NASA Technical Memorandum 101027

Three-Dimensional Self-Adaptive Grid Method for Complex Flows

M. Jahed Djomehri and George S. Deiwert

November 1988

National Aeronautics and
Space Administration

NASA Technical Memorandum 101027

Three-Dimensional Self-Adaptive Grid Method for Complex Flows

M. Jahed Djomehri
George S. Deiwert, Ames Research Center, Moffett Field, California

November 1988

NASA
National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

THREE-DIMENSIONAL SELF-ADAPTIVE GRID METHOD FOR COMPLEX FLOWS

M. Jahed Djomehri* and George S. Deiwert†

Ames Research Center

1. ABSTRACT

A self-adaptive grid procedure for efficient computation of three-dimensional complex flow fields is described. The method is based on variational principles to minimize the energy of a spring system analogy which redistributes the grid points. Grid control parameters are determined by specifying maximum and minimum grid spacings. Multidirectional adaptation is achieved by splitting the procedure into a sequence of successive applications of a unidirectional adaptation. One-sided, two-directional constraints for orthogonality and smoothness are used to enhance the efficiency of the method. Feasibility of the scheme is demonstrated by application to a multinozzle, afterbody, plume flow field. Application of the algorithm for initial grid generation is illustrated by constructing a three-dimensional grid about a "bump-like" geometry.

2. INTRODUCTION

The effective design of future aircraft, rockets, and aerospace vehicles which fly at high speeds requires realistic consideration of three-dimensional (3-D) complex flow fields. Analysis of these problems by means of simulation techniques demands enhanced grid distribution methodologies to obtain efficient and accurate resolution of rapidly varying solutions to the pertinent partial differential equations (PDE).

Studies of solution-adaptive grid methods are significant because of their potential for improving the efficiency and accuracy of the numerical methods and for reducing computer memory requirements. Continuing advances in solution algorithms and computer power have considerably improved the efficiency of the solution, but the accuracy of the solution still depends directly on the distribution of grid points over the physical space in question.

In recent years, a considerable amount of literature has been devoted to the subject of adaptive methods. General surveys by Anderson (ref. 1) in 1983, by Babuska et al. (ref. 2) in 1983, by Thompson (ref. 3) in 1985, and by Eiseman (ref. 4) in 1987 summarizes much of this activity. One of the important tools which has been used to cluster the grid points in the physical space and which has been reviewed in detail, in references 1 and 3, is to redistribute the grid points along a fixed line so that some positive weight function is equidistributed over the line. Similar adaptation along a family of fixed lines can be applied to multidimensional adaptation (ref. 5). Without additional constraints these methods can produce undesirable results in terms of grid skewness. By introducing an orthogonality constraint, Anderson (ref. 6) and Dwyer (ref. 7) have been able to control this problem to some extent in two-dimensional (2-D) applications; Eiseman (ref. 8) and Nakahashi and Deiwert (refs. 9,10) have introduced

*National Research Council Research Associate, Member AIAA.

†Chief, Aerothermodynamic Branch, Associate Fellow.

a practical multidimensional adaptation using successive applications of a one-dimensional (1-D) adaptation separately in each of the curvilinear coordinate directions. In the latter a torsion-tension spring system analogy is used incorporating resistance to movement away from orthogonality. The orthogonality and smoothness of the above spring system analogy have been controlled by several incorporated parameters and constants whose values are determined by the effort of the user through numerical experiment. To diminish the degree of empiricism intrinsic to this approach, particularly in 3-D, Nakahashi and Deiwert (ref. 11) have suggested some new measures to relate the value of the parameters directly to the solution data. This new improvement has significantly increased the accuracy and efficiency in the study of 2-D airfoil problems.

This paper presents a practical 3-D solution-adaptive algorithm for the redistribution of grid points to optimal positions and is suitable for modern computer architectures. This work extends the 2-D self-adaptive approach of reference 11 to 3-D. The control of grid skewness is achieved by one-sided torsion and control of spacing by two-sided tension springs. The procedure can be used independently of the PDE-algorithm and/or can be coupled with a variety of existing PDE-solvers, both for steady and unsteady cases. It is suitable for use with zonal approach. The multidirectional adaptation is achieved using the concept of splitting (or fractional steps) in all aspects of the approach.

A user friendly 3-D code which implements this algorithm has been developed. The code automatically calculates most of the incorporated parameters, which affects orthogonality and smoothness via user-specified constants denoting the maximum and minimum grid spacing. The user selects direction of adaptation along a curvilinear grid line in the physical space. Grid points are then redistributed to their optimal positions along a family of these lines whose collection sweeps a user-specified curvilinear grid plane. The procedure covers the entire physical space along each of these planes in a given marching direction. The feasibility of the method as a solution-adaptive scheme is illustrated by its application to a realistic three-dimensional problem concerning a generic two-nozzle afterbody plume flow. The outer flow field is complicated by the three-dimensional body shape interacting with multiple jet streams, three-dimensional barrel shocks, Mach disc, mixing shear layers and recompression shocks. Application of the scheme as an initial grid generation is illustrated by generating a 3-D grid about a bump.

3. METHODOLOGY

Here we briefly review the conceptual approach, based on variational principles that are the main ingredient of the adaptive grid scheme. Then a more relaxed version, although not in content, of some of the mathematical constraints are considered to produce an efficient and practical solution-adaptive algorithm.

The problem of grid genertion and adaptation is to establish the mapping relation $x(\xi)$ or $\xi(x)$ between the physical and computational space. An optimum distribution is one in which the solution error or some approximate measure of this error, here denoted by a positive weight function $w(x)$, is uniformly distributed over all grid points. The solution data should be adjusted on the new grid either by interpolation or incorporation of the grid speeds in conjunction with the transformation of the time derivatives. This concept is discussed in reference 1 in detail; here we will use the following relations taken from this reference for our immediate purpose:

$$\int_{x_i}^{x_{i+1}} w(x)dx = \text{constant}, \qquad \text{discrete case}; \qquad \Delta x_i w_i = \text{constant} \tag{1}$$

$$x_\xi w(\xi) = \text{constant}, \qquad I_1 = w(\xi)x_\xi^2 \, d\xi \tag{2}$$

The initial part of equation (2) is the Euler-Lagrange equation for minimization of the second integral there. It can also be interpreted as minimization of the energy of a system of springs with constant $w(\xi)$ between each pair. We also have

$$\Delta w_i = \left(w(\xi_i) N \int_0^1 \frac{d\xi}{w(\xi)} \right)^{-1} \tag{3}$$

where N is the total number of grid intervals.

The weighting function $w(x)$ is most commonly experssed as a linear combination of positive functions, M_i's, related to the first or higher gradient of the solution,

$$w(x) = M_0 + c_1 M_1 + c_2 M_2 + \ldots + c_m M_m\,, \quad c_i \geq 0 \tag{4}$$

See also suggestions by White (ref. 12).

In multiple dimensions, the adaptation in different directions should be coupled to maintain sufficient smoothness in the grid. Brackbill and Saltzman (refs. 13-15) in their variational approach minimize a linear combination of integrals where each is a measure of some grid properties; their formulation is given by:

$$I = \lambda_w I_w + \lambda_s I_s + \lambda_o I_o \tag{5}$$

$$I_w \equiv \int w(\underline{x}) J \, d\underline{x} = \int w(\xi) J^2 \, d\xi \tag{6}$$

$$I_s \equiv \int \left(\sum_{i=1}^{3} J_{ii} \right) d\underline{x} = \int \left(\sum_{i=1}^{3} J_{ii} \right) J \, d\underline{\xi} \tag{7}$$

$$I_0 \equiv \int [(J_{12})^2 + (J_{23})^2 + (J_{31})^2] d\underline{x} = \int [(J_{12})^2 + (J_{23})^2 + (J_{31})^2] J^2 \, d\underline{\xi} \tag{8}$$

$$J_{ij} = \underline{\nabla}\xi^i \cdot \nabla\xi^j\,, \quad J = (\det \mid J_{ij} \mid)^{1/2} \tag{9}$$

J is the Jacobian of the transformation, λ_w, λ_s, and λ_o are specified coefficients, $d\underline{x}$ and $d\underline{\xi}$ are differential volumes in physical and computational space, respectively. Larger values of these coefficients put more emphasis upon their corresponding weighted volume integrals. The I_w forces the cell size to be small where $w(\underline{x})$ is large and vice versa; I_s reinforces the smoothness of the grid; and finally, I_o constrains the orthogonal behavior of the grid.

Minimization of I yields the Euler-Lagrange equations and can be obtained by application of the following operators to the integrand of I:

$$E^i = \frac{\partial}{\partial \xi^i} - \frac{\partial}{\partial x_j}\left(\partial/\partial\left(\frac{\partial \xi^i}{\partial x_j}\right)\right), \quad i = 1, 2, 3 \qquad (10)$$

Equation (10), which establishes the mapping relation, would now be a system of three coupled elliptic PDEs whose solutions would be responsible for generating smooth, clustered or stretched, and orthogonal curvilinear coordinates in the physical space. The solution of the previous equations in practice is in itself a rather complicated task. Here we will show how a spring system analogy whose constants are functions of the solution itself can be used to relax multidimensional elliptic equation (10). The procedure illustrated is an extension of the self-adaptive method of Nakahashi and Deiwert (ref. 9) to 3-D problems.

4. UNIDIRECTIONAL SELF-ADAPTIVE SCHEME

In this section we will derive the basic equations for grid adaptation. For simplicity of illustration, consider a 3-D flow field with adaptation along the curvilinear coordinate ξ. Also for clarity of our notation assume that indices i, j, and k correspond to the ξ, η, and ζ coordinates in the computational space and let ξ_i denote the curvilinear plane ξ_i = constant, which is a transformation of the $\xi = i$ of the computational plane. The $\xi_{j,k}$ denotes the curvilinear line which is the intersection of the η_j and ξ_k-planes. Finally, let the point $p(i,j,k)$ of computational space correspond to the point $(x_{i,j,k}, y_{i,j,k}, z_{i,j,k})$ of the physical space and let $f_{i,j,k}$ be the value of a function f at p.

In figure 1, grid points on the $\xi_{j,k}$ line are free to move along a line whose configuration is fixed. Each point A on $\xi_{j,k}$ is suspended by two tension springs which connect A to points B and C, and whose spring constants (weighting functions) are $w_{i-1,j,k}$ and $w_{i,j,k}$ and w is

$$w_{i,j,k} = M_0 + rM_i^b, \quad M_i = \frac{f_{i,j,k} - f_{MIN}}{f_{MAX} - f_{MIN}} \qquad (11)$$

where f_{MIN} and f_{MAX} are the minimum and maximum of nonnegative driving function f with its discrete values denoted by $f_{i,j,k}$; r and b are positive constants; and M_0 is used to maintain grid smoothness and is equal to unity in most cases. The function $f \geq 0$ is related to the gradient of pressure, density, temperature, Mach number or other vital solution variables and/or a linear combination of these quantities. This choice of f is normally used to define the stretching and clustering of grid points, for other purposes, such as distributions across wall-bounded shear layers a geometric (exponential) function and around curved surfaces, the local curvature would be used.

The constants r and b in equation (11) are critical in obtaining the degree of grid clustering along the grid lines. Determining these constants by means of experiment, particularly in multidimensional application, is a tedious task. Here an automatic method of calculating them is considered. This would be one area of computation which makes the scheme "self-adaptive." In this approach (see also ref. 11) a desired maximum and minimum grid spacing Δs_{MAX} and Δs_{MIN}, respectively, are specified by the user; then the grid spacing is determined by equation (3) and written for three dimensions as:

$$\Delta s_{i,j,k} = L\left(w_{i,j,k} \sum_{n=1}^{N} (w_{n,j,k})^{-1} \right)^{-1} \tag{12}$$

where $s_{i,j,k}$ is the approximated arc secant length and calculated from point (i,j,k) along the line $\xi_{j,k}$. From equations (11) and (12), Max M_i the maximum of M_i is equal to unity and Max $w_{i,j,k} = 1 + r$ for some $i = i^*$, $1 \le i^* \le N$ when $M_0 \equiv 1$. The ratio of Max $w_{i,j,k}$ to Min $w_{i,j,k}$, from equation (12), is proportional to the ratio of Max $\Delta s_{i,j,k}$ to Min $\Delta s_{i,j,k}$. These observations all suggest that r should be defined as

$$r = \frac{\Delta s_{MAX}}{\Delta s_{MIN}} - 1 \tag{13}$$

For practical convenience Δs_{MAX} and Δs_{MIN} are given as multiples of the uniform spacing value L/N.

The constant b is iteratively found such that equation (12) is satisfied; this means we find some $b = \bar{b}$ so that $\psi(\bar{b}) = \text{Min } \Delta s_{i,j,k}$ has the value $\psi(\bar{b}) = \Delta s_{MIN}$. This can be done by application of the Newton-Raphson iterative scheme; for the (v + 1)th iteration of b we have:

$$b^{(v+1)} = b^{(v)} + \Delta b\,, \quad \Delta b = (\Delta s_{MIN} - \psi(b)) / \left(\frac{\partial \psi}{\partial b} \right) \tag{14}$$

$$\frac{\partial \psi}{\partial b} = D\psi_1 + D\psi_2 \tag{15}$$

$$D\psi_1 = -r \frac{\text{Min } \Delta s_{i,j,k}}{\text{Max } \Delta w_{i,j,k}} M_i^b \log M_i\,, \quad \text{at} \quad i = i^* \tag{16}$$

$$D\psi_2 = r\left(\frac{\text{Max } w_{i,j,k}}{L} \right) (\text{Min } \Delta s_{i,j,k}) \sum_{n=1}^{N} \frac{M_n^b \log M_n}{(w_{n,j,k})^2}\,, \quad \text{at} \quad i = i^* \tag{17}$$

In equations (16) and (17) we have used the fact that Min $\Delta s_{i,j,k}$ = Max $w_{i,j,k}$, at $i = i^*$ from equation (12). Also from Max $M_i = 1$ and Max $w_{i,j,k} = 1 + r$ we have $D\psi_1 = 0$, and $D\psi_2$ is given by

$$D\psi_2 = \frac{r(1+r)}{L} (\text{Min } \Delta s_{i,j,k}) \sum_{n=1}^{N} \frac{M_n^b \log M_n}{(w_{i,j,k})^2}\,, \quad i = i^* \tag{18}$$

A force to control inclinations of the η and ζ coordinates is given by the torsion springs attached at points D and E in figure 1; these torsions enforce the inclinations of lines DA and EA with respect to prescribed reference lines DA_1 and EA_2. If the coefficients for torsion springs are denoted by C_1 and C_2 a mathematical statement of the force is given by

$$F_{torsion} = -C_1 \theta_{i,j-1,k} - C_2 \theta_{i,j,k-1} \tag{19}$$

where $\theta_{i,j-1k}$ and $\theta_{i,j,k-1}$ are the angles between DA and DA_1 and EA and EA_2, respectively. The reference lines DA_1 and EA_2 are projections of prescribed lines DA_1^* and EA_2^* on the planes DAB and EAH, respectively. The DA_1^* itself is defined by a weighted average of 1) an extension of FD, to maintain smoothness along the η-line and 2) a line normal to the plane of $DG\tilde{D}$, $\tilde{D} = (i, j - 1, k + 1)$ on the $\xi_{j-1,k+1}$ line. Similarly, EA_2^* is a weighted average of an extension of QE and normal to the plane $EH\tilde{E}$, $\tilde{E} = (i, j + 1, k - 1)$ on the $\xi_{j+1,k-1}$ line. Other reference lines, such as streamlines or shocks, etc., could be used as well. These springs affect the orthogonality and smoothness constraints described by equations (7) and (8). The effects are incorporated in a one-sided manner without introduction of ellipticity in the η and ξ directions.

The distribution of grid points along the $\xi_{j,k}$-coordinate with new values of $s_{i,j,k}$ is determined by a balance equation for the complete spring system:

$$w_{i,j,k}(s_{i+1,j,k} - s_{i,j,k}) - w_{i-1,j,k}(s_{i,j,k} - s_{i-1,j,k}) - C_1\theta_{i,j-1,k} - C_2\theta_{i,j,k-1} = 0 \tag{20}$$

In the previous use of this scheme (ref. 10) C_1 and C_2 were held fixed for the entire flow field. The magnitude of these constants, large or small, makes the spring system highly damped or relatively undamped accordingly and the coordinate lines may oscillate. This increases the number of iterations for a desired convergence. An automatic calculation of C_1 and C_2 that keeps up with the "self-adaptivity" concept is possible, if they are specified along each ξ_{jk}-line as follows:

$$C_1 = \alpha\lambda_1 , \quad C_2\lambda_2 , \quad \alpha = \frac{1}{N}\sum_{n=1}^{N} w_{n,j,k} \tag{21}$$

where λ_1 and λ_2 are user specified constants and α is the average of tension spring constants along the ξ_{jk}-line. The adapted grid is less sensitive to (λ_1 and λ_2) than (C_1 and C_2) where the latter is fixed uniformly over the flow field.

The inclination angles are approximated by

$$\theta_{i,j-1,k} = \frac{s_{i,j,k} - s'_{i,j,k}}{DA_1} , \quad \theta_{i,j,k-1} = \frac{s_{i,j,k} - s''_{i,j,k}}{DA_2} \tag{22}$$

where $s'_{i,j,k}$ and $s''_{i,j,k}$ are the arc length to the intersection of reference lines DA_1 and DA_2, respectively. Equation (20) reduces to

$$\begin{aligned} w_{i,j,k}s_{i+1,j,k} - (w_{i+1,j,k} + w_{i-1,j,k} + T_{i,j-1,k} + T_{i,j,k-1})s_{i,j,k} + w_{i-1,j,k}s_{i-1,j,k} \\ = -T_{i,j-1,k}s'_{i,j,k} - T_{i,j,k-1}s''_{i,j,k} \longleftarrow \end{aligned} \tag{23}$$

$$T_{i,j-1,k} = (\alpha\lambda_1)/DA_1 , \quad T_{i,j,k-1} = (\alpha\lambda_2)/DA_2 \tag{24}$$

Equation (23) is a tridiagonal system of equations for $s_{i,j,k}$ along the $\xi_{j,k}$-line and can be readily solved. In this approach, only the torsion forces $T_{i,j-1,k}$ and $T_{i,j,k-1}$ on the upstream sides influence the distribution at the $\xi_{j,k}$ coordinate. This allows a simple marching scheme to be used along each $\xi_{j,k}$-line (on eighter η_j or ξ_k-plane) stepping along the j or the k index. However, if the influence of torsion forces from both sides ($T_{i,j-1,k}$ and $T_{i,j+1,k}$) and ($T_{i,j,k-1}$, $T_{i,j,k+1}$) are considered simultaneously, equation (23) is no longer a tridiagonal system of equations. This increases the computational effort without additional benefit.

It should be noted that equation (23) without inclusion of torsion forces is a 1-D elliptic equation, similar to equation (2), whose solution $s_{i,j,k}$ is a monotonically increasing function along the $\xi_{j,k}$-line. With torsion forces included, however, the equidistribution concept is disturbed. This may disturb the monotonicity of the solution when inappropriate values of the constants, λ_1 and λ_2, and the inclination angles are used. As a result the grid points may become extremely clustered, coarse and/or even crossed and overlapped. To avoid this, one should monitor the values of $s'_{i,j,k}$ and $s''_{i,j,k}$ for possible overlappings; e.g., $s'_{i,j,k} < s'_{i+1,j,k}$ for some $i = l$ and $s''_{i,j,k} < s''_{i+1,j,k}$ for some $i = m$. A good policy is to modify the values of $s'_{i,j,k}$ and $s''_{i,j,k}$ by equating them to the average of their local maximum and minimum in the overlapping region. Otherwise, in the case of severe clustering or coarseness weighting functions $w_{i,j,k}$ should be relaxed (decreased or increased) in proportion to a measure of the fractions $\Delta s_{i,j,k}/\Delta s_{MIN}$ or $\Delta s_{i,j,k}/\Delta s_{MAX}$, respectively. These considerations are automatically done in any course of adaptation; however, in the case where monotonicity of the solution values, $s_{i,j,k}$, fails, the procedure is terminated indicating some smaller values of the constants λ_1 and λ_2 be specified. Equation (23) is solved for each $s_{i,j,k}$, iterating until converged values of $s_{i,j,k}$ to some specified tolerance are obtained.

4.1 Outline of the Procedure

This section summarizes the basic steps for grid adaptation along principle adaptation directions; i.e., a family $\xi_{j,k}$-coordinate lines lying on the ζ_k-plane stepping along the j-index from $j = j_S$ to $j = j_E$ and then marching along the ζ_k-plane from $k = k_S$ to $k = k_E$ (definitions of these indices will be given next). The family of these adapted planes when marched along the k-index covers the entire field. The summary is as follows: 1) specify the rectangular box R of the computational space to be adapted, i.e., the boundary indices i_S, i_E, j_S, j_E, k_S, and k_E, where S and E denote starting and ending indices of R; 2) specify maximum and minimum allowable grid spacing Δs_{MAX} Δs_{MIN}, to control the density of the redistributed points; 3) specify the torsion spring coefficients λ_1 and λ_2; 4) march in the k direction (i.e., on each ζ_k-plane) beginning at k_S; 5) march in the j-direction (i.e., on each ξ_{jk}-line lying on the ζ-plane) beginning at j_S; 6) for each k and j calculate arc secant length $s_{i,j,k}$ and $w_{i,j,k}$ (from eq. (14)) for $i_S \sim i \sim i_E$; 7) calculate $s'_{i,j,k}$ and $s''_{i,j,k}$, $T_{i,j-1,k}$, $T_{i,j,k-1}$ and then correct for any overlapping of $s'_{i,j,k}$ and $s''_{i,j,k}$; 8) solve the tridiagonal system of equation (23) for new values of $s_{i,j,k}$ and monitor their monotonicity; if it fails the program is terminated, then return to step (3) and select a new set of parameters (λ_1 and λ_2) and repeat; 9) interpolate $f_{i,j,k}$ on $s_{i,j,k}$ from the flow solution data on the old grid points; and 10) check the convergence of $s_{i,j,k}$; if not converged keep returning to step 6.

4.2 Multidirectional Adaptations

Multidirectional adaptation is achieved by sequential application procedure into a sequence of unidirectional adaptations. For instance, the grid points are first adapted to the solution along each ξ-coordinate, followed by adaptation along each η-coordinate, and then for each ζ-coordinate. This is analogous to the fractional step schemes for PDEs. It should be recalled that the earlier concept is maintained because of the consideration of torsion forces from one side only.

The selected order of adaptation is arbitrary, but the resulting adapted grid is not necessarily unique. One reason is that equation (23) is nonsymmetric and the choice of the boundary of the region R and the order of adaptation direction affect the redistribution of grid points on all grid lines. Another reason is that torsion parameters λ_1 and λ_2 play a role in deciding the final grid.

A good policy for choosing the first adaptation direction would be choosing the coordinate for which gradients with flow properties are large. The solution field is interpolated into a newly adapted grid, using second-order, one-dimensional Lagrange interpolation after each unidirectional adaptation, and is more efficient than using multidimensional interpolation. For unsteady problems, no interpolation would be necessary if the grid speeds were registered after each adaptation and incorporated into transformed time derivatives of the solution variables.

The 3-D code which implements this algorithm enjoys a modular structure. The user can specify any direction of coordinate line for the adaptation and either of two planes of adaptation, as discussed earlier; then the grid and solution data are swapped so that the standard procedure of the previous section can be applied. This portion of the algorithm is another important user friendly feature of this code which automatically maps the initial input grid and solution data with user-specified directions of adaptation into newly ordered data. The adaptation can then proceed along the principal adaptation directions and that after completion the data are returned to their original order before output, by means of inverse mapping. The mapping and/or remapping procedure in 3-D takes place by the appropriate reordering of the indices i, j, and k and (x, y, and z) coordinates of the physical space.

The concept of splitting not only breaks complex 3-D flow field calculation into effective successive 1-D ones, but it also enables one to take advantage of the vector processing architectures of modern computers and treat large 3-D data sets by employing the block/pencil data base approach for large and complex 3-D data (e.g., Lomax and Pulliam (ref. 16) and Deiwert and Rothmund (ref. 17)) where the topological space is subdivided into blocks which interface with one another exactly.

4.3 Boundary Modifications

This section explains the adaptation procedure near or on the boundaries of the subdomain R. Here we address two issues: 1) the modification of equation (23) for the boundaries of R, and 2) the maintenance of the configuration of the initial grid lines near the boundaries. For definiteness, consider the principle marching directions described in the previous sections. For $k = k_S$ the torsion forces $T_{i,j,k-1}$ are equal to zero, because the $(k_S - 1)$st plane corresponds to an external plane of the region R. Equation (29) would be modified by setting $T_{i,j,k-1} = 0$. The adaptation along the $\xi_{j,k}$-line for $j = j_S$ (i.e., the initial line) would further require that torsion force $T_{i,j-1,k}$ from the η-direction be zero because $j_S - 1$ corresponds to the external boundary; therefore, both torsion forces in equation (23) should be considered zero; i.e., it then reduces to a simple equidistribution concept.

As for the second issue, it is essential to preserve the grid line configuration on or near the boundary. Redistribution of grid points along $\xi_{j,k}$-lines may affect the direction of the initial grid lines, $\eta_{i,k}$ or $\xi_{i,k}$, in the neighborhood of the boundaries of R. This may be undesirable for cases where, for instance, a boundary of R is an interface between two patched blocks of an initial grid, or when it corresponds to a physical boundary. The present algorithm tackles this problem by modifying torsion constants (λ_1 and λ_2) and the related direction of the reference lines.

For clarity of discussion below let $\vec{r}$, $\vec{g}$, and $\vec{n}$ denote unit vectors along a reference line, a grid line, and a line normal to the boundary, respectively, issued from a grid point on the $\xi_{j,k}$ lines on or near the boundary. As defined before, the reference line is a weighted average of vectors $\vec{g}$ and $\vec{n}$; then we can write:

$$r = \beta_1 \vec{g} + \beta_2 \vec{n} \tag{25}$$

where β_1 and β_2 are some user specified parameters on the interval [0,1], $\beta_2 = 1 - \beta_1$. Control of the vector $\vec{r}$ affects the inclination of the angles which in turn control the distribution of grid points along the adapted line $\xi_{j,k}$. To further clarify this point, suppose that in a given problem the user has to preserve the initial grid direction of the $\eta_{i,k}$-lines near the boundary, say for several ξ_{jk}-lines, stepping along the j-index from $j = j_S$ to $j = j_S + n_g$. This n_g, where $n_g > j_S$ is specified by the user and is a mean to define a desired neighborhood near the boundary. Then by a simple algorithm, such as

$$\tilde{\beta}_1 \rightarrow \beta_1 + (1 - \beta_1)\left(\frac{n_g - j + j_S}{n_g}\right) \tag{26}$$

one finds that for $j = j_S$ (i.e., at the boundary) β_1 is increased to unity shifting the emphasis to the direction of vector $\vec{g}$, then for $j = j_S + n_g$, β_1 retains its original value, and for $j_S < j < n_g$, β_1 and $\vec{g}$ are smoothly modified. This procedure allows preservations of the initial directions of the $\eta_{i,k}$-lines of $\vec{g}$. At the same time the value of the torsion constant λ_1 is modified in a similar way making the adjustment of β_1 more effective.

This argument can be repeated for problems where the initial boundary-conforming grid must be preserved by smooth modification of β_2. This would shift the emphasis to vector $\vec{n}$ near the boundary and smoothly transmit this effect to some specified neighbor of the boundary so that the orthogonality of grid lines near the boundary would be preserved. The procedure of modifying λ_1 and $\vec{r}$ just described for the $\eta_{i,k}$-lines can be repeated for all $\zeta_{i,j}$-lines near the boundary plane ζ_k, $k = k_S$ when stepping along the k-index. This approach makes the entire algorithm for treating the boundary consistent with the self-adaptive concept.

There are also two more concerns with the boundaries. One, when the subdomain R does not correspond to the entire flow field, is that after each adaptation on the ζ_k-plane the grid distribution along the lines $\xi_{j,k}$, $j = j_E$ may not match smoothly with those $\xi_{j,k}$-lines outside the boundaries, specifically the $\xi_{j,k}$-lines with $j > j_E$. This case can be cured by projecting the grid spacing of the $\xi_{j,k}$-lines, $j = j_E$ onto the $\xi_{j,k}$-lines, $j = j_E + 1$ and so forth, or by using algorithms such as discussed previously to preserve grid line configuration when $j \rightarrow j_E$ and $k \rightarrow k_E$. Another concern is in avoiding large differences in spacing at the boundaries, along the $\xi_{j,k}$-lines; in particular between $\Delta s_{i-1,j,k}$ and $\Delta s_{i,j,k}$ for $i = i_S$, and between $\Delta s_{i-1,j,k}$ and $\Delta s_{i,j,k}$ for $i = i_E$. These differences can be modified by the magnitude of tension forces like the weighting functions $w_{i,j,k}$ at both ends prior to adaptation according to the concept of equidistribution; that is, we let

$$w_{i,j,k}\,\Delta s_{i-1,j,k} = \bar{C}, \quad \text{at} \quad i = i_S \tag{27}$$

$$w_{i,j,k}\,\Delta s_{i+1,j,k} = \bar{C}, \quad \text{at} \quad i = i_E \tag{28}$$

where $\bar{C}$ is an average of the products $w_{i,j,k}\,\Delta s_{i,j,k}$ along the current $\xi_{j,k}$-line (or previously adapted line $\xi_{j-1,k}$). Then these values of $w_{i,j,k}$ at both ends are merged to alleviate the values of several weight functions in the neighborhood of $i = i_S$ and $i = i_E$. The related algorithm here is also compatible with the self-adaptive concept and will be effective upon request by specifying a related input parameter.

4.4 Example of a Solution-Adapted Grid

To demonstrate the feasibility of the algorithm, the 3-D flow field about a two-nozzle afterbody plume flow is considered. Complex flow structure is produced because of interaction between external supersonic flow and underexpanded supersonic jets in the plume region. Flow features consist of shocks, mixing shear layers, barrel shocks, and Mach disk, etc. This structure is further complicated due to 3-D geometric complexities and multiple jet stream interactions. Vankatapathy and Feiereisen (ref. 18) have computed multinozzle plume flows with a 3-D Navier-Stokes solver; their numerical simulations are performed with an upwind solver for an ideal gas. We have used the self-adaptive algorithm to adapt the initial grid used by these authors to the solution of the two-nozzle plume flow. The computation is made under such free-stream conditions as: Mach number = 4.22, temperature = 250 K, and pressure = 0.0028 atm. Figure 1 depicts the 3-D initial grid in the physical space, limited to the plume region downstream of the nozzle's exit plane. The flow is bisymmetric about the yx-plane and zx-plane, and thus the computational domain is limited to one quadrant. The grid has a nearly cylindrical structure whose centerline is parallel to the x-axis and located near it on the yx-plane. Figure 2 shows the converged solution in terms of the Mach number. The Mach contours present the flow features in the plume region.

Figure 3 shows a solution adapted grid. The grid is adapted to the solution of Mach numbers in the entire region; this means the function $f_{i,j,k}$ in equation (11) is equal to the absolute value of the gradient of the Mach number along the direction of ζ-lines. The grid consists of 97 x 25 x 47 grid points. The subdomain R of the adaptation is specified by the input parameters $i_S = 1$, $i_E = 97$, $j_S = 1$, $j_E = 25$, $k_S = 47$, and $k_E = 5$. Adaptation is performed along the $\zeta_{i,k}$-lines stepping along the k-index from the outer boundary at $k_S = 47$ toward the centerline but kept away from it at $k_E = 5$. After sweeping a family of $\zeta_{i,k}$-lines on an η_j-plane, the procedure continues by adapting the entire η_j-plane, stepping it along the j-index.

Minimum and maximum grid spacings and torsion constants are $\Delta s_{MIN} = 0.001$, $\Delta s_{MAX} = 1$, $\lambda_1 = 0.15$, and $\lambda_2 = 0.001$, respectively. On the boundary of R we have requested the preservation of the initial grid line direction that is for the $\xi_{j,k}$ and $\eta_{i,k}$-lines there. The overall adapted grid clearly follows the specifications of the algorithm (e.g., clustering grid points across the shocks, mixing layers, barrel shocks, etc.) depicting all the explicit flow features.

4.5 Example of an Initial Grid Generation

A secondary but important purpose of this work is to describe the use of this adaptive grid method as an effective initial grid generation scheme. The grid spacings are controlled by weighting functions which are related to the specific geometry of the functions rather than to the flow properties. The example presented here describes this procedure for a simple bump-like geometry. It should be noted that complicated geometries can be subdivided into patched blocks of simple geometry; then the following simple procedure may separately be applied on each block.

The surface geometry of the bump is defined by

$$z(x,y) = \begin{cases} [1 + \cos(4\pi y)][1 + \sin(4\pi(x - 0.5) - 0.5\pi] , & \\ \qquad\qquad \text{for } 0.5 \leq x < 1 \text{ and } 0 \leq y \leq 0.25 & \\ 0, \qquad\qquad \text{for } x \leq 0.5 \text{ or } 1 \leq x \text{ or } 0.25 \leq y \end{cases} \qquad (29)$$

Initially a rectangular grid (49 x 29 x 40) is generated with uniform spacing on each side of a rectangular box B of the physical space; this is shown in figure 5. The coordinates ξ, η, and ζ are now considered to be parallel with the direction of x, y, and z-axes. Next, the bottom surface of B is deformed to describe the bump, defined by equation (29), by shortening the z-coordinate accordingly. Figure 6 ~~shows this ste~~p of the grid construction in only one-half of the space. The computational domain R is ndices $i_S = 1, i_E = 49, j_S = 1, j_E = 29, k_S = 1$, and $k_E = 40$.

from figure 6, the ξ and η coordinates have conformed to the body shape, but have not the physical boundaries; ζ coordinates along the z-axis are not orthogonal to the surface of make this grid more suitable for a 3-D flow field type of calculations, we need to properly grid points in the physical space. In this instance, for a flow past a bump, we may cluster near the yz-symmetry plane along the η-coordinates, and near the surface of the bump ordinates and adjust the inclinations of the ζ-lines so that they become orthogonal to the bump. All this can be done by successive application of the described scheme as follows: the adaptation along the $\xi_{j,k}$-lines on the ζ_k-planes. Cluster the grid points using a f geometric weighting functions for regions of large curvature and exponentially stretched the boundaries. In equation (11) $f_{i,j,k}$ is replaced by a measure of the curvature and M_0 n exponential function given by

$$f_{i,j,k} = z_{i-1,j,k} - 2z_{i-1,j,k} + z_{i+1,j,k} , \quad M_0 \equiv \left(\frac{\Delta s_I}{\Delta s_{II}}\right)^{(k/m)} \qquad (30)$$

- i_S - 1, Δs_I and Δs_{II}, the mesh spacings at both ends, are also specified.

line inclinations are controlled by using large relative values for the torsion constants λ_1 the grid quasi-orthogonal near the surface of the bump.

nd application of the scheme proceeds as previously mentioned schemes except for the g the $\eta_{i,j}$-lines on the ζ-plane. The third application proceeds along the $\zeta_{j,k}$-lines on the $f_{i,j,k}$ in equation (30) set to zero because there is no curvature change along this direction. the final enhanced initial grid, which includes the features of clustering and orthogonality nd boundaries of the bump.

5. SUMMARY

ition-adaptive grid based on variational principles and spring analogies was described. A s tical splitting procedure for multidirectional adaptation is used. Grid skewness and [illegible] controlled by one-sided torsion springs. User-specified maximum and minimum grid spacings determine other important constants of the method and control the clustering and stretching of the grid. Two computed examples were used to confirm the feasibility of the method and to demonstrate the adaptive grid as robust and solution effective as well as a simplified initial grid generation scheme.

6. REFERENCES

1. Anderson, D. A.: Adaptive Grid Methods for Partial Differential Equations. In: K. N. Ghia and W. Ghia, eds., Advances in Grid Generation ASME-FED5, ASME, New York, 1983, pp. 1-15.

2. Babuska, I.; Chandra, J.; and Flaherty, J. E., eds.: Adaptive Computational Methods for Partial Differential Equations, SIAM, Philadelphia, PA, 1983.

3. Thompson, J. F.: A Survey of Dynamically-Adaptive Grids in the Numerical Solution of Partial Differential Equations. AIAA Paper 84-1606, 1984.

4. Eiseman, P. R.: Adaptive Grid Generation. Computer Methods in Applied Mechanics and Engineering, vol. 64, nos. 1-3, 1987, pp. 321-376.

5. Dwyer, H. A.: Adaptive Gridding for Finite Difference Solutions to Heat and Mass Transfer Problems. Numerical Grid Generation, J. Thompson, ed., North-Holland, 1987, p. 339.

6. Anderson, D. A.; and Rajendran, N.: Two Approaches Toward Generating Orthogonal Adaptive Grids, AIAA Paper 84-1610, 1984.

7. Dwyer, H. A.; and Onyejekwe: Generation of Fully Adaptive and/or Orthogonal Grids. Proceedings of the Ninth Int. Conf. on Num. Methods in Fluid Dynamics, Saclay, Soubbaramayer and Broujot, eds., Springer-Verlang, 1985.

8. Eiseman, P. R.: Alternating Direction Adaptive Grid Generation. AIAA 6th Computational Fluid Dynamics Conference, Danvers, Mass., 1983.

9. Nakahashi, K.; and Deiwert, G. S.: A Practical Adaptive Grid Method for Complex Fluid Flow Problems, NASA TM-85989, 1984.

10. Nakahashi, K; and Deiwert, G. W.: A Three-Dimensional Adaptive Grid Method. AIAA Paper 85-0486, 1985.

11. Nakahashi, K.; and Deiwert, G. S.: A Self-Adaptive Grid Method with Application to Airfoil Flow. AIAA Paper 85-1525-CP, 1985.

12. White, A. B., Jr.: On Selection of Equidistributing Meshes for Two Point Boundary-Value Problems, SIAM J. Numerical Analysis, vol. 16, no. 3, 1979, pp. 472-502.

13. Brakbill, J. U.; and Saltzman, G. S.: Adaptive Zoning for Singular Problems in Two Dimensions. J. of Comp. Phys., vol. 46, no. 3, 1982, pp. 342-368.

14. Saltzman, J.; and Brackbill, J.: Applications and Generalizations of Variational Methods for Generating Adaptive Meshes. Numerical Grid Generation, Thompson, J. F., ed., North-Holland, Amsterdam, 1982.

15. Saltzman, J.: A Variational Method for Generating Multi-Dimensional Adaptive Grids. Report DOE/ER/03077-174, Courant Mathematics and Computing Laboratory, New York University, New York, 1982.

16. Lomax, H.; and Pulliam, T. H.: A Fully Implicit Finite-Difference Factored Code for Computing Three-Dimensional Flows on the ILLIAC IV. Parallel Computations, G. Rodrigue, ed., Academic Press, New York, 1982, pp. 217-250.

17. Deiwert, G. S.; and Rothmund, H.: Three-Dimensional Flow Over a Conical Afterbody Containing a Centered Propulsive Jet: A Numerical Simulation. AIAA Paper 83-1709, 1983.

18. Venkatapathy, E.; and Feiereisen, W. J.: 3-D Plume Flow Computations with an Upwind Solver, AIAA Paper 88-3158, 1988.

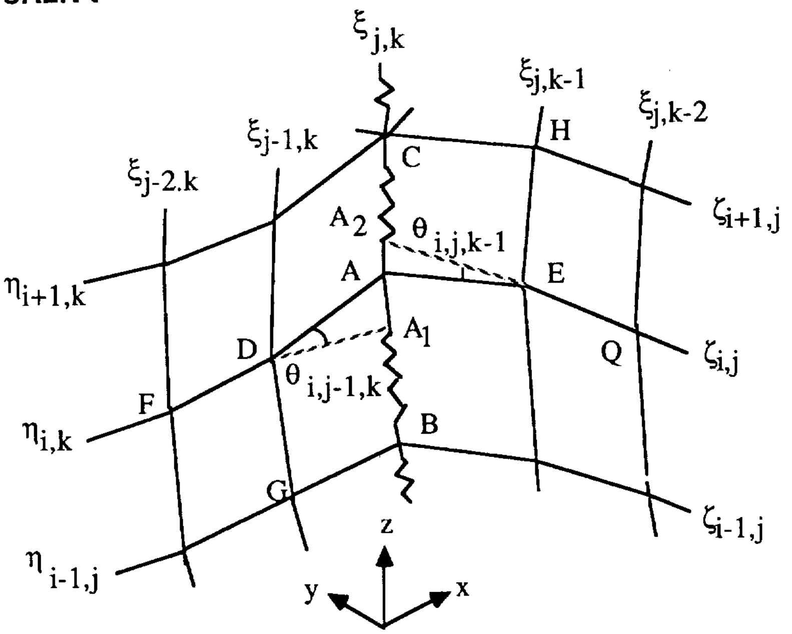

Figure 1.– Schematic of grid lines with tension and torsion spring analogy.

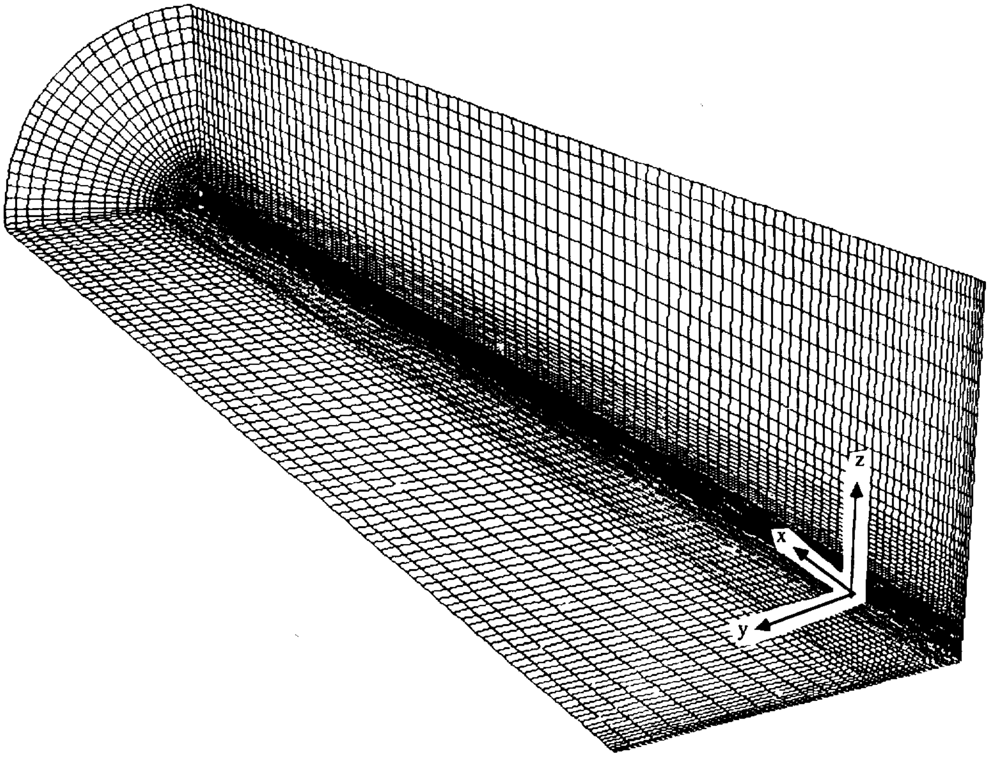

Figure 2.– 3-D initial grid–plume region.

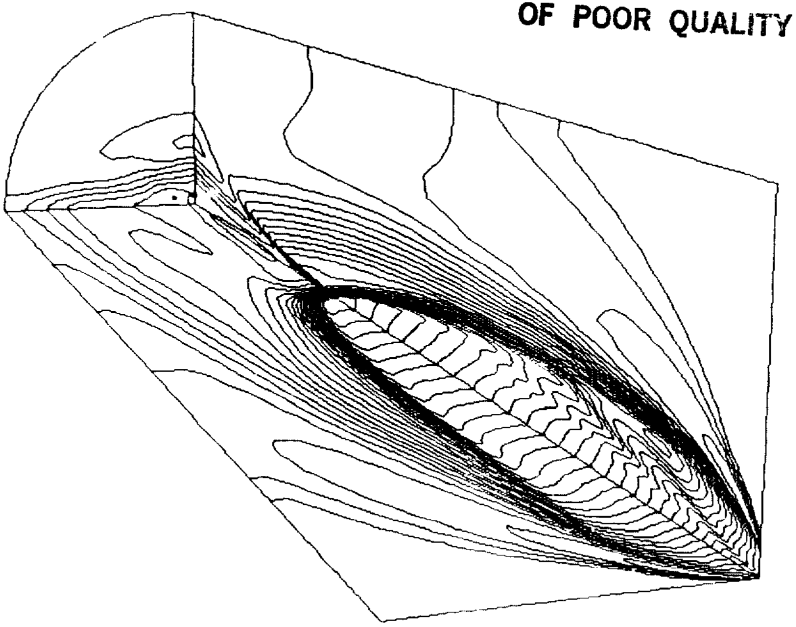

Figure 3.– Computed Mach contours–plume flow.

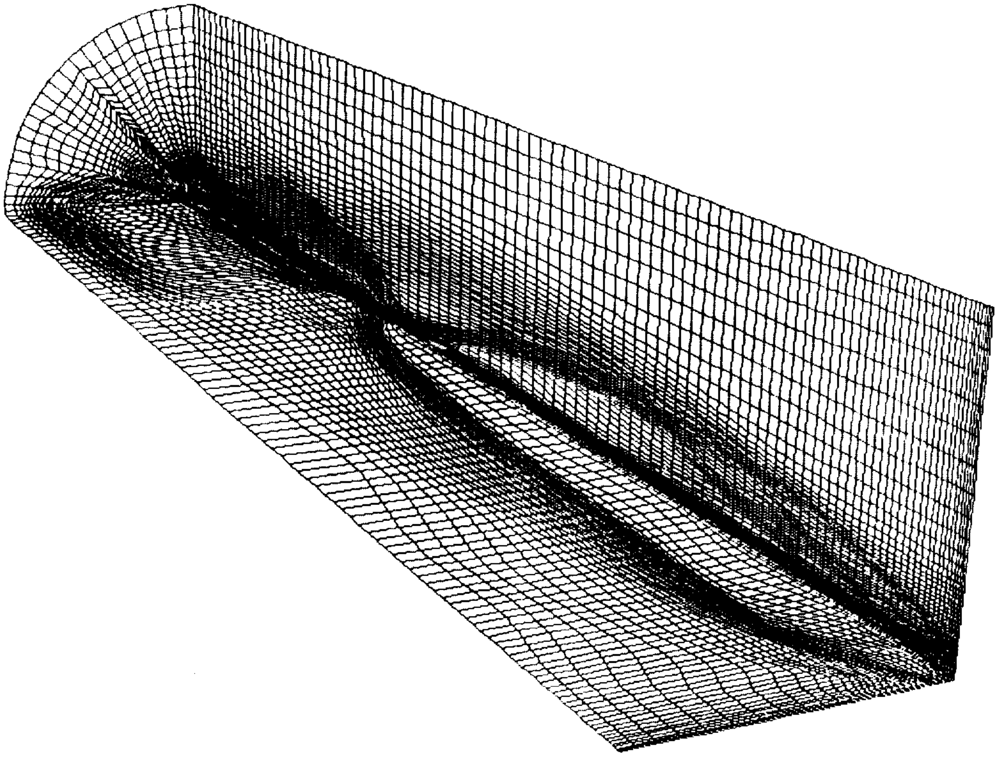

Figure 4.– 3-D solution-adapted grid, based on Mach numbers.

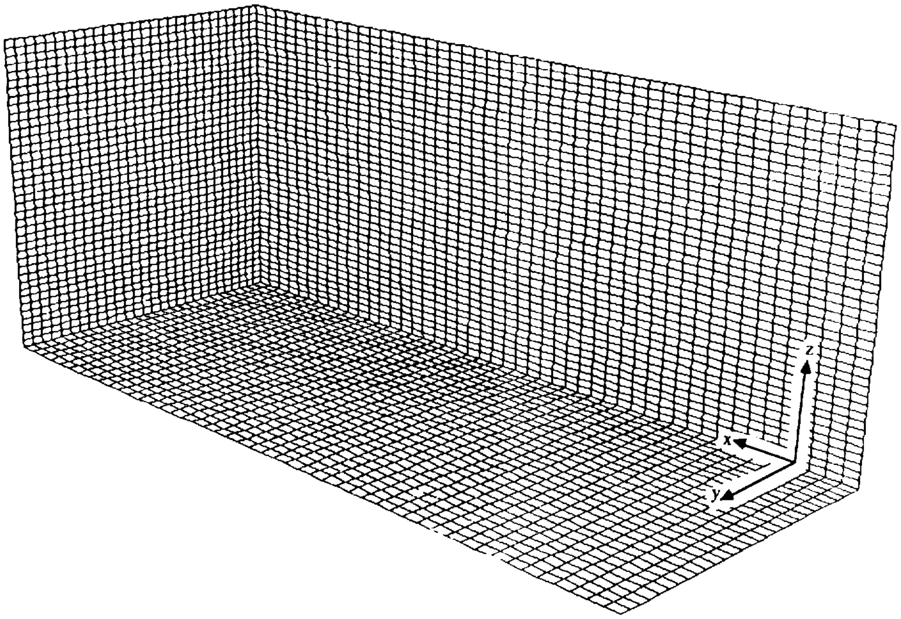

Figure 5.– Uniform spacing, rectangular grid.

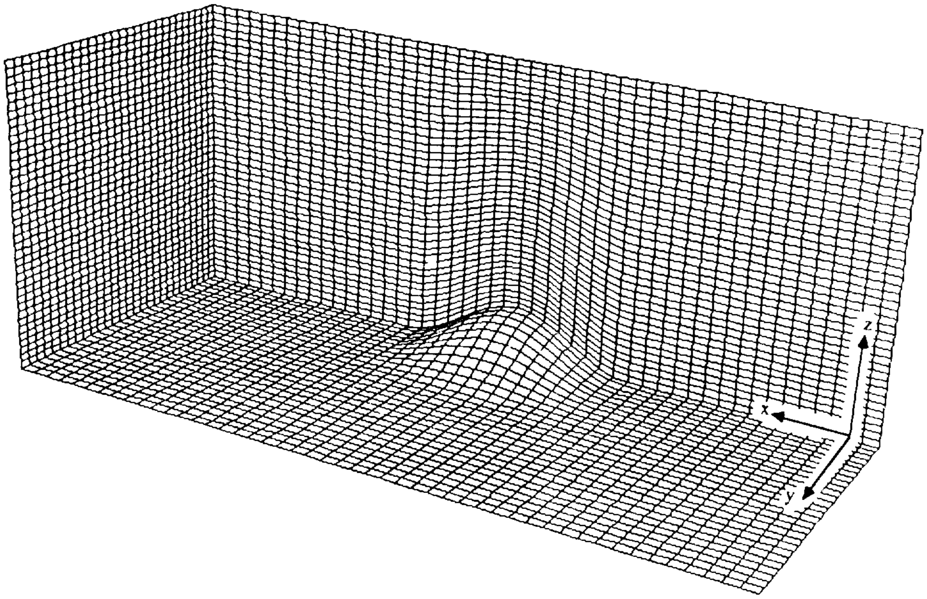

Figure 6.– Uniform spacing, 3-D grid about the bump.

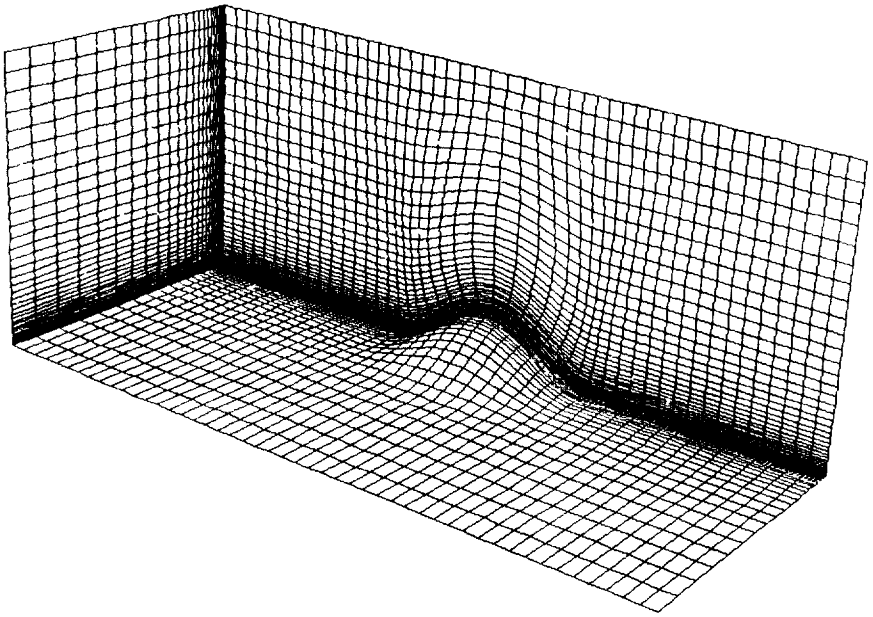

Figure 7.– Enhanced 3-D initial grid, based on curvature and stretch functions.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA TM-101027	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle Three-Dimensional Self-Adaptive Grid Method for Complex Flows	5. Report Date November 1988 6. Performing Organization Code
7. Author(s) M. Jahed Djomehri and George S. Deiwert	8. Performing Organization Report No. A-88277 10. Work Unit No. 506-40-91
9. Performing Organization Name and Address Ames Research Center Moffett Field, CA 94035	11. Contract or Grant No. 13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546-0001	14. Sponsoring Agency Code

15. Supplementary Notes

Point of Contact: M. Jahed Djomehri, Ames Research Center, MS 230-2, Moffett Field, CA 94035
(415) 694-9907 or FTS 464-9907

16. Abstract

A self-adaptive grid procedure for efficient computation of three-dimensional complex flow fields is described. The method is based on variational principles to minimize the energy of a spring system analogy which redistributes the grid points. Grid control parameters are determined by specifying maximum and minimum grid spacings. Multidirectional adaptation is achieved by splitting the procedure into a sequence of successive applications of a unidirectional adaptation. One-sided, two-directional constraints for orthogonality and smoothness are used to enhance the efficiency of the method. Feasibility of the scheme is demonstrated by application to a multinozzle, afterbody, plume flow field. Application of the algorithm for initial grid generation is illustrated by constructing a three-dimensional grid about a "bump-like" geometry.

17. Key Words (Suggested by Author(s)) Three-dimensional Self-adaptive Grid method	18. Distribution Statement Unlimited – Unclassified Subject Category – 02

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 19	22. Price A02

For sale by the National Technical Information Service, Springfield, Virginia 22161