Self-Adaptive-Grid Method with Application to Airfoil Flow

Kazuhiro Nakahashi*
National Aerospace Laboratory, Tokyo, Japan
and
George S. Deiwert†
NASA Ames Research Center, Moffett Field, California

A self-adaptive-grid method that is suitable for multidimensional steady and unsteady computations is described. Based on variational principles, a spring analogy is used to redistribute grid points in an optimal sense to reduce the overall solution error. User-specified parameters, denoting both maximum and minimum permissible grid spacings, are used to define the all-important constants, thereby minimizing the empiricism and making the method self-adaptive. Operator splitting and one-sided controls for orthogonality and smoothness are used to make the method practical, robust, and efficient. Examples are included for both steady and unsteady viscous flow computations about airfoils in two dimensions, as well as for a steady inviscid flow computation and a one-dimensional case. These examples illustrate the precise control the user has with the self-adaptive method and demonstrate a significant improvement in accuracy and quality of the solutions.

Introduction

SOLUTION-adaptive-grid methods are an important subject of study in computational fluid dynamics because of their potential for improving the efficiency and accuracy of numerical methods. Based on the idea of Gnoffo,[1] we have developed a practical multidimensional solution-adaptive-grid method which uses tension- and torsion-spring analogies, and have successfully applied it to two-dimensional[2] and three-dimensional[3] flow problems. Several constants and parameters are incorporated in this method in order to control such features as orthogonality and smoothness; until now, appropriate values for them had to be determined empirically. Some considerable flexibility is afforded by this method, but the appropriate determination of the various constants and parameters, as well as their relationship to one another, has been determined in an ad hoc trial-and-error manner, and direct control of an optimal grid adaption procedure has not been possible.

The purpose of this paper is to remove this empiricism and uncertainty and to describe a procedure that uses certain constraints on the grid to make it self-adaptive. In particular, an equidistribution concept based on a maximum and minimum permissible grid spacing is developed for line adaptation and applied to multidimensional flows. The technique described is not restricted to the present method alone; on the contrary, the concept and procedure are applicable to a variety of adaptive-grid methods.

The flow past an airfoil is used to illustrate the applicability of this method. Particular emphasis is placed on interacting viscous flow cases. The adaptive grid permits a more precise description of shock waves and separated shear layers.

Presented as Paper 85-1525 at the AIAA Seventh Computational Fluid Dynamics Conference, Cincinnati, OH, July 15–17, 1985; received Oct. 31, 1985; revision received Aug. 4, 1986.

*NRC Research Associate. Member AIAA.
†Research Scientist. Associate Fellow AIAA.

Method

In this section we first consider some of the conceptual formalisms, based on variational principles, that are used in the development of an adaptive-grid scheme. It is then shown how some of the mathematical constraints can be simplified in form, although not in content, to produce an adaptive-grid scheme that is practical, efficient, and robust, and how this scheme can be applied to multidimensional grid adaptation.

Conceptual Approach

Consider a grid as an organized set of points formed by the intersection of lines of a boundary-conforming coordinate system. The problem associated with both grid generation and grid adaptation is to establish the mapping relationship between physical space and computational space. In the computational space, the points are uniformly distributed in Cartesian coordinates. Here the computational domain will be assumed to be normalized to unity in all directions. In the physical space, the points are clustered and stretched over nonorthogonal curvilinear coordinates. Ideally, an optimum grid is one in which the numerical solution error (which results from discretization and is usually reflected by the residuals) is uniformly distributed over all grid points. Since the solution error is not known a priori, the only way to approach an optimum grid is to adapt the grid to the solution in a time-like manner. This time-like adaptation can be realized in either of two ways. One is to keep the computational space fixed and include the grid speed in the flowfield equations themselves. This is the ideal method to use for unsteady flows or for schemes in which the grid evolves with the solution at each time step. For simplicity of illustration, consider the one-dimensional transformation to computational space where

$$t \mapsto \tau \qquad x \mapsto \xi(x,t)$$

The time rate of change of the dependent variable u then becomes

$$\frac{\partial u}{\partial t} \mapsto \frac{\partial u}{\partial \tau} + \frac{\partial \xi}{\partial t}\frac{\partial u}{\partial \xi}$$

where $\partial\xi/\partial t$ is the grid speed which must be either specified or determined numerically. A convenient method for ap-

proximating the grid speed numerically has been to use backward first-order differences in time.

The second way to adapt the grid in a time-like manner is to set the grid speed $\partial\xi/\partial t$ equal to zero and interpolate the solution onto the new grid after each adaptation. This is equivalent to solving a sequence of initial boundary-value problems and is an economical way to treat steady flows whose solutions are approached asymptotically. Here it is generally sufficient to adapt the grid just a few times during the course of obtaining the steady-state solution.

There are several constraints to consider when adapting the grid to the solution. It is desirable to use a measure of the solution error as the driving mechanism for redistribution. At the same time there should be some means of controlling grid smoothness and orthogonality (or skewness). The points must also be free to concentrate, but it is not desirable to have regions completely devoid of points. And, finally, it must be possible to control the grid speed when it is not set equal to zero.

Following the variational approach described by Brackbill and Saltzman[4] for one-dimensional adaptation, the total error is reduced when the grid points are distributed such that

$$\int_{x_i}^{x_{i+1}} w(x)\,dx = \text{const} \tag{1}$$

for all i, where $w(x)$ is a positive weighting function. Ideally, $w(x)$ would be the magnitude of the truncation error that results from the discretized solution. Practically, however, this error is not easy to determine, and if it is approximated by higher-order differences it tends to be "noisy." Generally the largest numerical errors are found in regions where the solution is changing most rapidly. Hence, gradients of the solution itself are often used as a suitable driving mechanism for grid adaptation.

To maintain a uniform grid in ξ space (i.e., computational space), Eq. (1) implies that

$$x_\xi w(\xi) = \text{const} \tag{2}$$

where x_ξ is the metric coefficient and corresponds to the ratio of arc lengths in physical and computational space. Equation (2) is the Euler-Lagrange equation for the minimization of the integral

$$I_w = \int_0^1 w(\xi)\, x_\xi^2\, d\xi \tag{3}$$

The minimization of this integral is analogous to minimizing the energy of a system of springs with constants $w(\xi)$ between each pair of grid points. The constant in Eq. (2) can be evaluated, and an expression for grid-point spacing obtained (e.g., see Ref. 5)

$$\Delta x_i = 1\Big/\left[w(\xi_i) N \int_0^1 \frac{d\xi}{w(\xi)} \right] \tag{4}$$

where N is the total number of grid intervals.

Generalized forms for the weighting function w can be given by

$$w(x) = a_0 + a_1|u_x| + a_2|u_{xx}| + \ldots \tag{5}$$

or, as suggested by White,[6]

$$w(x) = \left[\alpha + |u^{(n)}|^2\right]^{1/2n} \tag{6}$$

From White's expression we have for $\alpha = 0$, $n = 1$, a constant change in the solution u over each interval; and for $\alpha = 1$, $n = 1$, a uniform distribution of arc length on the solution curve. For $n > 1$, the grid can be related to the truncation error.

For multidimensional adaptation, smoothness and orthogonality constraints must be considered. For example, in two-dimensions, smoothness can be enforced by minimizing the integral

$$I_s = \int_{\text{vol}} \left[(\nabla\xi)^2 + (\nabla\eta)^2 \right] d\text{vol}$$

$$= \int\int \frac{\left[(x_\xi)^2 + (x_\eta)^2 + (y_\xi)^2 + (y_\eta)^2 \right]}{J}\, d\xi\, d\eta \tag{7}$$

Here the smoothness is measured by integrating the change in computational coordinates over the physical domain. The mapping is given by the Laplace equation when the Euler-Lagrange equations are constructed.

Orthogonality can be enforced by minimizing the integral

$$I_o = \int_{\text{vol}} (\nabla\xi \cdot \nabla\eta)^2 J^3\, d\text{vol} = \int\int (x_\xi x_\eta + y_\xi y_\eta)^2\, d\xi\, d\eta \tag{8}$$

In actual practice, it is necessary to consider a combination of these constraints (i.e., error distribution, smoothness, and orthogonality). This can be achieved by minimizing the sum of the integrals

$$I_T = I_w + \lambda_s I_s + \lambda_o I_o \tag{9}$$

where λ_s and λ_o are weighting coefficients, and

$$I_w = \int_{\text{vol}} w(x, y)\, J\, d\text{vol} \quad \text{or} \quad \int_{\text{vol}} w(\xi, \eta)\, J^2\, d\text{vol} \tag{10}$$

The resulting Euler-Lagrange equations are given by

$$b_1 x_{\xi\xi} + b_2 x_{\xi\eta} + b_3 x_{\eta\eta} + a_1 y_{\xi\xi} + a_2 y_{\xi\eta} + a_3 y_{\eta\eta} = \frac{-J^2}{2w} w_x \tag{11a}$$

$$a_1 x_{\xi\xi} + a_2 x_{\xi\eta} + a_3 x_{\eta\eta} + c_1 y_{\xi\xi} + c_2 y_{\xi\eta} + c_3 y_{\eta\eta} = \frac{-J^2}{2w} w_y \tag{11b}$$

where the a_i, b_i, c_i are functions of the metric coefficients and the λ_s and λ_o. These equations are elliptic and to obtain their solution is as formidable a task as obtaining the solution to the flowfield equations themselves.

It is the purpose of the remainder of this paper to show how the driving mechanism for grid adaptation can be represented by a system of springs whose constants are functions of the solution itself, and how the smoothness and orthogonality constraints can be imposed in a straightforward manner without the multidimensional ellipticity implied by Eq. (11).

Self-Adaptive-Grid Scheme

When the weighting function $w(x)$ in Eq. (1) is held fixed over the interval between x_i and x_{i+1}, the equation can be rewritten as

$$w_i \Delta x_i = \text{const} \tag{12}$$

where Δx_i is the grid interval $(x_{i+1} - x_i)$.

The constant values used to define the weighting function [e.g., Eqs. (5) and (6)] are critical in determining the degree of grid clustering that is achieved. Heretofore these parameters have been determined empirically, in a time-consuming trial-and-error manner; moreover, that procedure is further complicated for two- and three-dimensional grids in which other constants used to control grid skewness appear. In the present work, the constants are related to the grid spacing directly,

affording the user precise control of the grid adaptation, removing the empiricism and uncertainty, and making the scheme "self-adaptive." In this approach, it is necessary only to specify the desired maximum grid spacing Δx_{MAX} and minimum spacing Δx_{MIN}.

The form used here for the weighting function is given by

$$w_i = 1 + A\bar{f}^B, \qquad \bar{f} = (f_i - f_{MIN})/(f_{MAX} - f_{MIN}) \quad (13)$$

which is normalized so that the first constant is unity. When the constant A is zero, the grid points will be equally spaced. The non-negative driving function f_i represents the flow solution, A and B are positive constants, and f_{MIN} and f_{MAX} are the minimum and maximum values of f_i. The constant A is determined by the ratio of the specified maximum and minimum grid spacings. For an equidistribution scheme, the grid spacing is determined by Eq. (4), rewritten here as

$$\Delta x_i = L \Big/ \left(w_i \sum_{j=1}^{N} \frac{1}{w_j} \right) \quad (14)$$

where L is the length of the line in physical space. The ratio of maximum and minimum grid spacings $\mathrm{Max}[\Delta x_i]/\mathrm{Min}[\Delta x_i]$ is equal to the ratio of weighting functions $\mathrm{Max}[w_i]/\mathrm{Min}[w_i]$. Thus, the constant A in Eq. (13) is given by Δx_{MAX} and Δx_{MIN} as

$$A = (\Delta x_{MAX}/\Delta x_{MIN}) - 1 \quad (15)$$

The constant B in Eq. (13) is chosen so that the minimum spacing $\mathrm{Min}[\Delta x_i]$ is equal to the specified Δx_{MIN} and is determined by the following iterative equation derived from Eqs. (13) and (14):

$$B^{(n+1)} = B^{(n)} + \Delta B^{(n)} \quad (16)$$

where

$$\Delta B^{(n)} = \left[\Delta x_{MIN} - \mathrm{Min}(\Delta x_i)^{(n)} \right] \Big/ \left[\frac{\partial \mathrm{Min}(\Delta x_i)}{\partial B} \right] \quad (17)$$

$$\frac{\partial \mathrm{Min}(\Delta x_i)}{\partial B} = -A \sum_{j=1}^{N} \left[\Delta x_i \left(\delta_{ij} \frac{1}{w_i} - \frac{w_i}{w_j^2} \frac{\Delta x_i}{L} \right) \bar{f}_j^B \log \bar{f}_j \right] \quad (18)$$

In Eq. (18), δ_{ij} is the Kronecker delta and is equal to 1 for $i = j$ and 0 for $i \neq j$.

Shown in Fig. 1 is an example of the one-dimensional adaptive grid determined by specifying maximum and minimum grid spacings. The function f in Eq. (13) is chosen as the absolute value of the gradient of the solution Q, which is defined by the curve in the figure. For this figure, the minimum spacing is set to half of the uniform spacing, and the maximum spacings are varied parametrically from two to four times the uniform spacing. As shown here, the resulting grid distributions are precisely controlled by the specified maximum and minimum spacings.

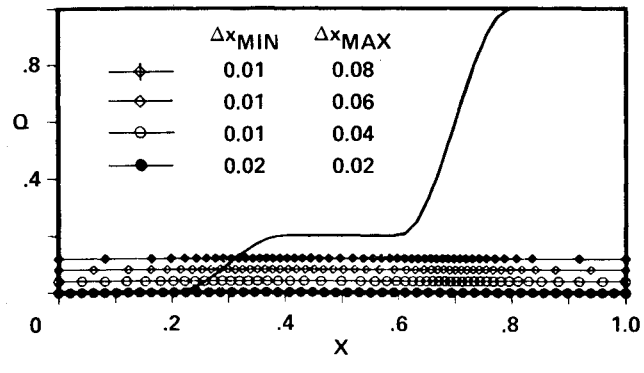


Fig. 1 One-dimensional self-adaptive grid.

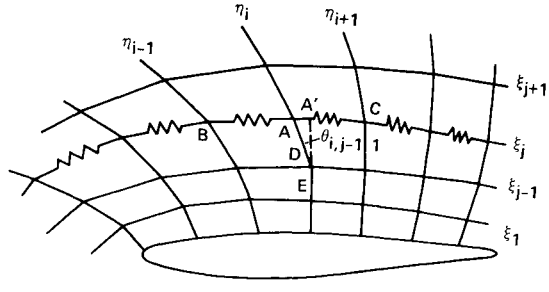
Two-Dimensional Adaptation

To extend the self-adaptive concept to two dimensions, we follow the procedure described in Refs. 2 and 3. In that method, two-dimensional adaptation is treated as a sequence of the one-directional operations.

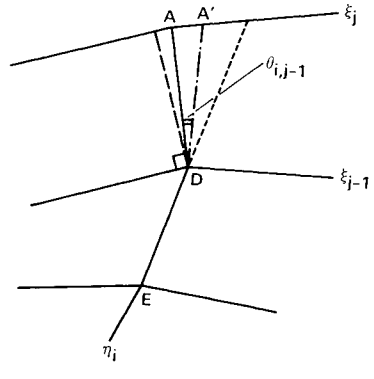
For simplicity of illustration, consider a two-dimensional airfoil flowfield with adaptation along the ξ coordinate (body-oriented) lines. To adapt the grid to the flowfield solution in the ξ direction, the procedure is started at the ξ coordinate line corresponding to the airfoil surface. The adaptation along this line can be treated one-dimensionally. By specifying the desired maximum and minimum grid spacings Δs_{MAX} and Δs_{MIN}, the grid point distribution along the surface is determined by the equidistribution scheme along a line as described previously. Here, Δs_{MAX} and Δs_{MIN} are specified so that the grid can be clustered near shock waves and near the leading edge without resulting in an extremely coarse mesh. For practical convenience, Δs_{MAX} and Δs_{MIN} are given as multiples of the uniform spacing value L/N.

From the ξ_2 to the ξ_{N+1} coordinate line, grid points are free to move along a ξ_j-line whose configuration is fixed. Grid point A on the ξ_j coordinate in Fig. 2 is suspended by two tension springs which connect point A to points B and C, whose spring constants are $w_{i-1,j}$ and $w_{i,j}$. The spring constants are determined by Eq. (13), written for two dimensions as

$$w_{i,j} = 1 + A \left(\frac{f_{i,j} - f_{MIN}}{f_{MAX} - f_{MIN}} \right)^B \quad (19)$$



a) Adaptive-grid algorithm with spring analogy.



b) Reference lines for torsion spring.

Fig. 2 Schematic of adaptive-grid algorithm.

and the grid spacing is determined by Eq. (14), written for two dimensions as

$$\Delta s_{i,j} = L \Big/ \left(w_{i,j} \sum_{k=1}^{N} \frac{1}{w_{i,k}} \right) \tag{20}$$

where $s_{i,j}$ is the arc length calculated from point $(1, j)$ along the ξ_j coordinate. In actual practice, the $s_{i,j}$ is approximated by arc secants for simplicity. The same values of the constants A and B determined at the surface line adaptation are used for Eq. (19).

A force to control inclinations of the η_i coordinate is given by a torsion spring attached at point D in Fig. 2a. The torsion spring enforces the inclinations of line DA with respect to a prescribed reference line DA′. This spring affects the orthogonality and smoothness constraints described by Eqs. (7) and (8), but does so in a one-sided manner so as not to introduce ellipticity into the equations in the η coordinate direction. It is this feature that contributes significantly to the robustness and practicality of the method.

If the coefficient for the torsion spring is denoted by C, a mathematical statement of the force is given by

$$F_{\text{torsion}} = -C\theta_{i,j-1} \tag{21}$$

where $\theta_{i,j-1}$ is the angle between line DA and DA′. The reference line DA′ (Fig. 2b) is prescribed by an average of 1) an extension of ED (dotted line), to maintain smoothness along the η-line, and 2) a line normal to the ξ_{j-1} coordinate (dashed line), to make the grid quasiorthogonal. Other reference lines, such as streamlines or shocks, could be used as well.

The distribution of grid points along the ξ_j coordinate, namely, new values of $s_{i,j}$, is determined by a balance equation for the complete spring system

$$w_{i,j}(s_{i+1,j} - s_{i,j}) - w_{i-1,j}(s_{i,j} - s_{i-1,j}) - C\theta_{i,j-1} = 0 \tag{22}$$

In the previous use of this scheme, C was held fixed for the entire flowfield. In keeping with the self-adaptive concept, C is now specified along each line. Here it is set proportional to the average of the tension-spring constants along a line and is determined automatically

$$C = \lambda \left(\frac{1}{N} \sum_{k=1}^{N} w_{i,k} \right) \tag{23}$$

where λ is a user-specified constant.

The adapted grid is less sensitive to a uniform constant value λ than it is to a uniform constant C. A value of $\lambda = 0.005$ was used for all computations in the present study.

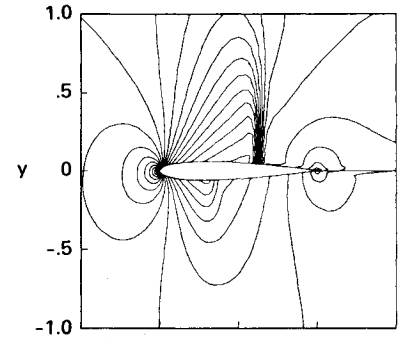
To facilitate solutions to Eq. (22), the third term is rewritten as

$$C\theta_{i,j-1} \mapsto T_{i,j-1}(s_{i,j} - s'_{i,j}) \tag{24}$$

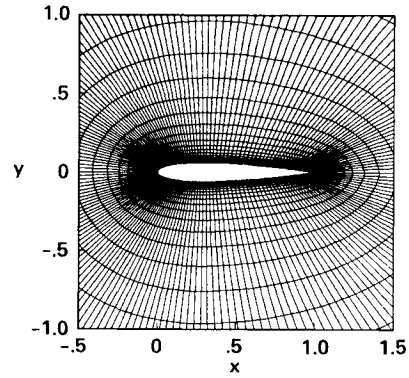
where $s'_{i,j}$ is the arc length to the intersection of reference line DA′ along the ξ_j coordinate, as depicted in Fig. 2. The torsion-spring constant $T_{i,j-1}$ is set equal to C divided by the length of DA′. Finally, Eq. (22) reduces to

$$w_{i,j}(s_{i+1,j} - s_{i,j}) - w_{i-1,j}(s_{i,j} - s_{i-1,j}) - T_{i,j-1}(s_{i,j} - s'_{i,j}) = 0 \tag{25}$$

This is a tridiagonal system of equations for $s_{i,j}$ along the ξ_j coordinate, and can be readily solved.

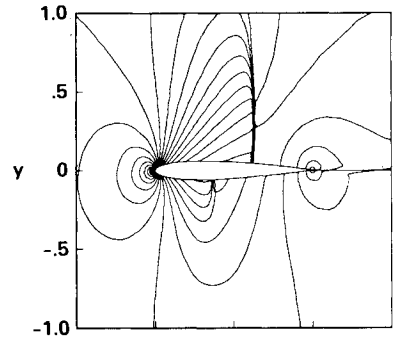


a) Initial grid (193 × 33).

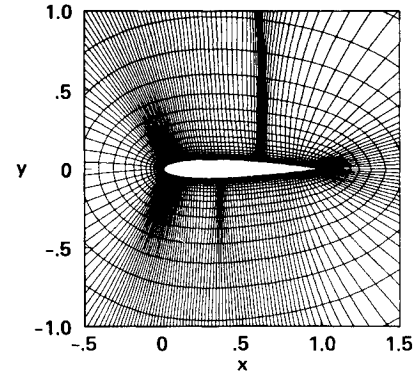


b) Computed Mach contours with initial grid.

Fig. 3 Grid and Mach contours of inviscid airfoil computation without adaptive grid: NACA-0012, $M_\infty = 0.8$, $\alpha = 1.25$ deg.



a) Adapted grid.



b) Computed Mach contours with adapted grid.

Fig. 4 Grid and Mach contours of inviscid airfoil computation with adaptive grid: NACA-0012, $M_\infty = 0.8$, $\alpha = 1.25$ deg.

Since the torsion force constraint is now considered along with the equidistribution constraint, it is not possible to precisely enforce equidistribution. Hence, the grid spacings are checked along each ξ coordinate line, and extreme clustering or coarseness is adjusted by modifying the weighting function locally. In this readjustment of $w_{i,j}$, the equidistribution concept of Eq. (12) can be applied locally.

In this analysis, only one-sided torsion forces ($T_{i,j-1}$) influence the distribution at the $\xi_{i,j}$ coordinate. This permits simple marching schemes to be used in the j direction without any loss of generality, and contributes to the simplicity and robustness of the method. Conversely, if the influence from both sides ($T_{i,j-1}$ and $T_{i,j+1}$) is considered simultaneously, Eq. (25) is no longer a tridiagonal system of equations, and the computational effort is increased considerably without any additional benefit.

After the grid has been adapted along the ξ coordinate, either a grid speed can be approximated for use in a time-accurate scheme, or the solution can be interpolated one-dimensionally along each ξ coordinate to be used as new initial boundary values. To adapt the grid in two dimensions, a similar procedure as described for ξ coordinate adapting can now be applied to η coordinate adapting. This split-operator procedure does not diminish the generality of the method; rather it enhances its flexibility and efficiency and eliminates multidimensional ellipticity.

Examples

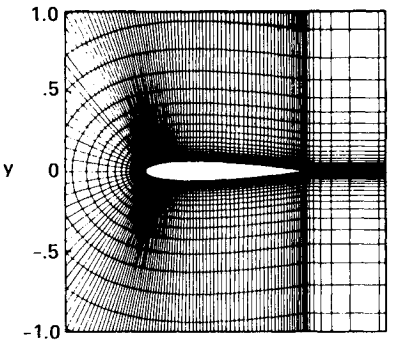
Steady Inviscid Flow

To demonstrate the self-adaptive mesh in a two-dimensional field, transonic flow about a NACA-0012 airfoil is computed using the ARC2D code[7] for the Euler equations. The grid adaptation is performed only in the ξ direction and is used to define the shock waves in the transonic regime. The density gradient is used as the reference flow property f for adaption along each ξ coordinate line. Minimum and maximum grid spacings are specified, respectively, as 0.2 and 2.0 times a uniform spacing, and the resolution of the shock wave after adaption is expected to be five times finer than the result with a uniformly spaced grid.

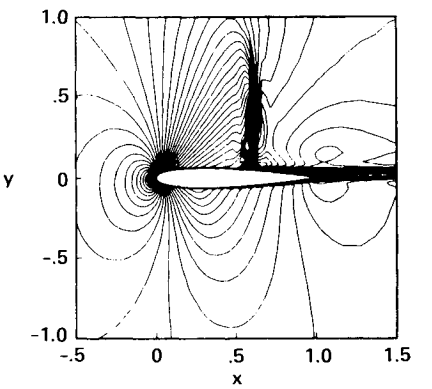
Shown in Fig. 3 are the initial grid and computed Mach contours for a Mach 0.8 flow obtained without an adapted grid; Fig. 4 shows the resulting self-adapted grid and recomputed Mach contours. The adapted-grid points are appropriately clustered to the shock and are smoothly distributed. The resolution of the upper-surface shock is much more precise than the smeared shock obtained using the unadapted grid. The lower-surface shock, which is difficult to resolve with a rather course grid, is also clearly depicted. The quality of the computed results using the adapted grid show significant improvement. The present method works equally well when the flow conditions are changed or when the total number of grid points is changed, without necessitating the respecification of new adaption-control parameters.

Steady Viscous Flow

In recent years, there has been considerable improvement in our ability to simulate interactive flows about airfoils; however, questions still remain concerning the dependency of solution accuracy on the grid. These concerns are particularly important in the transonic-flow regime where shock waves significantly influence the surface-pressure distributions and can induce boundary-layer separation. Physically, the surface-pressure distribution observed beneath the foot of the shock depends on the strength of the shock wave and on the thickness of the boundary layer. In addition, in numerical simulations the shock waves are "smeared" over several grid points and the grid distribution itself can affect the pressure distribution observed at the foot of the shock. Furthermore, if the boundary layer does in fact separate, a shear layer will leave the surface of the airfoil and may not be adequately

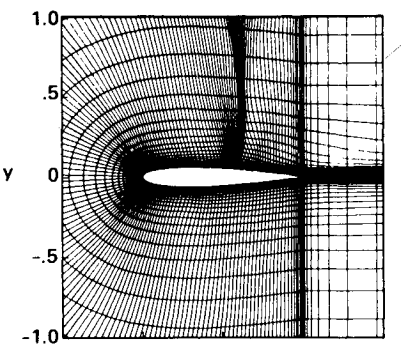


a) Initial grid (193 × 49).

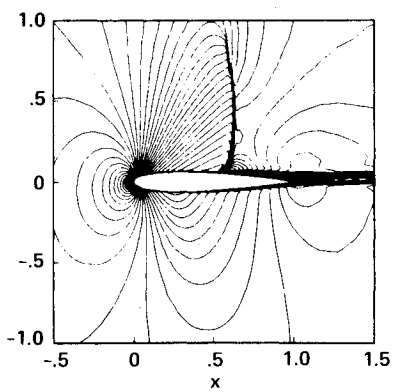


b) Computed Mach contours with initial grid.

Fig. 5 Grid and Mach contours of viscous airfoil computation without adaptive grid: NACA-0012, $M_\infty = 0.8$, $\alpha = 2$ deg, $Re = 1 \times 10^6$.



a) Adapted grid.



b) Computed Mach contours with adapted grid.

Fig. 6 Grid and Mach contours of viscous airfoil computation with adapted grid: NACA-0012, $M_\infty = 0.8$, $\alpha = 2$ deg, $Re = 1 \times 10^6$.

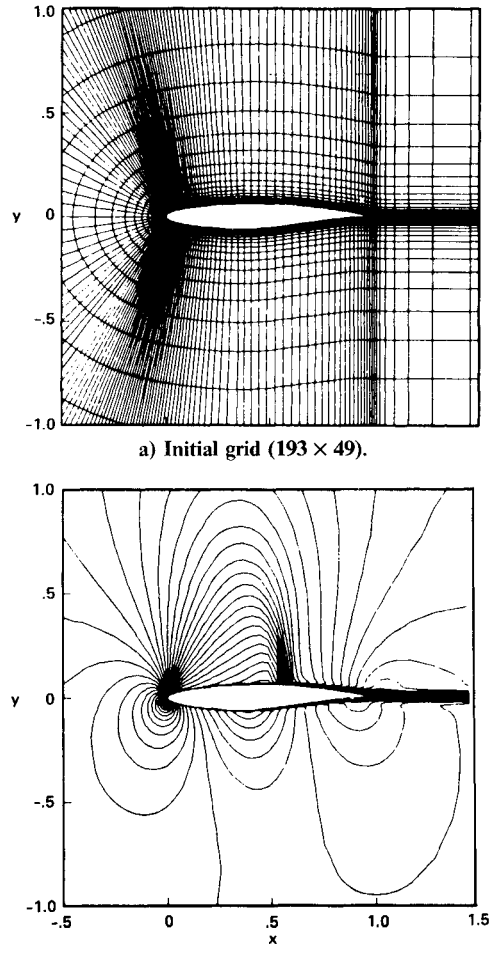
resolved by commonly used geometrically stretched grids. One way to eliminate questions about the dependency of the accuracy of the solution on grid resolution is to use many grid points in the vicinity of the interaction or, better yet, to adapt the grid to the solution and optimally distribute the available nodal points. It is this latter course that will be considered here.

We use the self-adaptive-grid method in simulating transonic interactive flowfields over two different airfoil sections: a NACA-0012 symmetrical section and an RAE2822 supercritical section. Grid points are adapted in both coordinate directions so as to resolve shock waves, boundary layers, and separated shear layers. The streamwise density gradient is used for the adapting parameter in the ξ direction, and the transverse gradient of the streamwise momentum ρu is used in the η direction. The flowfield is determined by solving a shock-capturing version of the thin-layer approximation of the time-dependent, compressible, Reynolds-averaged Navier-Stokes equations.[7]

Shown in Figs. 5 and 6 are results for the NACA-0012 airfoil where a C-grid (193×49) is used to discretize the flowfield. The freestream Mach number is 0.8, the angle of attack is 2.0 deg, and the chord Reynolds number is 1×10^6. Shown in Fig. 5 is the initial, unadapted grid and the resulting computed Mach contours. Because of relatively coarse spacing near the shock wave, the computed results show a highly smeared shock, with some apparent "noisiness." Shown in Fig. 6 is the solution-adapted grid and corresponding recomputed Mach contours. The improved quality of the solution in terms of shock definition and smoothness in Fig. 6b compared to that in Fig. 5b is obvious.

Computations for the supercritical RAE2822 sections are shown in Figs. 7–9. Shown in Fig. 7a is the initial grid (193×49, with 165 points on the airfoil surface). The computed Mach contours and pressure coefficients using this grid are shown in Fig. 7b for a freestream Mach number of 0.73, an angle of attack of 2.79 deg, and a chord Reynolds number of 6.5×10^6. The relatively coarse grid over the mid-portion of the section results in a poorly resolved shock and a correspondingly smeared surface-pressure distribution.

The same flowfield over the RAE2822 section has been recomputed using adaptive techniques and one-half the number of grid points over the surface that was used in the previous example (i.e., 83 points instead of 165). Shown in Fig. 8a is the solution-adapted grid and in Fig. 8b the resulting computed Mach contours. The quality of this solution is superior to the unadapted solution using twice as many points (Fig. 7).



a) Initial grid (193×49).

b) Computed Mach contours with initial grid.

Fig. 7 Grid and Mach contours of viscous airfoil computation without adaptive grid: RAE2822, $M_\infty = 0.73$, $\alpha = 2.79$ deg, $Re = 6.5 \times 10^6$.

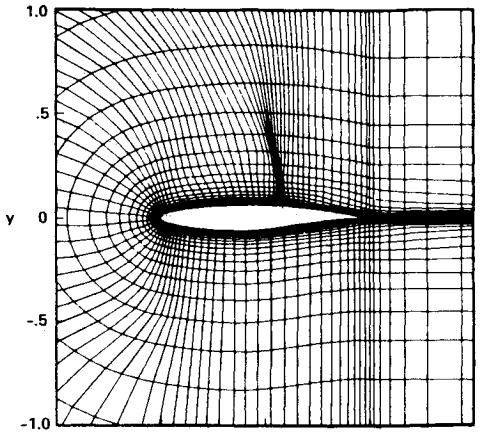
Unsteady Viscous Flow

An important area for adaptive-grid application are dynamic systems in which the grid points are moved with an unsteady flowfield. There are several key considerations for developing the dynamically adaptive grid. First, the points must concentrate without excessive grid-line distortions, but no region can be allowed to become devoid of points; too much declustering of grid points may change the unsteady flow features. The self-adaptive concept described above, which can maintain the maximum and minimum grid spacings, satisfies this requirement.

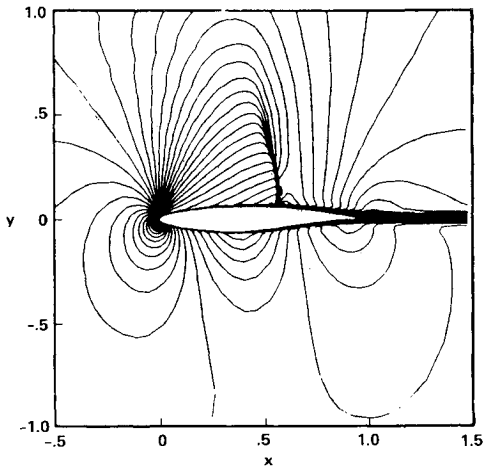
Second, the dynamically adaptive-grid method must be efficient. Although an adaptive mesh reduces the total number of grid points required, too much computational time can make the method impractical. In the present method, directional splitting and one-sided orthogonality and smoothness constraints remove multidimensional ellipticity and permit the use of marching techniques. The efficiency of the method is further enhanced by subdividing the flowfield into adapted and unadapted regions. Careful programming and the use of vectorization can further minimize the computational time for grid adaptation.

To illustrate a dynamic adaptive-grid application, the transonic flow over an 18% thick, biconvex, circular-arc airfoil at zero angle of attack is numerically simulated. A series of experiments[9] for this configuration indicate the existence of a self-excited unsteady flow in the Mach number range from 0.76–0.78. Here the flow varies alternately over the upper and lower surface, 180 deg out of phase, between fully attached flow and a flow with shock-induced separation. Numerical simulations by Levy,[10] using an explicit algorithm, and by Steger,[11] using an implicit algorithm, reproduce these findings, with Steger's results suggesting a slightly higher Mach number range for the unsteady regime. Using the method of Steger, the flowfield is computed here for a freestream Mach number of 0.783 and a chord Reynolds number of 11×10^6. The self-excited reduced frequency for this case is $\Omega = 0.5$.

Two selected sequences for one period of oscillation are shown in Fig. 10. Included are adapted grids and computed Mach contours. As in the steady-flow examples, the density gradient was used for streamwise adapting, which was done only over the aft portion of airfoil ($0.4 < x/c < 1.0$), and the gradient of the streamwise momentum component was used in the transverse adaption to capture the shear layers and moving wake. Here, the C-grid cut line extending downstream of the trailing edge was held fixed during the course of the solution. These results indicate excellent shock resolution and shear-layer definition using the dynamically adapted grid. The computation time for grid adaptation in this study was a third less than the time required for one time step of the implicit Navier-Stokes code. The present adaptive-grid program is not



a) Adapted grid (111 × 49).



b) Computed Mach contours with adapted grid.

Fig. 8 Grid and Mach contours of viscous airfoil computation with adaptive grid: RAE2822, $M_\infty = 0.73$, $\alpha = 2.79$ deg, $Re = 6.5 \times 10^6$.

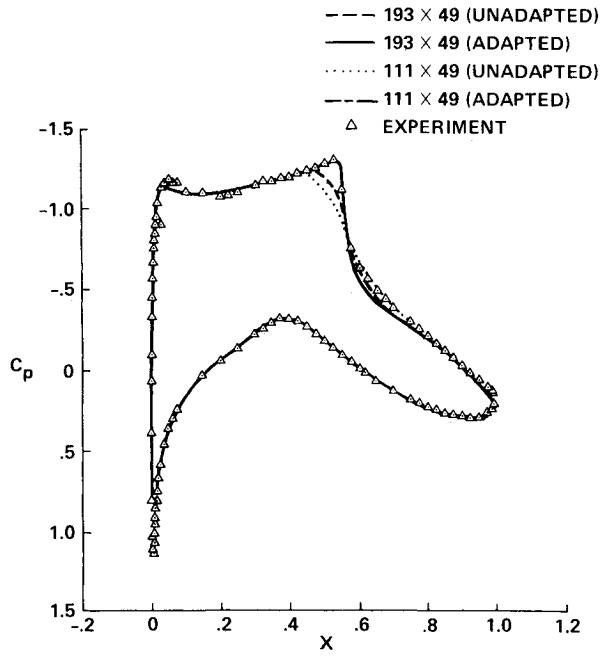
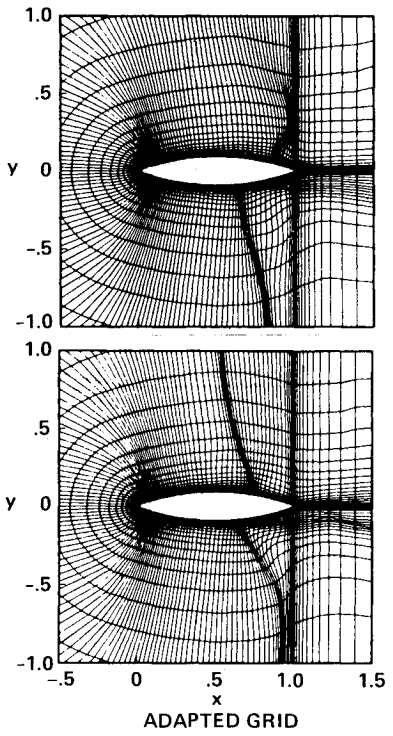


Fig. 9 Pressure coefficient: RAE2822, $M_\infty = 0.73$, $\alpha = 2.79$ deg, $Re = 6.5 \times 10^6$.



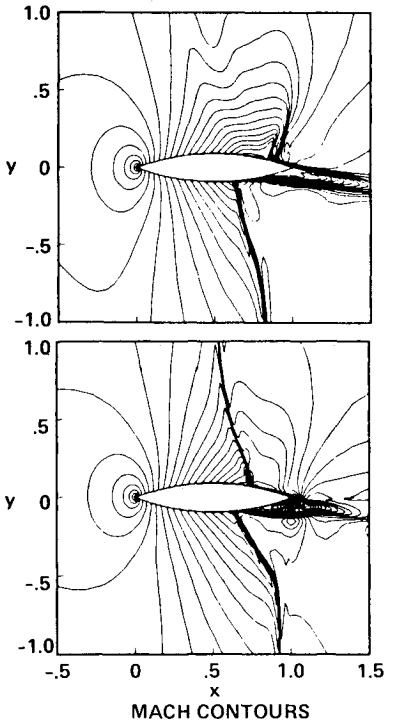


Fig. 10 Selected sequences for one period of oscillation of unsteady flow about an 18% biconvex wing with dynamic adaptive grid: $M_\infty = 0.783$, $\alpha = 0.0$ deg, $Re = 11 \times 10^6$.

vectorized, and it should to be possible to reduce the computational time by at least 50% by careful reprogramming.

Summary

A two-dimensional solution-adaptive-grid method based on variational principles and spring analogies was described. Directional splitting and one-sided orthogonality and smoothness constraints were used to produce a method that is simple, efficient, and robust. User-specified control parameters denoting maximum and minimum permissible grid spacings were used to define important constants and thus make the method self-adaptive.

Computed examples for two-dimensional transonic flowfields past airfoil sections were used to demonstrate the applicability of the method to both steady and unsteady interactive flows. It is not possible to provide precise comparisons of computation times and convergence rates for adapted vs nonadapted cases because of a strong dependency on the quality of the initial grid. However, it has always been observed that both the accuracy and the rate of convergence improve by using solution-adapted grids in this study. Sometimes this improvement is dramatic. Because of the error minimization variational concept employed, this improvement in accuracy and convergence is to be expected.

References

[1]Gnoffo, P.A., "A Finite-Volume, Adaptive Grid Algorithm Applied to Planetary Entry Flowfields," *AIAA Journal*, Vol. 21, Sept. 1983, pp. 1249–1254.

[2]Nakahashi, K. and Deiwert, G.S., "A Practical Adaptive-Grid Method for Complex Fluid-Flow Problems," *Lecture Notes in Physics*, Vol. 218, Springer-Verlag, 1984, pp. 422–426; also NASA TM-85989, 1984.

[3]Nakahashi, K. and Deiwert, G.S., "A Three-Dimensional Adaptive Grid Method," AIAA Paper 85-0486, Jan. 1985.

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

[5]Thompson, J.F., "A Survey of Dynamically-Adaptive Grids in the Numerical Solution of Partial Differential Equations," AIAA Paper 84-1606, June 1984.

[6]White, A.B. Jr., "On Selection of Equidistributing Meshes for Two-Point Boundary-Value Problems," *SIAM Journal of Numerical Analysis*, Vol. 16, No. 3, 1979, pp. 472–502.

[7]Pulliam, T.H. and Steger, J.L., "Recent Improvements in Efficiency, Accuracy, and Convergence for Implicit Approximate Factorization Algorithms," AIAA Paper 85-0360, Jan. 1985.

[8]Cook, P.H., McDonald, M.A., and Firmin, M.C.P., "Aerofoil RAE2822—Pressure Distributions, and Boundary Layer and Wake Measurements," AGARD-AR-138, 1979.

[9]McDevitt, J.B., Levy, L.L. Jr., and Deiwert, G.S., "Transonic Flow about a Thick Circular-Arc Airfoil," *AIAA Journal*, Vol. 14, May 1976, pp. 606–613.

[10]Levy, L.L. Jr., "Experimental and Computational Steady and Unsteady Transonic Flows about a Thick Airfoil," *AIAA Journal*, Vol. 16, June 1978, pp. 564–572.

[11]Steger, J.L., "Implicit Finite-Difference Simulation of Flow about Arbitrary Two-Dimensional Geometries," *AIAA Journal*, Vol. 16, July 1978, pp. 679–686.