AIAA JOURNAL
Vol. 32, No. 5, May 1994

New Multigrid Approach for Three-Dimensional Unstructured, Adaptive Grids

Vijayan Parthasarathy* and Y. Kallinderis†
University of Texas at Austin, Austin, Texas 78712

A new multigrid method with adaptive unstructured grids is presented. The three-dimensional Euler equations are solved on tetrahedral grids that are adaptively refined or coarsened locally. The multigrid method is employed to propagate the fine grid corrections more rapidly by redistributing the changes-in-time of the solution from the fine grid to the coarser grids to accelerate convergence. A new approach is employed that uses the parent cells of the fine grid cells in an adapted mesh to generate successively coarser levels of multigrid. This obviates the need for the generation of a sequence of independent, nonoverlapping grids as well as the relatively complicated operations that need to be performed to interpolate the solution and the residuals between the independent grids. The solver is an explicit, vertex-based, finite volume scheme that employs edge-based data structures and operations. Spatial discretization is of central-differencing type combined with special upwind-like smoothing operators. Application cases include adaptive solutions obtained with multigrid acceleration for supersonic and subsonic flow over a bump in a channel, as well as transonic flow around the ONERA M6 wing. Two levels of multigrid resulted in reduction in the number of iterations by a factor of 5.

I. Introduction

UNSTRUCTURED grids have evolved over the recent years as a very viable approach for modeling three dimensional geometries since they have been found to be very efficient means of generating body-conforming grids for complex three-dimensional configurations.[1] In the case of structured grids, their inherent order makes it very convenient to generate any number of successively coarser meshes by removing every other line of grid points in each coordinate direction. Hence, the implementation of the multigrid techniques have long been a feature of flow solvers that use structured grids. This classical multigrid approach used with structured grids has been very successful in accelerating convergence for numerical schemes, solving problems governed by elliptic equations.[2,3] The theory of multigrid method for problems involving hyperbolic equations is not so firmly established. However, significant success has been achieved in this area for accelerating the convergence to a steady state with explicit numerical schemes.[4,5]

The complete lack of order in unstructured grids precludes any possibility of generating successively coarser grids as is easily done with structured meshes. This poses an additional challenge for the implementation of a multigrid method on such grids. Hence, very little work has been done till now that explores the possibilities of implementing a coarse grid acceleration scheme for flow solvers that employ unstructured grids. One of the noted works in the area of unstructured multigrid techniques is by Mavriplis and Jameson,[6] who presented a scheme for coarse grid acceleration with a finite volume solver on a two-dimensional unstructured grid. This concept was later extended to three-dimensional unstructured grids.[7,8] The technique employs a sequence of independent and successively coarser grids that are generated a priori. Furthermore, the scheme involves extensive data structures for the intergrid transfer of the flow variables as well as the residuals.

Adaptive grid methods have evolved as an efficient tool to obtain numerical solutions without a priori knowledge of the nature and the resolution of the grid necessary to efficiently capture the flow features. These algorithms detect the regions that have prominent flow features and increase the grid resolution locally in such areas. Furthermore, they coarsen the grid by deleting the cells over the regions where flow features no longer exist. The data structures needed for the implementation of adaptive algorithms on three-dimensional unstructured grids are quite complicated. This has been a challenging topic in itself and until recently not many such schemes existed. Significant progress has since been made in the implementation of adaptive schemes for tetrahedral grids.[9–12]

The motivation for the new multigrid scheme developed in the present work comes from the work of Ni.[5] The coarser grids are used to propagate the fine grid solution's changes-in-time properly and rapidly throughout the flow domain, thus accelerating convergence to the steady state, while at the same time maintaining the low truncation error on the fine grid. The basic solver used for the discretization of the governing equations on the fine grid is a central space differencing, node-centered Euler scheme for tetrahedral grids. It is a one-step Lax-Wendroff-type scheme. The method employs special upwind-like smoothing operators for shock capturing and background smoothing. The scheme is formulated so that all operations are edge based, which reduces the computational work significantly.

The adaptive grids are created by division of tetrahedral cells. Isotropic as well as directional division of the cells provides considerable flexibility in increasing the resolution of the grid with the minimum possible number of cells. The cell-tree data structure is used to retrace the parents of the fine cells on the adapted grid, and these parent cells along with the cells in the unembedded region of the flow domain constitute the coarser level of the multigrid. The edge-tree data structure is used to interpolate the residuals back and forth between the fine and coarser level grids, and hence the scheme avoids the relatively complicated intergrid transfer algorithms employed in the other multigrid schemes.[7,8] The coarse grids are used only to propagate the solution changes obtained on the fine grid and hence create the corrections to the fine grid changes. The governing equations are not solved numerically on the coarser levels, which makes the scheme computationally less intensive. Application cases include adaptive solutions obtained with multigrid acceleration for supersonic flow over a bump in a channel and transonic flow around the ONERA M6 wing. Multigrid acceleration of convergence is monitored.

II. Basic Numerical Scheme

The governing equations to be solved is the system of time-dependent Euler equations for a perfect gas that combines the equa-

Received June 8, 1993; revision received Nov. 10, 1993; accepted for publication Nov. 26, 1993.

*Graduate Research Assistant, Department of Aerospace Engineering and Engineering Mechanics.

†Associate Professor, Department of Aerospace Engineering and Engineering Mechanics. Member AIAA.

tions of mass, momentum, and energy. The integral form of the equation is considered:

$$\int_{\Omega}\left(\frac{\partial \boldsymbol{U}}{\partial t}+\frac{\partial \boldsymbol{F}}{\partial x}+\frac{\partial \boldsymbol{G}}{\partial y}+\frac{\partial \boldsymbol{H}}{\partial z}\right)\mathrm{d}\Omega=0 \tag{1}$$

The state vector $\boldsymbol{U}$ and the flux vectors $\boldsymbol{F}$, $\boldsymbol{G}$, and $\boldsymbol{H}$ are expressed in terms of the conservation variables, namely, density ρ, x, y, and z momentum; and energy. The solution at any node 0 at time level $n+1$ can be expressed in terms of the solution at time level n using a Taylor series expansion:

$$\begin{aligned}\boldsymbol{U}_0^{n+1} &= \boldsymbol{U}_0^n+\delta\boldsymbol{U}_0^n\\ \delta\boldsymbol{U}_0^n &= \left(\frac{\partial \boldsymbol{U}}{\partial t}\right)_0^n\Delta t+\left(\frac{\partial^2\boldsymbol{U}}{\partial t^2}\right)_0^n\frac{\Delta t^2}{2}+\mathcal{O}(\Delta t^3)\end{aligned} \tag{2}$$

The temporal derivatives in the preceding expression are evaluated in terms of the spatial derivatives using the governing equations according to the Lax-Wendroff approach, and local time stepping is used based on the Courant-Friedrichs-Lewy (CFL) stability criterion.[16] The control volumes are defined around each node using the dual mesh, which is formed by constructing nonoverlapping volumes, referred to as dual cells. The dual cells represent the control volume associated with the respective node. The dual mesh for a tetrahedral grid is constructed by dividing each tetrahedron into four hexahedra of equal volumes by connecting the mid-edge points $(E_1,\ldots,E_6)$, face centroids $(F_1,\ldots,F_4)$, and the centroid of the tetrahedron (C), as shown in Fig. 1. The control volume around a node is constituted by a polyhedral hull that is the union of all such hexahedra that share the node.

The volume integral of the spatial terms in the Euler equation (1) has to be evaluated to compute the first-order part of the change $\delta\boldsymbol{U}$ in Eq. (2), at each node 0:

$$\int_{\Omega_0}\left(\frac{\partial \boldsymbol{F}}{\partial x}+\frac{\partial \boldsymbol{G}}{\partial y}+\frac{\partial \boldsymbol{H}}{\partial z}\right)\mathrm{d}\Omega=\int_{\partial\Omega_0}(\boldsymbol{F}n_x+\boldsymbol{G}n_y+\boldsymbol{H}n_z)\,\mathrm{d}S \tag{3}$$

which can be written in the discrete form as

$$\sum_k(\boldsymbol{F}S_x+\boldsymbol{G}S_y+\boldsymbol{H}S_z)_k \tag{4}$$

where the summation is over all of the dual mesh faces that constitute the boundary of the control volume around a node and S_x, S_y, and S_z are the area projections of the dual face.

A. Edge-Based Operations

The flux evaluation can be cast into edge-based operations. Consider an edge constituted by the nodes 0, $N(k)$. The quadrilateral faces of the dual mesh that are connected to the edge at its midpoint E are shown in Fig. 2. The number of such quadrilateral faces connected to an edge depends on the number of cells sharing that edge. The boundary of the control volume around a node is constituted by the union of such faces associated with each edge that shares the node. Thus, the summation over the dual mesh faces in Eq. (4) is equivalent to a summation over the edges of the grid, and the fluxes through the dual mesh faces associated with each edge are evaluated. This eliminates a significant amount of computational work since the number of edges is much less than the number of faces in an unstructured grid. Thus, the contribution of an edge to the fluxes across the faces of the dual volume surrounding a node is given by

$$\boldsymbol{F}_e(A_e)_x+\boldsymbol{G}_e(A_e)_y+\boldsymbol{H}_e(A_e)_z \tag{5}$$

where $(A_e)_x$, $(A_e)_y$, and $(A_e)_z$ are the projections of the area A_e associated with the edge E and $\boldsymbol{F}_e$, $\boldsymbol{G}_e$, and $\boldsymbol{H}_e$ are the flux vectors evaluated at the midpoint of the edge. The volume integral for evaluation of the second-order changes in Eq. (2) is

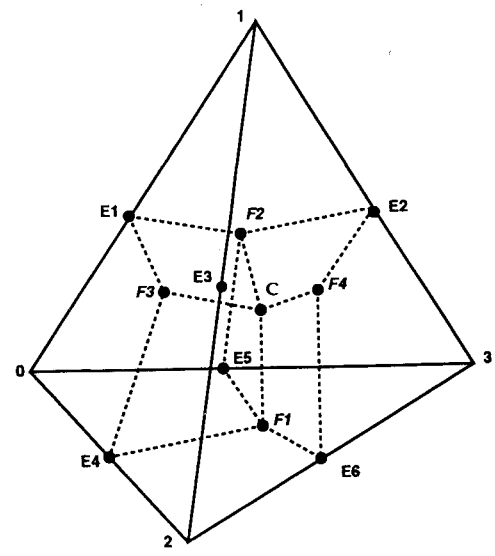


Fig. 1 Dual mesh, in dashed lines, for a tetrahedral grid.

$$\int_{\Omega_0}\frac{\partial}{\partial t}\left(\frac{\partial \boldsymbol{F}}{\partial x}+\frac{\partial \boldsymbol{G}}{\partial y}+\frac{\partial \boldsymbol{H}}{\partial z}\right)\mathrm{d}\Omega$$

$$=\int_{\partial\Omega_0}\left[\left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}}\frac{\partial \boldsymbol{U}}{\partial t}\right)n_x+\left(\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}\frac{\partial \boldsymbol{U}}{\partial t}\right)n_y+\left(\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{U}}\frac{\partial \boldsymbol{U}}{\partial t}\right)n_z\right]\mathrm{d}S \tag{6}$$

which can be written in the discrete form as

$$\sum_{k=1}^{n}(\tilde{\boldsymbol{A}}S_x+\tilde{\boldsymbol{B}}S_y+\tilde{\boldsymbol{C}}S_z)_k\left(\frac{\partial \boldsymbol{U}}{\partial t}\right)_k \tag{7}$$

The summation, similar to the first-order changes, is over the edges, and the contribution of each edge to the second-order change at a node is given by

$$[\tilde{\boldsymbol{A}}_e(A_e)_x+\tilde{\boldsymbol{B}}_e(A_e)_y+\tilde{\boldsymbol{C}}_e(A_e)_z]\left(\frac{\partial \boldsymbol{U}}{\partial t}\right)_e \tag{8}$$

where $\tilde{\boldsymbol{A}}_e$, $\tilde{\boldsymbol{B}}_e$, and $\tilde{\boldsymbol{C}}_e$ are the Jacobians evaluated using the average state vector $\boldsymbol{U}_e$ at the mid-edge point E.

The change $\delta\boldsymbol{U}_0^n$ of the state vector at any node 0 is obtained by substituting Eqs. (5) and (8) into Eq. (2):

$$\delta\boldsymbol{U}_0^n=\frac{-1}{\Omega_0}\sum_{k=1}^{n}\Delta t\,[\boldsymbol{F}_e(A_e)_x+\boldsymbol{G}_e(A_e)_y+\boldsymbol{H}_e(A_e)_z]_k$$

$$+\frac{-1}{\Omega_0}\sum_{k=1}^{n}\frac{\Delta t^2}{2}\left\{[\tilde{\boldsymbol{A}}_e(A_e)_x+\tilde{\boldsymbol{B}}_e(A_e)_y+\tilde{\boldsymbol{C}}_e(A_e)_z]\left(\frac{\partial \boldsymbol{U}}{\partial t}\right)_e\right\}_k \tag{9}$$

B. Upwind-Like Artificial Dissipation

The dissipation modeling in this work is formulated in such a manner as to simulate the implicit dissipation terms of the upwinding schemes without increasing the computation cost of the algorithm.

The numerical formula for the flux vector at any intermediate state I between two end states L and R can be expressed as

$$\boldsymbol{F}_I=(1/2)[(\boldsymbol{F}_L+\boldsymbol{F}_R)-\tilde{\boldsymbol{A}}_r(\boldsymbol{U}_R-\boldsymbol{U}_L)] \tag{10}$$

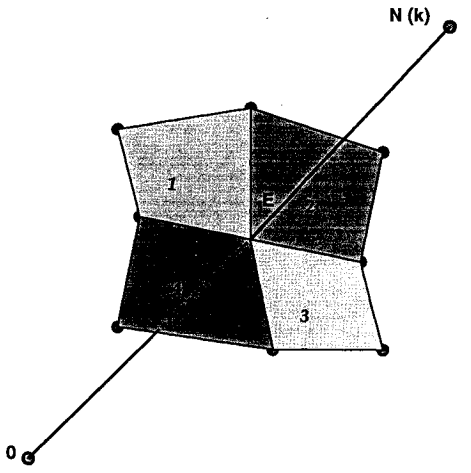


Fig. 2 Dual mesh faces attached to an edge.

where $\tilde{A}_r$ is Roe's matrix.[13] The dissipation terms are modeled so as to be similar to the second term of the preceding equation since this corresponds to the implicit smoothing term of the upwinding scheme. A simplified form of Eq. (10) is obtained by replacing $\tilde{A}$ with $\rho(\tilde{A}) = |u| + c$, the maximum eigenvalue of Roe's matrix. This ensures that the dissipation terms do not dwindle down to zero near the stagnation or the sonic points. The contribution $(\delta U_0^n)_{s2}$ of shock-smoothing terms to the change δU_0^n at the node 0 is given as follows:

$$(\delta U_0^n)_{s2} = \frac{\Delta t}{\Omega_0} \sum_{k=1}^{n} [\rho(\tilde{A}_r)S_x + \rho(\tilde{B}_r)S_y + \rho(\tilde{C}_r)S_z]_k (U_{N(k)} - U_0) \tag{11}$$

where $\rho(\tilde{A}_r)$, $\rho(\tilde{B}_r)$, and $\rho(\tilde{C}_r)$ are the spectral radii of Roe's matrices corresponding to the flux vectors $\boldsymbol{F}$, $\boldsymbol{G}$, and $\boldsymbol{H}$ evaluated at the middle of the edge k. The eigenvalues are evaluated at the Roe averaged quantities.[14] The shock-smoothing term is evaluated similar to the convective fluxes on an edge-wise basis. The fourth-order smoothing contribution $(\delta U_0^n)_{s4}$ is computed in a similar fashion. Instead of the first difference of state vectors as used in Eq. (11), a difference of the accumulated first difference over the edges sharing a node is used for background smoothing.

The change $(\delta U_0^n)_s$ at the node 0 due to second- and fourth-order smoothing is given by

$$(\delta U_0^n)_s = \sigma_2(\Delta P_0)(\delta U_0^n)_{s2} + \sigma_4(1 - \Delta P_0)(\delta U_0^n)_{s4} \tag{12}$$

The pressure switch ΔP is used to turn on the shock smoothing and the background smoothing at the appropriate regions. The coefficients σ_2 and σ_4 are empirical parameters that control the amount of shock and background smoothing.

III. Adaptive Grid Refinement

A dynamic grid adaptation algorithm has been previously developed for three-dimensional unstructured grids.[10] The algorithm is capable of simultaneously unrefining and refining the appropriate regions of the flow domain by detecting the local flow features. A feature detector senses the flow *features* that are present in different regions and guides the adaptive algorithm to refine or unrefine the grid, ensuring that sufficient grid resolution exists locally near the regions of significant flow variations. Velocity differences and gradients are used as detection parameters and threshold values are set for these parameters based on their distribution over the flow domain.[17]

The different strategies usually employed for embedding a tetrahedron and the methods of cell division employed in the adaptive algorithm used in the present work are discussed in detail in Refs. 10 and 15. The tetrahedra that are flagged for refinement are embedded by the octree cell division that divides the parent cell into eight children, as shown in Fig. 3, by inserting mid-edge nodes on the parent cell edges. The four corner child cells are similar to the parent cell. The four interior child cells are formed by dividing the interior octahedron, constituted by the nodes 5-6-7-8-9-10, by the shortest diagonal. Directional division of cells is also employed to divide "transition" cells at the border between different embedded regions that contain "hanging" nodes in the middle of some of their edges due to refinement of neighboring cells. This includes cases in which the mid-edge nodes appear on the same face or on only one of the six edges.

IV. Multigrid Method

A multigrid method for hyperbolic equations using structured grids was presented by Ni.[5] This uses the coarser grids to propagate, properly and more rapidly, the solution changes-in-time δU obtained based upon the discretization of the governing equations on the finest grid. A series of overlapping grids of varying spatial resolution is used. The fine grid calculations are performed to compute the changes to the solution at each node. These changes are transferred to the nodes of the coarse grid. The transferred changes are propagated on the coarse grid and henceforth create the coarse grid corrections. These corrections are then interpolated back to the fine grid nodes.

A. Coarse Grids for Multigrid

In the present work, the multigrid scheme is implemented in tandem with the grid adaptation scheme. A major hurdle in the implementation of the multigrid scheme using the unstructured grids is the generation of a sequence of successively coarser grids. In the structured grid approach, this is easily accomplished by removing every other line of grid points in each coordinate direction. This is not possible with the unstructured grids since there is no inherent order. The previous multigrid methods for unstructured grids[7,8] generate a sequence of completely independent, successively coarser and non-overlapping grids as shown in Fig. 4. The figure shows fine grid cells as well as coarse grid cells. A search algorithm is required to generate the data structures that determine the nodes A, B, and C to be associated with each node a on the next finer level and similarly the nodes 1, 2 and 3 to be associated with each node P on the next coarser level. Using these data structures, the multigrid scheme performs intergrid transfer of residuals and state variables. This method is intensive in terms of computing time and memory. Furthermore, generation of the successively coarser grids is quite difficult.

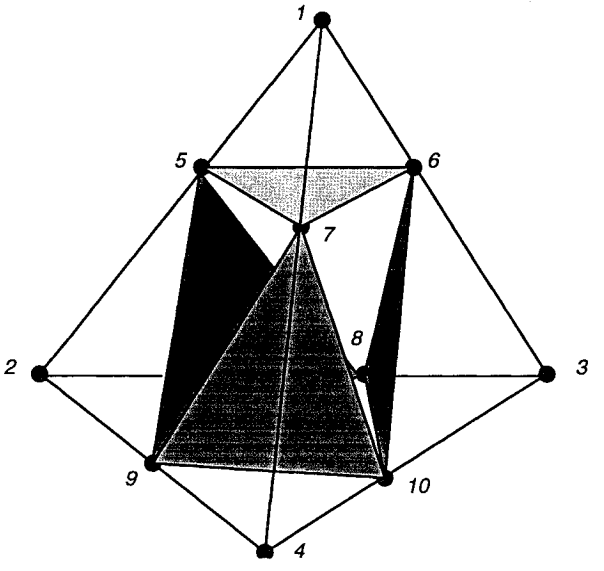


Fig. 3 Isotropic division of a tetrahedral cell into eight subcells.

The salient feature of the present work is that it circumvents the need for such complicated search algorithms and data structures to determine the intergrid communication by virtue of using overlapping coarse grids. This is illustrated in Fig. 5, which shows an example of a sequence of two unstructured two-dimensional grids obtained by adapting an initial grid once. The once-embedded grid (level 1) constitutes the fine grid on which the basic numerical scheme is implemented, and the other grid, namely, the unembedded initial grid (level 0), constitutes the coarser grid that shall be used to obtain the convergence acceleration. It must be noted that the level 0 grid is obtained directly out of the level 1 grid by using the parent cell pointers of the adaptive algorithm that give the successively coarser cells. The cells that constitute the unembedded zone of the fine (level 1) grid are included in the coarser level grid as well. Each time the grid is adaptively refined, an additional level is added to the multigrid system.

B. Coarse Grid Acceleration

The basic concept of the present multigrid method is that the corrections to the fine grid solution's changes-in-time obtained on the coarse grid should be driven by the fine grid residuals. The multigrid scheme is implemented by first discretizing the governing equations on the finest level. This consists of evaluating the changes-in-time of the solution at each fine grid node using the basic numerical scheme. This constitutes one *fine grid iteration.* The fine grid changes are then transferred to the coarse grid nodes. The coarse grid corrections are obtained by redistributing the transferred changes on the coarse grid nodes. These coarse grid corrections are then interpolated to the fine grid nodes. This set of operations constitutes a *coarse grid iteration.* In the present scheme a coarse grid iteration is performed after every fine grid iteration. Each *multigrid cycle* consists of a fine grid iteration and an iteration on each coarser level grid.

Consider that the coarse grid acceleration scheme involves an advance in time $\Delta t^{n+1} = t^{n+1} - t^n$. The Taylor series expansion in time for the correction $\delta \boldsymbol{U}^c$ obtained on a coarse grid node is written as

$$(\delta \boldsymbol{U}^c)^{n+1} = \boldsymbol{U}^{n+1} - \boldsymbol{U}^n = \left(\frac{\partial \boldsymbol{U}}{\partial t}\right)^n \Delta t^{n+1} + \left(\frac{\partial^2 \boldsymbol{U}}{\partial t^2}\right)^n \frac{(\Delta t^{n+1})^2}{2} \quad (13)$$

For the previous time step on the next finer grid, the expansion for the change in time of the solution $\delta \boldsymbol{U}^f$ obtained at that node is given by

$$(\delta \boldsymbol{U}^f)^n = \boldsymbol{U}^n - \boldsymbol{U}^{n-1} = \left(\frac{\partial \boldsymbol{U}}{\partial t}\right)^n \Delta t^n - \left(\frac{\partial^2 \boldsymbol{U}}{\partial t^2}\right)^n \frac{(\Delta t^n)^2}{2} \quad (14)$$

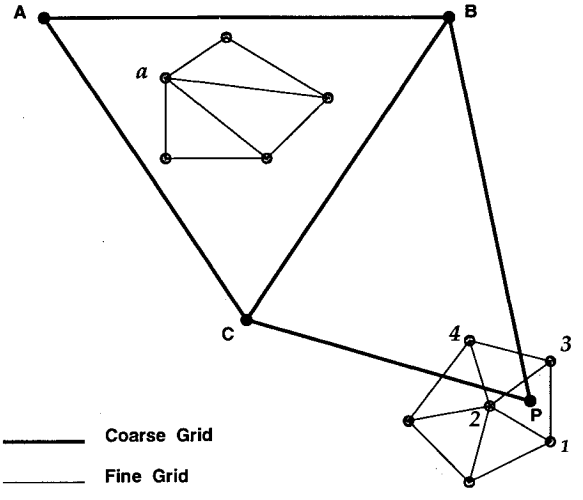


Fig. 4 Nonoverlapping grids used for multigrid scheme.

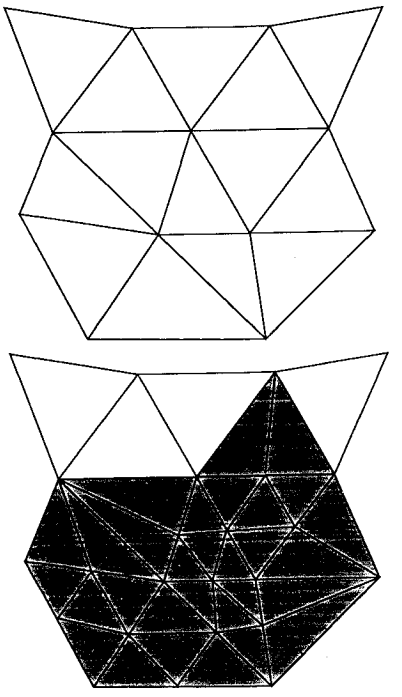


Fig. 5 Sequence of overlapping grids for the multigrid scheme, generated by the adapter, level 0 and level 1, respectively (shaded area denotes embedded region of the grid).

The coarse grid time step Δt^{n+1} and the fine grid time step Δt^n are alternatively represented by Δt^c and Δt^f, respectively. Combining the preceding equations, the correction obtained at the coarse grid node is expressed in terms of the fine grid change associated with that node as

$$(\delta \boldsymbol{U}^c)^{n+1} = (\delta \boldsymbol{U}^f)\frac{\Delta t^c}{\Delta t^f} - \frac{1}{2}\left(\frac{\partial^2 \boldsymbol{U}}{\partial t^2}\right)^n (\Delta t^c + \Delta t^f)\Delta t^c \quad (15)$$

The first term on the right-hand side in Eq. (15) represents the first-order part of the coarse grid correction. To compute this term, the change in time of the solution $\delta \boldsymbol{U}^f_{\text{node}}$ evaluated at that node in the previous time step on the finer grid is required. Instead of using a simple injection procedure, this term is evaluated by transferring to the node at the coarser level the fine grid changes associated with that node and its neighbors by means of a weighted averaging procedure. Referring to Fig. 6, the transferred change $\delta \boldsymbol{U}^{f\to c}_{\text{node}}$ at node 1 on the coarse grid is given as follows:

$$(\delta \boldsymbol{U}^{f\to c}_1)^n = \left[\frac{\Sigma_{\text{cell}=A,B,\ldots,E}(\delta \boldsymbol{U}^f_{\text{cell}})^n V_{\text{cell}}}{\Sigma_{\text{cell}=A,B,\ldots,E} V_{\text{cell}}}\right] \quad (16)$$

where $(\delta \boldsymbol{U}^{f\to c}_{\text{cell}})^n$ represents the average change in time of the solution computed at the previous time step for a cell of volume V_{cell} on the next finer level. For example, referring to Fig. 6, for the fine grid cell A we have $\delta \boldsymbol{U}^f_A = (1/3)(\delta \boldsymbol{U}^f_1 + \delta \boldsymbol{U}^f_2 + \delta \boldsymbol{U}^f_9)$. The first-order part $(\Delta \boldsymbol{U}^c_{\text{node}})^{n+1}$ of the coarse grid correction $\delta \boldsymbol{U}^c_{\text{node}}$ at any node is hence given by

$$(\Delta \boldsymbol{U}^c_{\text{node}})^{n+1} \equiv (\delta \boldsymbol{U}^{f\to c}_{\text{node}})^n \frac{\Delta t^c}{\Delta t^f} \quad (17)$$

To compute the second-order part of the coarse grid correction at the node, the temporal derivative $(\partial^2 \boldsymbol{U}/\partial t^2)$ in Eq. (15) is computed at the coarse grid nodes. Using the governing equation (1) to

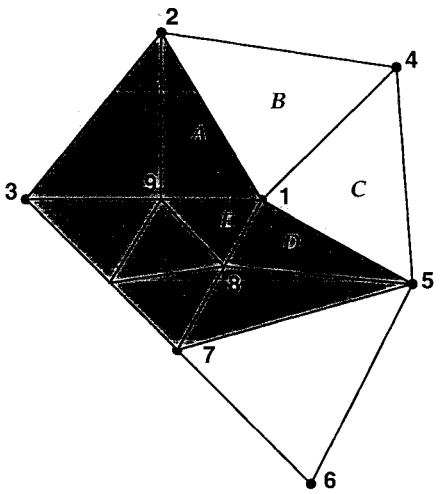


Fig. 6 Weighted averaging of the fine grid changes to obtain the coarse grid correction at node 1 (two-dimensional example).

replace the temporal derivatives by spatial derivatives and using the divergence theorem, we have for any coarse grid node

$$\left(\frac{\partial^2 \boldsymbol{U}}{\partial t^2}\right)^n_{\text{node}} \Omega_{\text{node}} = -\frac{1}{\Delta t^n}\int_{\partial\Omega_{\text{node}}} (\Delta \boldsymbol{F} n_x + \Delta \boldsymbol{G} n_y + \Delta \boldsymbol{H} n_z)^c_{\text{node}}\, \mathrm{d}S \tag{18}$$

where

$$\Delta \boldsymbol{F} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}}(\Delta \boldsymbol{U}^c_{\text{node}}), \quad \Delta \boldsymbol{G} = \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}(\Delta \boldsymbol{U}^c_{\text{node}}), \quad \Delta \boldsymbol{H} = \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{U}}(\Delta \boldsymbol{U}^c_{\text{node}})$$

The terms $\Delta \boldsymbol{F}$, $\Delta \boldsymbol{G}$, and $\Delta \boldsymbol{H}$ are computed at the coarse grid node using the first-order change $(\Delta \boldsymbol{U}^c_{\text{node}})^{n+1}$ evaluated at the node from Eq. (17).

No smoothing is applied on the coarser grids. The boundary conditions are applied at the fine grid nodes after each fine grid and coarse grid iteration.

The corrections obtained at the coarse grid nodes from Eq. (15) are interpolated to the exclusive fine grid nodes (fine grid nodes that do not appear on the coarser grid). This interpolation is easily accomplished on an edgewise basis by using the edge pointers that give the child edges of any coarse grid edge. For example, referring to Fig. 3, the tetrahedron 1-2-3-4 represents a coarse cell, and its child cells constitute the next finer grid cells. The coarse grid correction for the node 7 is obtained by linearly interpolating the corrections obtained at nodes 1 and 4. The sequence of operations for the multigrid algorithm is given as follows:

1) Compute the changes-in-time to the solution $(\delta \boldsymbol{U}^f_{\text{node}})$ at the finest grid nodes.

2) Transfer the fine grid changes to the next coarser level grid.

3) Redistribute these changes to the coarser level nodes to obtain the coarse grid corrections $(\delta \boldsymbol{U}^c_{\text{node}})$.

4) Interpolate the coarse grid corrections to the finer grid nodes to be added to the fine grid changes previously computed.

If the multigrid method employs more than one coarse level, then the solution changes-in-time on the fine grid that were corrected by the first coarse level contributions are transferred to the next coarser level and the same sequence of operations outlined earlier are performed to obtain the corrections on the next coarser level. Figure 7 shows the flow of operations for a multigrid cycle with two coarser level grids. This can be generalized to a multigrid scheme with n coarser levels.

V. Results

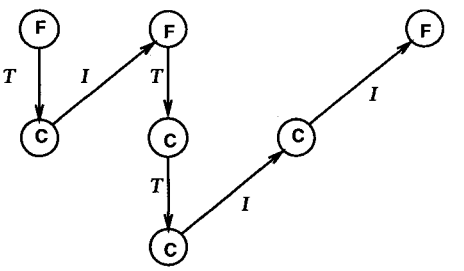
A. Adaptive Solution on an ONERA M6 Wing

The adaptive solution obtained for the transonic flow at M_∞ = 0.84 past an ONERA M6 wing at an angle of attack of 3.06 deg is shown here to validate the accuracy of the solution obtained with the basic numerical scheme employed. Computations were performed on a CRAY Y-MP. The initial grid is comprised of 231,507 cells and 42,410 nodes. The solutions are started from the freestream conditions being specified everywhere. The initial grid is adaptively embedded in the regions of the local flow features. Velocity differences and the velocity gradients are used as the detection parameters. The triangulation on the wing upper surface of the adapted grid is shown in Fig. 8 with the embedded regions of the grid denoted with the darker shades. The adapted grid has 833,613 cells and 144,722 nodes.

The solution obtained on the adapted grid is shown in Fig. 8. Mach number contour lines on the upper surface of the wing are plotted using an increment of $\Delta M = 0.02$. The contours show the λ shock standing on the upper surface of the wing. Both the fore and the aft shocks are seen to be captured sharply. The pressure coefficients comparison of the adapted grid solution with the experimental observations is shown in Fig. 9 at two spanwise locations. It is seen that there is a good agreement of the computed solution with the experimental observations. The leading-edge suction peak is captured well. It is observed that, at station $\eta = 0.44$, the aft shock location in the final solution is downstream of the experimentally measured location. This discrepancy has been observed in solutions with other Euler schemes.[19,20] The pressure coefficient distribution on the lower surface of the wing is also seen to be in good agreement with the experimental results.

B. ONERA M6 Wing—One-Level Multigrid

Solution for transonic flow at M_∞ = 0.84 is obtained using the multigrid technique. The initial grid employed in this case comprises 35,008 cells and 6910 nodes. The grid is adapted once, and the solution is obtained by using coarse grid acceleration. Figure



T - Transferring the changes from a finer to a coarser level

I - Interpolating the corrections from a coarser to a finer level

Fig. 7 Sequence of operations for a two-level multigrid scheme.

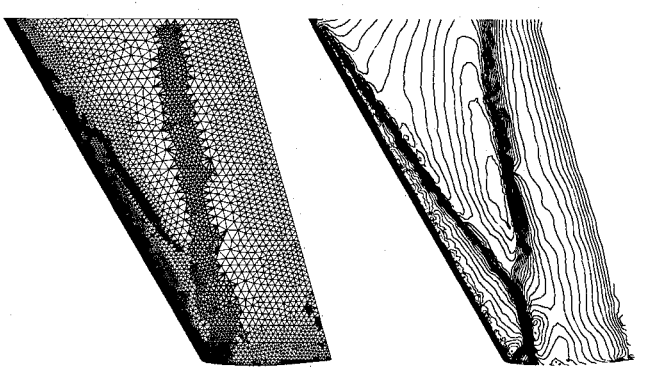


Fig. 8 Triangulation and the Mach number contour lines on the wing upper surface. Transonic flow solution past an ONERA M6 wing (M_∞ = 0.84) obtained on the once-adapted grid. Mach number contour lines plotted with an increment of ΔM = 0.02.

10 shows the triangulation on the wing upper surface of the embedded fine grid and the initial coarse grid that constitute the multigrid system. The finest level grid comprises 150,333 cells and 27,308 nodes. The coarser grid has approximately one-fourth the number of fine grid points. Comparison of the convergence histories for the solution obtained by performing only the fine grid iterations without any acceleration and the one obtained with coarse grid acceleration is shown in Fig. 11. The coarse grid acceleration with the one-level multigrid reduces the number of fine grid iterations by a factor of 2.5. Computing time comparisons made for the one-level multigrid scheme on a Cray Y-MP are shown in Table 1. The coarse grid acceleration reduced the CPU time roughly by a factor of 2.00.

Table 1 CPU time comparison for multigrid solution on an ONERA M6 wing (M_∞ = 0.84)

Time taken for one fine grid iteration, s	2.52	Convergence in 4045 fine grid iterations
Time taken for one multigrid cycle with one coarser level, s	3.02	Convergence in 1742 multigrid cycles

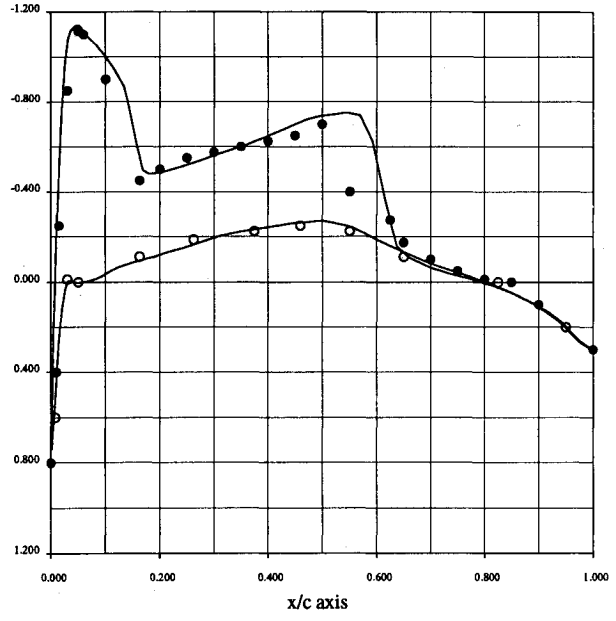


Fig. 9 Pressure coefficients comparison on the wing surface at η = 0.90 spanwise location; •, experimental values (upper surface); o, experimental values (lower surface); ——, one-level adapted grid.

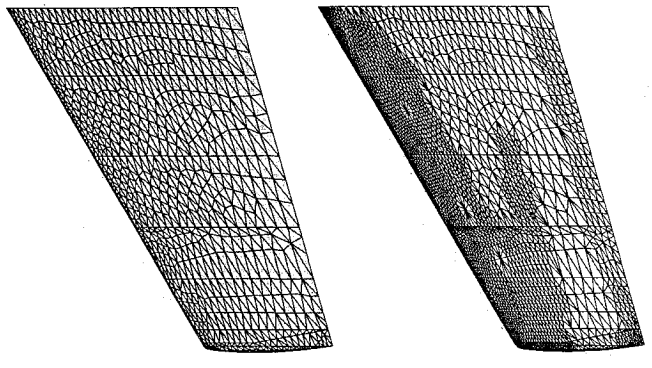


Fig. 10 Upper wing surface of the overlapping set of meshes used for multigrid acceleration. Transonic flow past an ONERA M6 wing (M_∞ = 0.84).

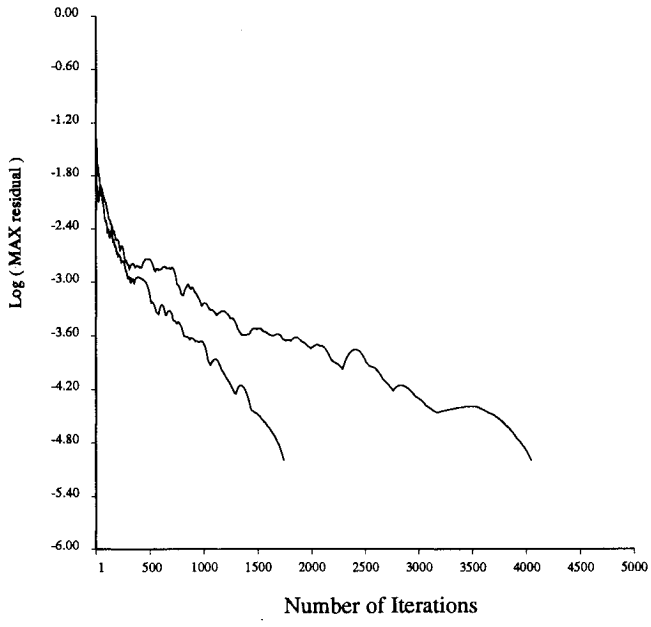


Fig. 11 Comparison of the convergence rates. Convergence without multigrid in 4045 fine grid iterations. Convergence with multigrid acceleration in 1742 multigrid cycles.

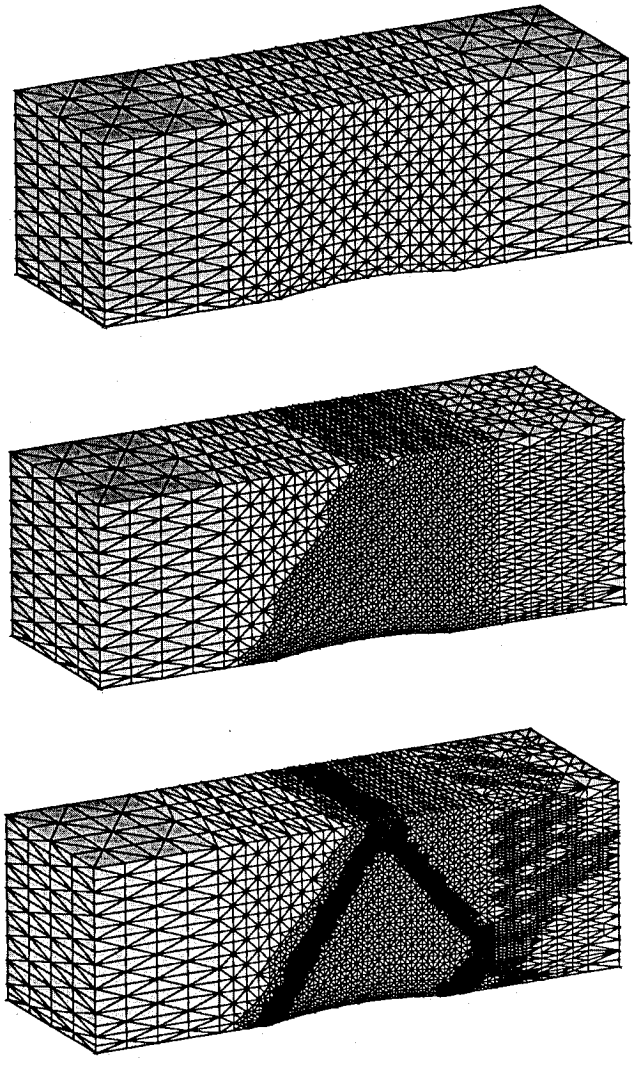


Fig. 12 Isometric view of the sequence of grids generated by successive passes of adaptation, employed by the two-level multigrid scheme.

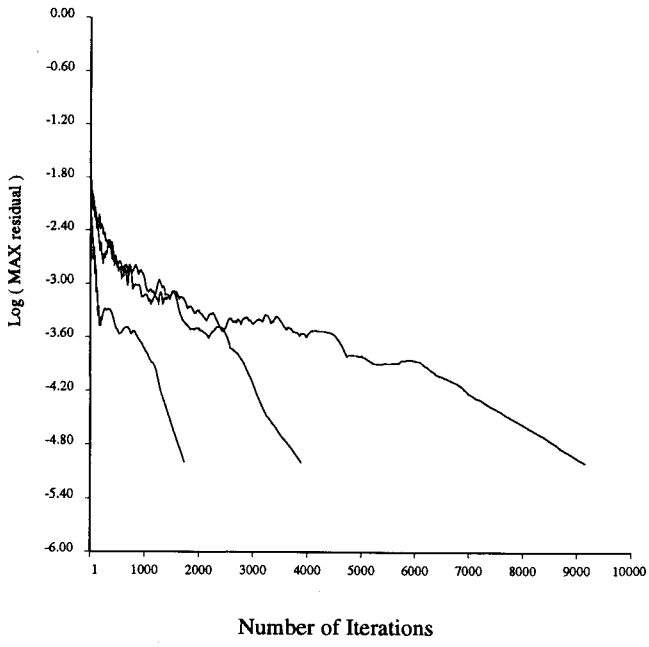


Fig. 13 Comparison of the convergence rates. Convergence without multigrid in 9151 fine grid iterations. Convergence with one-level multigrid acceleration in 3885 multigrid cycles. Converegence with two-level multigrid acceleration in 1730 multigrid cycles.

C. Supersonic Flow in a Duct—Two Level Multigrid

The solution for supersonic flow at a Mach number of 1.4 in a channel with a 4% bump is obtained using adaptive grid embedding coupled with coarse grid acceleration. Isometric views of the sequence of overlapping grids that are obtained at successive passes of adaptation are shown in Fig. 12.

The adapted grids obtained constitutes a two-level multigrid with the finest level represented by the twice-embedded grid and the two coarser levels represented by the once-embedded and the unembedded initial grids, respectively. The initial grid comprises 10,240 cells and 2805 nodes and constitutes the coarsest level. The once-embeddeded grid comprises 60,325 cells and 12,756 nodes The finest level of the multigrid system on which the basic numerical scheme is implemented comprises of 206,789 cells and 38,430 nodes.

Comparison of the convergence histories for the solution obtained by three cases, namely, no multigrid acceleration, one-level multigrid acceleration, and two-level multigrid acceleration, is shown in Fig. 13. It is seen that with multigrid acceleration, the number of fine grid iterations required is reduced by about 5.3 times. The coarsest level grid has roughly one-fourteenth the number of grid points at the finest level, and the next finer level has roughly one-third the number of nodes at the finest level. Hence, the coarse grid iterations are significantly less expensive compared with the fine grid iteration. In view of this, the reduction brought forth in the number of fine grid iterations is seen to be considerable. The effect of employing the two-level coarse grid acceleration is easily seen from the progressive steepening of the residual history curves. It is also observed from the figure that the effect of the multigrid acceleration is not very pronounced in the initial stages during which the high- frequency components are predominant. Convergence acceleration is seen to markedly improve once the low-frequency components become more dominant.

CPU time comparisons made on a Cray Y-MP as well as comparison of convergence acceleration obtained with different levels of multigrid are illustrated in Fig. 14. Figure 14a shows that there is a significant speed-up in convergence in terms of the number of iterations. Figure 14b shows that the computing time per iteration increases only by a small factor for every additional level of grid used. Hence, cumulatively these two factors obtain a significant speed-up in terms of computing time. It is seen that multigrid acceleration with just one coarse grid level reduces the total computing time roughly by a factor of 2. Using two coarser levels in the multigrid brings about a reduction by a factor of 4.2 for this case.

D. Subsonic Flow in a Duct—One-Level Multigrid

Solution is obtained for subsonic flow at a Mach number of 0.5 in the duct coupled with coarse grid acceleration using a one-level multigrid. Planar sections of the once-embedded grid at midspan are shown in Fig. 15. The initial grid comprises 10,240 cells and 2805 nodes and constitutes the coarser level, whereas the embedded grid with 58,084 cells and 10,946 nodes constitutes the fine grid.

Comparison of the convergence histories for the solution obtained without multigrid acceleration and the one obtained with coarse grid acceleration is shown in Fig. 16. It is seen that with multigrid acceleration, the number of fine grid iterations required is reduced by about 2.55 times. The number of grid points on the coarse grid is roughly one-fourth the number of fine grid points. Computations time comparisons were made on an IBM-375 RISC/6000 and are shown in Table 2. The coarse grid acceleration with the one-level multigrid reduced the CPU time by a factor of 2.35.

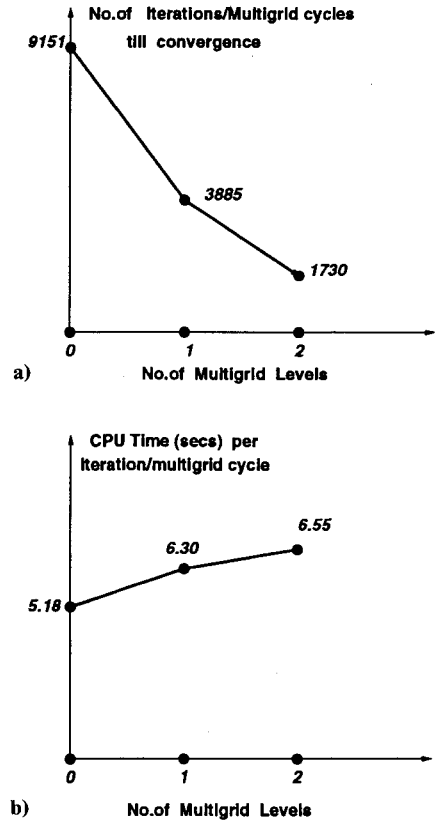


Fig. 14 Multigrid efficiency: a) reduction in the total number of iterations and b) computing time required for each additional level of multigrid.

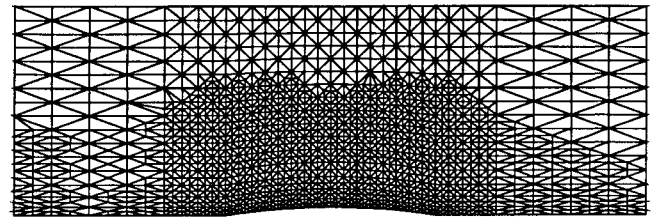


Fig. 15 Planar section, at midspan, of the once-embedded mesh used for the multigrid. Subsonic flow over a 4% bump in a channel at M• = 0.5.

Table 2 CPU time comparison for one-level multigrid solution for subsonic flow past a bump in a channel ($M_\infty = 0.5$)

Time taken for one fine grid iteration, s	4.00	Convergence in 3452 fine grid iterations
Time taken for one multigrid cycle with one coarser level, s	4.35	Convergence in 1364 cycles

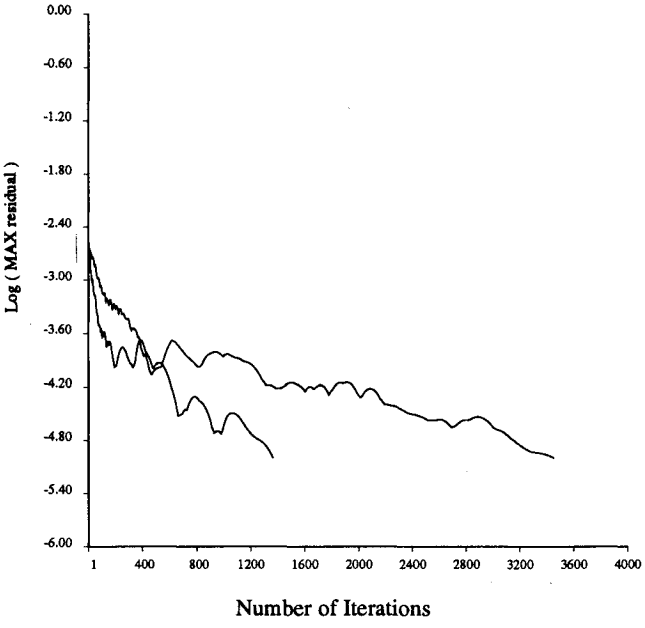


Fig. 16 Comparison of the convergence rates. Convergence without multigrid in 3452 fine grid iterations. Convergence with one-level multigrid acceleration in 1364 multigrid cycles.

Conclusions

A new multigrid approach for unstructured grids has been presented. The multigrid method was shown to obtain significant speed-up in computing time as well as convergence rates. The multigrid scheme avoided generation of independent coarser grids and also circumvented the need for search algorithms and extensive data structures to interpolate solution and residuals between the grids. The interpolation of solution and residuals between the grids was easily done since the grids employed for the multigrid scheme are overlapping. The multigrid technique was implemented in tandem with a grid adaptation scheme that enabled successively coarser levels to be generated conveniently from the parent cells of the adapted grid. The basic numerical scheme coupled with the multigrid acceleration has been shown to be robust by applying it to different flow situations in all three flow regimes, namely, subsonic, transonic, and supersonic.

Acknowledgments

This work was supported by DARPA Grant DABT 63-92-0042, and NASA Grant NAG1-1459. Supercomputing time was provided by the NAS Division of NASA Ames Research Center and the National Science Foundation Pittsburgh Supercomputing Center.

References

[1]Baker, T. J., "Three Dimensional Mesh Generation by Triangulation of Arbitrary Point Sets," AIAA Paper 87-1124, June 1987.

[2]Brandt, A., "Multilevel Adaptive Computations in Fluid Dynamics," *AIAA Journal*, Vol. 18, No. 10, 1980, pp. 1165–1172.

[3]Hackbush, W., "Multigrid Convergence Theory," *Multigrid Methods*, Lecture Notes in Mathematics, Springer-Verlag, Berlin, 1982, pp. 177–219.

[4]Jameson, A., "Solution of the Euler Equations for Two-Dimensional Transonic Flow by a Multigrid Method," *Applied Mathematics and Computation*, Vol. 13, June 1983, pp. 327–356.

[5]Ni, R. H., "A Multiple Grid Scheme for Solving the Euler Equations," *AIAA Journal*, Vol. 20, No. 11, 1982, pp. 1565–1571.

[6]Mavriplis, D. J., and Jameson, A., "Multigrid Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes," AIAA Paper 87-0353, Jan. 1987.

[7]Mavriplis, D. J., "Three Dimensional Unstructured Multigrid for the Euler Equations," AIAA Paper 91-1549, June 1991.

[8]Peraire, J., Peiro, J., and Morgan, K., "A 3D Finite Element Multigrid Solver for the Euler Equations," AIAA Paper 92-0449, Jan. 1992.

[9]Lohner, R., and Baum, J., "Numerical Simulation of Shock Interaction with Complex Geometry Three-Dimensional Structures Using a New Adaptive H-Refinement Scheme on Unstructured Grids ," AIAA Paper 90-0700, Jan. 1990.

[10]Kallinderis, Y., and Parthasarathy, V., "An Adaptive Refinement Coarsening Scheme for 3-D Unstructured Meshes," *AIAA Journal*, Vol. 31, No. 8, 1993, pp. 1440–1447.

[11]Rausch, R. D., Batina, J. T., and Yang, H. T. Y., "Spatial Adaptation Procedures for Tetrahedral Meshes for Unsteady Aerodynamic Flow Calculations," AIAA Paper 93-0670, Jan. 1993.

[12]Biswas, R., and Strawn, R., "A New Procedure for Dynamic Adaptation of Three Dimensional Unstructured Grids," AIAA Paper 93-0672, Jan. 1993.

[13]Roe, P. L., "Characteristic-Based Schemes for the Euler Equations," *Annual Review of Fluid Mechanics*, Vol. 18, 1986, pp. 337–365.

[14]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357-373.

[15]Parthasarathy, V., "A Dynamic Grid Adaptation Scheme for 3-D Unstructured Grids," M.S. Thesis, Dept. of Aerospace Engineering and Engineering Mechanics, Univ. of Texas at Austin, Austin, TX, May 1992.

[16]Kallinderis, Y., Parthasarathy, V., and Wu, J., "A New Euler Scheme and Adaptive Refinement Coarsening Algorithm for Tetrahedral Grids," AIAA Paper 92-0446, Jan. 1992.

[17]Kallinderis, J. G., and Baron, J. R., "Adaptation Methods for a New Navier-Stokes Algorithm," *Proceedings of the AIAA 8th Computational Fluid Dynamics Conference* (Honolulu, HI), AIAA, New York, 1987 (AIAA Paper 87-1167); see also *AIAA Journal*, Vol. 27, No. 1, 1989, pp. 37–43.

[18]Hall, M. G., "Cell Vertex Multigrid Schemes for Solution of the Euler Equations," *Proceedings of the Conference on Numerical Methods for Fluid Dynamics* (Reading, England, UK), Oxford Univ. Press, Oxford, England, UK, 1985.

[19]Frink, N. T., Parikh, P., and Pirzadeh, S., "A Fast Upwind Solver for the Euler Equations on Three-Dimensional Unstructured Meshes," AIAA Paper 91-0102, Jan. 1991.

[20]Batina, J. T., "Accuracy of an Unstructured Grid Upwind-Euler Algorithm for the ONERA M6 Wing," Accuracy of Unstructured Grid Techniques Workshop, NASA Langley Research Center, Hampton, VA, Jan. 1990.