AIAA Journal
Vol. 48, No. 11, November 2010

Hybrid Grid Generation Method for Complex Geometries

Hong Luo* and Seth Spiegel†
North Carolina State University, Raleigh, North Carolina 27695
and
Rainald Löhner‡
George Mason University, Fairfax, Virginia 22030

DOI: 10.2514/1.J050491

A hybrid mesh generation method is described to discretize complex geometries. The idea behind this hybrid method is to combine the orthogonality and directionality of a structured grid, the efficiency and simplicity of a Cartesian grid, and the flexibility and ease of an unstructured grid in an attempt to develop an automatic, robust, and fast hybrid mesh generation method for configurations of engineering interest. A semistructured quadrilateral grid is first generated on the wetted surfaces. A background Cartesian grid, covering the domain of interest, is then constructed using a Quadtree-based Cartesian Method. Those Cartesian cells overlapping with the semistructured grids or locating outside of computational domain are then removed using an Alternating Digital Tree method. Finally, an unstructured grid generation method is used to generate unstructured triangular cells to fill all empty regions in the domain as a result of the trimming process. The automatic placement of sources at the geometrical irregularities is developed to render these regions isotropic, thus effectively overcoming the difficulty of generating highly stretched good-quality elements in these regions. The self-dividing of the exposed semistructured elements with high aspect ratio and the adaptation of the background mesh using the cell size information from the exposed semistructured elements for generating Cartesian cells are introduced to improve the quality of unstructured triangular elements and guarantee the success of the unstructured grid generation in the void regions. The developed hybrid grid generation method is used to generate a hybrid grid for a number of test cases, demonstrating its ability and robustness to mesh complex configurations.

I. Introduction

COMPUTATIONAL fluid dynamics (CFD) has become an indispensable tool for a variety of applications in science and engineering. One of the greatest obstacles for widespread use of CFD methods is the level of expertise and the time required to generate a grid for a complex geometry. General speaking, the grid generation methods can be classified by the mesh they generate as structured grid generation methods, unstructured grid generation methods, and Cartesian grid generation methods. The structured grid generation methods alone are not practical for engineering applications, as they have a disadvantage for griding complex geometries. More often, they are used in the context of Chimera or overlapping approaches [1,2] to simplify the grid generation process for a complex configuration. The difficulty of generating a structured grid for complex geometries and the desire in the engineering community to simulate numerically flows past increasingly complex geometries have fueled interest in the development of unstructured grid methods [3–7]. Unstructured grids provide great flexibility in dealing with the complex geometries encountered in practice and offer a natural framework for solution-adaptive mesh refinement. Over the course of last decade, significant progress has been made on developing unstructured triangular/tetrahedral grid generation methods for complex geometries. However, the computational costs and memory requirements for unstructured triangular/tetrahedral grids are generally higher than for structured grids. Although the solution accuracy may not be strongly affected by element type even in the boundary layers, computational efficiency can benefit substantially through the use of prismatic elements in the boundary layers and Cartesian cells in the inviscid regions. This is due to a simple fact that approximately, 5–6 times more tetrahedra than hexahedra are required to fill a given region with a fixed number of nodes. Although the boundary-layer regions occupy only a small portion of the computational domain, it is not uncommon for more than half of the mesh resolution to be packed into this small region, and thus the quadrilateral elements in 2D and prismatic elements in 3D can lead to a significant saving in both memory requirements and computational costs. From the computational efficiency point of view, unstructured triangular/tetrahedral elements should be kept minimal. The advantages of the Cartesian grid approaches [8–12] include ease of grid generation, lower computational storage requirements, and significantly less operational count per cell. However, the main challenge in using Cartesian methods is how to deal with arbitrary boundaries, as the grids are not body-aligned. The cells of a Cartesian mesh near the body can extend through surfaces of boundaries. Accurate means of representing boundary conditions in cells that intersect surfaces are essential for successful Cartesian methods. Because each grid type has its own advantages and disadvantages, the best grid approach is clearly a hybrid one that combines the advantages and strengths of all these three grid types. As a result, grid generation efforts have been focused on the development of robust, reliable, and fast hybrid mesh generation methods and research work in this area has been widely published in the literature [13–19]. Furthermore, in most flow problems, different flow regions will require different types of grids to model flow physics accurately and obtain numerical solutions efficiently. At the high Reynolds numbers encountered in typical aerodynamic applications, viscous effects are important in relatively thin boundary layers, in close proximity to the solid surfaces. In these regions, the normal length scale can be many orders of magnitude smaller than the tangential length scale. Consequently, the viscous regions are best represented by highly stretched quadrilateral/prismatic elements, which possess an inherent feature of orthogonality and directionality. In contrast to the viscous predominant flow regions, local isotropy is desired in the mainly inviscid flow regions

Presented as Paper 2008-531 at the 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 7–10 January 2008; received 6 February 2010; revision received 3 June 2010; accepted for publication 9 August 2010.

*Associate Professor, Department of Mechanical and Aerospace Engineering. Senior Member AIAA.

†Ph.D. Student, Department of Mechanical and Aerospace Engineering.

‡Distinguished Professor, Department of Computational and Data Sciences. Member AIAA.

away from the bodies, which do not exhibit the directionality. Therefore, the inviscid regions are best discretized by Cartesian cells.

The objective of the efforts presented in this paper is to develop a hybrid mesh generation method for complex geometries in 2D, designed to achieve the maximum efficiency and minimum user intervention. The idea behind this hybrid grid generation method is to combine the orthogonality and directionality of a structured grid, the efficiency and simplicity of a Cartesian grid, and the flexibility and ease of an unstructured grid in an attempt to obtain an automatic, robust, and fast mesh generation method for complex configurations. In the anisotropic viscous regions of flows, high aspect ratio quadrilaterals are generated, and in the isotropic inviscid regions, Cartesian squares are generated, while triangles are generated to bridge these two grids and in the vicinity of boundaries. Because a majority of computational cells consist of the Cartesian cells and quadrilateral cells, the resulting hybrid grid is very efficient in terms of both computational costs and storage requirements for flow solutions. As the unstructured triangular cells are used in the vicinity of boundaries, the developed hybrid mesh generation method can easily be used for complex geometries without losing the robustness. The method uses as input the boundary curve definition and a background grid and/or sources specifying the desired spatial distribution of the mesh size in the computational domain. The boundary curves are first discretized in such a way that will render geometric singularities isotropic using the automatic placement of sources at these singularities, thus avoiding the difficulty of generating a good-quality highly stretched elements in these regions. A semistructured quadrilateral grid is then generated on the wetted boundary faces. A background Cartesian grid, covering the computational domain, is then automatically constructed using a Quadtree-based Cartesian Method. Those Cartesian cells overlapping with the semistructured grids or locating outside of computational domain are then removed using an Alternate Digital Tree method. Finally, an unstructured grid generation method is used to generate unstructured triangular cells to fill all voids in the domain as a result of the trimming process. Particular attention is focused on developing strategies that will improve the quality of unstructured triangular elements and ensure the success of the triangulation in the void regions. This is accomplished by self-dividing of the exposed semistructured elements with high aspect ratio, and the adaptation of the background mesh using the cell size information from the exposed semistructured elements for generating Cartesian cells. The developed hybrid grid generation method is used to generate a hybrid grid for a number of test cases, demonstrating its ability and robustness to mesh complex configurations.

II. Hybrid Grid Generation Method

The hybrid grid generation method will automatically generate a hybrid semistructured quadrilateral, unstructured Cartesian, and triangular grid, where high aspect ratio quadrilateral elements will be used in the boundary layers, cells in inviscid regions will be of Cartesian type, and unstructured triangular cells will be only used to fill the resulting voids between these two grids in the vicinity of boundaries and interfaces, where they are really needed and especially advantageous. Consequently, this hybrid grid represents the best grid as the semistructured character of cells near the surface of a body markedly reduces the number of cells needed for the boundary layers, and unstructured Cartesian cells significantly reduces the number of elements in the inviscid regions, thus achieving the maximum efficiency. The generation of a hybrid mesh consists of the following four steps starting from a boundary curve definition and a background grid and/or sources prescribing the desired spatial distribution of the mesh size in the computational domain:

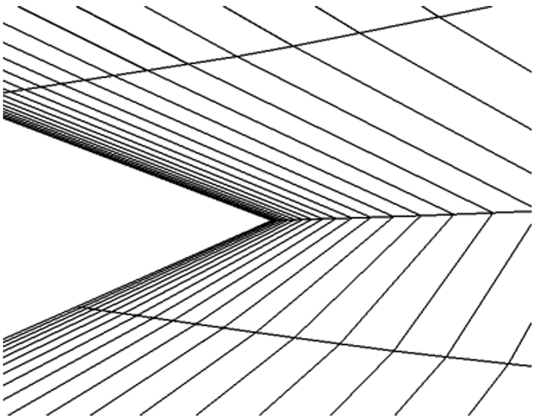

Fig. 1 Stretched elements are connected at sharp angles.

A. Discretization of Boundary Curves

The first step is to discretize the boundary curves, which are used to define a domain to be gridded. This involves the creation of edges and nodes on the boundary curves. It should be recognized that a boundary mesh, that is perfect and suitable for generating an isotropic Euler grid, is no longer appropriate for generating an anisotropic Navier–Stokes grid. In fact, a stretched viscous mesh with high aspect ratio of cells is only reasonable for smooth surfaces. In the case of a singularity like a corner, the stretched mesh should not be generated in the same way as for smooth surfaces. This is illustrated in Fig. 1, where the singularity unavoidably leads to extremely poor quality of elements at a sharp trailing edge, as long as one normal direction from the trailing edge is used to generate the grid. Even sophisticated elliptic or hyperbolic structured grid generators cannot guarantee an adequate mesh resolution in such regions. This issue can be addressed by introducing multiple normal directions at the corners. However, large discrepancies in neighboring cell sizes will still occur, leading to an inadequate quality of elements in the vicinity of the corners. This problem can be effectively overcome by simply generating isotropic elements at the corner, which will be gradually transferred to the stretched mesh away from the corner. This approach not only significantly improves the quality of grids in those regions, but also enhances the accuracy of flow solutions by providing a sufficient mesh resolution and robustness of numerical methods by preventing convergence of flow solution from stalling. The implementation of this approach is straightforward and easy. The corner points can be easily identified by measuring angles between the adjacent surfaces. Sources with a cell-size consistent with the boundary-layer thickness and the geometric progression normal to the surface are then automatically placed at these corner points. In this way, the resulting surface mesh is consistent with the topology of the viscous volume mesh, thus making the volume mesh generation more reliable and robust and greatly improves the quality of the overall grid. Figure 2 illustrates the resulting surface mesh for an RAE2822 airfoil, where an adequate clustering toward the trailing edge is clearly visible. This clustering allows the generation of isentropic elements, and thus avoids the difficulty of generating good-quality highly stretched quadrilateral elements in the vicinity of the trailing edge.

B. Generation of High Aspect Ratio Quadrilateral Elements in the Viscous Regions

After the generation of a surface mesh, a structured grid is then generated in the viscous regions. A preliminary step toward the generation of the structured volume mesh is the computation of an average surface normal for all the boundary grid points. The surface normals at the boundary nodes are computed by looping over the edges and scattering the contribution due to the edge to each of its two nodes. Then a mesh of quadrilateral cells is formed using the normals at boundary points, the information about the boundary-layer thickness, the number of boundary points in the boundary layers, and the geometric progression normal to the surface, which is provided by a user in the background grid. Finally, various criteria are used to remove the unacceptable elements that overlap other cells, do not have the desired shape or size or cross boundaries. Clearly, the development of good element removal criteria is critical to generate good semistructured quadrilateral grids. The criteria to be considered

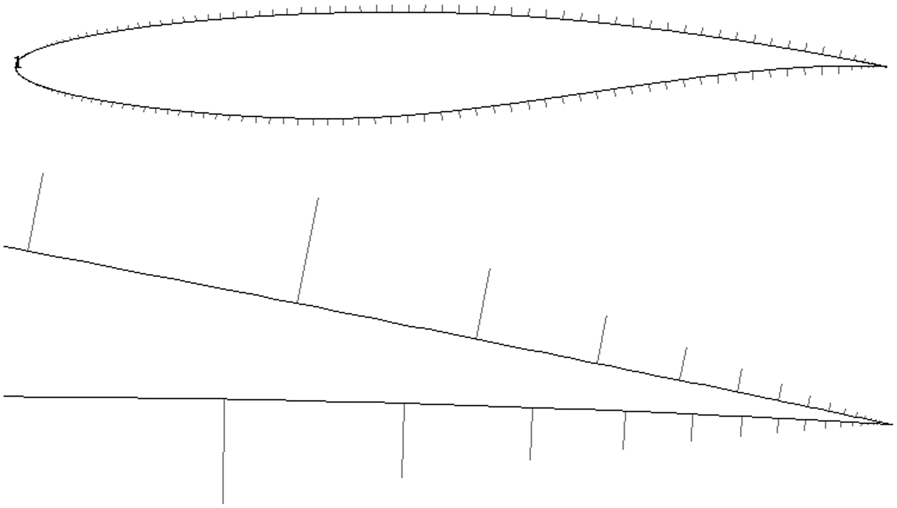

Fig. 2 Surface mesh for RAE2822 airfoil and close-up of surface mesh at the trailing-edge region.

are: element size, element shape, element overlap, and element crossing of boundary faces.

1. Element Shape

As a result of folding away from concave surface, elements with negative Jacobians may appear. All elements with negative volumes are marked for deletion. One has observed that typically the elements adjacent to negative elements tend to be highly deformed. Therefore, all elements that have points in common with negative elements are also removed. Obviously, this one-pass procedure can be extended to several passes, i.e., neighbors of neighbors, etc. Our experience indicates, however, that one pass is sufficient for most cases.

2. Element Shape

The aim of a semistructured mesh close to a wall is to provide elements with very small size normal to the wall and reasonable size along the wall. Because of different meshing requirements along the wall, elements that are longer in the direction normal to the wall than along the wall may appear. For the semistructured grids, the element and point numbering is known. Therefore, a local element analysis can be performed to determine whether the element has side-ratios that are consistent with boundary-layer gridding. All elements that do not satisfy this criterion are marked for deletion.

3. Overlapping Elements

Overlapping elements will occur in regions close to concave surfaces with high curvature, or when the semistructured grids of two close objects overlap. Another possible scenario is the overlap of the semistructured grids of mixing wakes. The main criterion employed is to keep the smaller elements whenever an overlap occurs. In this way, the small elements close to surfaces are always retained.

4. Elements Crossing Boundary Faces

In regions where the distance between surfaces is very small, the crossing of boundary faces by elements from the semistructured region is likely to occur. As this test is performed after the element crossing tests are conducted, the only boundaries that need to be treated are those that have no semistructured grid attached to it. To detect if overlapping occurs, we loop over the surface faces, seeing if any element crosses it. The face-crossing check looks essentially the same as the check for overlapping elements.

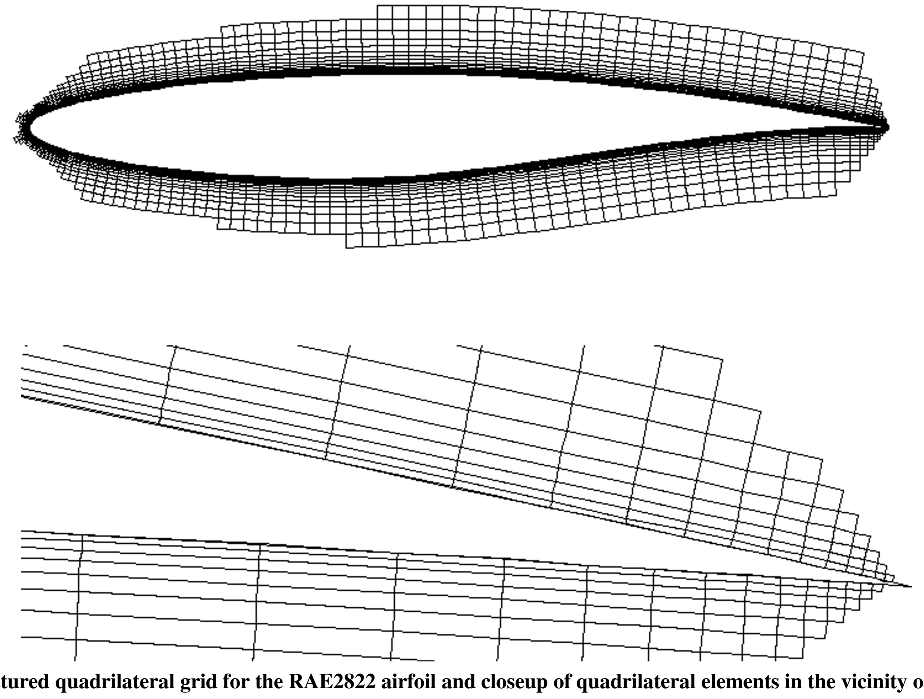

Fig. 3 Semistructured quadrilateral grid for the RAE2822 airfoil and closeup of quadrilateral elements in the vicinity of the trailing edge.

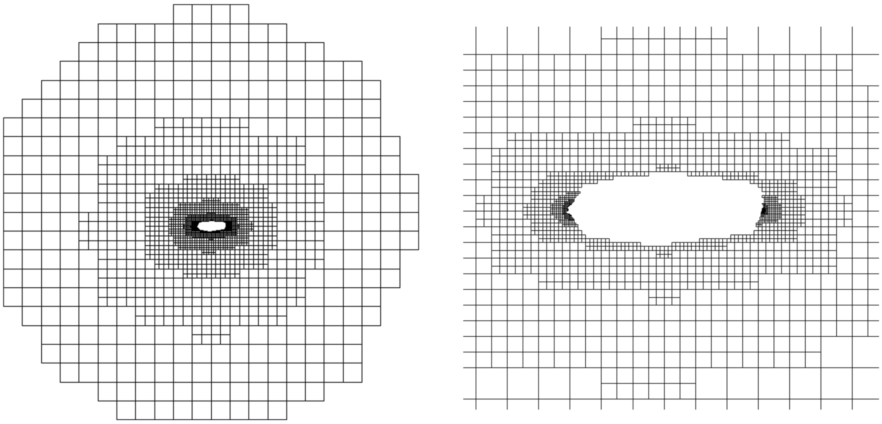

Fig. 4 Unstructured Cartesian grid in the inviscid regions for the RAE2822 airfoil and closeup of Cartesian cells in the vicinity of the airfoil.

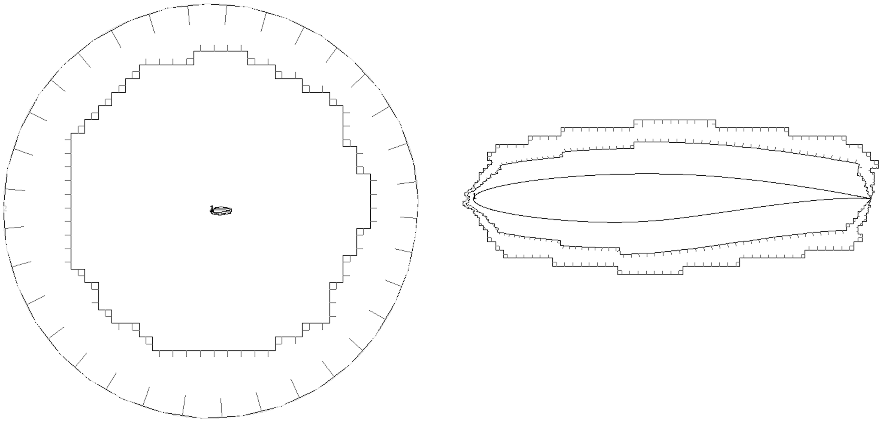

Fig. 5 The *empty* regions as a result of the trimming process and the resulting boundary triangulation, i.e., the initial front for these regions.

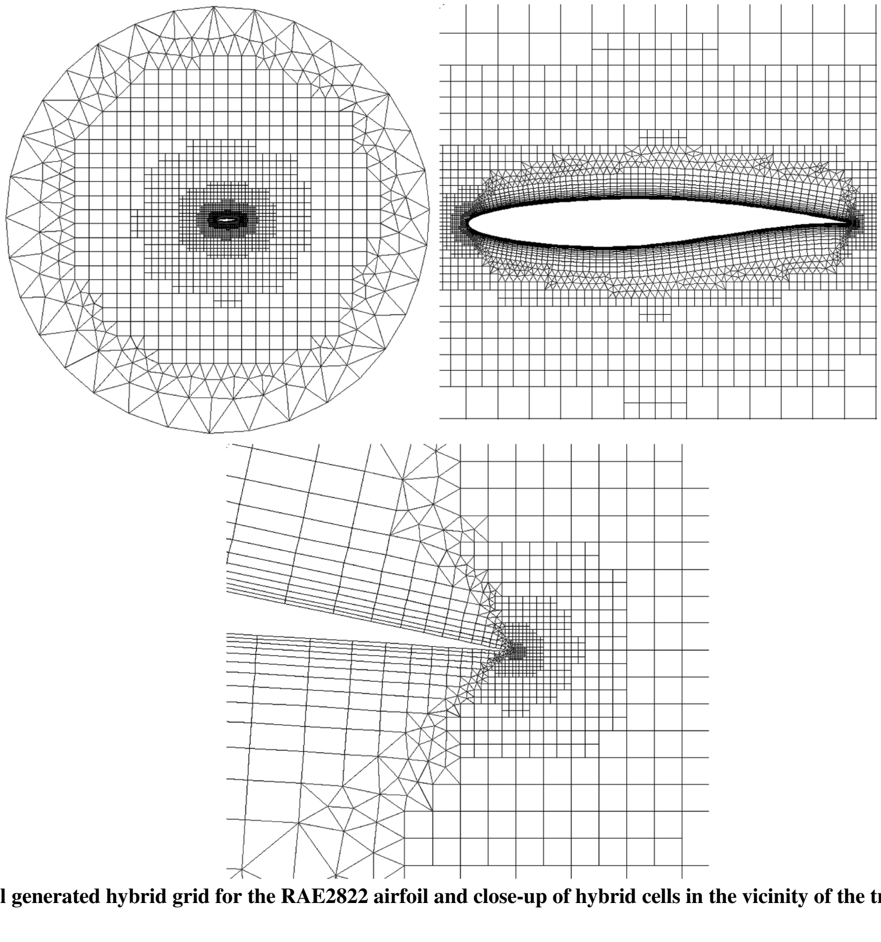

Fig. 6 Final generated hybrid grid for the RAE2822 airfoil and close-up of hybrid cells in the vicinity of the trailing edge.

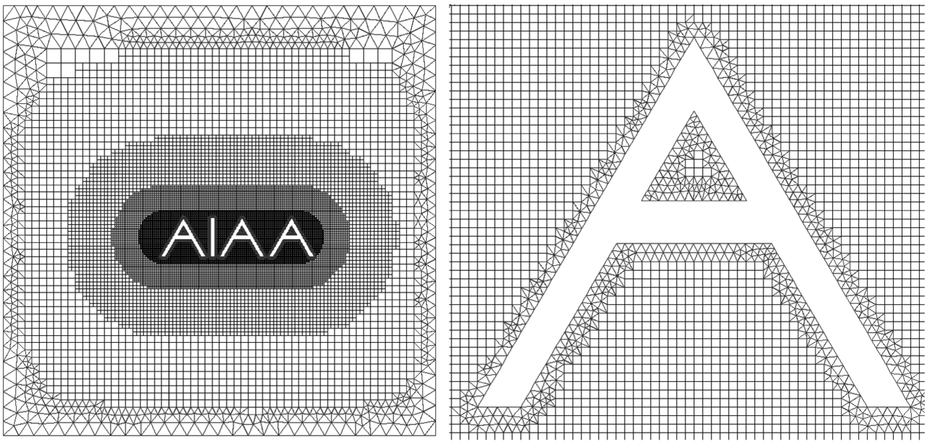

Fig. 7 Hybrid unstructured Cartesian and triangle grid for four letter AIAA configuration.

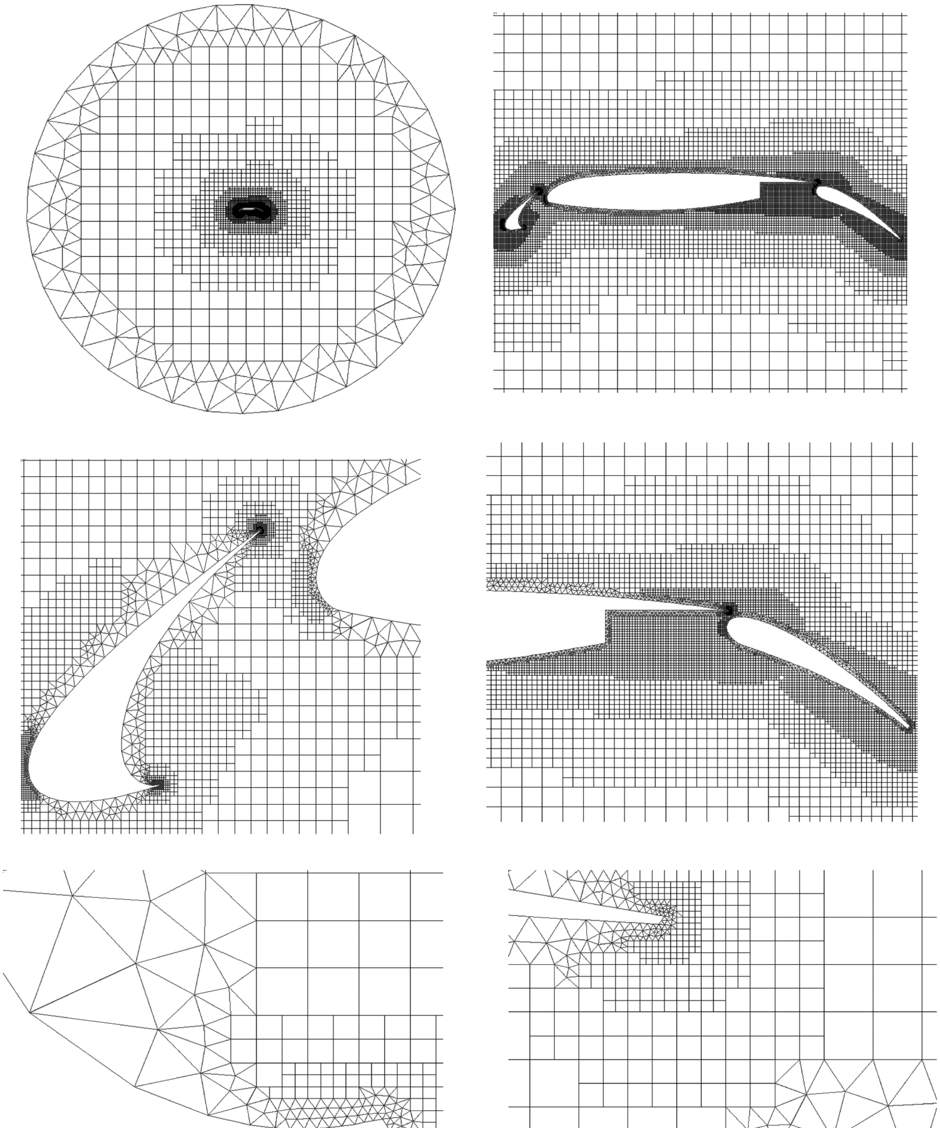

Fig. 8 Hybrid unstructured Cartesian and triangle grid for a three-element airfoil configuration.

Figure 3 shows the resulting quadrilateral grids for the RAE2822 airfoil. Notice that quadrilateral cells are removed in the vicinity of the trailing edge due to the automatic placement of a source, rendering the region around the trailing edge isotropic and avoiding the difficulty of generating good-quality highly stretched quadrilateral elements. The exposed faces in 2D will be used as the boundary faces to remove the baseline Cartesian cells outside of domain of interest.

C. Generation of Unstructured Cartesian Cells in the Inviscid Regions

The generation of unstructured Cartesian cells in the inviscid regions consists of the following two steps:

Generation of a Baseline Cartesian Grid: A baseline Cartesian cell (root cell) covering the computational domain is first generated, which is then subdivided into four Cartesian cells of equal size. In turn, these Cartesian cells are refined again until a desired distribution of cell sizes is achieved, which is determined by the grid sizes specified via a background mesh and/or sources provided by the user. To minimize the numerical errors between cells, a smoothing of the size difference among cells is performed, only allowing the size difference between two adjacent cells to two.

Cutoff of the Cartesian Cells in the Vicinity of the Computational Boundary: As the exposed faces and surface mesh are immersed in the baseline Cartesian grid, the cells that have at least one node out of the computational domain will be removed. This task is a specific application of the classic “point in polyhedron“ problem, which is resolved using the Alternating Digital Tree method and ray-casting approach. The description of Alternating Digital Tree can be found in [20]. Aftosmis et al. detailed an implementation of the ray-casting method in [21]. To have adequate space for the unstructured grid generation method to perform a triangulation, the neighboring cells to the cutoff cells are also removed. Obviously, this one-pass procedure can be extended to several passes, i.e., neighbors of neighbors, etc. Our experience indicates, however, that two passes are sufficient for most cases. Figure 4 shows the resulting Cartesian grids for the RAE2822 airfoil as an illustrative example.

D. Generation of Unstructured Triangular/Tetrahedral Elements in the Voids

The final step is to fill any voids in the computational domain as a result of the trimming process using unstructured triangular cells. Figure 5 shows the *empty* regions as a result of the trimming process, which need to be triangulated. The resulting boundary triangulation, i.e., the initial front for these regions is also illustrated in Fig. 5, where the short red lines represent the normals to the boundary faces. Both advancing front and Delaunay triangulation methods, which have reached a quite mature stage, have been implemented to generate triangular elements for these voids, although that the advancing front method is used in this work. The details of the advancing front method can be found in [4], where the generation of internal points is described in detail. The final hybrid grid generated by the present method is presented in Fig. 6.

The critical element of the hybrid mesh generation method described above is a valid triangulation for the voids in the domain. Ideally, all the boundary faces in close proximity for a void should have the same length scale, which, together with an adequate space as a result of trimming process, will lead to a good-quality triangulation. For the generation of an inviscid isotropic grid where the quadrilateral/prismatic elements do not exist, gridding these voids is simple and straightforward. Because both boundary triangulation and unstructured Cartesian cells are generated with the desired element size specified via a background mesh and/or sources, the size of the surface faces is consistent and does not exhibit widely differing length scales. However, for the generation of an anisotropic grid, where highly stretched quadrilateral elements exist, some of the

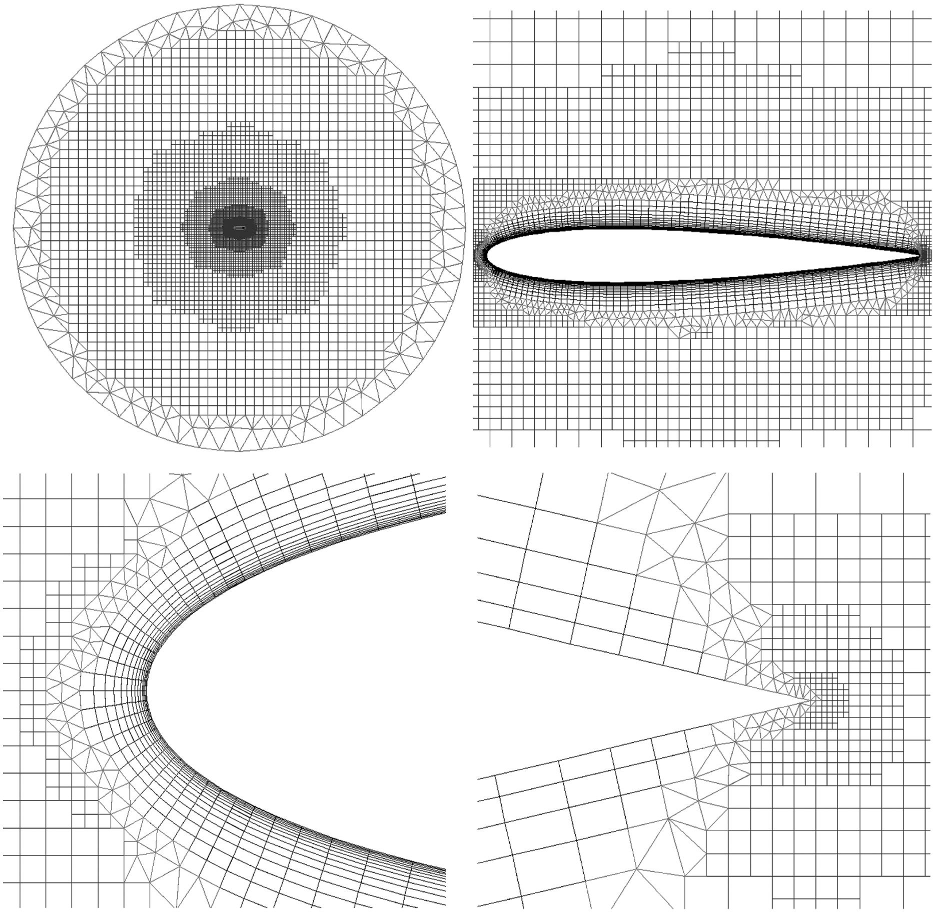

Fig. 9 Hybrid semistructured quadrilateral, unstructured Cartesian, and triangular grid for a NACA0012 airfoil.

exposed elements may have high aspect ratio. This can occurs if the number of boundary points is not sufficient, if the local background grid spacing is not prescribed properly, and if the interbody separation is not large enough in case of existence of multibodies. As a result, the length scales of some initial fronts, that are close to 1 another, may vary rapidly. This will have a disastrous effect on the quality of elements to be generated in the voids and can even lead to a failure of triangulation. The fundamental cause of this problem is that the background grid spacing function, which uses background and/or sources, can not completely control the size of semistructured quadrilateral/prismatic elements. To overcome this problem, two strategies are presented in this work. First the exposed elements having a high aspect ratio are subsequently divided into two by introducing a point in the middle of the exposed face until the aspect ratio for all the exposed elements is smaller than a prescribed value. This will avoid the abrupt variation of the length scales for all the exposed boundary faces from the semistructured quadrilateral grid, which will be served as a part of initial front for the triangulation of the voids. Second, the background grid is adapted by inserting the newly exposed points with the corrected spacing size, which will be used to generate the Cartesian cells. This will guarantee that the exposed boundary faces from the Cartesian cells will have the consistent and compatible spacing size as those from the semistructured grid and the boundary definition.

III. Examples

A. Four Letter AIAA Configuration

A hybrid unstructured Cartesian and triangular grid is generated for a square domain containing a four letter AIAA configuration. The hybrid grid, consisting of 16,628 Cartesian cells and 5347 triangles is shown in Fig. 7. Observe the smooth transition between the unstructured Cartesian and triangular grid regions.

B. Three-Element Airfoil Configuration

In this test case, an isotropic grid is generated for a three-element airfoil configuration. The hybrid unstructured Cartesian and triangle grid is shown in Fig. 8 and consists of 13,892 Cartesian cells and 3981 triangular elements. As in the previous example, a very smooth transition between the unstructured Cartesian and triangular grid regions is noticed everywhere for this complex configuration,

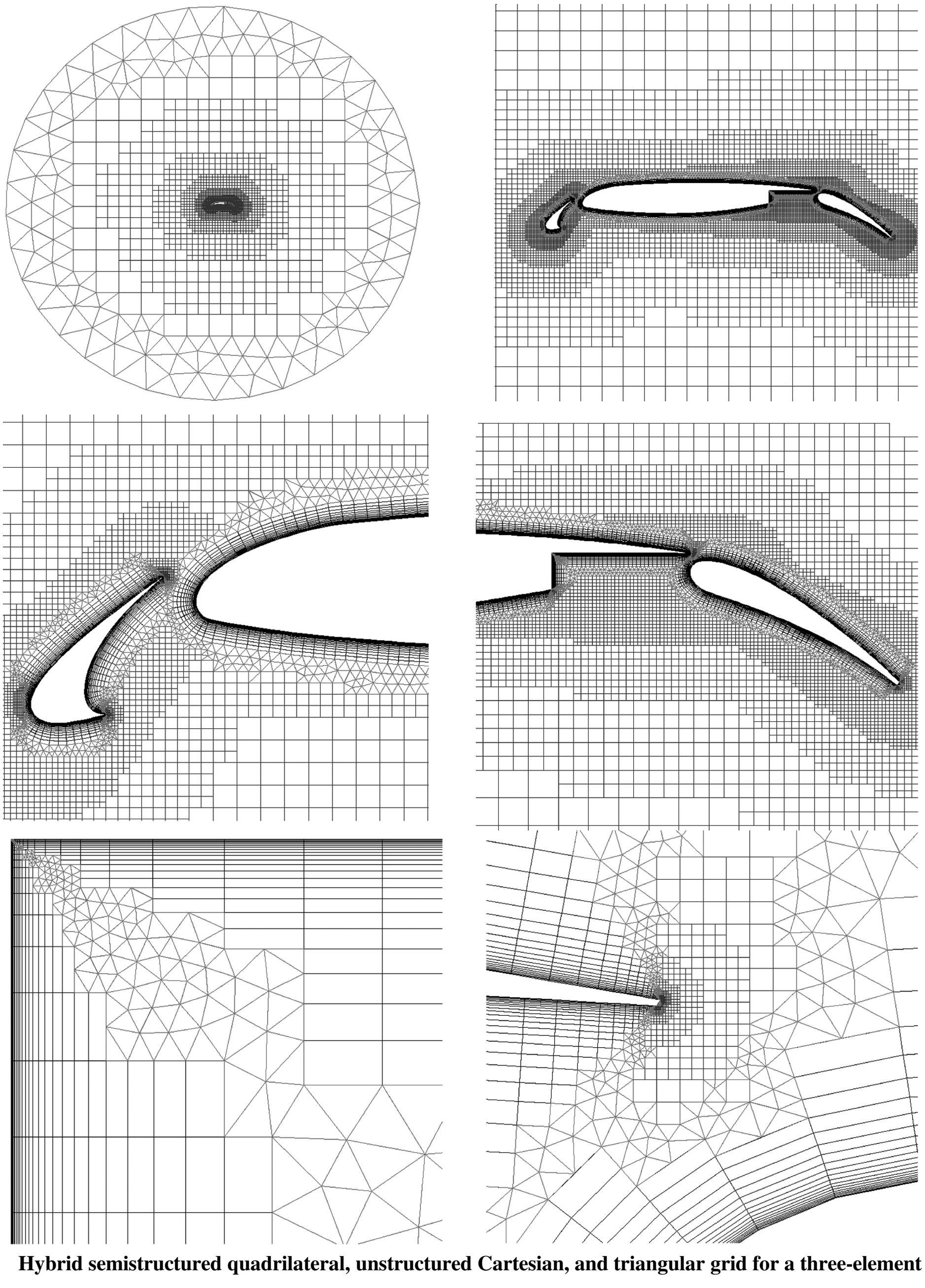

Fig. 10 Hybrid semistructured quadrilateral, unstructured Cartesian, and triangular grid for a three-element airfoil.

indicating the ability of the present hybrid grid generation method for generating a good-quality, isotropic, and mixed unstructured Cartesian and triangular grid for complex geometries.

C. NACA 0012 Airfoil Configuration

A hybrid quadrilateral, Cartesian, and triangular grid is generated for a NACA0012 airfoil using the present hybrid grid generation method. The resulting mixed grid is shown in Fig. 9 and consists of 2930 quadrilaterals, 9197 Cartesian cells, and 1201 triangles. The trailing edge as a geometrical singularity is identified and a source is automatically added, rendering the region around the trailing edge isotropic where the isotropic triangular and Cartesian cells are generated and avoiding the difficulty of generating good-quality highly stretched quadrilateral elements. Thus, the current hybrid grid generation method has the ability to provide the much-needed grid resolution for such singularity points. One can observe that the transition between the semistructured quadrilateral and unstructured triangular grid regions is as smooth as the one between unstructured Cartesian and triangular grid regions everywhere for this simple configuration, demonstrating the ability of the present hybrid grid generation method for generating a good-quality, and mixed semistructured quadrilateral, unstructured Cartesian, and unstructured triangular grid.

D. Multielement Airfoil Geometry

The same three-element airfoil geometry considered in test case B is reconsidered in this case for the generation of a hybrid quadrilateral, Cartesian, and triangular grid. The input file is practically the same as the one used in the test case B except the three additional parameters which specify the boundary thickness, the number of points in the boundary layer, and the geometric progression factor. The sources that are responsible for proper clustering toward the corners are defined and added automatically. The generated hybrid quadrilateral, Cartesian, and triangular grid is shown in Fig. 10 and consists of 12,179 quadrilateral elements, 13,245 Cartesian cells and 7877 triangular elements, where the quadrilateral elements in the boundary layers are intentionally left undivided to demonstrate that points are introduced in the middle of the exposed faces having a much larger length scale than the one prescribed by the background grid. One can observe the good-quality of elements in the concave region of the main element and the region between the trailing edge of the main element and the trailing-edge slat, due to the self-dividing the exposed quadrilateral elements and adaptation of the background mesh using the cell size information from the exposed semistructured elements. As in the previous example, a very smooth transition between Cartesian and triangular grid regions and quadrilateral and triangular grid regions is noticed everywhere for this complex configuration, indicating the ability of the present hybrid grid generation method for generating a hybrid, good-quality, quadrilateral, Cartesian, and triangular grid for complex geometries.

IV. Conclusions

A hybrid grid generation method for generating a hybrid quadrilateral, Cartesian, and triangular grid has been presented for complex geometries. The developed hybrid grid generation method is based on a combination of semistructured quadrilateral, quad Cartesian, and unstructured triangular meshing techniques that makes use of the strengths of these three approaches while avoiding their respective weaknesses. This hybrid grid generation method generates good-quality high aspect ratio quadrilateral elements in the anisotropic regions of flow, Cartesian cells in the isotropic regions of flow, and unstructured triangular elements for the transition between these two regions. The automatic placement of sources at the geometrical singularities has been found to be effective to avoid the difficulty of meshing these regions using anisotropic elements. The self-dividing of the exposed semistructured elements with high aspect ratio and the adaptation of the background mesh using the cell size information from the exposed semistructured elements have been found to be important and critical to generate good-quality triangles and ensure the success of triangulation in the voids, to provide a smooth transition between the semistructured grid and the Cartesian grid, and ultimately improve the robustness of this hybrid grid generation method. The developed hybrid mesh generation method has been illustrated for several examples, demonstrating its ability to generate hybrid unstructured quadrilateral, Cartesian, and triangular grids for complex geometries. Further effort will be focused on extending this hybrid method for three dimensional problems.

Acknowledgments

The first author would like to acknowledge the financial support provided by North Carolina State University new faculty startup fund and North Carolina State University Faculty Research and Development Fund.

References

[1] Benek, J. A., Buning, P. G., and Steger, J. L., "A 3D Chimera Grid Embedding Technique," AIAA Paper 1985-1523, 1985.

[2] Benek, J. A., Donegan, T. L., and Suhs, N. E., "Extended Chimera Grid Embedding Scheme with Application to Viscous Flows," AIAA Paper 1997-1126, 1997.

[3] George, P. L., *Automatic Mesh Generation*, Wiley, New York, 1991.

[4] Löhner, R., and Parikh, P., "Three-Dimensional Grid Generation by the Advancing Front Method," *International Journal for Numerical Methods in Fluids*, Vol. 8, No. 10, 1988, pp. 1135–1149. doi:10.1002/fld.1650081003

[5] Prizadeh, S., "Unstructured Viscous Grid Generation by the Advancing Layers Method," *AIAA Journal*, Vol. 32, No. 8, 1994, pp. 1735–1737. doi:10.2514/3.12167

[6] Sharov, D., Luo, H., Baum, J. D., and Löhner, R., "Unstructured Navier–Stokes Grid Generation at Corners and Ridges," *International Journal for Numerical Methods in Fluids*, Vol. 43, Nos. 6–7, 2003, pp. 717–728. doi:10.1002/fld.615

[7] OConnel, S. D., and Braaten, M. E., "Semi-Structured Mesh Generation for 3D Navier–Stokes Calculations," AIAA Paper 1995-1679, 1995.

[8] Clarke, D. K., Salas, M. D., and Hassan, H. A., "Euler Calculations for Multielement Airfoils Using Cartesian Grids," *AIAA Journal*, Vol. 24, No. 3, 1986, pp. 353–358. doi:10.2514/3.9273

[9] Coirier, W. J., and Powell, K. G., "An Accuracy Assessment of Cartesian Mesh Approaches for the Euler Equations," *Journal of Computational Physics*, Vol. 117, No. 1, 1995, pp. 121–131. doi:10.1006/jcph.1995.1050

[10] Berger, M. J., and LeVeque, R. J., "An Adaptive Cartesian Mesh Algorithm for the Euler Equations in Arbitrary Geometries," AIAA Paper 1989-1930, 1989.

[11] Aftosmis, M. J., Melton, J. E., and Berger, M. J., "Adaptation and Surface Modeling for Cartesian Mesh Methods," AIAA Paper 1995-1725, 1995.

[12] Luo, H., Baum, J. D., and Löhner, R., "A Hybrid Cartesian Grid and Gridless Method for Compressible Flows," *Journal of Computational Physics*, Vol. 214, No. 2, 2006, pp. 618–632. doi:10.1016/j.jcp.2005.10.002

[13] Weatherill, N. P., "Mixed Structured-Unstructured Meshes for Aerodynamic Flow Simulations," *Aeronautical Journal*, Vol. 84, No. 934, 1990, pp. 111–123.

[14] Noack, R. W., and Steinbrenner, J. P., "A Three-Dimensional Hybrid Grid Generation Technique," AIAA Paper 1995-1684, 1995.

[15] Khawaja, A., McMorris, H., and Kallinderis, Y., "Hybrid Grids for Viscous Flows around Complex 3-D Geometries Including Multiple Bodies," AIAA Paper 1995-1685, 1995.

[16] Löhner, R., "Matching Semi-Structured and Unstructured Grids for Navier–Stokes Calculations," AIAA Paper 1993-3348, 1993.

[17] Y. Zheng, Y., and Liou, M. S., "A Novel Approach of Three-Dimensional Hybrid Grid Methodology: Part 1. Grid Generation," *Computer Methods in Applied Mechanics and Engineering*, Vol. 192, No. 37, 2003, pp. 4147–4171. doi:10.1016/S0045-7825(03)00385-2

[18] Sharov, D., and Nakahashi, K., "Hybrid Prismatic/Tetrahedral Grid Generation for Viscous Flow Applications," *AIAA Journal*, Vol. 36, No. 2, 1998, pp. 157–162. doi:10.2514/2.7497

[19] Löhner, R., Luo, H., Baum, J. D., and Rice, D., “Selective Edge Deactivation for Unstructured Grids with Cartesian Cores,” AIAA Paper 2005-5232, 2005.

[20] Bonet, J., and Peraire, J., “An Alternating Digital Tree (ADT) Algorithm for 3D Geometric Searching and Intersection Problems,” *International Journal for Numerical Methods in Engineering*, Vol. 31, No. 1, 1991, pp. 1–17. doi:10.1002/nme.1620310102

[21] Aftosmis, M., Berger, M., and Melton, J., “Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry,” AIAA Paper 1997-0196, 1997.

S. Fu
Associate Editor