ANISOTROPIC CARTESIAN GRID METHOD FOR VISCOUS TURBULENT FLOW

Z.J. Wang* (zjw@cfdrc.com), R.F. Chen†
CFD Research Corporation, Huntsville, AL 35805

ABSTRACT

A 2^n tree based Cartesian grid generation method has been developed recently for complex geometries to simulate viscous flows. The "viscous" Cartesian grid is capable of resolving boundary layers with high-aspect ratio projected viscous layer grids. Compared with an Octree data structure, the 2^n tree data structure supports anisotropic grid adaptations in any of the coordinate directions in an arbitrary manner. This capability enables flow features such as shocks, shear layers and wakes to be resolved very efficiently. In this paper, the high-resolution capability are demonstrated with several well-documented viscous turbulent flow cases.

INTRODUCTION

The unstructured grid-based CFD methodology has undergone considerable development in the last decade in term of both grid generation and solution algorithm development. It is generally recognized that unstructured grid–based CFD algorithms offer the best promise for automated fluid flow simulations. There are many types of unstructured grids currently in use by CFD researchers and practitioners, which include triangular and tetrahedral grids[1–5], quadrilateral or hexahedral grids[6], prismatic grids[7] or mixed grids[8–9]. The most appealing properties of unstructured grids include the geometric flexibility and the ease in which the grid can be adapted according to flow features. Tetrahedral grids are the easiest to generate. Many well–known grid generation algorithms, such as the advancing front[10] and the Delauney triangulation method[11] have been developed to generate tetrahedral grids for complex geometries. However, experiences have indicated that tetrahedral grids are not as efficient and/or accurate as hexahedral or prismatic grids for viscous boundary layers. On the other hand, prismatic grids and hexahedral grids can resolve boundary layers more efficiently but they are more difficult to generate than tetrahedral grids. Many CFD researchers have come to the conclusion that mixed grids (or hybrid grids) are the way to go.

Recently, there has been a renewed interest in using Cartesian grids for complex geometries[12-17]. Coupled with a tree–based data structure and grid adaptation these methods have been demonstrated to be very viable tools for inviscid flows, with very complex geometry. One of the most appealing properties of a Cartesian grid is its efficiency in filling space with a minimum number of cells and faces given a certain grid resolution. The extension of the Cartesian grid approach to viscous flows was achieved with either the adaptive Cartesian/ prism grids[18-20] or with projected viscous layer grids from the Cartesian grid[21-23] (the so-called "viscous" Cartesian grid method). Another development in the adaptive Cartesian grid method is the use of 2^N tree data structure[24] instead of the Octree. The 2^N tree data structure supports anisotropic grid adaptations of Cartesian cells naturally. The use of anisotropic grid adaptation (vs. isotropic grid adaptations) offers the potential of dramatic reduction in the total number of cells to achieve a given level of solution accuracy since most high-gradient flow features as shock waves, slip lines, vortex sheets, and wakes are anisotropic.

In this paper, we attempt to demonstrate the benefits of anisotropic grid adaptations with a variety of flow problems, in particular turbulent flows with large separation regions. In the following sections, we will summarize the viscous Cartesian grid method and the flow solver, describe the adaptation criteria, and present several flow problems. Conclusions and future work will then follow.

SUMMARY OF VISCOUS CARTESIAN GRID APPROACH

Given a geometric entity, the following steps are taken to generate the computational grid.

1. Cartesian Grid Generation

One of the popular data structures for adaptive Cartesian grid is the Octree. The drawback of Octree is that only

* Author, AIAA membership status

† Co-Author, AIAA membership status

isotropic grid refinement is supported. In this study, a 2^n tree data structure has been used. The 2^n tree supports binary, Quadtree and Octree type of subdivisions, and therefore allows the adaptive Cartesian grid to be refined in a non-isotropic manner. 2^n tree is a hierarchical data structure in which each non-leaf tree node can have either 2, 4 or 8 child nodes. All possible Cartesian cell subdivisions supported by the 2^n tree are illustrated in Figure 1. Note that a Cartesian cell can be sub-divided in an arbitrary manner.

The adaptive Cartesian grid is generated by recursively subdividing a coarse root grid. Since the root grid must cover the entire computational domain, the surface geometry is contained in the root Cartesian cells. The size of the Cartesian cells intersecting the geometry is controlled by two parameters, disT and disN. Parameter disN controls the Cartesian cell size in the geometry normal direction, whereas disT specifies the Cartesian cell size in the geometry tangential direction. The ratio disT/disN determines the maximum aspect ratio in the Cartesian grid. The recursive sub-division process stops when all the Cartesian cells intersecting the geometries satisfy the length scale requirements. For the sake of solution accuracy, it is very important to ensure that the Cartesian grid is smooth. In the present study, the sizes of any two neighboring cells in any coordinate direction cannot differ by a factor exceeding 2. The use of the 2^ntree data structure makes high aspect ratio Cartesian cells possible. This property can translate into considerable efficiency gains when anisotropic grid adaptations are used to resolve flow features.

2. Cartesian Grid Front Generation and Smoothing

In order to "insert" a viscous layer grid between the Cartesian grid and the body surfaces, Cartesian cells intersected by the geometry must be removed, leaving an empty space between the Cartesian grid and the body surface. Both the 2^n tree (for the Cartesian grid) and the Alternating Digital Tree[25] data structures (for the surface triangles) are extensively utilized for efficient search operations. After this step, all the exposed Cartesian faces are gathered together and form the so-called Cartesian grid front. The exposed Cartesian faces are then smoothed using a Laplacian smoother to form a smoother front, which is to be projected to the body surface.

3. Projection of the Cartesian Front to the Body Surface

After the smoothed front in the Cartesian grid is obtained, each node in the front needs to be connected to the body surface to form the viscous layer grids. The "foot prints" of the layer grids on the body surface have the same topology (or connectivity) as the Cartesian front. With this assumption, the viscous layer grids are naturally "blended" with the adaptive Cartesian grid, eliminating the need of cell-cutting currently adopted by many Cartesian grid generators. Another major advantage of the approach over cell cutting is that nearly all computational cells generated are convex, boosting the stability and convergence properties of flow solvers. The projection from the Cartesian grid front to the body surface is performed according to the minimum distance rule.

4. Geometric Feature Preservation

The front projection based on the minimum distance criterion may smear critical geometric features, especially near non-convex corners. This is shown in Figure 2. The geometrically "critical" concave corner never connects with a front node, and is therefore not represented in the projected surface grid. In order to eliminate this problem, a geometric feature recovery algorithm was developed. The algorithm first detects all the critical features in the geometry automatically. Then all the critical features are preserved through a feature recovery technique, in which front nodes are reconnected to the critical features.

5. Surface Grid Smoothing and Layer Grid Generation

The projected surface grid from the Cartesian grid front is usually not smooth because the projections are based on the minimum distance to the triangulated surfaces. A Laplacian smoothing algorithm is therefore applied to improve the grid quality. By connecting each point on the Cartesian grid front to the corresponding projected point on the surface, only a single layer of viscous grids is produced. To perform a meaningful viscous flow simulation, many more layers of viscous grids are necessary. Therefore this single layer of viscous grids is further divided into (user specified) arbitrary number of layers with arbitrary grid point distributions.

FLOW SOLVER AND SOLUTION BASED GRID ADAPTATION

A cell-centered finite volume flow solver supporting arbitrary cell types has been used to perform the flow simulations. The flow solver is second-order accurate in space with a cell-wise least-squares linear reconstruction technique. Roe's approximate Riemann solver[26] was used to compute the inviscid flux. The viscous flux is computed with an efficient technique with very compact data support without a separate reconstruction[20]. A very fast and memory efficient LU-

SGS scheme (improvement LU-SGS[27]) was used for time integration. The improved LU-SGS scheme can rival the convergence rate of a fully implicit scheme with a CGS or GMRES solver while requiring a fraction of the memory. To simulate flow turbulence, the classical two-equation k-ε turbulence model with wall function was used[28]. Details of the flow solver can be found in Reference 27.

To achieve automation in flow simulation, solution-based grid adaptation is essential. To take full advantage of the anisotropic grid adaptation capability offered by the 2^n tree, the three coordinate directions of each Cartesian cell are examined independently for possible grid adaptation. Since the viscous layer grid is generated by projecting the Cartesian front to the geometry, it cannot be independently adapted. However, the number of viscous layers, and the grid clustering factor can be changed based on local flow features. Both first and second derivative based grid adaptation criteria have been developed. The following cell-wise parameters are used as the adaptation indicators in x-direction

$$\tau_{ix} = \left|\frac{\partial q_i}{\partial x}\right| \Delta x_i^{1.5}$$

$$\pi_{ix} = \left|\frac{\partial^2 q_i}{\partial x^2}\right| \Delta x_i^{3}$$

where q can be any flow variable (pressure, total velocity, or density), τ_{ix} is a first-derivative based indicator, while π_{ix} is a second-derivative based indicator. The adaptation indicators in other directions can be computed similarly. The standard deviation of the parameter is computed as:

$$\tau = \left(\frac{\sum_{i=1}^{N} \tau_{ix}^2 + \sum_{i=1}^{N} \tau_{iy}^2 + \sum_{i=1}^{N} \tau_{iz}^2}{3N} \right)^{\frac{1}{2}}$$

where N is the total number of Cartesian cells. The the following conditions are used for grid adaptation:

1) if $\tau_{ix} > c*\tau$, cell i is to be refined in x direction;
2) if $\tau_{iy} > c*\tau$, cell i is to be refined in y direction;
3) if $\tau_{iz} > c*\tau$, cell i is to be refined in z direction;

where c is a control parameter determining the total number of cells to be refined. In this study, c is chosen to be 1.

TEST CASES

Octree versus 2^N Tree for Transonic Flow over ONERA M6 Wing

This first test case is transonic flow over an ONERA M6 wing configuration[29]. The M6 wing has a leading-edge sweep angle of 30 degrees, an aspect ratio of 3.8, and a taper ratio of 0.562. The airfoil section of the wing is the ONERA "D" airfoil, which is a 10% maximum thickness-to-chord ratio conventional section. The flow was first computed at a Mach number of 0.84, an angle of attack of 3.06 degrees and with inviscid flow assumption. This case was selected to demonstrate the superiority of the 2^N tree over Octree in efficiently capturing flow features. The starting coarse computational grid was generated using the Octree data structure, and is shown in Figure 3. The mesh consists of 14,141 cells and 47,904 faces and two layers of projected layer grids. Three levels of solution based grid adaptations were then performed. The adaptation criteria are pressure and Mach number gradients. With the Octree data structure, if a cell needs to be refined in any of the coordinate direction, the cell is refined in all directions. The level 3 Octree cartesian grid and 2^N tree Cartesian grid are shown in Figure 4. The Octree Cartesian grid is composed of 342,289 cells and 1,105,732 faces, while the 2^N tree Cartesian grid has only 88,296 cells and 300,573 faces. However the solution on the adapted grids are essentially identical, as shown in Figures 5 and 6. Figure 5 compares the pressure contours on the level 3 grids using Octree and 2^N tree, and Figure 6 presents the pressure coefficients on the wing surface at 44% semispan station. Note that the Cp profiles on the adapted grids of the same level using Octree and 2^N tree are essentially the same. With the 2^N tree, a 75% saving in CPU can be achieved over with the Octree.

This case was also computed assuming fully turbulent flow. The Reynolds number is 1.814×10^7. The k-e model with a wall function was used to simulate flow turbulence. Based on prior experiences, an average y+ value of 30-50 usually gives good results. Therefore during the grid adaptation process, the average target y+ value is around 40. The initial and final computational grids for the turbulent flow case is shown in Figure. The initial grid has 40,480 cells, and the final level 3 grid has

413,551 cells. The pressure contours on the initial grid are compared with the contours on the final grid in Figure 8. Note the dramatic difference in the resolution of the overall flow features. The computed surface Cp profiles are compared with experimental data at four spanwise sections in Figure 9. It is observed that better agreement was obtained with grid adaptations. The surface Cp profile at 44% spanwise section is also compared to the inviscid simulation, and the experimental data in Figure 10. Note that the turbulent flow simulation has slightly better agreement with the experimental data. Both the inviscid and turbulent calculations over-predicted the expansion region on top of the wing.

High Angle of Attack Missile Aerodynamics

The second test case is performed for an ogive/cylinder configuration[30], for which extensive experimental data is available for validation. The model configuration of interest is a 3-caliber ogive with a 10-caliber cylindrical afterbody. The geometry of the configuration, and the initial viscous Cartesian grid is displayed in Figure 11. The flow was computed at the following conditions: Mach = 1.8, α = 14 degrees, Re = 6.56×10^6/m. The diameter of the cylinder is 3.7 inches. The flow field is characterized by large separation regions, and is a considerable challenge for turbulence models. In this study, the k-e model with wall function was used in the simulation. Several levels of grid adaptations were performed after the solutions were converged on the coarse grids. Again the target average Y+ value is 25, which was achieved on the level 2 grid. The adaptation criterion used is total pressure gradient, which is designed to capture the complex vortex pattern of the flow. The level 1 to 4 grids are displayed in Figure 12. The final level 4 grid has 473,153 cells and 1,516,791 faces. Note that the development of the complex vortex pattern is evident on the level 4 grid. It is seen that the grid was also refined near the shock wave. A picture of the flow field is shown in Figure 13, which displays total pressure contours, and the computational grid. The surface is colored with pressure distributions. Computed pressure coefficient profiles are compared with experimental data at several cross section stations in Figure 14. Note that at least visibly the turbulent solutions are approaching grid independence because the difference in Cp computed with the level 3 and level 4 grids is very small. This is the first evidence that grid independent turbulent solutions can be achieved with anisotropic grid adaptations. Generally speaking the agreement between the numerical simulation and the experimental data is fairly good. It can be said that improvement in turbulence models is needed to accurately capture the large separation in this case.

CONCLUSIONS

The 2^n tree adaptive viscous Cartesian grid method has been tested with turbulent flows. It is confirmed that the 2^n tree is much more efficient in capturing flow features than the octree data structure. In the inviscid flow case with the M6 wing geometry, a 75% saving in total number of cells can be achieved. With anisotropic grid adaptations for turbulent flow simulation, it is clearly demonstrated that grid independent turbulent solutions are achievable. Drastic improvements in solution accuracy are demonstrated with solution-based grid adaptations for both the M6 wing and the high angle of attack missile cases.

The adaptation criteria are shown to be very effective in capturing shock waves, wakes, and complex vortex structures. The use of anisotropic grid adaptations allows these features to be captured very efficiently.

It is also found that improvements in turbulence models are required to allow more accurate predictions of highly separated flows. Future work includes demonstrating the viscous Cartesian method for complete aircraft configurations and improve the grid quality in highly skewed regions of the computational grid.

ACKNOWLEDGMENT

The authors gratefully acknowledge valuable discussions with many colleagues in CFDRC, in particular, N. Hariharan, A. Habchi, and V. Adumitroaie. The technical guidance and advice of A.J. Przekwas and A.K. Singhal of CFDRC were very valuable. The study has been funded under a Navy SBIR project from NAWC/AD, the U.S. Navy. We thank D. Grove of NAWC/AD for very helpful discussions and for serving as the Technical Monitor. G. Laskowski of Stanford University is acknowledged for performing the M6 wing turbulent flow calculations. The National Science Foundation is acknowledged for funding the flow solver development.

REFERENCES

1. Barth, T.J., and Frederickson, P.O., Higher–Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction, AIAA Paper 90–0013, 1990.

2. Batina, J.T., Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex Aircraft Aerodynamic Analysis, AIAA Journal, vol. 29, pp. 327–333, 1991.
3. Venkatakrishnan, V., Convergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters," J. Comput. Phys., vol. 118, 1995, pp. 120–130.
4. Anderson, W.K. and Bonhaus, D.L., An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids, Computers Fluids, 23 (1994), pp. 1–21.
5. Connell, S.D. and Holmes, D.G., A 3D Unstructured Adaptive Multigrid Scheme for the Euler Equations, AIAA paper 93–3339–CP, 1993.
6. Schneiders, R., "Automatic Generation of Hexahedral Finite Element Meshes," in Proceedings of 4th International Meshing Roundtable '95, October 1995, Albuquerque, NM.
7. Nakahashi, K., "Adaptive–Prismatic–Grid Method for External Viscous Flow Computations," in Proceedings of 11th AIAA Computational Fluid Dynamics Conference, July 1993, Orlando, FL. AIAA–93–3314–CP.
8. Kallinderis, Y., Khawaja, A. and McMorris, H., "Hybrid Prismatic/Tetrahedral Grid Generation for Complex Geometries," AIAA Paper No. 95–0210, 1995.
9. Coirier, W.J. and Jorgenson, P.C.E., "A Mixed Volume Grid Approach for the Euler and Navier–Stokes Equations," AIAA Paper No. 96–0762, 1996.
10. Lohner, R. and Parikh, P., "Generation of Three–Dimensional Unstructured Grids by Advancing Front Method," International J. Numerical Method in Fluids, vol. 8, pp. 1135–1144, 1988.
11. Baker, T.J., "Three–Dimensional Mesh Generation by Triangulation of Arbitrary PointSets," AIAA Paper No. 87–1124, 1987.
12. Berger, M.J. and LeVeque, R.J., An Adaptive Cartesian Mesh Algorithm for the Euler Euations in Arbitrary Geometries, AIAA–89–1930–CP, June 1989.
13. DeZeeuw, D. and Powell, K., An Adaptively Refined Cartesian Mesh Solver for the Euler Equations, AIAA–91–1542–CP, 1991.
14. Coirier, W.J. and Powell, K.G., Solution–Adaptive Cartesian Cell Approach for Viscous and Inviscid Flows, AIAA J., vol. 34, no.5, pp. 938–945, 1996.
15. Bayyuk, A.A., Powell, K.G., and van Leer, B., A Simulation Technique for 2D Unsteady Inviscid Flows Around Arbitrarily Moving and Deforming Bodies of Arbitrarily Geometry, AIAA Paper 93–3391–CP, 1993
16. Melton, J.E., Enomoto, F.Y., and Berger, M.J., 3D Automatic Cartesian Grid Generation for Euler Flows, AIAA Paper 93–3386–CP.
17. Aftosmis, M.J., Berger, M.J. and Melton, J.E., "Robust and Efficient Cartesian Mesh Generation for Component–Based Geometry," AIAA Paper No. 97–0196, 1997.
18. Karman, S.L.,"SPLITFLOW: A 3D Unstructured Cartesian/Prismatic Grid CFD Code for Complete Geometries," AIAA–95–0343, 1995.
19. Wang, Z.J., "Proof of Concept – An Automatic CFD Computing Environment with a Cartesian/ Prism Grid Generator, Grid Adaptor and Flow Solver", in Proceedings of 4th International Meshing Roundtable, pp 87–99, October 1995, Albuquerque, New Mexico.
20. Wang, Z.J., “A Quadtree-Based Adaptive Cartesian/ Quad Grid Flow Solver for Navier-Stokes Equations,” Computers & Fluids, vol. 27, No. 4, pp. 529-549, 1998.
21. Smith, R.J., and Leschziner, M.A., “A Novel Approach To Engineering Computations for Complex Aerodynamic Flows,” Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Field Simulations, Mississippi State University, MS, 1996.
22. Tchon, K.F., Hirsch, C. and Schneiders, R., “Octree-Based Hexahedral Mesh Generation for Viscous Flow Simulations,” AIAA-97-1980-CP, 1997.
23. Wang, Z.J., “An Automated Viscous Adaptive Cartesian Grid Generation Method for Complex Geometries,” in Proceedings of the 6th International Conference on Numerical Grid Generation in Computational Field Simulations, University of Greenwich, London, England, 1998.
24. Wang, Z.J., Chen, R.F., Hariharan, N. and Przekwas, A.J., “A 2N Tree Based Automated Viscous Cartesian Grid Methodology for Feature Capturing,” in Proceedings of 14th AIAA CFD Conference, AIAA-99-3300-CP.
25. Bonet, J.A. and Peraire, “An Alternating Digital Tree (ADT) Algorithm for 3D Geometric Searching and Intersection Problems,” International J. Numerical Methods in Engineering, vol. 31, pp. 1-17, 1991.
26. Roe, P.L., Approximate Riemann Solvers, Parameter Vectors, and DifferenceSchemes, Journal of Computational Physics, vol. 43, pp. 357, 1983.
27. Chen, R.F. and Wang, Z.J., “An Improved LU-SGS Scheme with Faster Convergence for Unstructured Grids of Arbitrary Topology,” AIAA-99-0935, 1999.

28. Launder, B.E and Spalding, D.B., Computer Methods in Applied Mechanics and Engineering, 1974, vol. 3, pp 269.
29. Schmitt, V. and Charpin, F., "Pressure Distributions on the ONERA M6-wing at Transonic Mach Numbers," Experiment Data Base for Computer Program Assessment, AGARD AR-138, 1979.
30. Sturek, W.B., Birch, T., Lauzon, M., Housh, C., Manter, J., Josyula, E. and Soni, B., "The Application of CFD to the Prediction of Missile Body Vortices," AIAA-97-0637, 1997.
31. IIeim, E.R., "CFD Wing/Pylon/Finned Store Mutual Interference Wind Tunnel Experiment, AEDC-TSR-91-P4, Arnold Engineering Development Center, Arnold AFB, TN, Jan. 1991.

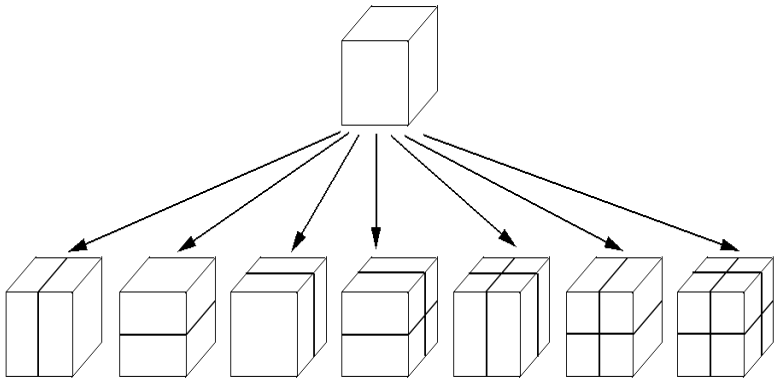

Figure 1. Cartesian Cell Subdivisions Supported by 2^n tree Data Structure

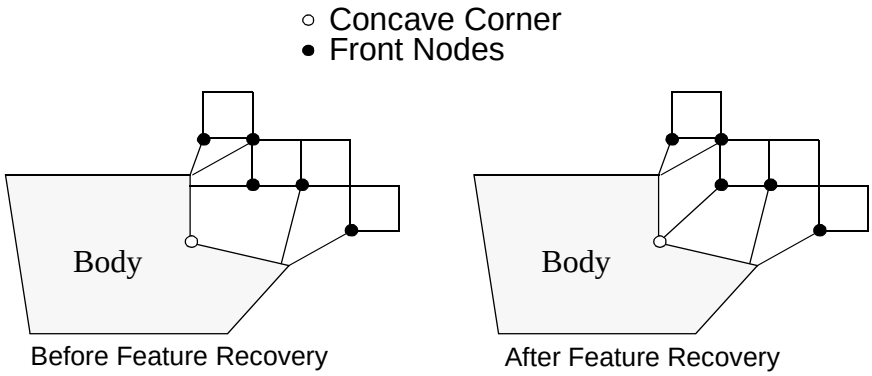

Figure 2. Illustration of a Smeared Concave Corner in Front Projection

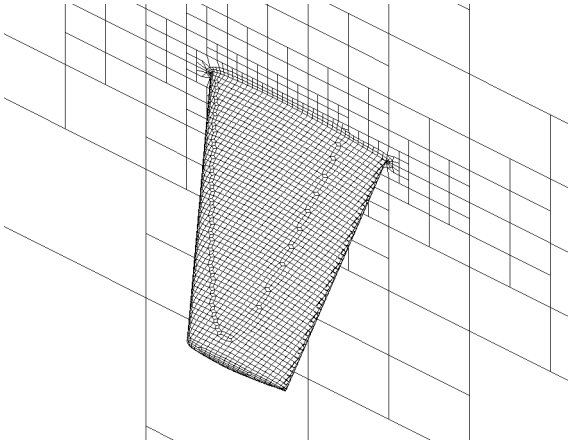

Figure 3. Initial Viscous Cartesian Grid for the ONERA M6 Wing (14141 cells and 47904 faces)

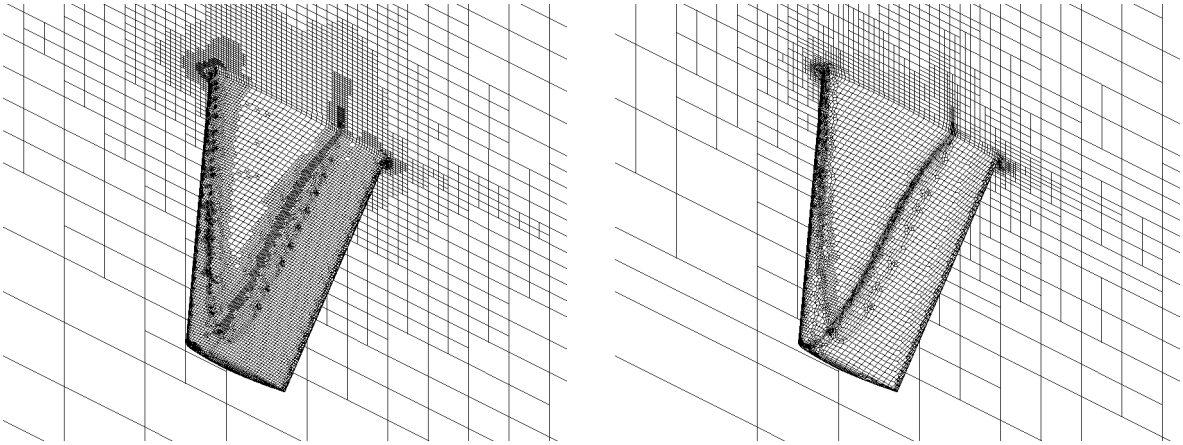

Figure 4. Level 3 Solution-Adaptive Cartesian Grids Using Octree and 2^N Tree

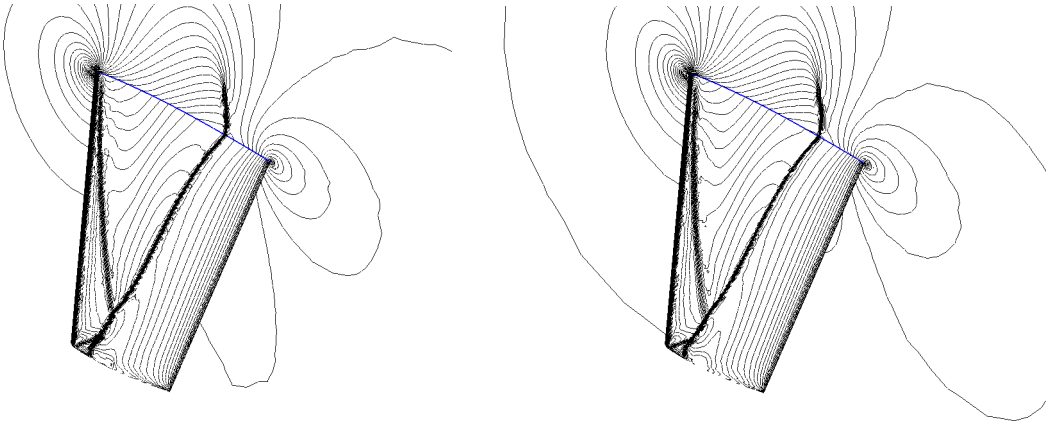

Figure 5. Computed Pressure Contour on the Level 3 Solution-Adaptive Cartesian Grids Using Octree (a) and 2^N Tree (b)

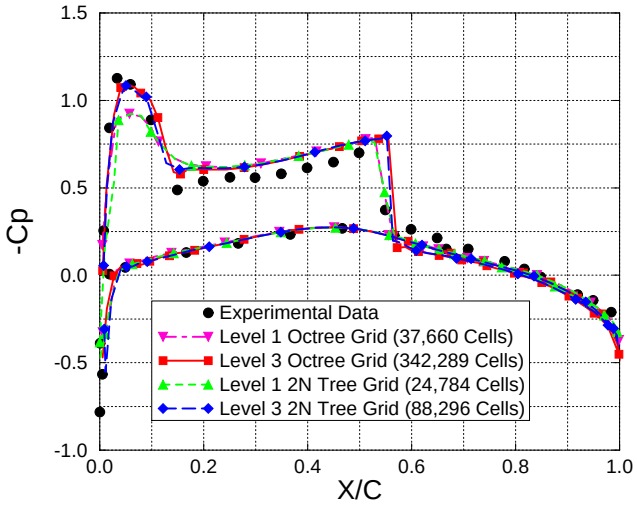

Figure 6. Comparison between Computed and Experimental Surface Pressure Coefficient at 44% Semispan Station

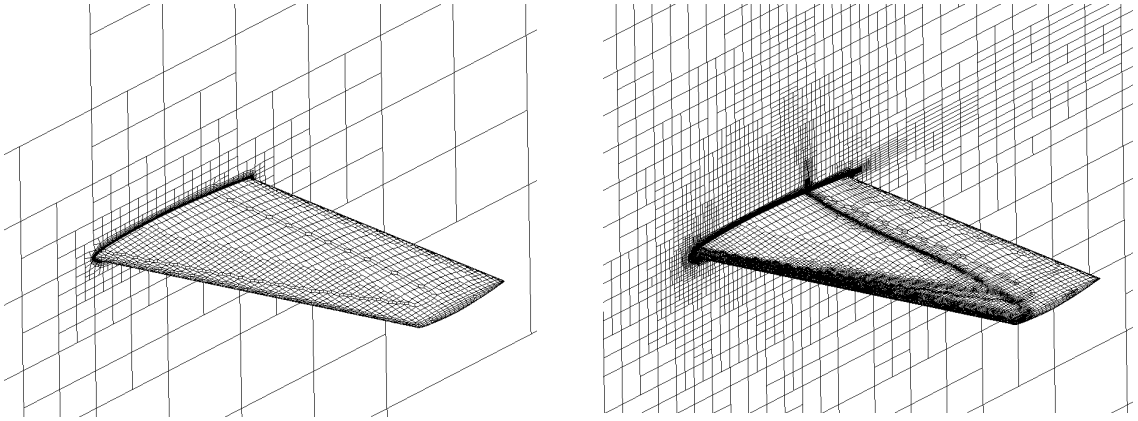

Figure 7. The Initial and Level 3 Adaptive Grids for Turbulent Flow Case

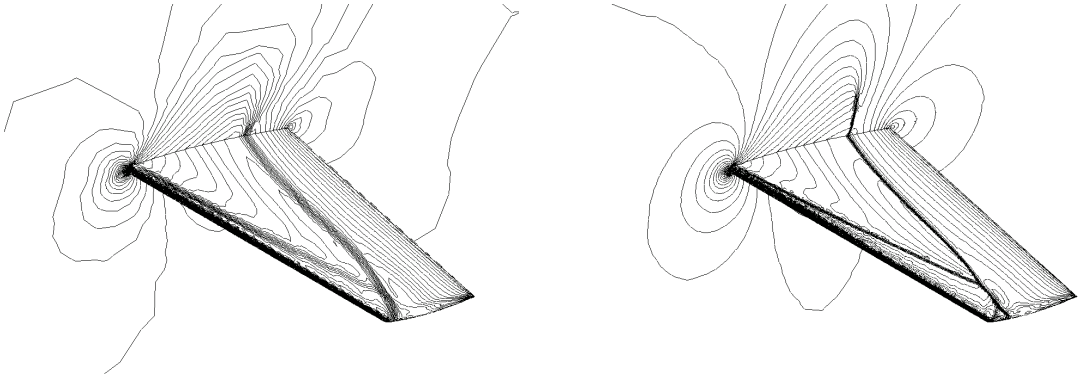

Figure 8. The Pressure Contours on the Initial and Level 3 Grids

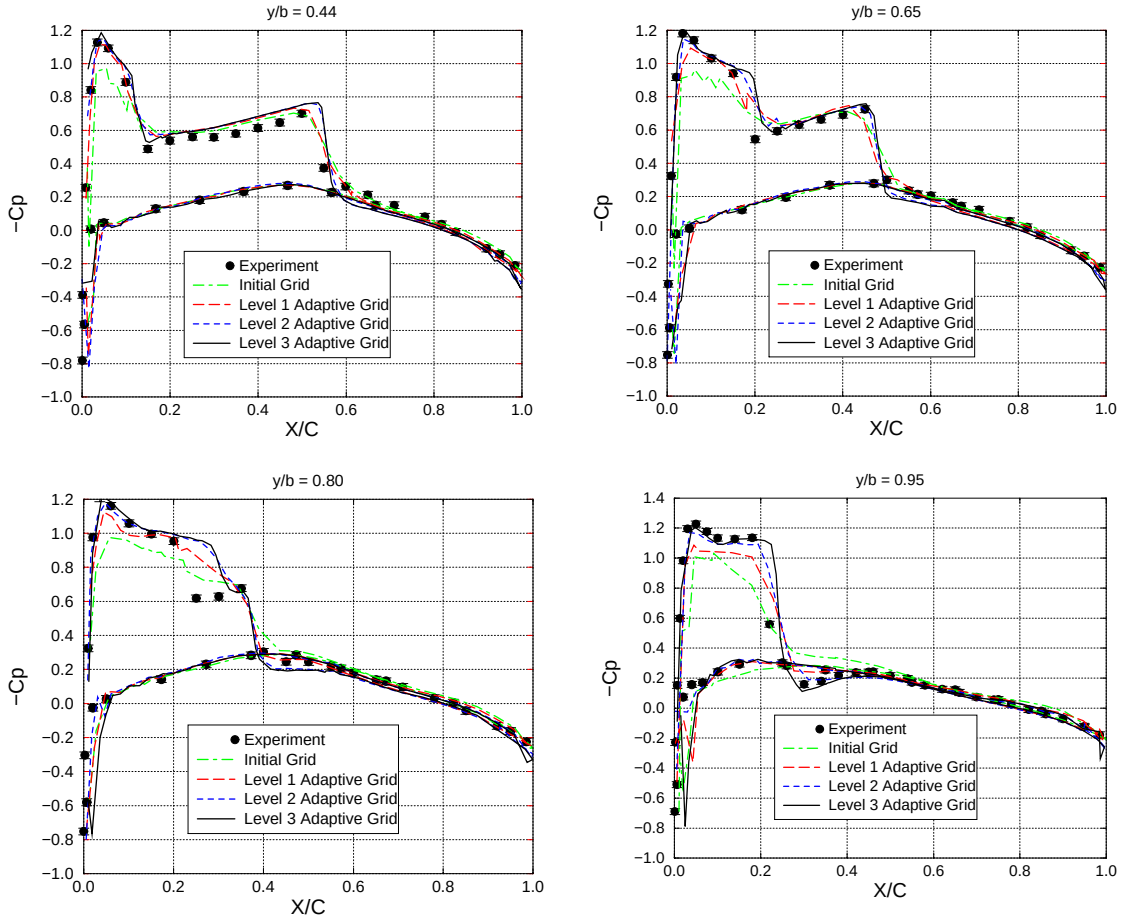

Figure 9. Comparison of Computed Cp Profiles with Experimental Data

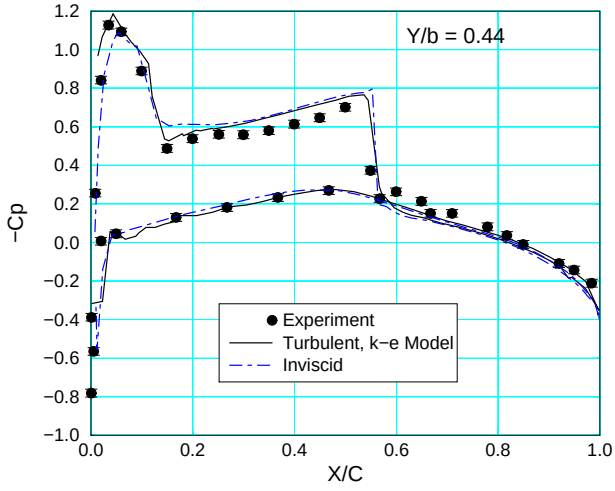

Figure 10. Comparison of Cp Profiles between Inviscid, Turbulent Simulations and Experiment

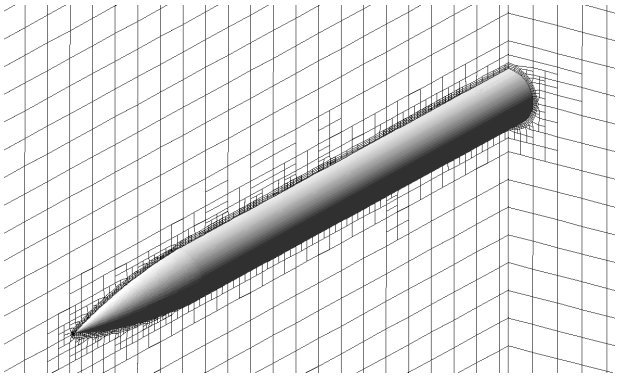

Figure 11. The Initial Viscous Cartesian Grid for the Ogive/Cylinder Missile Configuration

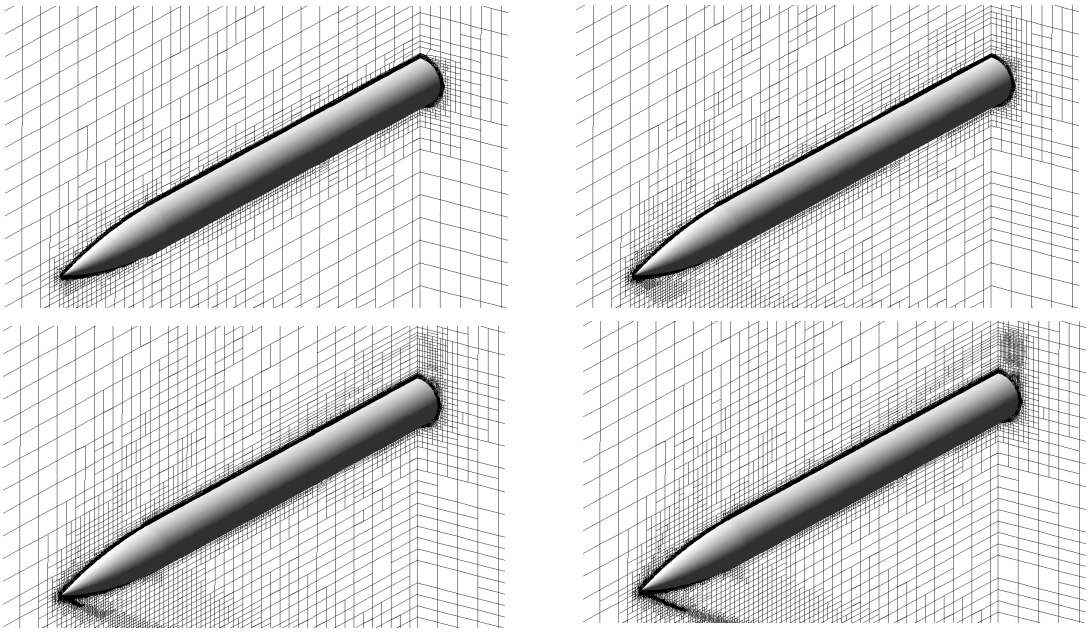

Figure 12. The Level 1-4 Solution Adaptive Grids for the Ogive/Cylinder Missile Configuration

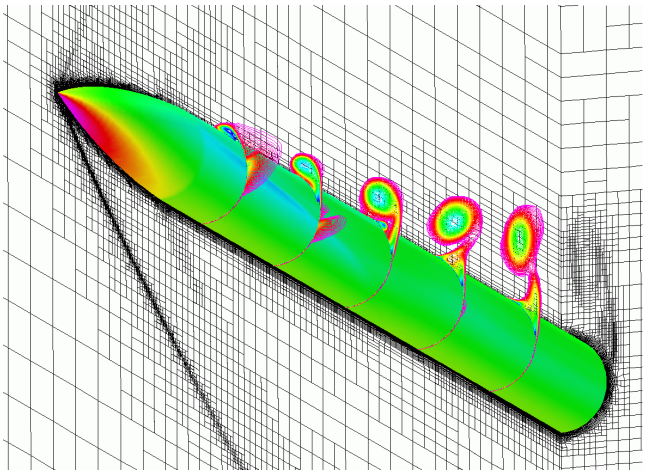

Figure 13. Computed Total Pressure Contours and Surface Pressure Distribution at Mach = 1.8, a = 14 Degrees

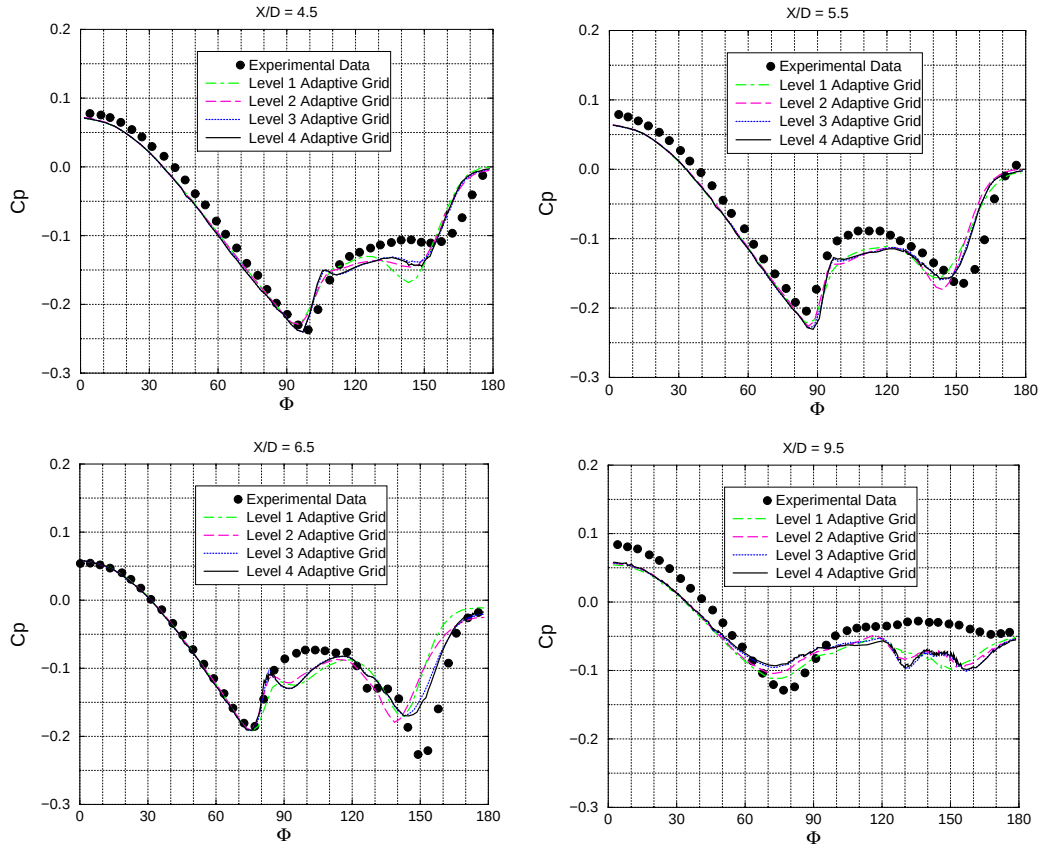

Figure 14. Comparison Between Computed and Experimental Pressure Coefficient Profiles at Several Cross Section Stations