

---

# A Solution-Based Adaptive Redistribution Method for Unstructured Meshes

Yasushi Ito, Alan M. Shih, Roy P. Koomullil and Bharat K. Soni

Dept of Mechanical Engineering  
University of Alabama at Birmingham, Birmingham, AL, U.S.A.  
{yito, ashih, rkoomul, bsoni}@uab.edu

**Abstract.** We propose an unstructured mesh redistribution method without using skewed elements for steady-state problems. The regions around solution features are indicated by a sensor function. The medial axes of the strong feature regions are calculated so that elements can be clustered around the most important solution features efficiently. Two approaches, a discrete surface-based approach using a Delaunay triangulation method and a mathematical-representation approach using least square fitting, are shown to calculate the medial axes. Remeshing of an initial volume mesh is performed around the medial axes using an advancing front method and/or an advancing layer method. Two examples are shown to present how our approach works.

**Key words:** Solution-based; mesh redistribution; remeshing; unstructured mesh

## 1. Introduction

Computational fluid dynamics (CFD) has become a crucial tool for prediction and analysis of flow field within a domain, providing engineers with a reliable means of understanding complex flow patterns. However, in order to obtain accurate results for highly complex flow fields, meshes must be clustered near the areas where the solution gradients are high. This is an arduous task the engineer must perform prior to the completion of the calculation. The meshes can be clustered in two ways; either a very fine

mesh is generated, or some solution-based mesh clustering is performed. The first approach can be very expensive in terms of computational costs. Although surface meshes can be adapted geometrically based on surface curvature and local volume thickness [1], it is often difficult to choose adaptation criteria for volume meshes before numerical simulations. The second approach can be achieved by mesh adaptation.

There are three mesh adaptation approaches: mesh refinement/de-refinement [2, 3], mesh redistribution [4] and the combination of these [5, 6]. Since structured meshes are not flexible for adding or deleting nodes locally, the mesh redistribution approach is widely used to move nodes toward solution features while the connectivity of the mesh is maintained.

Although solution features are adapted by unstructured meshes relatively easily, there are two issues needed to be addressed. One is the maintenance of valid elements. Hanging nodes can be created during a mesh refinement process. Local refinement of hybrid meshes for viscous flow simulations, which contain regular elements such as tetrahedra, prisms, pyramids and hexahedra, is difficult without creating low-quality elements to eliminate hanging nodes. To overcome this issue, an approach using generalized elements is promising [3].

The other issue is the quality of resulting refined meshes. Stretched elements may affect solution accuracy and cause a stiffness problem in numerical simulations. Mavriplis reports spanwise grid stretching, which is widely used in aircraft CFD simulations, may have substantial repercussions on overall simulation accuracy even at very high levels of resolution [7]. Since typical refinement and redistribution algorithms for unstructured meshes create highly stretched tetrahedra around solution features, the validation of the simulation process may be required. If a refined mesh does not have elements that have too small or too large angles even near solution features, we do not need to worry about these issues.

In this paper, we propose a solution-based redistribution method for unstructured volume meshes. The structured mesh redistribution methods only allow nodes to move towards solution features, while maintaining the mesh connectivity. In our unstructured mesh redistribution method, a mesh is remeshed around the solution features detected. The main objective here is to extract strong solution features as smooth surfaces (Section 2.3) based on sensor values (Section 2.2) and then to create high quality elements around them (Section 2.4). The entire domain can be remeshed with the embedded surfaces using an advancing front method with tetrahedra and an advancing layer method with prisms or hexahedra if needed. Alternatively, elements around the feature surfaces are removed from the initial volume mesh and only the resulting voids are remeshed to reduce

the required CPU time. Two examples are shown to present how our approach works (Section 4).

## 2. Meshing Methods

In this section, the methods for creating an initial mesh and redistributing nodes around solution feature surfaces are described.

### 2.1. Surface/Volume Mesh Generation

To generate surface meshes based on computer-aided design (CAD) defined geometries, a direct advancing front method is employed [8]. A modified decimation method is used for image-based geometries [1].

Tetrahedral meshes are created using an advancing front method [9]. For viscous flow simulations, a modified advancing layer method is used for the near-field mesh generation [10], which is followed by tetrahedral mesh generation to fill the rest of the domain using the advancing front method. The quality of the tetrahedral elements are enhanced using angle-based node smoothing, face swapping based on the Delaunay property, and removal of nodes that have an insufficient number of tetrahedra. A user can specify a stretching factor to control the mesh density.

The hybrid mesh generation method can be used to create layered meshes on solution features discussed in the next section to create high quality anisotropic adaptive meshes.

### 2.2. Feature Detection

After a numerical simulation using an initial mesh, the next step is the detection of solution features. The location of solution features is indicated by the weight function by Soni *et al.* [11] or the shock sensor by Lovely and Haines [12]. The weight function is calculated based on the conserved variables and indicates the regions of important flow features. It is defined at each element as follows:

$$\begin{aligned}
 W &= 1 + \frac{W^1}{\max(W^1, W^2, W^3)} \oplus \frac{W^2}{\max(W^1, W^2, W^3)} \oplus \frac{W^3}{\max(W^1, W^2, W^3)} \\
 W^k &= \sum_{\oplus i=1}^{nq} \left[ \frac{q_{\xi k}^i}{|q^i| + \varepsilon} \Bigg/ \left[ \frac{q_{\xi k}^i}{|q^i| + \varepsilon} \right]_{\max} \right] \quad (k = 1, 2, 3)
 \end{aligned} \tag{1}$$

where  $W^k$  ( $k = 1, 2, 3$ ),  $q_{\xi k}^i$  and  $q^i$  are  $x$ ,  $y$  and  $z$  components of normalized gradient, the  $k^{\text{th}}$  component of the gradient calculated using  $i^{\text{th}}$  variable and the average variable at the centroid of the element, respectively. The symbol  $\oplus$  represents the Boolean sum, which, for two variables  $q_1$  and  $q_2$ , is defined as

$$q_1 \oplus q_2 = q_1 + q_2 - q_1 q_2 \quad (2)$$

The shock sensor is based on the fact that the normalized Mach number  $M_n = 1$  at a shock.

$$M_n = \frac{\mathbf{V} \cdot \nabla p}{a |\nabla p|} = 1 \quad (3)$$

where  $a$ ,  $\mathbf{V}$  and  $\nabla p$  are the speed of sound, velocity vector and pressure gradient, respectively.

### 2.3. Extraction of Solution Feature Surfaces

To adapt high-quality elements around strong solution features, the next step is recognition of feature surfaces. Although this approach may need more meshing steps than a typical mesh redistribution method, much better quality elements can be generated around the solution feature surfaces. Marcum and Gaither propose a pattern recognition algorithm in 2D and mention the difficulty of extending it to 3D [13]. Although our approach needs user interaction during the process (to be discussed in Section 4), it enables feature surface extraction.

The direct extraction of solution feature surfaces is difficult from the initial mesh and solution data. At least two steps are needed. First, regions around the solution features are specified by selecting a certain sensor value. Although elements can be subdivided in the entire regions, the number of elements in the resulting mesh may become too big. The regions can be very thick if an initial volume mesh is coarse at the solution feature locations. To avoid this problem, the medial axis (also known as skeleton) of each region is extracted in the following step. Elements are clustered around the medial axes.

Two approaches can be considered to extract solution feature surfaces. One is a discrete surface-based approach. A medial axis is extracted from a triangulated closed surface using Delaunay triangulation [14]. Triangulated isosurfaces at a certain sensor value can be calculated easily and robustly, which enclose regions around solution features. For example, the shock features are surrounded by the isosurfaces at  $M_n = 1$ . A Delaunay

tetrahedral mesh can be obtained from a triangulated isosurface. The center of the circumsphere of each tetrahedron is considered to represent the medial axis. The quality of the resulting medial axes depends on the smoothness of the isosurfaces. However, isosurfaces are not usually smooth, and they may have bumps and holes due to truncation errors in the entire simulation process. User interaction is often required to fix the resulting surfaces.

The other approach is a mathematical-representation approach. A medial axis can be estimated using least square fitting directly from the nodes on an isosurface. Least square fitting methods often minimize the vertical offsets from a surface function instead of the perpendicular offsets to simplify an analytic form for the fitting parameters. Consequently, the least square fitting does not estimate the surface function well when the region defined by an isosurface is thick. Although a set of coordinates of nodes near a solution feature is needed as an input for a least square fitting method, the connectivity of the nodes is not required. Therefore, we define a solution feature as a set of nodes based on the following process:

1. Select nodes of a volume mesh that have a certain range of sensor values.
2. Also select nodes that are one-ring neighbors of the nodes in Step 1 to eliminate noise due to truncation errors.
3. Number each cluster of selected nodes, which can be defined as their connectivity, if the mesh has more than one solution features.
4. Calculate distance from the closest boundary at each selected node. The boundary is represented by the selected nodes that have at least one unselected node as their one-ring neighbor. The distance is defined as the number of edges from the boundary.
5. The nodes that have local maxima of the distance values are considered to form medial axes.

The coordinates of the nodes in Step 5 are fitted to functions, such as a plane, quadric and cone, using a least square fitting method. Local mesh size can be considered to be the error range of a data point. The reciprocal of the local mesh size is used for weighing. Suppose that a cluster of selected nodes  $\mathbf{x}_{m_j}$  ( $j = 1, 2, \dots, n_m$ ) is fitted to a function  $z = f(x, y)$ .

$$E \equiv \sum_{j=1}^{n_m} \left( \frac{z_j - f(x_j, y_j)}{l_j} \right)^2 \quad (4)$$

where  $l_j$  is the maximum edge length connected to node  $j$ .  $E$  should be minimized.

The resulting function should be trimmed to define a surface in the computational domain.

## 2.4. Remeshing with Embedded Surfaces

Next step is to remesh the initial volume mesh (defined as  $V$ ) around the medial axes created in the previous section. First, surface meshes (defined as  $S$ ) are created on the medial axes based on user-specified mesh size. Second, elements of  $V$  are removed if they are close to  $S$ . Each node of  $S$ ,  $S_i$ , should be in an element of  $V$ ,  $V_i$ . A node of  $V$ ,  $V_j$ , is removed if it is in a sphere defined at node  $S_i$  ( $i = 1$  to the number of nodes in  $S$ ) with the radius of  $3 \max(l_{si}, l_{vj})$ , where  $l_{si}$  is the maximum length of the edges connected to  $S_i$  and  $l_{vj}$  is the maximum length of the edges in  $V_j$  and if  $V_j$  is visible from  $S_i$  (*i.e.*, there is no boundary surface of  $V$  between  $V_j$  and  $S_i$ ). Third, the void regions around the medial axes are filled using the meshing method described in Section 2.1.

The entire domain is remeshed with the embedded surfaces if a layered mesh is needed for no-slip walls or the shape of the outer boundary needs to be changed.

## 3. Flow Solver

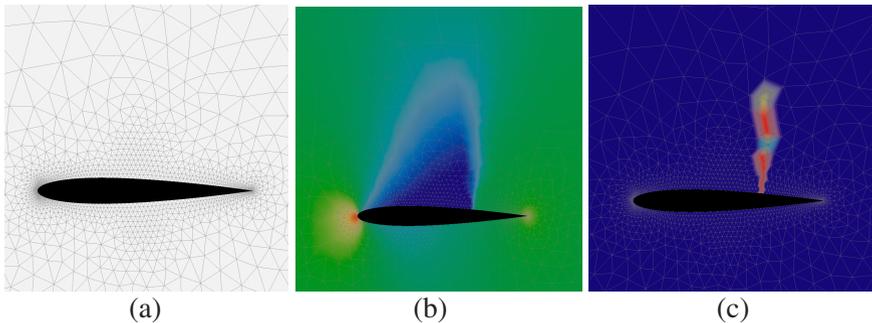
The flow simulation system that is used for the current study is developed for a generalized grid framework, in which the discretization of the physical domain can be of structured, unstructured or an agglomeration of cells with an arbitrary number of faces (polytops) [15, 16]. The integral form of the Navier-Stokes equations is taken as the governing equations for the fluid flow. The spatial discretization of the governing equations is based on a cell-centered, finite volume upwind scheme. The convective fluxes at the cell-faces are evaluated using Roe's approximate Riemann [17]. Higher-order accuracy in the spatial domain is achieved using a Taylor series expansion of flow variables. The gradients at the cell center for the Taylor series expansion is estimated using either the Gauss theorem together with a weighted averaging procedure or a least-square fit of the variables in the neighboring cells. The least-square system resulting from the later approach is solved using the Gram-Schmidt method. A limiter function is added to the Taylor's series expansion to avoid the creation of local extrema during the reconstruction process. Limiters by Venkatakrisnan [18], and Barth and Jespersion [19] are implemented in the generalized grid framework.

## 4. Applications

In this section, two examples are shown to demonstrate and to discuss the solution-based mesh redistribution approach.

### 4.1. NACA0012 Wing

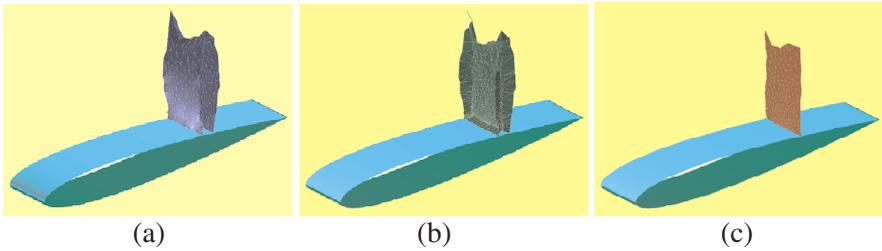
Figure 1a shows a mesh around a NACA0012 wing (50k nodes). An inviscid flow simulation is carried out at a freestream Mach number of 0.799 and an angle of attack,  $\alpha$ , of  $2.26^\circ$ . Figures 1b and 1c illustrate pressure coefficient ( $C_p$ ) distribution and weight function value distribution based on Eq. 1; the results indicate a shock on the wing.



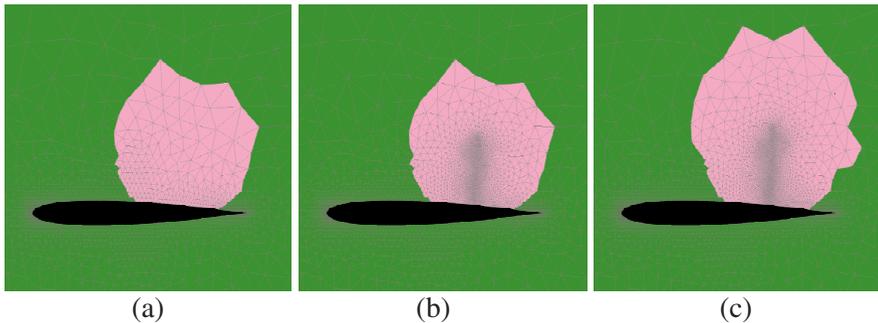
**Figure 1.** NACA0012 wing: (a) a tetrahedral mesh; (b)  $C_p$  distribution ( $M = 0.799$ ,  $\alpha = 2.26^\circ$ ); (c) weight function value distribution.

In this example, the shock location is estimated using the discrete surface-based approach based on Delaunay triangulation (Figure 2). First, an isosurface of a weight function value of 0.2 is extracted and smoothed using Visualization Toolkit (VTK) [20] (Figure 2a). Second, the medial axis of the isosurface is calculated (Figure 2b). The symmetry planes prevent creating a single medial axis. The medial axis is modified there manually (Figure 2c). It is sometimes difficult to generate an expected medial axis as single surface using existing algorithms even if a solution feature is simple.

Once the feature surface is computed, the surface mesh generation algorithm is applied to create a high quality mesh on it. Elements of the initial volume mesh,  $V^0$ , near the solution feature are removed and the void is remeshed using the advancing front method. Figure 3b shows the resulting volume mesh (110k nodes;  $V^1$ ). As a result, a high quality redistributed mesh is produced with alignment to the major flow feature. Figure 3c shows another redistributed volume mesh after the second simulation cycle (130k nodes;  $V^2$ ).



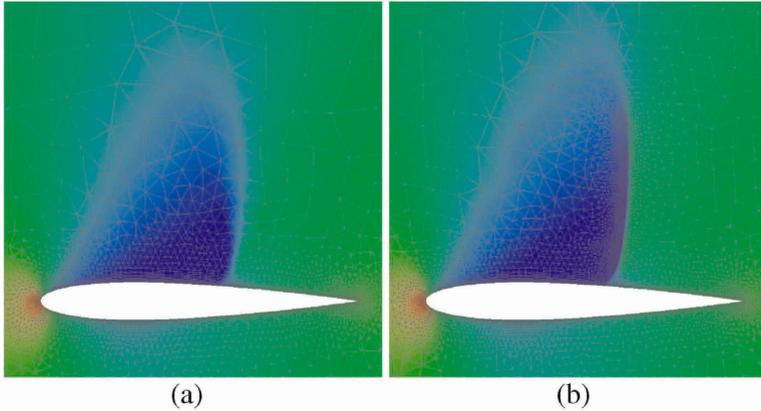
**Figure 2.** Extraction of a flow feature: (a) isosurface of a weight function value of 0.2 at the shock location; (b) medial axis of the isosurface; and (c) modified medial axis.



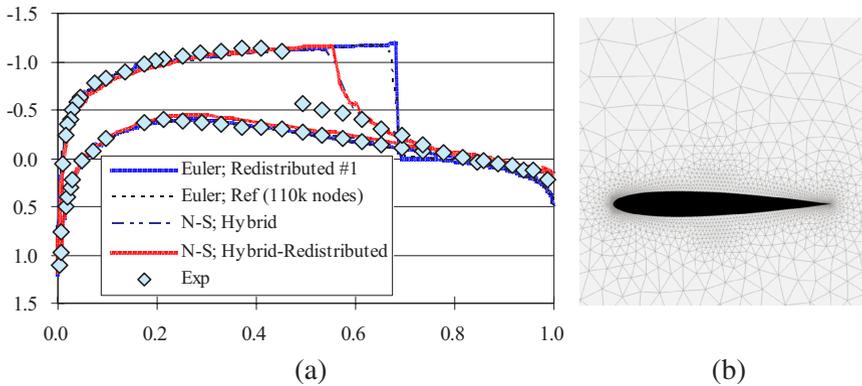
**Figure 3.** Redistributed volume meshes: (a) initial mesh (50k nodes); (b) redistribution #1 (110k nodes) – elements around the shock are replaced with finer elements; (c) redistribution #2 (130k nodes).

Figure 4 illustrates hybrid meshes for the same wing geometry to perform viscous flow simulations and  $C_p$  distribution. The shock location is estimated using the same approach from the initial hybrid mesh (Figure 4a), and then the entire domain is remeshed with the embedded surface (Figure 4b). To avoid creating skewed elements around the intersection between the wing upper surface and the embedded surface, the near-field mesh around the wing is generated first. The embedded surface close to or within the near-field mesh is trimmed automatically, and then the rest of the domain is filled with tetrahedral elements.

Figure 5 shows  $C_p$  distribution based on the redistributed mesh  $V^{n1}$  (Figure 3b), a reference tetrahedral mesh (110k nodes; Figure 5b), the hybrid meshes (Figure 4), and an experimental result. The shock locations of the numerical results do not agree with that of the experimental data well. The viscous flow simulations give better result, but further investigation for the turbulence model is required. In the inviscid flow case, although the reference mesh has almost the same number of nodes as  $V^{n1}$ , it is not adapted to the shock feature. The redistributed mesh  $V^{n1}$  represents the shock more clearly. In the viscous flow case, the redistributed mesh also represents the shock more clearly.



**Figure 4.** Hybrid meshes for the NACA0012 wing and  $C_p$  distribution (-1.0 to 1.0): (a) initial hybrid mesh; (b) redistributed hybrid mesh.

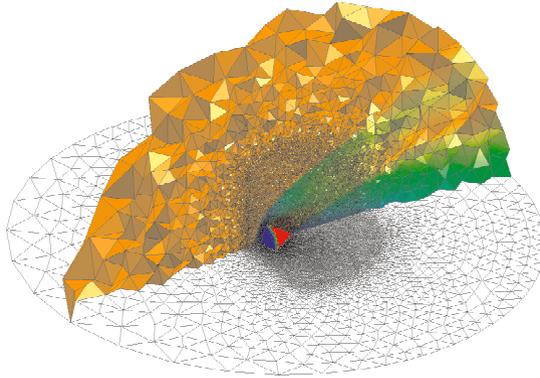


**Figure 5.** Comparison of  $C_p$  distribution for the NACA0012 wing ( $M = 0.799$ ,  $\alpha = 2.26$ ): (a)  $C_p$  distribution; (b) reference mesh (110k nodes).

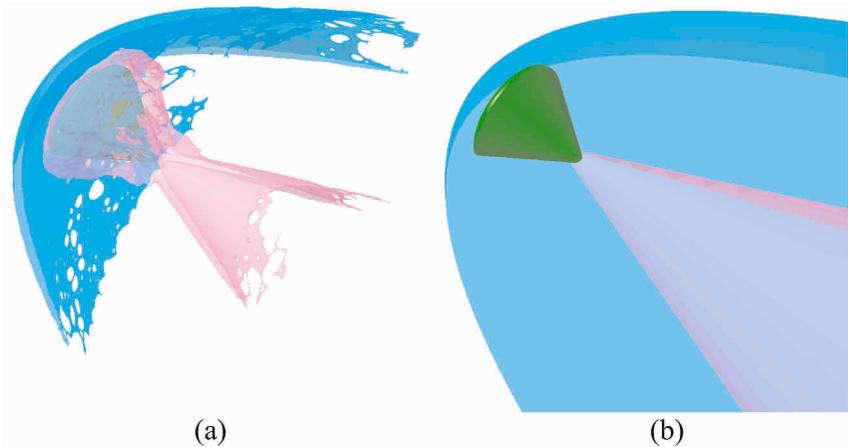
## 4.2. Capsule Model

Figure 6 shows an initial tetrahedral mesh around a re-entry capsule model and Mach number distribution on a cross-section. The bow shock in front of the capsule becomes steady, but the flow solution is not fully converged. The shape of the outer boundary is a hemisphere so that the mesh can be used for flows at different angles of attack. Isosurfaces of a certain weight function value can be extracted as triangulated surfaces (Figure 7a), the medial axes of which are considered to represent the most important locations. An approach to obtain medial axes using a Delaunay trian-

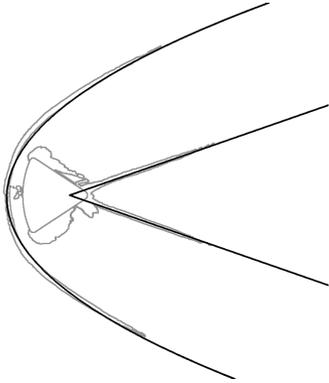
gulation method from the triangulated surfaces can be considered. However, it is difficult to obtain medial axes automatically as smooth surfaces as discussed in the previous example. Although the isosurfaces shown in Figure 7a are smoothed using a Laplacian method, many holes and small features prevent extracting smooth medial axes.



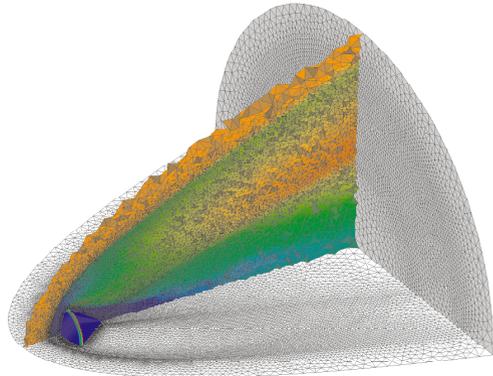
**Figure 6.** Initial mesh for a capsule model and Mach number distribution on a cross-section ( $M = 1.0\text{--}4.0$ ).



**Figure 7.** Flow features: (a) extracted isosurfaces at a weight function value of 0.05; (b) estimated features using a least square fitting method.



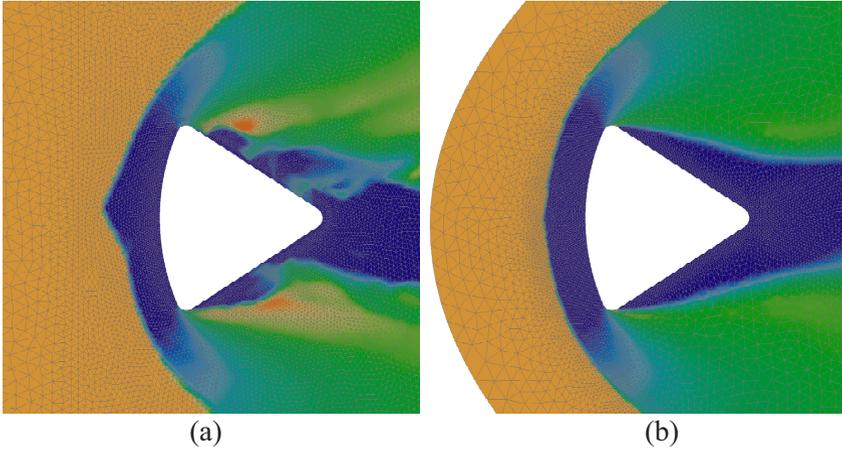
**Figure 8.** Flow features on the symmetry plane.



**Figure 9.** Redistributed mesh for a capsule model and Mach number distribution on a cross-section ( $M = 1.0-4.0$ ).

The other approach using the least square fitting method is more appropriate in this case. After a user specifies one of the template functions, such as a cone, quadratic and quartic, and the  $z$  axis of the function, a corresponding medial axis is obtained as a mathematical function (Figure 7). The bow shock in front of the capsule is fitted to a quadratic, and the shock from the aft of it is fitted to a cone. Figure 8 shows the obtained surfaces (Figure 7b) and the isosurfaces for reference (Figure 7a) on the symmetry plane. The least square fitting method estimates the medial axes well. One of the disadvantages using unstructured meshes is that flow features diverge quickly. This approach enables us to estimate missing flow features.

Figure 9 shows a redistributed mesh, which has 0.74 million nodes. In this case, the entire mesh is regenerated because the shape of the outer boundary is changed to remove extra elements. The initial mesh shown in Figure 6 can be used for cases at different angles of attack, but it has 0.89 million nodes. In addition, the elements around the shocks in the far field are coarse. Figure 10 illustrates Mach number distribution on the symmetry planes of the initial and redistributed meshes. Both flows are fully converged. The initial mesh gives a carbuncle phenomenon on the bow shock, while the redistributed mesh gives better result. One of the solution feature surfaces shown in Figure 7b fits the bow shock well.

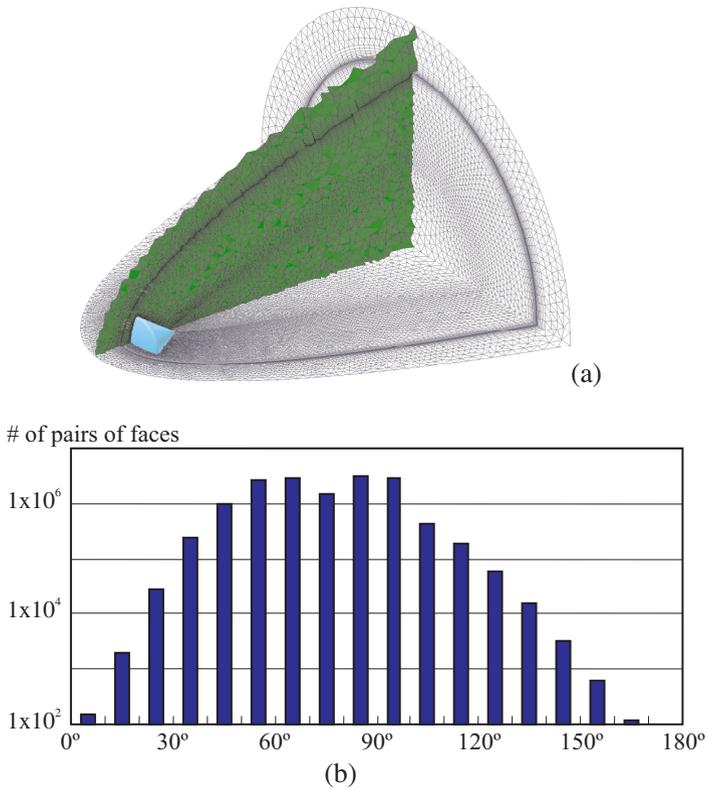


**Figure 10.** Mach number distribution for the capsule model on the symmetry planes ( $M = 1.0-4.0$ ): (a) the initial mesh showing a carbuncle phenomenon; (b) the redistributed mesh.

The most notable advantage of the surface-based mesh redistribution method is that anisotropic nonsimplicial elements can be used around the feature surfaces to avoid creating skewed elements. Figure 11 shows a redistributed hybrid mesh, which has 0.62 million nodes, based on the same numerical result. Prismatic layers are placed around the bow shock. The quality of the mesh is excellent as shown in the dihedral angle distribution (Figure 11b).

## 5. Conclusion and Future Work

In this paper, we propose a solution-based mesh redistribution method for strong solution features. Solution features are indicated by a weight function and a shock sensor. The feature locations are estimated by medial axes of isosurfaces at a certain sensor value. To compute medial axes, two approaches are discussed. The discrete surface-based approach using a Delaunay triangulation method may not be suitable to estimate solution features as smooth surfaces. The mathematical-representation approach using least square fitting can represent solution features easily and can interpolate missing features due to truncation errors. The remeshing method with embedded surfaces enables anisotropic nonsimplicial elements to be placed around the features to avoid creating skewed elements.



**Figure 11.** Redistributed mesh for a capsule model using anisotropic elements: (a) hybrid mesh; (b) dihedral angle distribution of the mesh.

The proposed approach, however, does not work well if a solution feature is difficult to be represented as a single surface, such as vortex breakdown. In future work, the solution-based mesh redistribution method will be combined with a mesh refinement method [3] to adapt a mesh to all the solution features efficiently [21].

## Acknowledgments

This research is supported in part by the NASA Constellation University Institutes Project (CUIP) No. NCC3-994. Part of this material is based upon work supported by the National Science Foundation under the following NSF programs: Partnerships for Advanced Computational Infrastructure, Distributed Terascale Facility (DTF) and Terascale Extensions: Enhancements to the Extensible Terascale Facility.

## References

1. Ito, Y., Shum, P. C., Shih, A. M., Soni, B. K. and Nakahashi, K., "Robust Generation of High-Quality Unstructured Meshes on Realistic Biomedical Geometry," *International Journal for Numerical Methods in Engineering*, Vol. 65, Issue 6, 2006, pp. 943-973.
2. Cavallo, P. A. and Grismer, M. J., "Further Extension and Validation of a Parallel Unstructured Mesh Adaptation Package," AIAA Paper 2005-0924, 43<sup>rd</sup> Aerospace Sciences Meeting and Exhibit, Reno, NV, 2005.
3. Senguttuvan, V., Chalasani, S., Luke, E. and Thompson, D., "Adaptive Mesh Refinement Using General Elements," AIAA Paper 2005-0927, 43<sup>rd</sup> AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 2005.
4. Soni, B. K., Thornburg, H. J., Koomullil, R. P., Apte, M. and Madhavan, A., "PMAG: Parallel Multiblock Adaptive Grid System," Proceedings of the 6<sup>th</sup> International Conference on Numerical Grid Generation in Computational Field Simulation, London, UK, 1998, pp. 769-779.
5. Shephard, M. S., Flaherty, J. E., Jansen, K. E., Li, X., Luo, X., Chevaugon, N., Remacle, J.-F., Beall, M. W. and O'Bara, R. M., "Adaptive Mesh Generation for Curved Domains," *Applied Numerical Mathematics*, Vol. 52, Issue 2-3, 2005, pp. 251-271.
6. Suerich-Gulick, F., Lepage, C. and Habashi, W., "Anisotropic 3-D Mesh Adaptation for Turbulent Flows," AIAA Paper 2004-2533, 34<sup>th</sup> AIAA Fluid Dynamics Conference and Exhibit, Portland, OR, 2004.
7. Mavriplis, D. J., "Grid Resolution Study of a Drag Prediction Workshop Configuration Using the NSU3D Unstructured Mesh Solver," AIAA Paper 2005-4729, 23<sup>rd</sup> Applied Aerodynamics Conference, Toronto, Canada, 2005.
8. Ito, Y. and Nakahashi, K., "Direct Surface Triangulation Using Stereolithography Data," *AIAA Journal*, Vol. 40, No. 3, 2002, pp. 490-496.
9. Ito, Y., Shih, A. M. and Soni, B. K., "Reliable Isotropic Tetrahedral Mesh Generation Based on an Advancing Front Method," Proceedings of the 13<sup>th</sup> International Meshing Roundtable, Williamsburg, VA, 2004, pp. 95-105.
10. Ito, Y., Shih, A. M., Soni, B. K. and Nakahashi, K., "An Approach to Generate High Quality Unstructured Hybrid Meshes," AIAA Paper 2006-0530, 44<sup>th</sup> AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 2006.
11. Soni, B. K., Koomullil, R., Thompson, D. S. and Thornburg, H., "Solution Adaptive Grid Strategies Based on Point Redistribution," *Computer Methods in Applied Mechanics and Engineering*, Vol. 189, Issue 4, 2000, pp. 1183-1204.
12. Lovely, D. and Haines, R., "Shock Detection from Computational Fluid Dynamics Results," AIAA Paper 1999-3285, 14<sup>th</sup> AIAA Computational Fluid Dynamics Conference, Norfolk, VA, 1999.
13. Marcum, D. L. and Gaither, K. P., "Solution Adaptive Unstructured Grid Generation Using Pseudo-Pattern Recognition Techniques," AIAA Paper 97-1860, 13<sup>th</sup> AIAA Computational Fluid Dynamics Conference, Snowmass Village, CO, 1997.
14. Sheehy, D. J., Armstrong, C.G. and Robinson, D. J., "Shape Description by Medial Surface Construction," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 2, Issue 1, 1996, pp. 62-72.
15. Koomullil, R. P., Thompson, D. S., and Soni, B. K., "Iced Airfoil Simulation Using Generalized Grids", *Journal of Applied Numerical Mathematics*, Vol. 46, Issues 3-4, 2003, pp 319-330.
16. Koomullil, R. P., and Soni, B. K., "Flow Simulation Using Generalized Static and Dynamics Grids", *AIAA Journal*, Vol. 37, No. 12, 1999, pp 1551-1557.
17. Roe, P. L., "Approximate Riemann Solvers, Parameter Vector, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357-372.

18. Venkatakrishnan, V., "On the Accuracy of Limiters and Convergence to Steady State Solutions," AIAA Paper 93-0880, 31<sup>st</sup> AIAA Aerospace Sciences Meeting, Reno, NV, 1993.
19. Barth, T. J. and Jespersen, D. C., "The Design and Application of Upwind Schemes on Unstructured Meshes," AIAA Paper 89-0366, 27<sup>th</sup> AIAA Aerospace Sciences Meeting, Reno, NV, 1989.
20. Visualization Toolkit (VTK), <http://www.vtk.org/>.
21. Shih, A., Ito, Y., Koomullil, R., Kasmai, T., Jankun-Kelly, M., Thompson, D., and Brewer, W., "Solution Adaptive Mesh Generation using Feature-Aligned Embedded Surface Meshes," 45<sup>th</sup> AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 2007, submitted.

