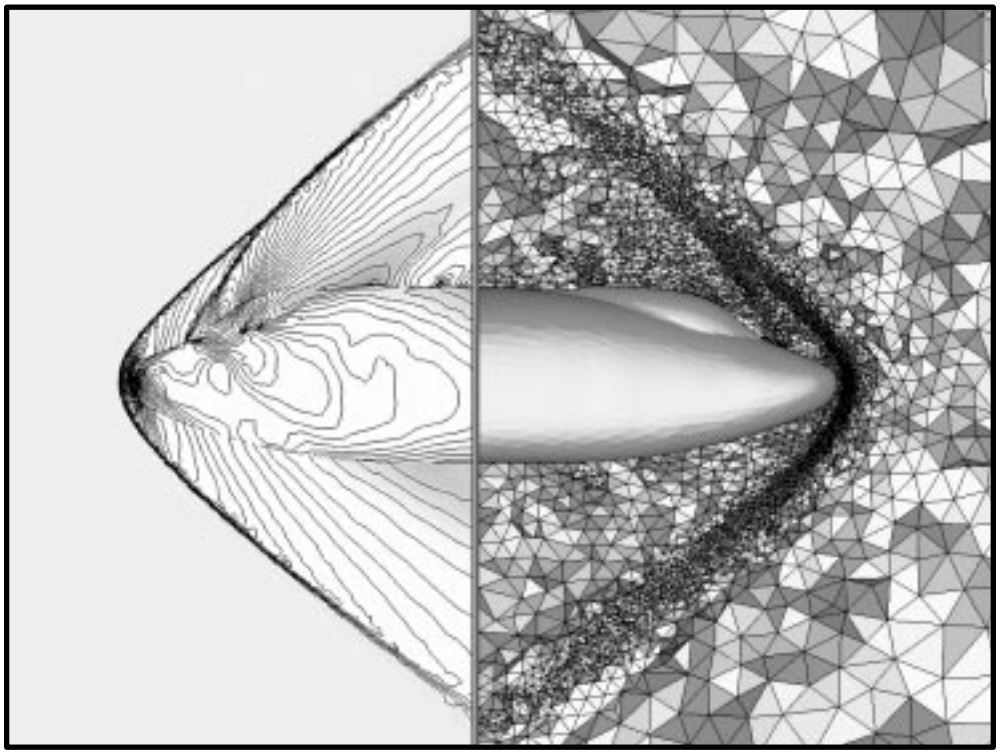
AIAA 99−3255

AN ADAPTIVE UNSTRUCTURED GRID METHOD BY GRID SUBDIVISION, LOCAL REMESHING, AND GRID MOVEMENT

S.Z. Pirzadeh
Configuration Aerodynamics Branch
MS 499, NASA Langley Research Center
Hampton, VA 23681−2199



14th AIAA Computational Fluid Dynamics Conference
June 28−July 1, 1999 / Norfolk, Virginia

AN ADAPTIVE UNSTRUCTURED GRID METHOD BY GRID SUBDIVISION, LOCAL REMESHING, AND GRID MOVEMENT

Shahyar Z. Pirzadeh*
Configuration Aerodynamics Branch
MS 499, NASA Langley Research Center
Hampton, VA 23681−2199

Abstract

An unstructured grid adaptation technique has been developed and successfully applied to several three dimensional inviscid flow test cases. The approach is based on a combination of grid subdivision, local remeshing, and grid movement. For solution adaptive grids, the surface triangulation is locally refined by grid subdivision, and the tetrahedral grid in the field is partially remeshed at locations of dominant flow features. A grid redistribution strategy is employed for geometric adaptation of volume grids to moving or deforming surfaces. The method is automatic and fast and is designed for modular coupling with different solvers. Several steady state test cases with different inviscid flow features were tested for grid/solution adaptation. In all cases, the dominant flow features, such as shocks and vortices, were accurately and efficiently predicted with the present approach. A new and robust method of moving tetrahedral "viscous" grids is also presented and demonstrated on a three−dimensional example.

Introduction

Generation of good quality computational grids on complex geometries remains a difficult task (or even a limiting factor) for the routine and timely application of Computational Fluid Dynamics (CFD). In addition, generation of an efficient grid usually requires some prior knowledge of the flow behavior in order to match the grid resolution to the essential features of the flowfield. While such information may not be always available in advance, a number of "trial−and−error" iterations between the solution and grid generation are often required to tailor the grid to the specifications of the problem at hand. Alternatively, an overly fine grid is usually generated to guarantee the desired solution accuracy. In both cases, the amount of time, effort, and computational resources may become excessive.

The solution adaptive grid technology is a powerful tool in CFD which alleviates some of the complexities associated with the generation of high quality grids. Although grid adaptation does not address the crucial and time−consuming issues of geometry set up, domain decomposition (for structured grids), etc., they automate the process of adjusting the grid resolution to the flow and the geometric characteristics of the problem. Since the distribution of grid points is efficiently determined by the adaptation process, the final adapted grid contains a far fewer number of points than an unadapted grid with similar resolution at the desired locations. This important feature of grid adaptation results in a substantial savings in the amount of computational time and memory requirement.

In general, most adaptive methods fall into three broad categories: 1) grid movement (r−refinement), 2) grid enrichment (h−refinement), and 3) local solution enhancement (p−refinement). While the methods in the first two classes modify local grid clustering in order to improve the solution accuracy (grid adaptation), those under the third category enhance the order of numerical approximation at locations where the solution undergoes abrupt variations (solution adaptation). Most adaptive techniques used in the CFD applications fall into the first two classes. In the present work, several grid adaptation methods are employed in order to exploit their advantages collectively. A brief discussion of these techniques are presented here as follows.

In the grid movement approach, nodes are redistributed and moved towards regions where higher degree of accuracy is needed. Since the grid topology remains unchanged throughout the grid adaptation, the process of grid movement can be simply incorporated into the solver in a modular fashion. Also, no data transfer (e.g., interpolation) is required since the grid structure remains intact during the process, and thus no solution accuracy is lost from one adaptation cycle to the next. The method is especially advantageous for the transient problems involving moving surfaces. However, since the number of grid nodes remains constant, transferring nodes from one part of the grid to another may cause local "depletion" of grid elements, and thus severe distortion of the grid may be introduced.[1] Adaptation by grid movement has mainly been applied to structured grids and also 2D triangular meshes. In this work, grid movement is employed for the geometric adaptation of grids only.

In the grid enrichment technique, more nodes are added to regions where higher accuracy of the solution is desired. Nodes can also be removed from locations where the solution is smooth and requires less grid resolution. Due to node addition (deletion), the topology (connectivity) of the grid changes from one adaptation cycle to another. As a result, some type of data transfer among the consecutive grids is required. Adaptive methods by grid enrichment are particularly attractive for their flexibility, especially when applied in conjunction with unstructured grids.

Among the adaptive grid methods by enrichment, two techniques are notable: grid subdivision and grid

*Senior Research Engineer, Senior Member AIAA.

remeshing. In grid subdivision, new nodes are added to the edges of the cells identified for refinement, and the "parent" cells are divided into several smaller cells. The method is efficient and fast, and once a systematic data structure is maintained prior to the adaptation cycles, both refinement and de−refinement can easily be implemented. Also, the transfer of data from the "parent" to "children" cells (and vice versa) can be accomplished conveniently. The grid subdivision methods have been best demonstrated on Cartesian meshes[2] and can also be readily implemented in triangular grids. However, their applications to tetrahedral grids involve complex data structures and possibly grid distortion.[3]

Global and partial remeshing have also been successfully employed for adaptive grid refinement.[4,5] Two significant advantages of these methods are: 1) flexibility for refinement and unlimited coarsening (in subdivision methods, for example, grids cannot be de−refined beyond the initial grid resolutions) and 2) good quality grids generated in each refinement cycle. Since several grids are generated during the adaptation process, the grid generation time and the cost of solution interpolations are extensive in these methods, especially in the global remeshing. Also, methods by global remeshing lack the desired degree of automation as the processes of grid generation and flow solution remain uncoupled.

As there is no single grid type (e.g., structured, unstructured, etc.) or generation method suitable for all classes of computational problems, there neither exists an individual adaptive method which can be universally applied to a wide variety of problems. Different methods offer certain advantages to different classes of grids and problems.[2−5] For example, methods employing h−refinement are well suited for Cartesian and 2D triangular grids; mesh refinement based on local remeshing can be easily applied to unstructured grids due to their inherent flexibility; methods based on grid movement are ideal for transient problems involving moving bodies, etc. Therefore, it is beneficial to exploit the advantages of several techniques in a hybrid adaptive grid method for solving complex problems.[1] In this work, an attempt has been made to combine the efficiency of h−refinement, the flexibility of remeshing, and the convenience of grid movement for resolving different aspects of the grid adaptation problem.

Approach

The proposed grid adaptation strategy is summarized in the flow chart shown in Fig. 1. Two distinct approaches are taken in the present method: 1) solution adaptive grid refinement, which is presented on the right hand side of the flow chart, and 2) grid adaptation to moving geometries, shown on the left. The method employs a local remeshing technique to refine (coarsen) unstructured tetrahedral "Euler" grids for adaptation to the flow. The surface triangulation is partially "h−refined" as part of the local remeshing. A mesh movement strategy is used to perturb "viscous" and/or "Euler" tetrahedral grids for adaptation to deforming or moving surfaces.

There are two main components in any adaptation technique. First, a strategy is employed to determine where in the field the grid (solution) needs to be modified, e.g., by means of error estimation or flow feature detection. Secondly, a mechanism is used to either change the grid density or modify the solution method. The focus of this paper is mostly on the alteration of grids as adapted to flows or geometry perturbations. A brief discussion of error estimation and flow feature detection follows. Further in−depth study of the subject is planned for future work.

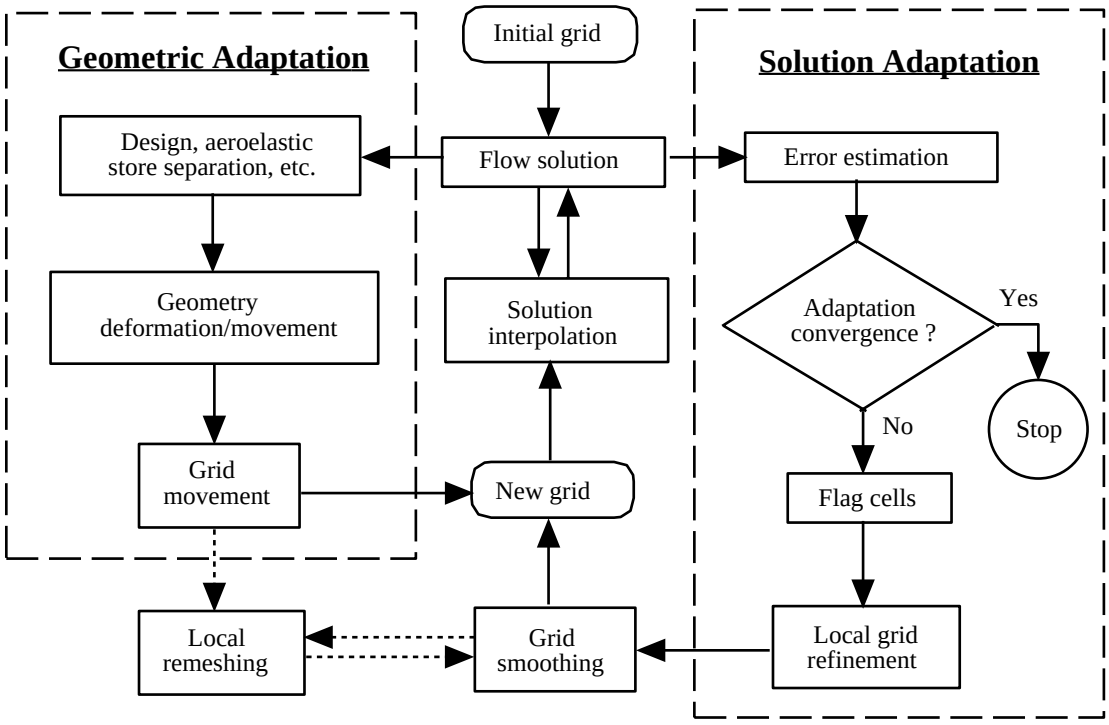


Figure 1. Flow chart of the proposed grid adaptation strategy by local remeshing, grid subdivision, and grid movement.

Error Estimation and Feature Detection

In most grid adaptation techniques, the question of where to modify the grid or the solution accuracy in a computational domain is directly or indirectly addressed through the concept of error equidistribution. This principle states that nodes should be clustered in a grid in such a way that the computational error is uniformly distributed throughout the grid elements. In other words, the grid should be denser where the solution incurs larger error, e.g., where the flow undergoes rapid changes, and vice versa. The principle of error equidistribution is strictly applied in methods by r−refinement (and to some degree in global remeshing techniques) to optimally redistribute grid points in the field. The magnitude of the computed errors directly determines the grid spacings in these methods.

In the methods based on h−refinement, however, an error estimation only serves as a means to locate large computational errors. A separate mechanism then modifies the distribution of grid nodes at these locations without considering the magnitude of the estimated errors. Unlike the r−refinement which aims for a converged grid with equal error at each node, the h−refinement fulfills the objective of reducing the maximum error through several preset steps without considering the error equidistribution criterion at each adaptation cycle. The role of error estimation in these methods practically reduces to the detection of flow features rather than accurate indication of errors. Therefore, the concept of error estimation appears to be less critical for h−refinement as it does not influence the grid characteristics directly.[6]

Most error (feature) indicators in use are based on some physical flow quantities such as density, pressure, entropy, etc. Functions of the first or second gradient of these quantities are usually used to estimate errors or detect flow features. For h−refinement, even a crude indicator such as a simple increment of a flow quantity is sufficient as long as it correctly detects the desired flow feature. In this work, the following simple indicator based on the static pressure difference is used to detect expansion and shock waves.

$$\vartheta_i = \left(1 + \delta_i / \delta_a\right) \left|\Delta p_i\right| / p_i \qquad (1)$$

where p_i and Δp_i are the local static pressure and its increment associated with the i^{th} grid element, respectively, δ_i is the local grid spacing, and δ_a is an average grid spacing. The inclusion of a grid spacing correction factor (first parenthesis in Eq. 1) results in a better detection of weak flow discontinuities in larger grid cells away from the surfaces.

Functions based on flow vorticity or simply entropy generation in the flow field have been used as indicators for detecting vortices and adapting grids to vortical flows. In this work, the following simple measure of entropy is used to capture vortices.

$$\varepsilon_i = \left(\gamma p_i / \rho_i^{\gamma}\right) - 1 \qquad (2)$$

where p_i is the local static pressure, ρ_i is the local density, and γ is the specific heat ratio. The indicators computed by Equations (1) and (2) are compared with the user's prescribed threshold values at each grid element. If an indicator is greater than the threshold value, the corresponding grid element is flagged for refinement.

A challenge in the practical implementation of adaptive methods for complex problems is the choice of appropriate indicating functions. While a particular indicator may work for certain class of flow features, it may not be as effective in recognizing other flow phenomena. Usually, a prior knowledge of the flow characteristics is needed in order to select relevant functions. Since the information about the flow is generally not available in advance, an indicator based on a combination of several flow functions is desirable for capturing all dominant flow features. Such an indicator must be "smart" enough to distinguish between the actual flow variations and the numerical "noise" present in the solution. Otherwise, the grid may be refined in wrong locations. A comprehensive study of the universal flow feature indicators is planned for future work.

Adaptive Refinement

The inherent irregularity of unstructured grids offers two important advantages: 1) high degree of flexibility to handle complex shapes and 2) ease of mesh alteration. The lack of a regular structure in tetrahedral grids, for example, results in arbitrary cell groupings which, in effect, makes every part of a grid independent of the rest. Consequently, any section of a tetrahedral grid can be removed and locally remeshed without disturbing the rest of the grid. Furthermore, the local resolution of a grid can be arbitrarily changed when the grid is partially remeshed. This important property makes unstructured grids particularly suitable for adaptive local refinement.

In this work, an unstructured tetrahedral grid generation system, VGRIDns, is used to generate initial grids. The grid generation method is based on the Advancing−Front[7] and Advancing−Layers[8] techniques. The Advancing−Front method is a marching process in which tetrahedral cells grow on surface triangles (initial front) and gradually advance in the field. The front, made of the exposed triangular faces of the tetrahedra, continuously evolves as new cells are created and added to the field. The process continues until the entire domain is filled with tetrahedra. At this point, no exposed frontal face remains in the field. The grid characteristics, used during the marching process, is prescribed through a set of source elements included in a "transparent" Cartesian background grid.[9] The data, to be used for clustering unstructured grid points, are smoothly distributed from sources onto the background grid nodes by solving an elliptic equation.

An important feature of the advancing front technique, like any other marching method, is that the solution process can be restarted at any time. Since a grid segment, once constructed, does not influence the rest of the mesh yet to be generated, the process can be stopped and restarted without "carrying" the grid segment already generated in the previous run(s). The only data required to restart the generation process are those defining the current front, excluding information about the field grid underneath. An efficient grid restart capability along with a local remeshing technique have previously been developed and incorporated into the

VGRIDns system for grid post−processing.[10] In this work, the existing local remeshing capability is extended for adaptive grid refinement.

The process of local grid refinement is outlined in the flowchart shown in Fig. 2 and is demonstrated schematically for a simple 2D example on a hypothetical grid in Fig. 3. In this example, a transonic flowfield around a simple airfoil is assumed. An initial coarse grid along with the corresponding flow solution are supplied to the adaptive refinement scheme (Fig. 3a). An appropriate flow detector, such as that given in Eq. (1), is used to detect the dominant flow features. For example, a diffused shock wave and a rapid flow gradient at the leading edge of the airfoil are assumed in the present example. The grid elements experiencing large variations in the flow properties are then identified for removal (Fig. 3b). In the next step, the flagged elements along with an additional layer of cells are deleted to create voids (empty pockets) in the mesh (Fig. 3c). The remaining grid points and cells are then renumbered, and those faces exposed in the pockets are grouped to form a new front in the grid. If any portion of the geometry is exposed in the voids, as shown in the example, the corresponding surface grid is h−refined, and the newly inserted nodes are projected onto the geometry model. The remaining faces on the front (those in the field) remain unrefined. Next, the resolution of the grid to be generated in the pockets are readjusted, and the voids are remeshed by the Advancing−Front method as in a regular grid generation restart. The newly generated grid elements are then renumbered and assembled with the rest of the mesh (Fig. 3d).

In an h−refinement, new grid nodes are added to the mid−point of the surface line segments (in 2D) and triangle edges (in 3D). On a surface in 3D space, each interior triangle is then divided into four smaller triangles. The "transitional" triangles (those between the refined and unrefined regions) are divided into two or three triangles, depending on their number of edges exposed in the voids. The process of local h−refinement is depicted in Fig. 4 for a portion of a 3D triangular surface mesh. In this figure, the shaded triangles are assumed to be covered with tetrahedra. The unshaded triangles are exposed in a void after a segment of the volume grid is removed as a result of a discontinuity present in the flow.

Since the length of a mesh edge on the surface is cut in half by the h−refinement, the spacing parameters defined by the background grid are also reduced by

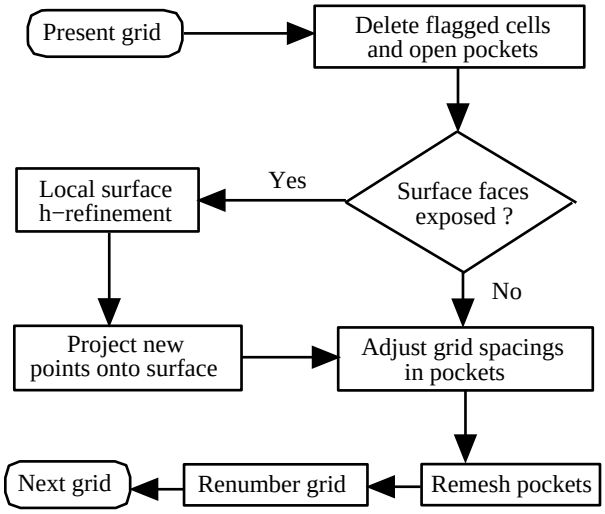


Figure 2. Flow chart of local grid refinement.

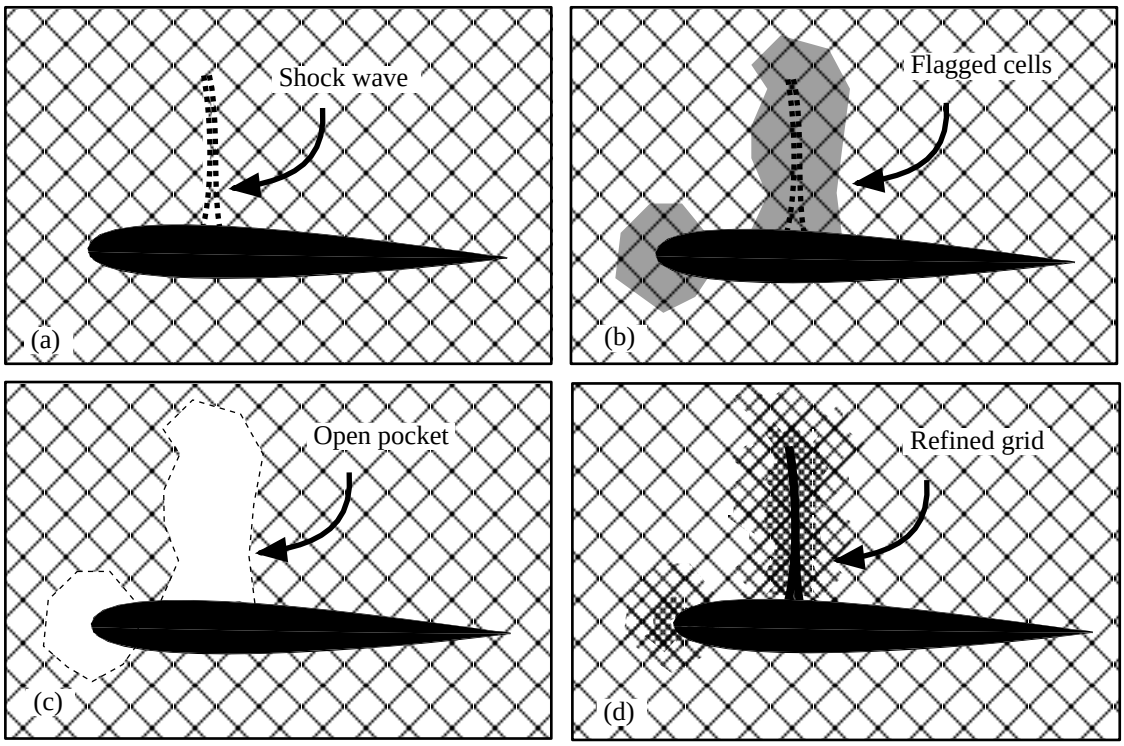


Figure 3. Adaptive refinement steps by local remeshing: (a) initial grid/solution, (b) flagged cells in regions of rapid rapid flow gradients, (c) flagged cells removed, and (d) locally refined grid showing a smooth transition from fine to coarse cells.

50% for regenerating the volume grid in the voids. The modified background grid spacing would provide the required compatibility between the h−refined surface and locally remeshed volume grids. An average grid spacing based on the actual size of the interior front faces (those which are not h−refined) and the modified background grid parameters is used to generate the first layer of tetrahedra. The average spacing provides a smooth transition from the coarse tetrahedra of the original grid to the fine cells generated in the pockets as indicated in Fig. 3d schematically.

Grid movement

A grid movement strategy is employed in this work to perturb volume grids adapting to moving and/or deforming surfaces. Although the method allows substantial grid perturbations, it is not designed for solution adaptation by grid movement (r−refinement). Two different techniques are employed to redistribute unstructured grid nodes in the field as a result of surface motion. While the perturbation of tetrahedral "inviscid" grids is relatively straightforward and has been implemented for design and aeroelastic analyses in the past, moving thin layered unstructured "viscous" grids is more complicated and requires different strategies than those used for inviscid grids.

Regular inviscid grids generated with the Advancing Front method are moved, in the present work, using a modified version of the "spring" method reported in Ref. 11. In this method, the grid nodes are assumed to be interconnected with a system of tension springs which is further assumed to be in an equilibrium state initially. Once the geometry along with the boundary nodes are moved, the disturbed spring system rebounds to a new equilibrium state which can be modeled by

$$\sum_{j=1}^{J_i} \frac{\lambda_{ij}}{l_{ij}^2} (\Delta s_i - \Delta s_j) = 0 \qquad (3)$$

for all nodes i in the field. In Eq. (3), Δs_i and Δs_j are the coordinate displacements of node i and its neighboring node j, respectively, J_i is the number of neighboring nodes connected to node i, l_{ij} is the length of the edge (spring) connecting node i to node j, and λ_{ij} is a spring stiffness function defined as

$$\lambda_{ij} = \lambda_{ij} \ (L_{ij}, J_{is}) \qquad (4)$$

where L_{ij} is a parameter proportional to the distance from spring $i-j$ to the surface and J_{is} is the number of springs connecting node i to the surface. The correcting function given by Eq. (4) equalizes forces exerted on the nodes directly connecting to the surface regardless of the number of springs attached. It also distributes stiffness quantities among springs in such a way that the value is maximum for the springs attached to the moving surfaces and gradually diminishes to a constant value as the distance from the surface increases. The use of spring stiffness (λ_{ij} / l_{ij}^2) in Eq. (3), rather than l_{ij} (as used in Ref. 11 and some other references in the literature), has significantly improved the robustness of the method for problems involving large amounts of movement. Figure

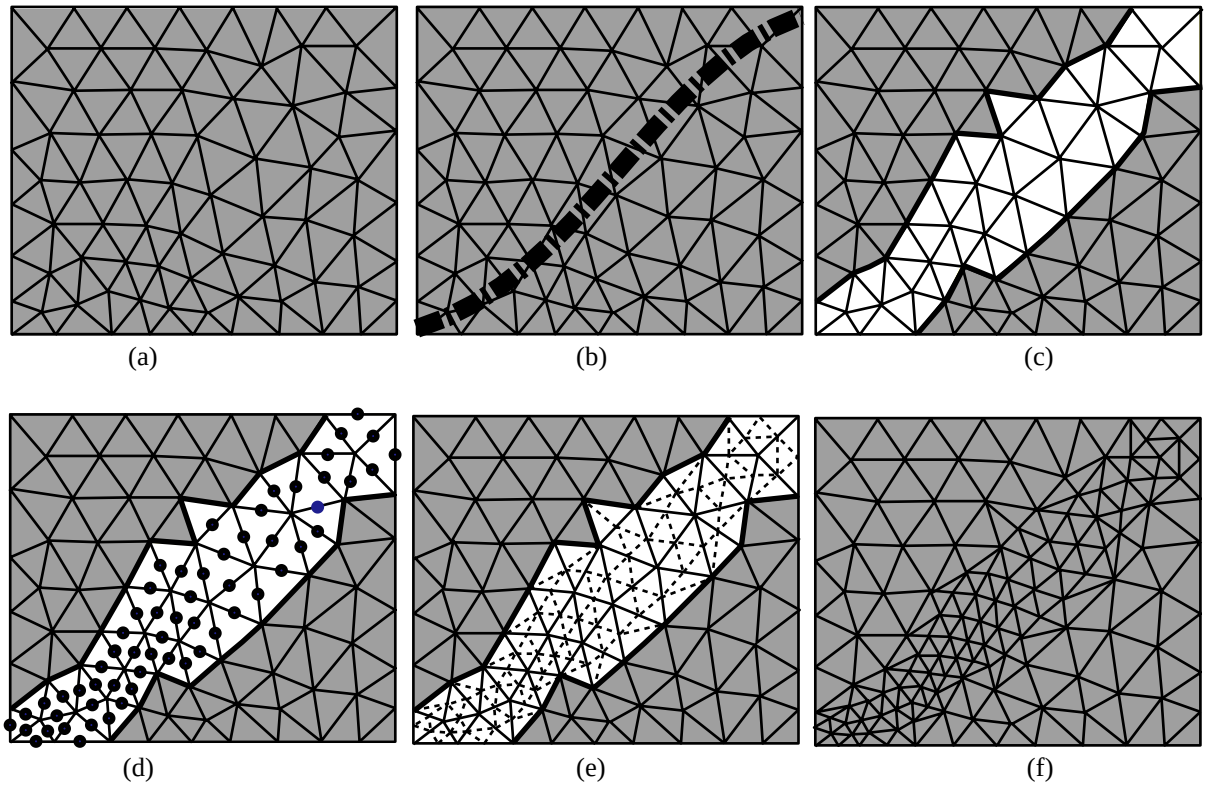
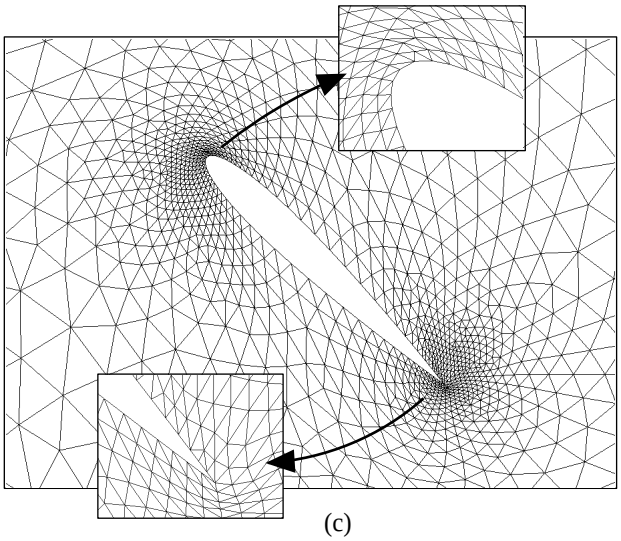


Figure 4. Process of surface mesh local h−refinement in 3D: (a) initial coarse surface grid, (b) footprint of flow discontinuity on surface, (c) exposed triangular faces in a void, (d) insertion of new points on edges of flagged faces, (e) subdivision of flagged faces, (f) final adapted grid.

5 demonstrates the effect of the stiffness correction on a grid undergoing substantial perturbation. A triangular mesh around the NACA 0012 airfoil is shown in Fig 5a in an initial position of zero incidence. The geometry is then rotated about the mid−chord by 45 degrees clockwise while the outer boundary is held fixed. With the conventional spring method, the perturbed grid folds at the leading and trailing edges creating 30 negative volume cells as shown in Fig. 5b. The modified method moves the grid correctly with no negative volume cells introduced as a result of stiffer springs around the geometry (see Fig. 5c). In both cases, the grids were moved in ten steps.

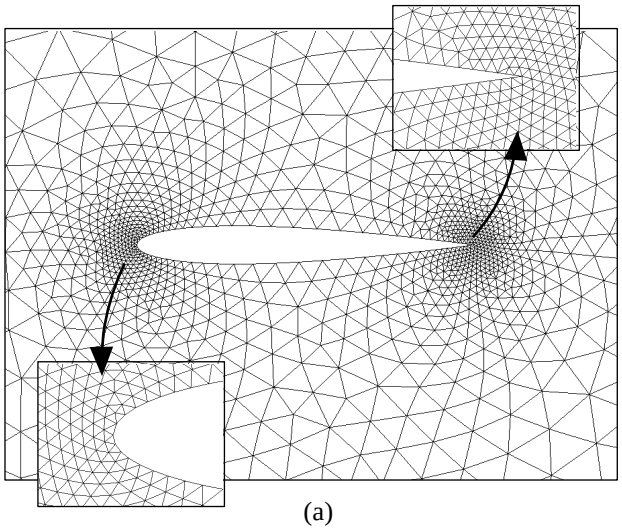
While the spring method is effective to move inviscid triangular and tetrahedral grids, it often fails when applied to viscous grids containing highly stretched cells adjacent to the body. Instead, a different technique is proposed in this work for moving viscous grids. The new technique is compatible with the method by which viscous grids are generated in VGRIDns.

The viscous portion of tetrahedral grids are generated with VGRIDns using the Advancing Layers technique which is also a marching method. Unlike the Advancing Front method which introduces grid nodes in an irregular manner, the advancing layers distribute nodes in the boundary layer along a set of predetermined surface vectors. Layers of thin tetrahedral cells grow on surface triangles and systematically advance in the direction of surface vectors.[8] When the cell heights, determined by a prescribed stretching function, become compatible with the local background grid spacings, or when two opposing fronts approach each other, the layers stop growing locally. At the end of the Advancing−Layers process, the triangular faces exposed on the outer layers form an initial front for the Advancing−Front method to fill the rest of the domain with regular tetrahedra as explained before.

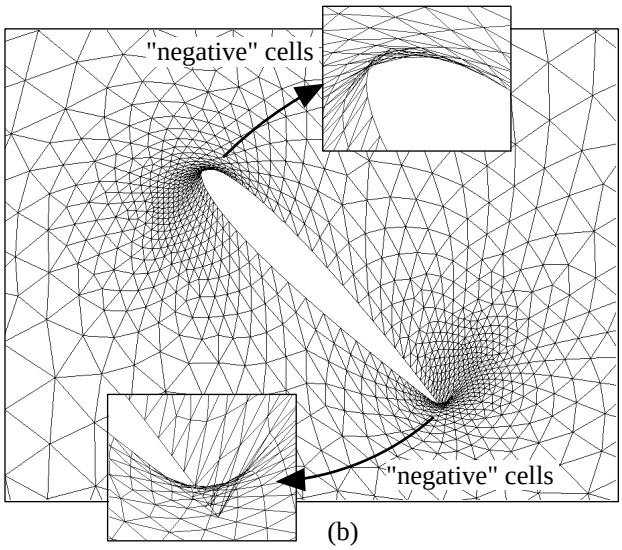


(c)

Figure 5. Moving grid around a NACA 0012 airfoil: (a) grid in the original position, (b) grid moved with the conventional method, (c) grid moved with the modified method.



(a)



(b)

An important element in the Advancing−Layers method is the computation of surface vectors. It is crucial that the computed vectors satisfy a "visibility" criterion in order to prevent cells at sharp corners from folding and creating negative volume cells. A robust iterative scheme has been developed earlier (see Ref. 8) which satisfies the visibility condition for all surface vectors. The vectors are computed once and are smoothed with a Laplacian operation before the process of volume grid generation starts.

To move grid nodes in the boundary layer, the same algorithm for the computation of surface vectors is employed in this work. After the geometry as well as the surface mesh are moved/deformed, a new set of vectors based on the displaced or deformed surface geometry is computed. The existing volume grid nodes in the boundary layer are then redistributed along the new vectors while preserving the original cell topology. The remaining grid points outside the boundary layer (those generated with the Advancing−Front method) are then moved by the spring method with the viscous portion of the grid held fixed. With this approach, the fidelity of highly stretched viscous grids remains intact as proper displacement of grid nodes in the boundary layer is guaranteed by the recomputation of surface vectors.

Results

Three steady state test cases with different inviscid flow features were considered in this study: 1) an ONERA M6 wing at a transonic speed resulting in a "λ" shock wave on the upper surface of the wing, 2) the Modular Transonic Vortex Interaction (MTVI) model at a subsonic speed featuring a strong vortical flow, and 3) an X−38 forebody configuration at supersonic speed creating a strong detached "bow" shock in front of the geometry. A three−dimensional moving viscous grid is also presented to demonstrate the geometric grid adaptation using the present directional node redistribution approach. All inviscid flow solutions, presented in this paper, were obtained using the upwind, cell−centered, finite−volume, unstructured grid solver USM3D.[12]

ONERA M6 Wing

An ONERA M6 wing configuration has been used to demonstrate the transonic shock capturing capability of the present solution adaptive grid method. The flow condition is at Mach number 0.84 and an incidence of 3.06º.

A coarse grid with a nearly uniform grid distribution chordwise has been generated which is shown in Fig. 6(a). The grid contains 2,615 boundary nodes, 15,432 total nodes, and 83,356 tetrahedral cells and serves as the initial grid for adaptation. An inviscid flow computation on this grid indicates the presence of a "λ" shock wave on the upper surface of the wing. As expected, the shock wave is diffused due to the coarseness of the grid. The surface pressure contours are also illustrated in Fig. 6(a).

Using the grid subdivision and local remeshing procedure described earlier, three levels of adaptive refinements were performed for this case. The indicator given by Eq. (1) was used to detect regions of large flow variations. In each refinement, cells with 20% or higher increment of the indicator were deleted and remeshed. The surface triangles exposed in the opened pockets were h−refined as described earlier. The final surface grid and the pressure contours are shown in Fig. 6(b). The adapted grid has 9,739 boundary nodes, 54,385 total nodes, and 288,739 tetrahedra. As indicated, the grid is efficiently refined at the shock locations and the leading edge of the wing where there is a large pressure gradient at the suction peak. The effect of grid refinement is clearly indicated on the surface pressure contours showing a sharp "λ" shock definition.

To investigate the effect of the present grid adaptation method on the accuracy of solutions, a uniformly fine grid with a resolution similar to that of the adaptively refined grid was generated. The grid contains 40,424 boundary nodes, 394,155 total nodes, and 2,217,001 tetrahedra. The surface grid and the corresponding pressure contours are shown in Fig. 6(c). A comparison of this solution with that of the adapted grid reveals that the differences between the two are negligible, and that the grid adaptation has produced an almost identical result with about an order of magnitude smaller grid size. Figure 7 illustrates several chordwise surface C_p distributions for the initial coarse, adapted, and unadapted

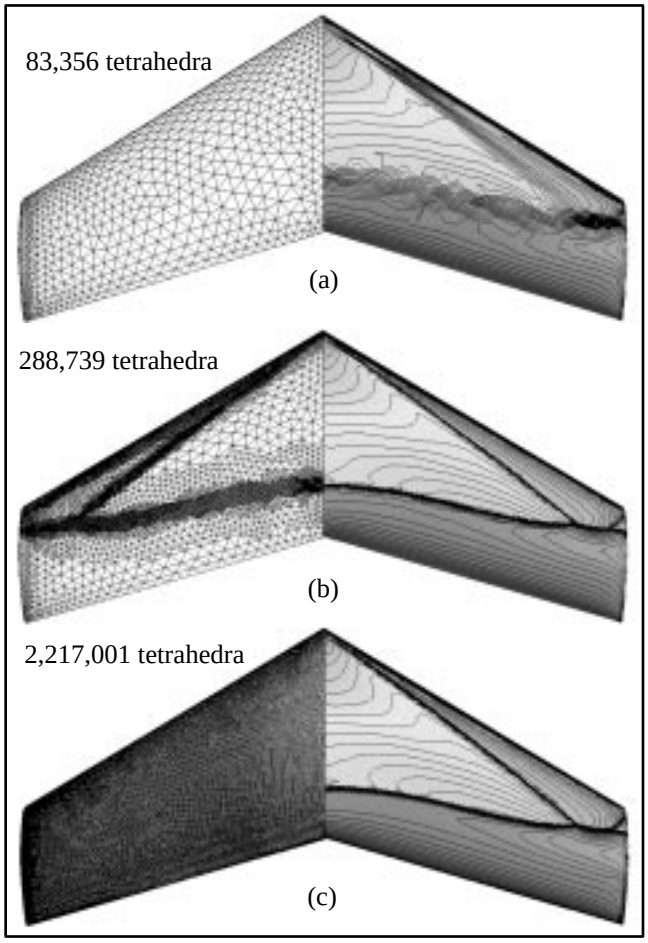


Figure 6. Grids and surface pressure contours on ONERA M6 wing, M=0.84, α=3.06º: (a) initial, (b) adapted, and (c) fine unadapted grids and solutions.

fine grids as compared with the experimental data at five different span locations. As expected, there are insignificant differences between the adapted and the fine grid results. Although the result of the coarse grid seems to be in good agreement with the experimental data at the shock locations, it is well known that inviscid solutions predict stronger shocks further downstream as indicated by both the fine and adapted grid curves in Fig. 7. Addition of viscous effects to inviscid solutions usually weaken and move shock waves upstream to the correct locations.

The initial grid and solution, as well as the adaptive refinements and solutions, were all generated using a Silicon Graphics Octane workstation with a R10000 Processor in this case. While the mesh for the fine grid was also generated on the same workstation, the corresponding flow was computed on a CRAY C90 supercomputer. A converged solution on the fine grid took about 36,548 CPU seconds on the CRAY C90, whereas a total of 40,335 CPU seconds of the SGI workstation were spent to obtain the adapted solutions. For all cases presented in this paper, the adapted solutions were started from the freestream condition at each adaptation cycle. The development of an efficient unstructured grid interpolator is planned for future work.

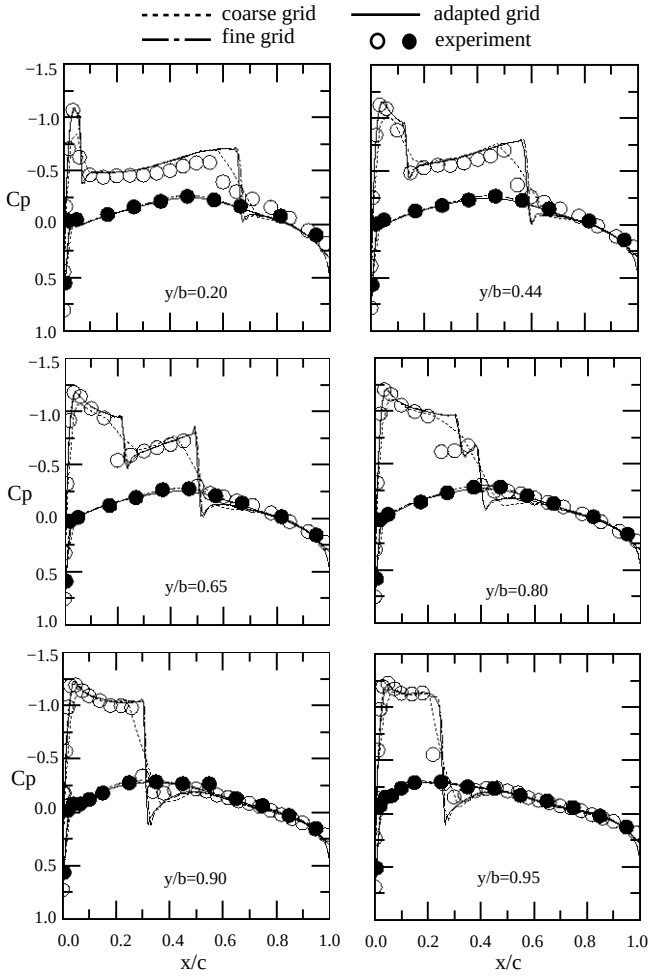


Figure 7. Unstructured grid inviscid solutions on ONERA M6 wing (M=0.84, α=3.06º).

With such a utility, the converged solution of a coarse grid can be transferred onto the fine grid of the next adaptive cycle to restart the solution. Interpolation of data would further reduce the adaptive solution time.

Modular Transonic Vortex Interaction Configuration

To demonstrate the effectiveness of the present solution adaptive method for predicting vortical flows, a generic fighter model referred to as the Modular Transonic Vortex Interaction (MTVI) has been employed. The geometry features a chine forebody with an included angle of 30 degrees, sixty−degree cropped delta wings, partially deflected wing leading−edge flaps, and twin vertical tails. All edges of the geometry are sharp, inducing flow separations and vortices which are independent of viscous effects. Inviscid solutions were obtained at Mach 0.4 and a 20− degrees angle of attack.

The initial coarse grid contains 31,565 nodes and 163,619 tetrahedra. After three levels of grid adaptation, the numbers of grid nodes and cells increase to 108,014 and 564,727, respectively. Figure 8 shows the surface triangulations for the initial coarse grid (port) and after three cycles of adaptation (starboard). The refinement of surface grid, as adapted to the dominant flow features, is clearly indicated.

The local refinements of volume grids at the vortex locations are depicted in Figure 9 at two different stages of adaptation. The refinement of the initial grid (Fig. 9a), determined by the first solution, indicates a chine

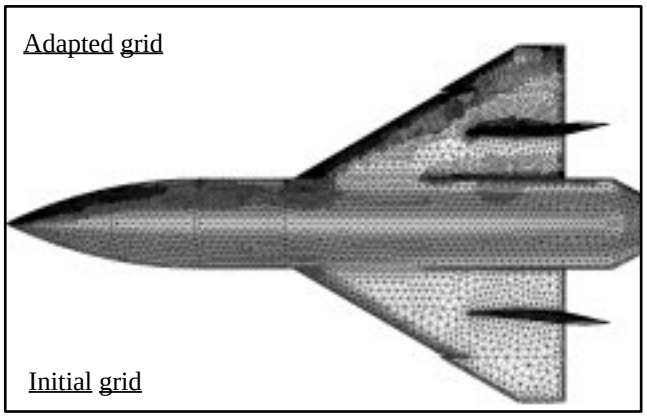


Figure 8. Surface triangulations on the twin−tail MTVI configuration: initial grid (port) and adapted grid (starboard).

vortex extending beyond the aircraft tail. The final refined grid correctly predicts a chine vortex breakdown ahead of the vertical tail as indicated in Fig. 9b. A similar vortex break down phenomenon on this geometry has been observed experimentally. Thus, the initial coarse grid actually yields a misleading solution which underscores the importance of grid adaptation for providing adequate grid resolution automatically.

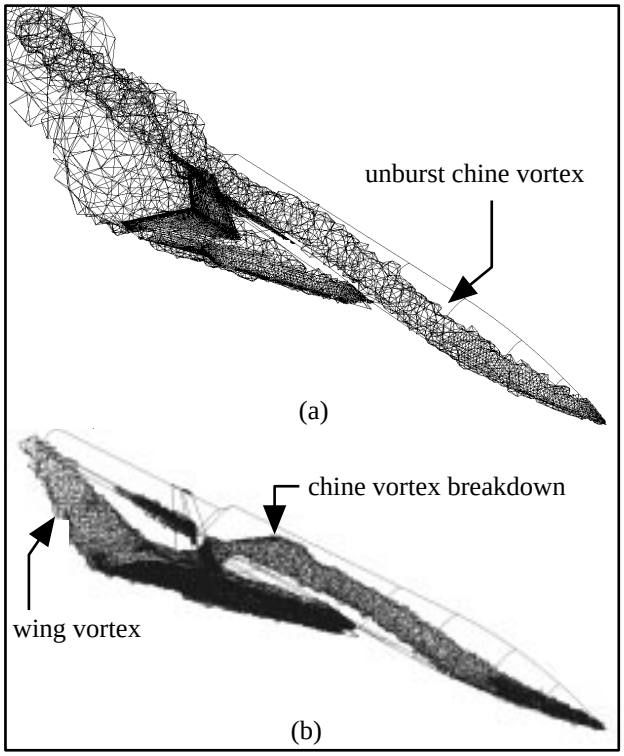


Figure 9. Local refinement of the MTVI grid: (a) first adaptation cycle indicating an unburst chine vortex, (b) third adaptation cycle showing a chine vortex breakdown.

Figure 10 compares surface pressure distributions before and after grid adaptation on the port and starboard sides of the aircraft, respectively. The pressure distributions in the field, showing the vortices in a cross−sectional plane and on the surface, are portrayed in Fig. 11. As evident, the wing leading−edge vortices have been sharply captured with grid adaptation (starboard). Also, the adapted solution indicates a crisper chine vortex footprint which does not extend as far downstream as that of the unadapted solution (indicative of the vortex burst phenomenon). A corresponding image of the volume grids (before and after adaptation), as well as the surface grids, are shown in Fig. 12.

To detect vortices, the feature detector given by Eq. (2) was used in this example. Those cells with an entropy production level (ε) above a threshold value (typically a small fraction of the maximum entropy produced in the field) are flagged for removal at each adaptation cycle. In the present example, a threshold value of 0.01 has been used.

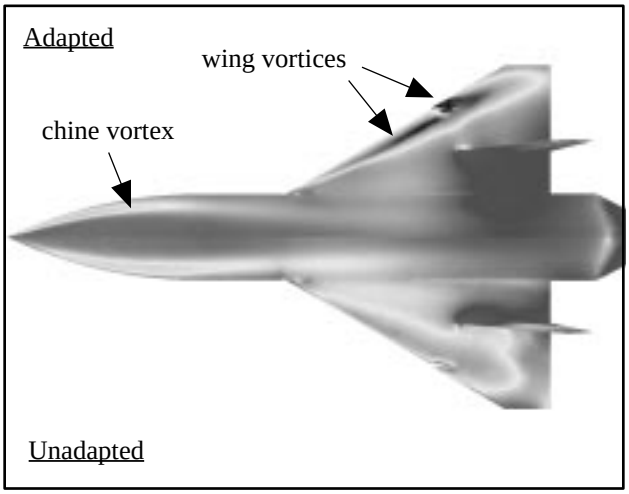


Fig. 10. Surface pressure distribution on the MTVI showing footprints of the wing leading−edge and chine vortices: initial solution (port) and adapted solution (starboard).

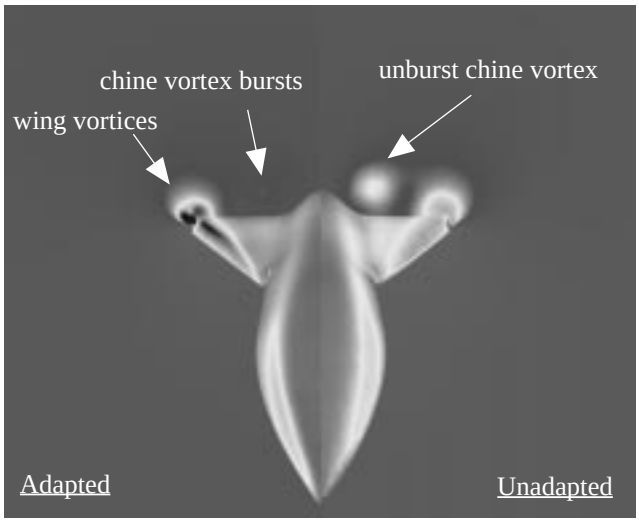


Fig. 11. Surface and field pressure distributions on the MTVI showing the wing leading−edge and chine vortices: initial solution (port) and adapted solution (starboard).Ì

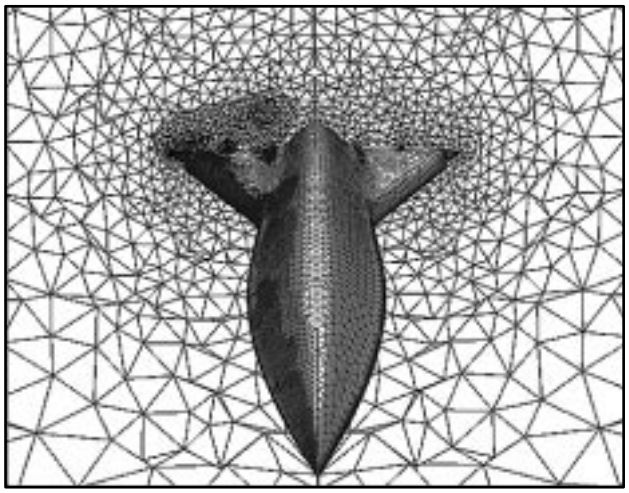


Fig. 12. Surface and field grid distributions on the twin−tail MTVI showing grid refinement at the vortex locations: initial grid (port) and adapted grid (starboard).

X−38 Forebody Configuration

The last case to demonstrate the solution adaptive capability of the present method concerns a supersonic flow creating a detached shock wave in front of a blunt−nose geometry. The configuration selected for this purpose is the front portion of an experimental aerospace vehicle referred to as X−38. The flow condition is at Mach 2 and zero incidence angle. This case represents a classic example for which the generation of an efficient unadapted grid becomes challenging due to the presence of a conical detached shock wave extending far into the field. Even with a prior knowledge of the shock location, it is difficult to control the concentration of grid points at a curved surface in a 3D space. Often, the generated grids are either too coarse away from the geometry, which fail to capture the discontinuities accurately, or globally too fine making the computational cost excessive. The advantages of solution adaptive grid methods become more tangible for such applications.

In Fig. 13, two separate grid cross−sections (before and after adaptation) are illustrated. The initial grid with 87,806 cells, shown on the left−hand side of the figure, represents a typical unadapted grid which is adequately resolved around the geometry but is too coarse in the field. The grid after three levels of adaptive refinement, shown on the right−hand side of the figure, includes 840,135 cells. As evident, the adapted grid is efficiently refined in the field at the 3D conical shock location. Even a weaker shock in front of the canopy is automatically detected, and the grid is refined. Also, note the smooth transition of the grid spacing from the original coarse sections to the refined regions where the shock waves are formed. An unadapted globally fine grid, resolving the shock

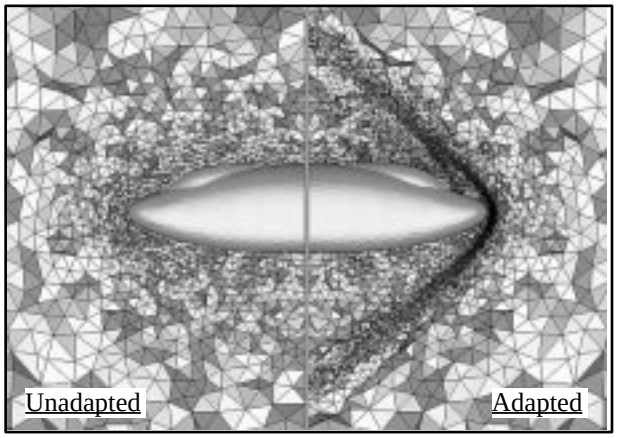


Figure 13. Adapted and unadapted tetrahedral grids on the X−38 forebody configuration.

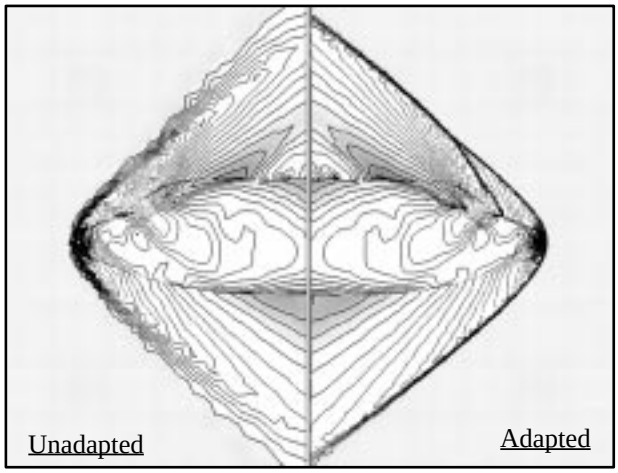


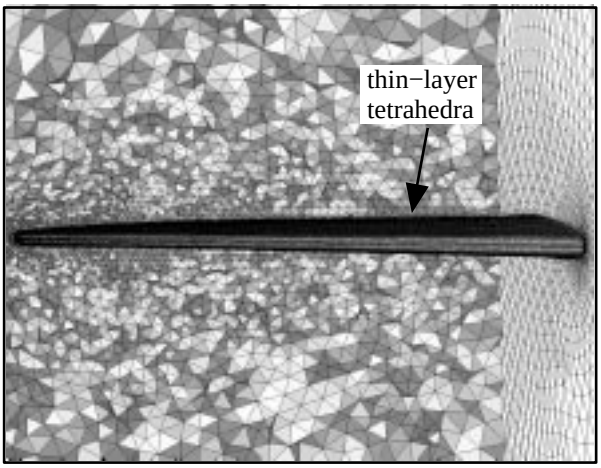
Figure 14. Comparison of adapted and unadapted pressure contours on X−38 forebody configuration, M=2, α=0.

locations similarly to the adapted grid, was also generated for comparison. The fine grid (not shown) contains 11,786,137 cells.

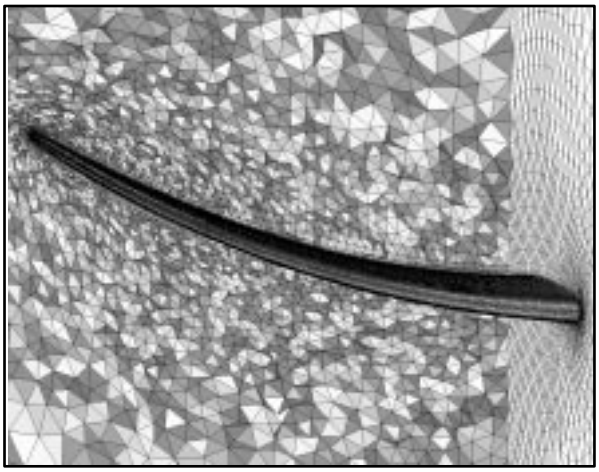
Equation (1) is used to detect the shock waves in this case. The effect of the grid spacing correction factor in Eq. (1) has resulted in a better detection of small pressure differences away from the geometry. Consequently, the larger cells which have hardly experienced flow discontinuities in the initial solution are flagged and refined, resulting in crisper adapted shocks extended farther away in the field. Figure 14 shows pressure contours on the surface and in the field reflected on a cutting plane passing through the geometry. The solution on the unadapted grid (left−hand side of Fig. 14) contains a diffused shock wave which quickly dissipates away from the body. The adapted solution, on the right, predicts a sharp bow shock extending farther out to the outflow boundary. A secondary shock wave in front of the canopy is also well captured with adaptation, whereas it appears as a weak compression wave in the unadapted solution.

Elastic Wing

The moving grid capability is demonstrated on a viscous grid around a generic wing undergoing substantial movement and deformation. The grid contains thin layers of tetrahedral cells generated with the Advancing Layers method. Figure 15a depicts the front view of the undisturbed wing which is attached to a wall. The surface triangles and a cross−section of the volume grid around the wing are also shown.



(a)

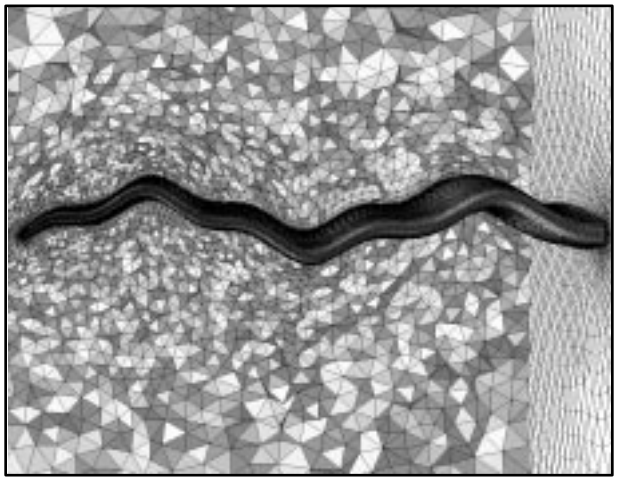


(b)

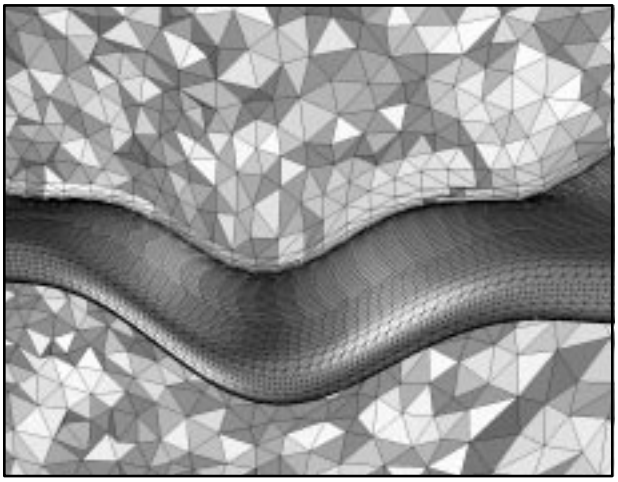
Figure 15. Moving viscous grid around a generic wing: (a) initial grid, (b) perturbed grid.

In the first case of grid movement, the wing is bent upward simulating deformation due to an aeroelastic effect as shown in Fig. 15b. The distance by which the wing tip is moved is comparable to the chord length at the wing mid−section. As explained earlier, the volume grid is moved by first recomputing the surface vectors and redistributing the grid points in the boundary layer in the direction of vectors. The inviscid portion of the grid is then moved by the spring method as described earlier.

Figure 16a illustrates the wing undergoing a different type of deformation in which segments of grid move in opposite directions. A close−up view of



(a)



(b)

Figure 16. Moving viscous grid around a deforming wing: (a) farfield, (a) close−up view showing viscous layers .

the wing/grid at the mid−section, where a large amount of movement occurs, is shown in Fig. 16b. As evident, the viscous layers along with the outer portion of the grid are adapted to the deformed surface properly. The quality of the grid after movement is checked to confirm that no grid anomaly such as negative volume cells were introduced as a result of grid movement.

Concluding Remarks

An adaptive unstructured grid approach is developed and tested on several three−dimensional cases. The method, based on a surface grid subdivision, volume grid local remeshing, and grid movement, has demonstrated promising potential for solution adaptive refinement and movement of unstructured grids around complex configurations. While the objective of this paper is to demonstrate the "pilot" technology, further work is required to fully automate the method and extend its capabilities for realistic, complex problems. Among further developments planned for future work are: implementation of better error/feature indicators for accurate adaptation of solutions involving multiple dominant flow features, automatic projection of new surface grid points onto the geometry model, solution interpolation, and extension of the method for solution adaptive refinement of viscous grids.

Acknowledgements

The author would like to thank Mr. Farhad Ghaffari, Subsonic Aerodynamics Branch at NASA Langley Research Center (LaRC) and Dr. Neal T. Frink, Configuration Aerodynamics Branch at LaRC for many inspiring technical discussions during the course of this study. The support and encouragement of Dr. James L. Pittman, High Performance Aircraft Office at LaRC and Mr. Laurence D. Leavitt, Configuration Aerodynamics Branch at LaRC are gratefully acknowledged.

References

[1] Soni, B.K., Weatherill, N.P., and Thompson, J.F., "Grid Adaptive Strategies in CFD," Invited Paper, International Conference on Hydro Sciences & Engineering, Washington, D.C., June 7−11, 1993.

[2] Coirier, W.J. and Powell, K.G., "A Cartesian, Cell−Based Approach for Adaptively−refined Solutions of the the Euler and Navier−Stokes Equations," *Proceeding of the Surface Modeling, Grid Generation and Related Issues in Computational Fluid Dynamics Workshop*, NASA Conference Publication 3291, NASA Lewis Research Center, Cleveland, OH, May 1995.

[3] Mavriplis, D.J., "Adaptive Meshing Techniques for Viscous Flow Calculations on Mixed Element Unstructured Meshes," AIAA Paper 97−0857, January 1997.

[4] Peraire, J., Peiro, J., and Morgan, K., "Adaptive Remeshing for Three−Dimensional Compressible Flow Computations," *Journal of Computational Physics*, 103, 1992, pp. 269−285.

[5] Baum, J.D., Luo, H., and Lohner, R., "A New ALE Adaptive Unstructured Methodology for the Simulation of Moving Bodies," AIAA Paper 94−0414, January 1994.

[6] Baker, T.J., private communications.

[7] Lohner, R. and Parikh, P., "Three−Dimensional Grid Generation by the Advancing−Front Method," *International Journal of Numerical Methods in Fluids*, 8, 1988, pp. 1135−1149.

[8]Pirzadeh, S., "Three−Dimensional Unstructured Viscous Grids by the Advancing−Layers Method," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 43−49.

[9] Pirzadeh, S., "Structured Background Grids for Generation of Unstructured Grids by Advancing Front Method," *AIAA Journal*, Vol. 31, No. 2, 1993, pp. 257−265.

[10] Pirzadeh, S., "Recent Progress in Unstructured Grid Generation," AIAA Paper 92−0445, January 1992.

[11] Batina, J.T., "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes," AIAA Paper 89−0115, January 1989.

[12] Frink, N.T., "Tetrahedral Unstructured Navier−Stokes Method for Turbulent Flows," *AIAA Journal*, Vol. 36, No. 11, 1998, pp. 1975−1982.