1}{2},j}}\, \underline{g}^{\xi}_{i+\frac{1}{2},j} \cdot \left[\underline{g}^{\xi}_{i+\frac{1}{2},j} (\tfrac{\partial w}{\partial \xi})_{i+\frac{1}{2},j} + \underline{g}^{\eta}_{i+\frac{1}{2},j} (\tfrac{\partial w}{\partial \eta})_{i+\frac{1}{2},j} \right] \\ & - \sqrt{(g)_{i-\frac{1}{2},j}}\, \underline{g}^{\xi}_{i-\frac{1}{2},j} \cdot \left[\underline{g}^{\xi}_{i-\frac{1}{2},j} (\tfrac{\partial w}{\partial \xi})_{i-\frac{1}{2},j} + \underline{g}^{\xi}_{i-\frac{1}{2},j} (\tfrac{\partial w}{\partial \eta})_{i-\frac{1}{2},j} \right] \\ & + \sqrt{(g)_{i,j+\frac{1}{2}}}\, \underline{g}^{\eta}_{i,j+\frac{1}{2}} \cdot \left[\underline{g}^{\xi}_{i,j+\frac{1}{2},j} (\tfrac{\partial w}{\partial \xi})_{i,j+\frac{1}{2}} + \underline{g}^{\eta}_{i,j+\frac{1}{2}} (\tfrac{\partial w}{\partial \eta})_{i,j+\frac{1}{2}} \right] \\ & - \sqrt{(g)_{i,j-\frac{1}{2}}}\, \underline{g}^{\eta}_{i,j-\frac{1}{2}} \cdot \left[\underline{g}^{\xi}_{i,j-\frac{1}{2}} (\tfrac{\partial w}{\partial \xi})_{i,j-\frac{1}{2}} + \underline{g}^{\eta}_{i,j-\frac{1}{2}} (\tfrac{\partial w}{\partial \eta})_{i,j-\frac{1}{2}} \right] \Big\} \\ = {} & h_{ij} - 1, \end{aligned} \tag{11}$$

where $(\frac{\partial w}{\partial \xi})_{i-\frac{1}{2},j}$, $(\frac{\partial w}{\partial \eta})_{i-\frac{1}{2},j}$, etc., are to be approximated by standard central differencing formulas on the computational domain. Equation (11) is then solved by the standard SLOR method.

In Steps 2 and 3, the deformation equations form a series of ordinary differential equations (ODEs), a standard fourth-order Runge–Kutta scheme is used to integrate from the initial point, and the time step used is 1/20.

4. Numerical solution of the Euler equations. The two-dimensional compressible Euler equations can be written in integral form as

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{W}\, dV + \oint_{\partial \Omega} [\mathbf{E}\, dS_x + \mathbf{F}\, dS_y] = 0 \tag{12}$$

for a fixed region Ω with boundary $\partial\Omega$,

$$\mathbf{W} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} \rho u \\ \rho u u + p \\ \rho u v \\ \rho u H \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v v + p \\ \rho v H \end{bmatrix},$$

where p, ρ, u, v, E, and H denote the pressure, density, Cartesian velocity components, total energy, and total enthalpy. For a perfect gas,

$$E = \frac{p}{(\gamma-1)\rho} + \frac{1}{2}(u^2+v^2), \qquad H = E + \frac{p}{\rho},$$

where γ is the ratio of specific heats.

Jameson, Schmidt, and Turkel [23] developed a finite volume method with multigrid for solving the above equations. In this method, equation (12) is applied to each cell of the computational grid, for example, the shaded cell marked by its upper right corner index (i, j) in Figure 1. A system of ODES is then obtained by approximating the surface integral of (12) by a finite volume scheme. This yields a semidiscrete equation

$$\frac{d}{dt}(\Omega_{ij}\mathbf{W}_{ij}) + \mathbf{Q}_{ij} = 0, \tag{13}$$

where Ω_{ij} is the cell volume, $\mathbf{W}_{ij}$ is the average flow variable over the cell, and $\mathbf{Q}_{ij}$ is the finite volume approximation for the net flux out of the cell. $\mathbf{Q}_{ij}$ can be evaluated as

$$\mathbf{Q}_{ij} = \sum_{k=1}^{4} [\mathbf{E}_k(\Delta S_x)_k + \mathbf{F}_k(\Delta S_y)_k], \tag{14}$$

where $\mathbf{E}_k$ and $\mathbf{F}_k$ denote values of the flux vectors $\mathbf{E}$ and $\mathbf{F}$ on the kth face of the cell, and $(\Delta S_x)_k$ and $(\Delta S_y)_k$ are the x and y components of the face area vector. $\mathbf{E}_k$ and $\mathbf{F}_k$ can be evaluated by taking the averages of $\mathbf{E}$ and $\mathbf{F}$, respectively, on either side of the cell face.

The scheme constructed in this manner reduces to a central difference scheme on Cartesian grids and is formally second-order accurate in space. For a fixed Ω_{ij}, equation (13) can be written as

$$\frac{d\mathbf{W}_{ij}}{dt} + \mathbf{R}_{ij}(\mathbf{W}) = 0, \tag{15}$$

where $\mathbf{R}_{ij}$ is the residual

$$\mathbf{R}_{ij}(\mathbf{W}) = \frac{1}{\Omega_{ij}}(\mathbf{Q}_{ij} - \mathbf{D}_{ij})$$

and $\mathbf{D}_{ij}$ is an artificial dissipation term which is added to prevent odd and even decoupling and to capture shock waves without oscillations in a transonic flow field. $\mathbf{D}_{ij}$ is of higher order than the convective term $\mathbf{Q}_{ij}$ except near shock waves where first order is needed to prevent nonphysical oscillations across shocks.

Equation (15) is integrated in time by an explicit multistage scheme. Let $\mathbf{W}^n$ be the value of $\mathbf{W}_{ij}$ after n time steps. Dropping the subscripts i, j, a general m-stage hybrid scheme to advance a time step Δt can be written as

$$\begin{aligned}
\mathbf{W}^{(0)} &= \mathbf{W}^n, \\
\mathbf{W}^{(1)} &= \mathbf{W}^{(0)} - \alpha_1 \quad \Delta t\mathbf{R}^{(0)}, \\
&\cdots \\
\mathbf{W}^{(m-1)} &= \mathbf{W}^{(0)} - \alpha_{m-1}\, \Delta t\mathbf{R}^{(m-2)}, \\
\mathbf{W}^{(m)} &= \mathbf{W}^{(0)} - \quad \Delta t\mathbf{R}^{(m-1)}, \\
\mathbf{W}^{n+1} &= \mathbf{W}^{(m)}.
\end{aligned}$$

For time-accurate problems, the coefficients α_m can be chosen to obtain desired order of accuracy in time. For steady state calculations, time marching is used to reach the steady state solution as fast as possible so that α_m can be optimized to yield the best convergence rate rather than trying to maintain high order of time accuracy. For our calculations we use a five-stage scheme with the coefficients

$$\alpha_1 = \frac{1}{4}, \quad \alpha_2 = \frac{1}{6}, \quad \alpha_3 = \frac{3}{8}, \quad \alpha_4 = \frac{1}{2}.$$

In order to further increase the rate of convergence to steady state, locally varying time steps, residual averaging, and multigrid methods are used. The readers are referred to Jameson, Schmidt, and Turkel [23]; Jameson [24]; Liu and Jameson [25]; and Liu and Ji [22] for the details of the finite volume discretization, time marching and the multigrid method, and the various well-validated computational results by this method for two- and three-dimensional transonic flows over airfoils and through turbomachinery cascades.

5. Application and numerical results.

5.1. Deformation from uniform grid. For simplicity of presentation, a two-dimensional square domain is chosen to demonstrate the deformation method.

Let $\Omega = (0,1)^2$. We start from a uniform mesh with 57×57 grid points. We want to construct a coordinate transformation $\underline{\phi} : \Omega \to \Omega$ such that the adapted grid concentrates points around the circle $(x-0.5)^2 + (y-0.5)^2 = 0.2^2$. The nodes are deformed to a new grid with the following prescribed cell sizes.

Define f:

$$\frac{1}{f} = \begin{cases} A & \text{if} \quad 0 \le r < a, \\ A + (B-A)\frac{r-a}{b-a} & \text{if} \quad a \le r < b, \\ B + (1-B)\frac{r-b}{0.25-b} & \text{if} \quad b \le r < 0.25, \\ 1 & \text{if} \quad r \ge 0.25, \end{cases}$$

where

$$r = |0.20 - \sqrt{(x-0.5)^2 + (y-0.5)^2}|,$$
$$a = 0.05, \quad b = 0.1, \quad A = 1.5, \quad B = 0.625.$$

The corresponding $\underline{\phi}$ satisfies (2). Figure 3 shows the contours of the analytically defined f in the domain. Figure 4 shows the grid after deformation. It can be seen in Figure 4 that the grid cell sizes are much small near the circle $(x-0.5)^2+(y-0.5)^2 = 0.2^2$ so the grid points are concentrated as designed. Figure 5 is the contour plot of the grid size distribution of the calculated grid. Clearly, it agrees with the specified f function contours shown in Figure 3. This demonstrates that the computed grid does yield the desired cell sizes as specified by the function f.

5.2. Euler flow over an airfoil. The cell volume control method is applied successfully to calculating transonic Euler flows with shock waves. The method is applied to computing the flow field over an airfoil. Figure 6 shows the initial C-mesh of an NACA0012 airfoil (see, for example, reference [26] for the geometric definition of the airfoil). This grid is obtained by an algebraic method based on transfinite interpolation (see, for example, reference [1]). Two flow cases were calculated over the initial grid. The first is a transonic case with free stream Mach number $M_\infty = 0.85$

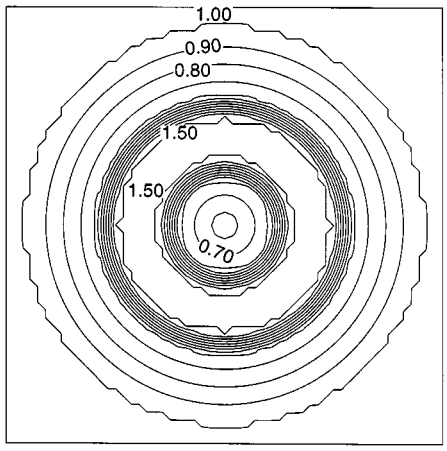

FIG. 3. *Contours of the specified Jacobian.*

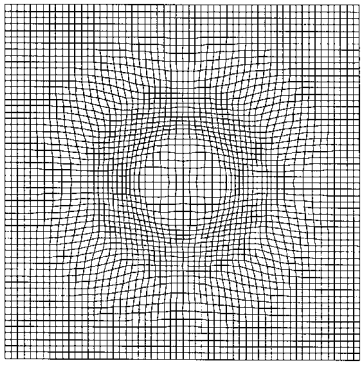

FIG. 4. *Grid based on the specified Jacobian.*

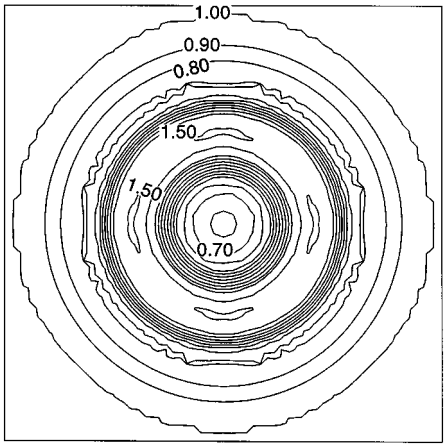

FIG. 5. *Contours of the cell areas of the computed grid.*

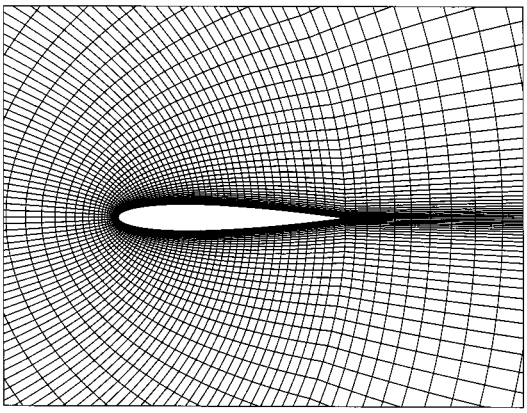

FIG. 6. *Mesh around NACA*0012 *airfoil without adaptation.*

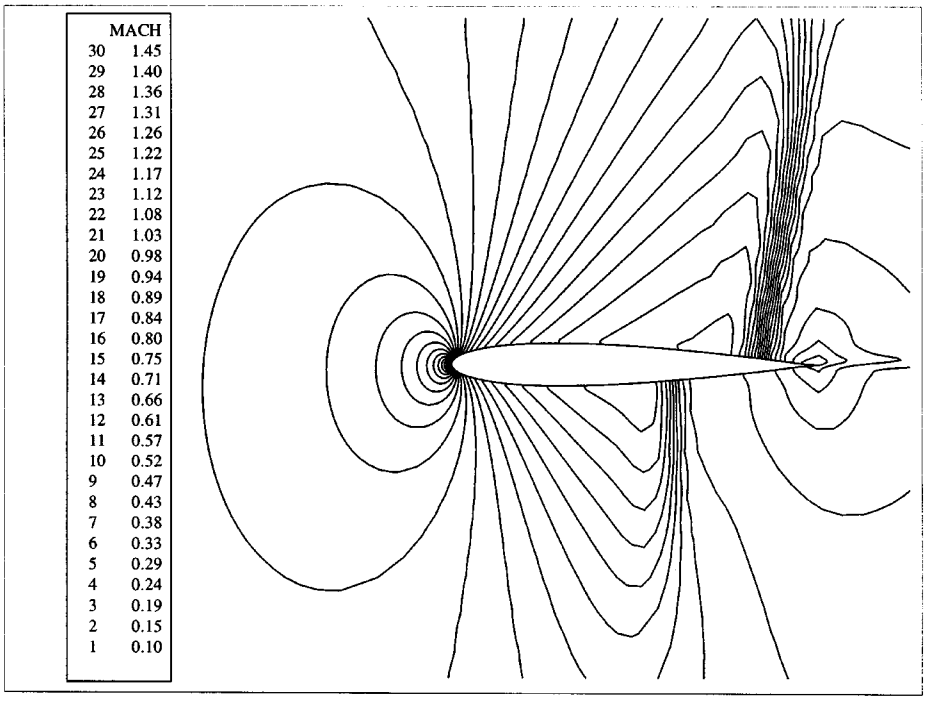

FIG. 7. *Mach number contours of transonic flow around NACA*0012 *airfoil computed on grid without adaptation.*

and an angle of attack $\alpha = 1^\circ$. Figure 7 plots the Mach number contours computed on the initial unadapted grid shown in Figure 6. The Mach number levels of the plotted contour lines are listed on the left of the figure. There is one strong shock wave on the upper side of the airfoil and a weaker one on the lower side. It can be seen from Figure 7 that the computed shock waves are rather thick. Shock waves are fronts of large gradients in the flow field. In fact, a theoretical shock should have

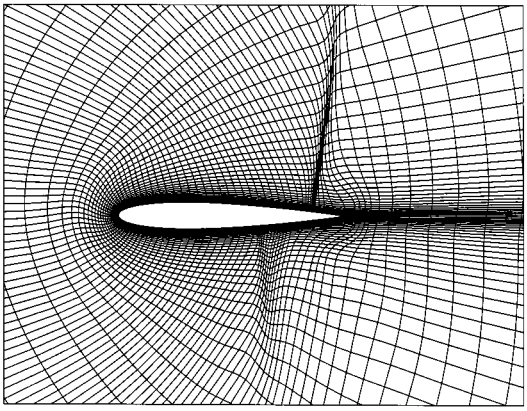

FIG. 8. *Mesh around NACA*0012 *airfoil after adaptation for transonic flow calculations.*

zero thickness in the inviscid limit. To get better computational results, particularly to capture the shock waves more accurately, one would like to concentrate grid points around the shock waves. The deformation method is applied to get a new grid with prescribed distribution of cell sizes based on gradients of the flow field. The adaptive criterion here is to detect the shock waves. This suggests choosing the monitor function f of the form

$$\frac{1}{f} = C_1(1 + C_2 \nabla P), \tag{16}$$

where P is the pressure and C_1 and C_2 are constants. In the computation, $C_2 = 20$, and C_1 is determined by satisfying equation (1). Figure 8 shows the new grid after adaptation based on the above criterion. It can be seen that grid points are clustered closely in the areas where the two shock waves occur, although grid lines are somewhat skewed in the clustered regions because the deformation method does not guarantee orthogonality. However, since our flow solver is based on a finite volume scheme which does not require the use of an orthogonal grid, we are content with the locally reduced cell sizes. Figure 9 shows the Mach number contours calculated on this new grid. Clearly, much better shock wave resolutions are obtained with the new adapted grid. The computed shock waves are better defined than those computed on the initial unadapted grid.

The computation is done on a mesh with 161×49 grid points. The Euler flow solver takes 1160 seconds on an SGI Indigo 2 with a 150 MHz R4000 CPU to reach convergence with 7 orders of magnitude reduction in the root-mean-squared residuals. Since the grid adaptation scheme using the proposed deformation method does not add or subtract grid points and it does not change the data structure of the flow solver, the computational effort for the flow solver is the same with or without the grid adaptation. The grid adaptation procedure takes 180 seconds on the same machine, only a fraction of the computational time for the flow solver. Most of that time is spent on solving the Poisson equation to obtain the grid deformation velocity. We maintain a residual reduction of 8 orders of magnitude for the SLOR solution of the Poisson equation. In practice, it is not necessary to reach such a low level of residuals for the Poisson solver. Consequently, the time for the grid adaptation can be further reduced.

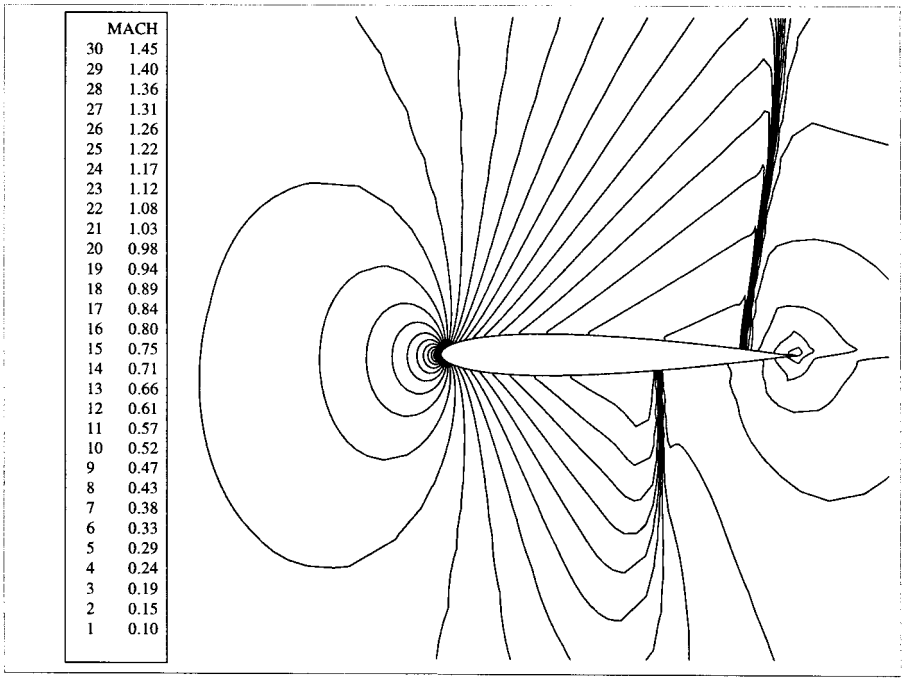

FIG. 9. *Mach number contours of transonic flow around NACA*0012 *airfoil computed on adapted grid.*

Another test case is the supersonic flow over the same airfoil with a free stream Mach number $M_\infty = 1.5$, and $\alpha = 0°$. Figure 10 shows the Mach number contours computed on the initial unadapted grid (Figure 6). As can be seen, a strong bow shock wave appears in front of the airfoil leading edge. In addition, there are two weak shocks emanating from the trailing edge of the airfoil. By applying the above adaptive grid procedure to the same initial grid in Figure 6, a new grid shown in Figure 11 is generated by the deformation method. Grids are automatically concentrated where the bow shock appears. Clustering of grid points around the weaker trailing edge shocks are less obvious in Figure 11 because the shocks are not strong enough to increase adequately the adaptation function $\frac{1}{f}$ in equation (16). Figure 12 shows the Mach number distribution computed on the adapted grid. It can be seen that a sharper front of the bow shock is captured compared with that on the initial unadapted grid. The resolution of the two trailing edge shocks is also slightly increased. The computational time needed for the grid adaptation and the flow solver for the supersonic case is the same as that for the transonic case.

6. Concluding remarks. The deformation method of grid generation is a structured grid generator which controls the cell size directly and precisely. The adaptive remeshing procedure moves each node from an initial grid to a new position according to a system of n ODEs, where n is the spatial dimension. The control over the cell size is achieved by making the Jacobian determinant equal to a positive monitor function, which is constructed according to the fine structures of the flow.

The application of this method to grid adaptation for Euler flows demonstrates that the cell control deformation method is an effective scheme to obtain sharp resolution of flow features with large gradients such as shock waves. Grid points are

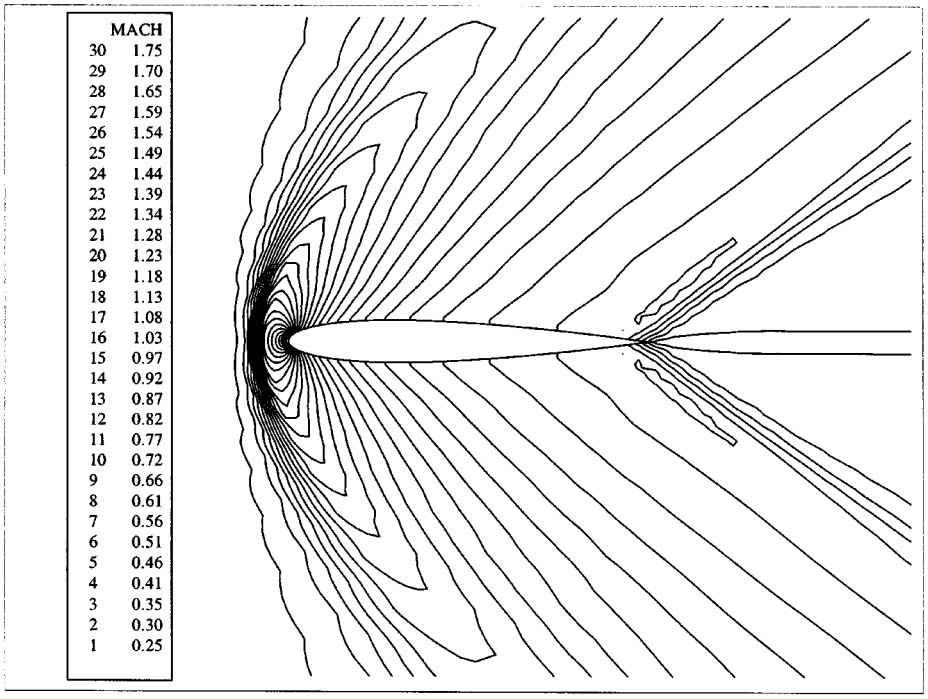

FIG. 10. *Mach number contours of supersonic flow around NACA*0012 *airfoil computed on grid without adaptation.*

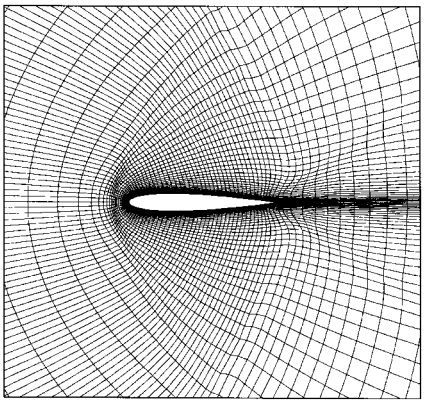

FIG. 11. *Mesh around NACA*0012 *airfoil after adaptation for supersonic flow calculations.*

automatically clustered around shocks without adding extra grid points and changing the data structure of the flow solver. Therefore, the resolution of shocks or other sharp fronts can be greatly improved with only a small increase in computation time needed for adapting the grids. Currently, the authors are working on applying the real time deformation this method to dynamic grid adaptation for time dependent PDEs.

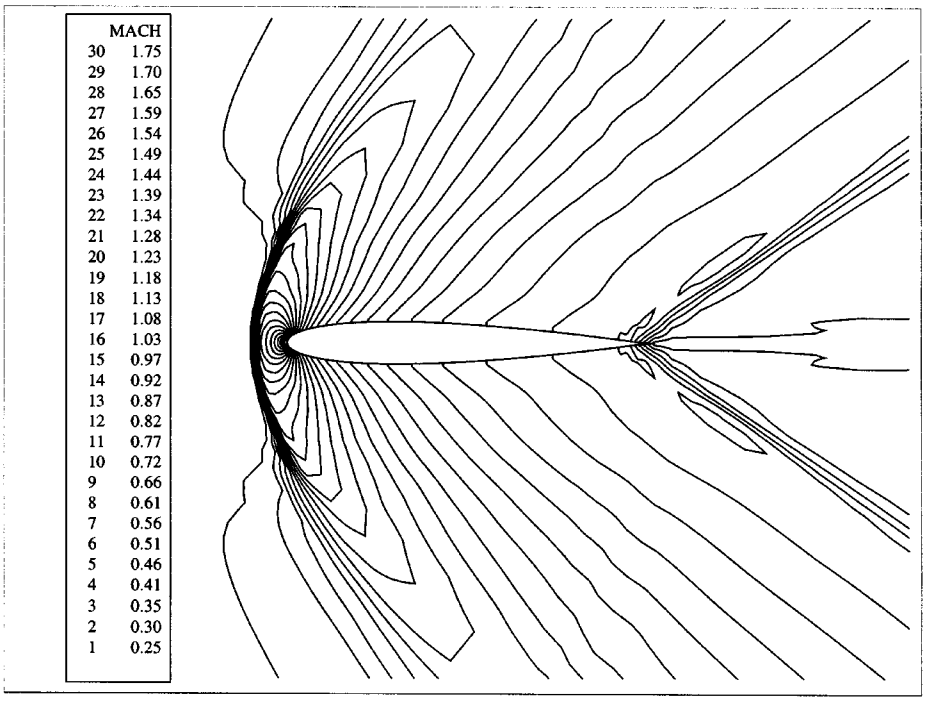

FIG. 12. *Mach number contours of supersonic flow around NACA*0012 *airfoil computed on adapted grid.*

Acknowledgments. We would like to thank the reviewers of this paper for their detailed comments which helped us improve the presentation of the paper. Liao would also like to thank Professor Antony Jameson for his valuable suggestions and encouragement in this work.

REFERENCES

[1] J. F. THOMPSON, Z. U. A. WARSI, AND W. C. MASTIN, *Numerical Grid Generation: Foundations and Applications*, Elsevier, New York, 1985.

[2] P. M. KNUPP AND S. STEINBERG, *The Fundamentals of Grid Generation*, CRC Press, Boca Raton, FL, 1993.

[3] I. BABUSKA, O. C. ZIENKIEWICZ, J. CAGO, AND E. R. OLIVEIRA, EDS., *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, I. Babuska et al., eds., Wiley, New York, 1986.

[4] Y. KALLINDERIS AND J. R. BARON, *Adaptation methods for a new Navier-Stokes algorithm*, AIAA J., 27 (1989), pp. 37–43.

[5] D. J. MAVRIPLIS, *Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes*, AIAA J., 28 (1990), pp. 213–221.

[6] D. DE ZEEUW AND K. G. POWELL, *An adaptive refined Cartesian mesh solver for the Euler equations*, J. Comput. Phys., 104 (1993), pp. 56–68.

[7] F. LIU AND L. WANG, *A solution adaptive unstructured mesh method for cascade flow calculations*, in Proc. ASME Symposium on Computational Fluid Dynamics for Aeropropulsion, ASME AD, Vol. 49, November 1995.

[8] D. C. ARNEY AND J. E. FLAHERTY, *An adaptive mesh-moving and local refinement method for time dependent partial differential equations*, ACM Trans. Math. Software, 16 (1990), pp. 48–71.

[9] A. WINSLOW, *Numerical solution of the quasi-linear Poisson equation*, J. Comput. Phys., 1 (1966), pp. 149–172.

[10] J. F. Thompson, F. D. Thames, and C. W. Mastin, *Automatic numerical generation of body-fitted coordinate system for fields containing and number of arbitrary two-dimensional bodies*, J. Comput. Phys., 15 (1974), pp. 229–319.
[11] D. A. Anderson and J. Steinbrenner, *Generating Adaptive Grids with a Conventional Grid Scheme*, AIAA paper 86-0427, AIAA, New York, 1986.
[12] D. A. Anderson, *Equidistribution schemes, Poisson generators, and adaptive grids*, Appl. Math. Comput., 24 (1987), pp. 211-277.
[13] D. A. Anderson, *Grid cell volume control with an adaptive grid generator*, Appl. Math. Comput. 35 (1990), pp. 209–217.
[14] G. Liao and D. Anderson, *A new approach to grid generation*, Appl. Anal., 44 (1992), pp. 285–298.
[15] J. Moser, *On the volume elements of a manifold*, Trans. AMS 120, pp. 286–294, 1965.
[16] A. Banyaga, *Formes volumesur les varietes a bord*, Enseign. Math., 20 (1974), pp. 127–131.
[17] B. Dacorogna and J. Moser, *On a partial differential equation involving Jacobian determinant*, Ann. Inst. H. Poincaré Anal. Non Linéaire, 7 (1990), pp. 1–26.
[18] G. Liao and J. Su, *A direct method in Dacorogna-Moser's approach of grid generation*, Appl. Anal., 49 (1993), pp. 73–84.
[19] B. Semper and G. Liao, *A moving grid finite-element method using grid deformation*, Numer. Methods Partial Differential Equations, 11 (1995), pp. 603–615.
[20] G. Liao, T.-W. Pan, and J. Su, *A numerical grid generator based on Moser's deformation method*, Numer. Methods Partial Differential Equations, 10 (1994), pp. 21–31.
[21] P. Bochev, G. Liao, and G. de la Pena, *Analysis and computation of adaptive moving grids by deformation*, Numer. Methods Partial Differential Equations, 12 (1996), p. 489.
[22] F. Liu and S. Ji, *Unsteady flow calculations with a multigrid Navier-Stokes method*, AIAA J., 34 (1996), pp. 2047–2053.
[23] A. Jameson, W. Schmidt, and E. Turkel, *Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes*, AIAA paper 81-1259, AIAA, New York, 1981.
[24] A. Jameson, *Transonic Flow Calculations*, MAE Report 1651, Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ, 1983.
[25] F. Liu, and A. Jameson, *Cascade flow calculations by a multigrid Euler method*, J. Propulsion and Power, 9 (1993), pp. 90–97.
[26] I. H. Abbott and A. E. von Doenhoff, *Theory of Wing Sections*, Dover, New York, 1959.
[27] G. Carey, *Computational Grids: Generation, Adaptation, and Solution Stategies*, Taylor & Francis, Philadelphia, 1997.