AIAA JOURNAL
Vol. 47, No. 8, August 2009

Solution Adaptive Mesh Generation Using Feature-Aligned Embedded Surface Meshes

Y. Ito,* A. Shih,† and R. Koomullil‡
University of Alabama at Birmingham, Birmingham, Alabama 35294
and
N. Kasmai,§ M. Jankun-Kelly,§ and D. Thompson¶
Mississippi State University, Mississippi State, Mississippi 39762

DOI: 10.2514/1.39378

In this paper, we present a technique for generating solution adaptive meshes that uses feature-aligned surfaces embedded in the computational domain. A feature-detection algorithm is first employed to identify features of interest. Surfaces associated with these features are then meshed, and a volume mesh that respects these embedded surface meshes is then generated. This approach is promising because it provides a mechanism for controlling local grid quality as well as automatically aligning cell faces with the feature. For flows that include topologically and/or geometrically complex features, we describe a mesh refinement technique that can be used in tandem with the embedded surface meshes to provide a mechanism for refining throughout the computational domain. We include results from two cases to illustrate how the method works.

I. Introduction

MESH adaptation has long been a subject of interest in computational fluid dynamics (CFD). Mesh adaptation methods can be categorized into three approaches: mesh redistribution, in which nodes are repositioned; mesh refinement/derefinement, in which nodes are added or removed, respectively; and the combination of both [1,2]. Although the mesh redistribution is the natural choice for structured meshes, Amato et al. [3] attempted refinement by generating a mesh with different spacings in each subblock based on the strengths of selected features. For tetrahedral and hybrid meshes, both redistribution [4–8] and refinement [9–16] approaches have been explored with different levels of success.

There are a couple of issues that need to be addressed in mesh adaptation. The first issue is the ability to produce anisotropic simplicial meshes (triangular meshes in two dimensions and tetrahedral meshes in three dimensions) for easy implementation. It may seem counterintuitive, as the common wisdom is that elements should not have angles that are too small or too large. Anisotropic adaptation can produce highly skewed elements that would be considered to be of "poor" quality. However, as mentioned by Laflin and McRae [17], mesh quality should not be restricted to only geometric measures, but should also consider the error associated with the solution. As demonstrated by Yang and Soni [5,18] and Yang and Lie [19], for example, impressive solutions were produced on extremely skewed adaptive meshes. However, skewed elements can cause a stiffness problems in some flow solvers. In addition, Mavriplis reported that 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 [20]. Highly stretched elements from mesh adaptation may cause the same problem. Therefore, a mesh adaptation strategy that does not produce highly skewed elements would seem to be a better choice.

The second issue concerns the additional computational cost induced by the adaptation. For redistribution schemes, the original nodes are moved according to some error function; thus, the same number of points are maintained. The computational overheads induced during each adaptation cycle are the one-time costs for mesh adaptation, interpolation of the solution from the original mesh to the new adapted mesh, and the extra iterations required to overcome the convergence spikes after the interpolation. In contrast, for most refinement approaches, even those with a derefinement capability, the overall mesh size tends to increase significantly. This induces considerable computational cost in each iteration of the computation on top of those experienced by the redistribution algorithms.

To overcome these issues, we have developed a method for generating solution adaptive meshes with embedded surfaces aligned with flow features [21]. This approach is advantageous because it provides more control over volume mesh density around the features. Because prisms or hexahedra can be placed around the features, no highly skewed elements are generated. Another advantage is that mesh faces aligned with the features can improve the resolution of the features for Riemann-based flow solvers. The primary disadvantage of this approach is that, for some geometrically and/or topologically complex features, it may not be possible to generate a quality embedded surface mesh or a volume mesh. In these cases, we propose to employ the isotropic refinement technique described by Senguttuvan et al. [9] and Chalasani et al. [10] to perform solution adaptive mesh refinement. Our approach is illustrated schematically in Fig. 1.

This paper describes the algorithms employed to identify the shock (Sec. II.A) and vortex (Sec. II.B) features, a three-dimensional mesh adaptation approach that combines a regeneration algorithm using embedded surfaces [21] and an isotropic mesh refinement algorithm [9,10]. Volume mesh generation is performed that respects the embedded feature surfaces (Sec. III). The resulting meshes are then refined around geometrically and/or topologically complex features using a mesh refinement strategy that exploits general elements to limit refinement propagation (Sec. IV). To demonstrate the method, we consider two cases (Sec. V). The first case, a wing tip vortex from a NACA 0012 wing, is topologically simple and does not require additional mesh refinement. The second case, the high-speed

Presented as Paper 558 at the 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno Nevada, 8–11 January 2007; received 26 June 2008; revision received 21 April 2009; accepted for publication 22 April 2009.

*Research Assistant Professor, Department of Mechanical Engineering. Senior Member AIAA.

†Research Professor, Department of Mechanical Engineering. Associate Fellow AIAA.

‡Associate Professor, Department of Mechanical Engineering. Member AIAA.

§Graduate Research Assistant, Center for Advanced Vehicular Systems Computational Simulation and Design Center, Box 9627. Student Member AIAA.

¶Associate Professor, Department of Aerospace Engineering, Box A. Associate Fellow AIAA.

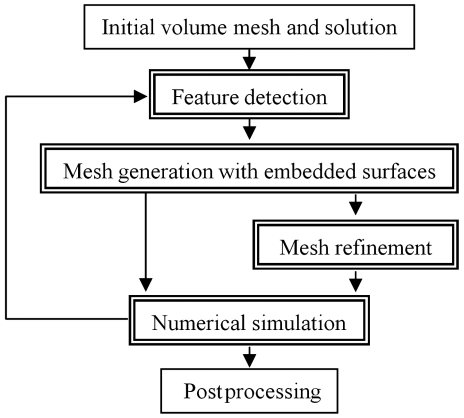


Fig. 1 Outline of the solution adaptive mesh generation process.

flow past a notional crew return vehicle in proximity to a booster, does require additional mesh refinement.

II. Feature Detection for Surface Embedding

The first step in the process is to detect the feature of interest in the flowfield. In the two problems we consider here, we want to employ embedded surface meshes to represent vortices and shock waves, respectively. The approaches taken to identify the embedded feature surfaces are briefly described in this section.

A. Shocks

Shocks can be detected using any one of several techniques [22]. Shock detection algorithms rely on two observations: 1) the Mach number normal to a stationary shock has a value of at least 1 just before the shock, and 2) the pressure gradient is normal to the shock. A normalized directional Mach number is defined as follows:

$$M_n = \frac{\mathbf{V}}{a} \cdot \frac{\nabla p}{|\nabla p|} \tag{1}$$

where a, $\mathbf{V}$, and ∇p denote the speed of sound, velocity vector, and pressure gradient, respectively. Therefore, the isosurfaces at $M_n = 1$ surround the shock features. However, the thickness of the detected shocks depends on the coarseness of the mesh or, more precisely, it is a linear function of element size [22]. To cluster nodes around the shocks, constant-size smaller elements can be placed inside the region specified by the isosurfaces of $M_n = 1$. However, this approach wastes computational resources when the initial mesh is coarse. Therefore, we use either the medial axis or part of the isosurface as the feature-aligned surface to cluster nodes around a shock more efficiently [21]. Because each medial axis or isosurface can be represented as a single surface, we can easily generate anisotropic, nonsimplex elements (i.e., prisms and hexahedra) if desired.

B. Vortices

Vortices are detected using the approach described by Jankun-Kelly et al. [23,24] Their technique employs a predictor-corrector algorithm to extract the location of a vortex core line from a specified scalar field. The predictor-corrector algorithm exploits the fact that, for certain scalar fields, a vortex core line coincides with a line-type extremum in the field variable. Subcell resolution of the vortex core line position is achieved by assuming a conical variation of the scalar field in the region near the core line. The position of the apex of the cone is determined based on a least-squares fitting of surrounding data. For the results included here, the scalar field we used was the swirl parameter [25], which is based on the eigenvalues of the velocity gradient tensor. Other scalar fields that have extreme values along the vortex core line may also be used.

Once the core line is extracted, the extent of the vortex is computed. Unfortunately, the extent of a vortex is an ambiguous concept. We employ the procedure outlined by Garth et al. [26], who computed a two-dimensional extent in the plane containing the swirling motion by determining a set of radial positions at which the tangential velocity is a maximum. This set of radial positions forms a closed curve in the swirl plane. A surface is developed by lofting between each of the two-dimensional extent curves. A sample extent surface is shown in Fig. 2. The extent surface is an appropriate choice for our surface embedding approach because it defines a region in which the tangential velocity has high curvature and therefore needs to be well resolved.

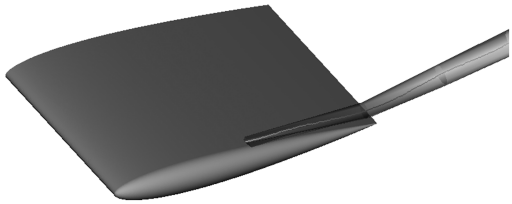


Fig. 2 Detected core line and extent surface for a wing tip vortex. The extent surface is colored by local tangential velocity to illustrate that it is not an isosurface. We use the extent surface as the embedded surface for the solution adaptive mesh generation.

III. Mesh Generation with Embedded Surfaces

To create a triangulated surface mesh on each of the embedded surfaces, a direct advancing front method is used [27,28]. A discrete model described using the stereolithography format is used as a background mesh upon which a new surface mesh suitable for computational simulations is created. The embedded surfaces discussed in Sec. II are easily represented as discrete entities. Geometrical features are extracted based on a folding angle at each edge. The user specifies the node distributions on the geometrical features. They form an initial front for the advancing front algorithm, so that the user can control the mesh density easily. Surface triangulation is performed in the physical three-dimensional space to ensure the quality of triangles.

Tetrahedral or hybrid meshes are created automatically from the surface meshes based on a few user-specified parameters [29,30]. The proximity of an embedded surface may prevent the growth of a near-field mesh around a model. For example, the extent surface for the wing tip vortex shown in Fig. 2 may be too close to the wing. This situation results in the poor transition of neighboring elements, which may cause divergence in solution. To prevent this, the mesh generation is organized as follows:

1) Create an initial volume mesh and obtain a solution.

2) Extract solution features as discrete surfaces.

3) Refine the embedded surfaces and the initial surface mesh if the embedded surface is in close proximity to the surface mesh and if there is a significant disparity in element size. Create a near-field mesh around the model if a viscous flow simulation is needed.

4) Remove nodes on the embedded surfaces automatically if they are in the near-field mesh or if they are close to the top of the near-field mesh. The closeness is determined based on the maximum length of the connecting edges at each node on the top of the near-field mesh.

5) Create a near-field mesh around the trimmed embedded surfaces if needed.

6) Create tetrahedral elements to fill the rest of the domain.

The benefits inherent to this approach are that 1) high-quality meshes can be created, and 2) the user can control the size of the mesh easily.

IV. Mesh Refinement

In some cases, the geometry or topology of a feature may preclude generating a mesh with an embedded surface. We employ isotropic mesh refinement for these features. We have previously presented a mesh refinement strategy that provides isotropic refinement of any element type [9,10]. We call this a face-based approach because it is the refinement of the faces that determines how the cells are refined.

An n-sided face is subdivided into n four-sided faces by inserting a new vertex at the centroid of the face and connecting it to the midpoint of each edge. These newly created edges form faces that divide the interior of the cell. However, as it is purely a refinement strategy, this approach does not facilitate the alignment of cell faces with flow features.

Our refinement strategy can be thought of as analogous to refinement for a Cartesian mesh. Consider two adjacent quadrilaterals in a two-dimensional Cartesian mesh. One of the cells is selected for refinement and the other is not. Rather than subdividing the adjacent cell, the "hanging" node effectively produces a cell with five edges. This suggests a strategy whereby other element types may be subdivided in a similar manner, thereby converting neighboring elements into general elements. A step-by-step procedure follows for the refinement of any type of element:

1) A criterion is applied to determine if a cell should be subdivided. A cell is also marked for subdivision if a neighboring cell has two or more additional levels of subdivision. The following steps are done for all the cells marked for subdivision.

2) Every edge in the cell is subdivided by inserting a vertex at its midpoint.

3) For every n-sided face in the cell,

a) A new vertex is created at its centroid.

b) The vertex at the midpoint of every edge on the face is connected to the vertex at the centroid to form n new edges.

c) Each vertex on the face lies exactly on two edges and it forms a four-sided face with the midpoints of those two edges and the centroid of the face. n four-sided faces are formed.

4) For the n-faced cell with e edges and v vertices,

a) A new vertex is created at its centroid.

b) The vertex at the centroid of every face on the cell is connected to the new vertex to form n new edges.

c) Each edge lies exactly on two faces, and the vertex at the midpoint of the edge forms a four-sided face with the centroid of the two faces and the centroid of the cell. e four-sided faces are formed.

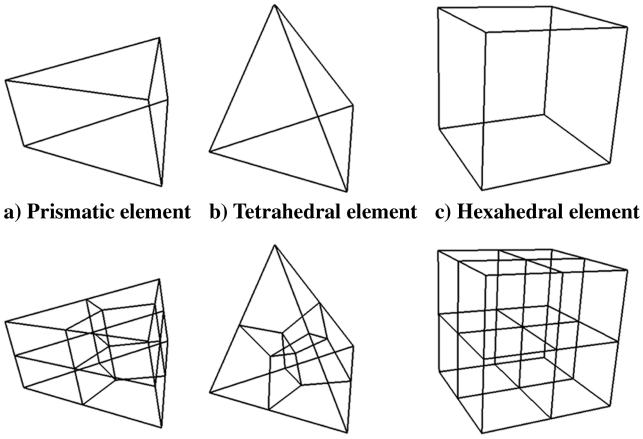
d) For each vertex, a cell is formed. New faces on the surface containing this vertex are added to the cell. Then, all the newly formed faces are checked. If three of the four vertices of a face (the fourth is the centroid of the cell) are part of the formed cell, the face is added to the cell. This creates v cells.

5) Finally, if any face contains a refined edge, that edge is removed and replaced by the subdivided and newly formed edges. If any cell contains a refined face, that face is removed and replaced by the subdivided and newly formed faces. This updates the connectivity of the grid.

The refinement of several element types using the face-based strategy is shown in Fig. 3. The refinement of a standard element such as a prism, tetrahedron, or hexahedron produces hexahedral elements. The general elements are produced in the neighboring elements due to the hanging nodes. The refinement of a pyramid, actually, of any element that contains four or more edges on a vertex, produces a combination of hexahedra and general elements.

This approach has many of the same advantages and disadvantages as Cartesian mesh refinement. The newly created vertices and edges can increase the size of the mesh significantly. Further, because no smoothing or reconnection is employed, the isotropic refinement strategy employed here tends to maintain element quality, either good or bad, in the sense that existing face angles are not subdivided. Regardless, poor element quality in the unrefined mesh is not likely to be improved by mesh refinement alone. Introduced faces may have large ($\sim$180 deg) included face angles, however. In fact, it is not clear as to what constitutes a high-quality general element, as even notions of convexity are lost for nonsimplex elements. We have previously approached element quality in terms of the quality of the faces [31]. However, as noted by Knupp [32], the assessment of element quality based on face quality alone may not be adequate.

Additionally, there are unanswered questions regarding the accuracy of the solution in regions that contain general elements. The resolution of these issues is an ongoing effort. We have reported results from one such effort in a previous paper [33]. In general, we have not noted significant artifacts near general elements except in cases in which highly anisotropic elements were refined, for example, isotropic refinement in a boundary layer. In these cases, we have found that the necessary limiting reduces the order of accuracy of the method.



a) Prismatic element **b) Tetrahedral element** **c) Hexahedral element**

d) Refined prismatic element **e) Refined tetrahedral element** **f) Refined hexahedral element**

Fig. 3 Isotropic refinement strategies for basic element types.

V. Results

In this section, we present results from two different cases that illustrate the feasibility and efficacy of this approach. The first case focuses on adaptation in the region containing a wing tip vortex, thereby improving the simulation results. The other case, a crew return vehicle in the proximity of a rocket booster, focuses on improved shock capturing.

A. Feature-Based Embedded Surfaces: Wing Tip Vortex

The specific configuration [34] considered here is a low-aspect-ratio rectangular planform wing (3 ft span and 4 ft chord) with a constant NACA 0012 section. The half-wing model is mounted in a rectangular cross-sectional (4-ft-wide by 2.667-ft-tall) wind tunnel at an angle of attack of 10 deg with respect to the horizontal. The wing has a rounded tip in both the experiment and the simulation. The reference velocity is 170 ft/s, which, at the given conditions, results in a Reynolds number of 4.6×10^6 and a Mach number of 0.1986. The computational domain employed in our simulation is indicated in Fig. 4 and is significantly longer than the domain employed by Dacles-Mariani et al. [34]. The boundary conditions used at the inflow and exit in [34] were obtained from wind-tunnel data, whereas, in our simulation, characteristic boundary conditions were employed. At the inlet, the mass flow rate was specified along with the total temperature and the pressure was allowed to float. At the exit, a nonreflecting, inflow/outflow characteristic-based boundary condition was employed with the exterior pressure, temperature, and velocity specified to be the reference values. To reduce the cost of the simulation, the tunnel's bottom, top, and side walls are modeled as slip walls.

The incompressible flowfield was computed using the CHEM flow solver [35]. CHEM has formal second-order spatial and temporal accuracy and is designed to work with meshes that contain arbitrary elements. The Spalart–Allmaras turbulence model [36] was employed with the modification to the production term suggested by Dacles-Mariani et al. [34] that reduces the eddy viscosity in regions in which the vorticity exceeds the strain rate, such as a vortex core.

For problems of this type, numerical error due to the inadequate resolution of the vortex manifests itself as increased dissipation. The vortex does not develop properly and dissipates in a nonphysical manner. Thus, to simulate the vortex in an efficient manner, the nodal density should be increased in the region near the vortex. One of the

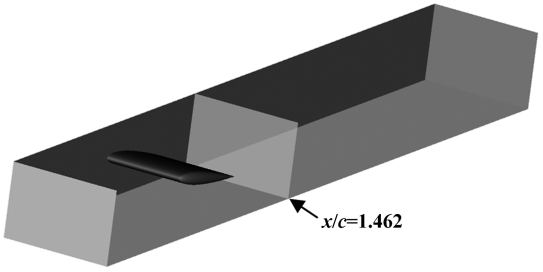


Fig. 4 Computational domain employed for the wing tip vortex problem described by Dacles-Mariani et al. [34]. The inflow boundary was located approximately 1.6 chord lengths upstream of the wing leading edge, and the outflow boundary was located 4.5 chord lengths downstream of the wing trailing edge. Comparisons with experimental data are made in the plane at $x/c = 1.462$, shown just downstream of the wing trailing edge.

complicating factors associated with this problem is the fact that the flow near the wing tip must be adequately resolved if the vortex is to be accurately computed downstream of the wing tip. In this case, we have refined the mesh on the wing surface at the tip in the baseline mesh to ensure the proper development of the wing tip vortex, as shown in Fig. 5. We do not consider this limitation significant because it is well known where the refinement needs to be performed. However, it is not clear to what level the mesh needs to be refined to adequately capture the vortical flow at the wing tip.

Figures 6a, 6c, 6e, and 6g show cross sections of the meshes employed for the baseline computation, after one adaptation cycle, after two adaptation cycles, and after three adaptation cycles, respectively. The vortex extent surface extracted from each solution is shown in Figs. 6b, 6d, 6f, and 6h. The wing surface mesh was generated using SolidMesh [37], whereas the other boundary/embedded surface and volume meshes were generated using the techniques described in Sec. III. The wing surface mesh remains fixed for each regeneration cycle. Each mesh was generated using the vortex extent surface computed on the previous mesh in the sequence as an embedded surface upon which a surface mesh was generated. The following spacings were employed to generate the embedded surface meshes:

1) Baseline: no embedded surface.

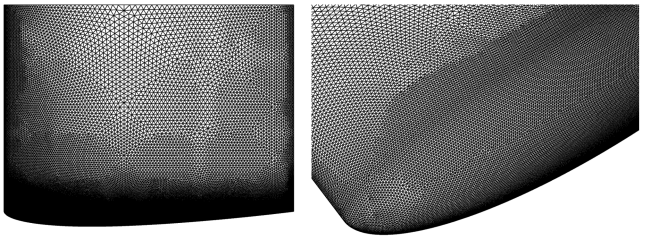
2) Cycle 1: mesh spacing on the embedded surface varied from 0.005 over the wing to a spacing of 0.01 at the downstream boundary.

3) Cycle 2: mesh spacing on the embedded surface varied from 0.002 over the wing to a spacing of 0.005 at the downstream boundary.

4) Cycle 3: mesh spacing on the embedded surface varied from 0.002 over the wing to a spacing of 0.005 at the downstream boundary and the nodal density inside of the embedded surface was increased.

Note that, due to the poor resolution of the vortex on the coarse baseline mesh shown in Fig. 6a, the vortex extent surface is initially quite diffused. However, the quality of the extent surface improves dramatically with increased resolution, as evident in the images of the resulting meshes shown in Figs. 6d, 6f, and 6h. Note that, as the resolution is improved, the volume of the vortex decreases. A statistical description of each of the four meshes employed here is given in Table 1. Although the number of nodes is quite large, it should be noted that a significant region of the domain, that is, the flowfield near the extent surface of the wing tip vortex, has been refined (see Figs. 6b, 6d, 6f, and 6h). In this problem, the wing tip vortex extends to the downstream boundary, which is more than 4.5 chord lengths downstream of the wing trailing edge.

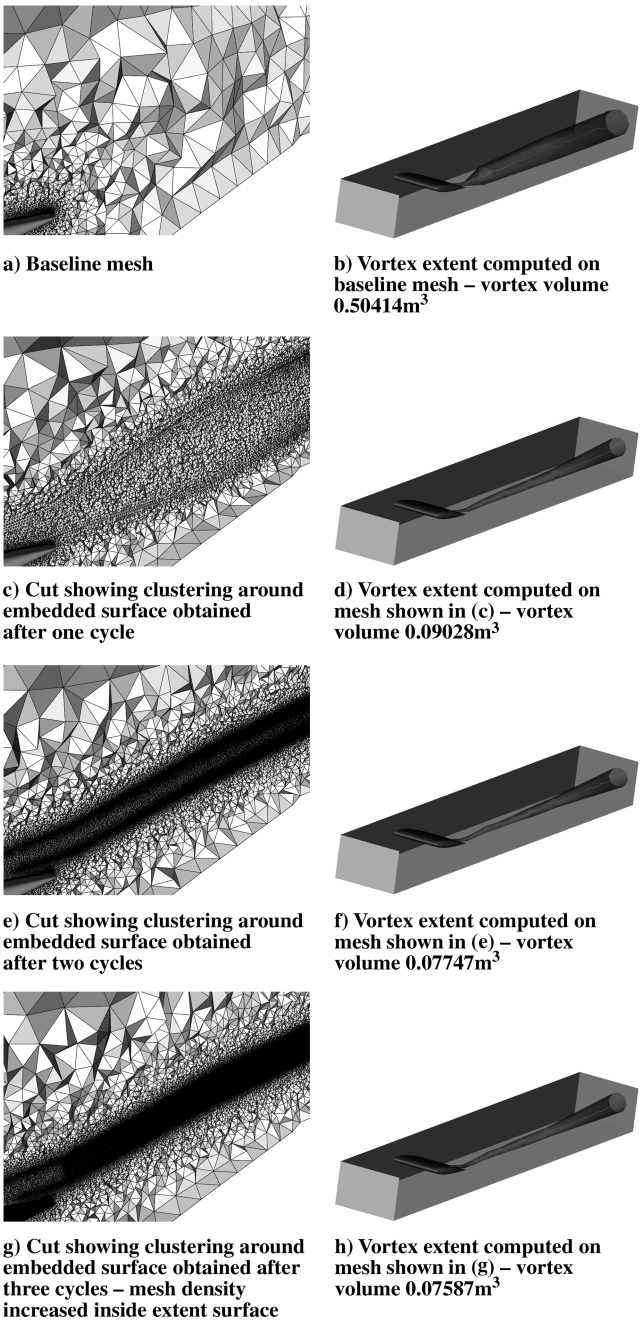
Figure 7a shows a shaded surface plot of the crossflow velocity in a plane located at $x/c = 1.462$ (see Fig. 4), predicted on the mesh designated cycle 3. In Fig. 7, all velocities and distances were nondimensionalized by the freestream velocity and the wing chord, respectively. Note the well-defined core and region of maximum crossflow velocity. Figure 7b shows a comparison between the computed and experimentally determined nondimensional circumferential velocity component in a plane located at $x/c = 1.462$.



a) Top view of wing

b) Zoom of wing leading edge at wing tip

Fig. 5 Mesh distribution on the upper surface of the wing and near the leading edge at the tip.



a) Baseline mesh

b) Vortex extent computed on baseline mesh – vortex volume 0.50414m^3

c) Cut showing clustering around embedded surface obtained after one cycle

d) Vortex extent computed on mesh shown in (c) – vortex volume 0.09028m^3

e) Cut showing clustering around embedded surface obtained after two cycles

f) Vortex extent computed on mesh shown in (e) – vortex volume 0.07747m^3

g) Cut showing clustering around embedded surface obtained after three cycles – mesh density increased inside extent surface

h) Vortex extent computed on mesh shown in (g) – vortex volume 0.07587m^3

Fig. 6 Meshes and computed extent surfaces for baseline and cycles 1, 2, and 3. The vortex extent surface for the solution computed on the baseline mesh produces only a crude representation of the vortex. As the computations are performed, the quality of the representation of the vortex improves dramatically.

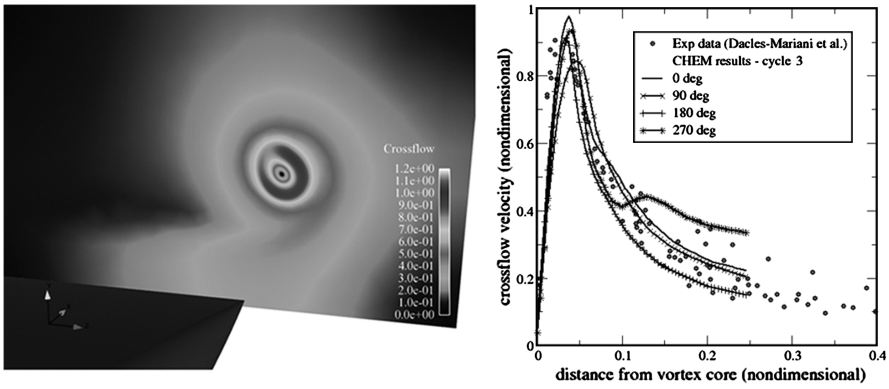
Table 1 Statistics for wing meshes

Meshes	Tetrahedra	Hexahedra	Prisms	Pyramids	Total
Baseline	5,605,165	42,262	7,092,684	18,692	12,758,803
Cycle 1	8,458,059	42,262	7,092,730	18,648	15,611,699
Cycle 2	23,496,122	42,262	7,092,730	18,648	30,639,762
Cycle 3	30,773,011	42,262	7,092,730	18,648	37,926,651

The computed results are plotted along rays emanating from the core line oriented at 0, 90, 180, and 270 deg with respect to the horizontal. There is variation in both the experimental data and predicted results because of the asymmetry of the flow, as illustrated in Fig. 7b. Although the peak velocities were reasonably well predicted, the slope in the core region is significantly underpredicted. We attribute this result to the inadequate resolution of the flow in the core region of the vortex over the wing. Because of the interference between the embedded mesh and the wing surface mesh, the nodal density in the vortex core over the wing could not be increased (see Fig. 6g). This is an inherent limitation of the current approach. We anticipate that increased nodal density in the vortex core in the region over the wing will result in improved agreement with the experimental data.

B. Feature Fitting with Mesh Refinement: Notional X-38 in Proximity to Booster

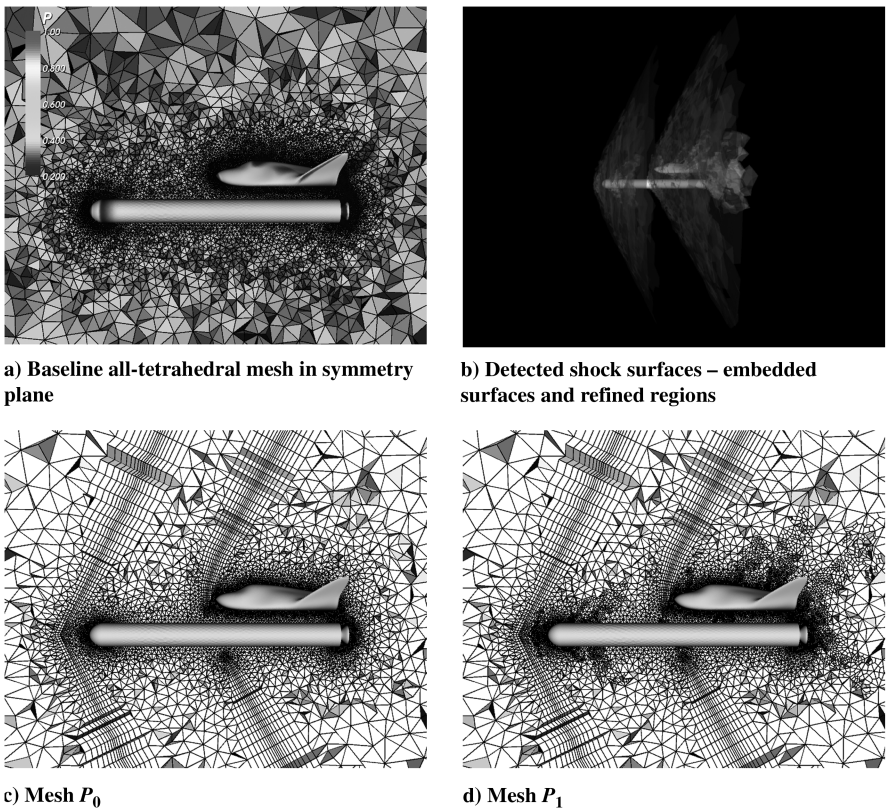
This test case configuration consists of a booster rocket and a notional X-38 geometry. The rocket is 2.17 times longer than the X-38. Inviscid transonic flow simulations were performed at a freestream Mach number of 1.2. The angle of attack is taken as 0 deg for the following simulations. The solution was computed using the HYB3D flow solver [38–40]. HYB3D is also designed to work with meshes that contain arbitrary elements. Figure 8 illustrates the adaptation process for this case.



a) Nondimensional crossflow velocity in the plane at x/c=1.462

b) Comparison of computed results with the experimental data of Dacles-Mariani et al. [34]

Fig. 7 Results computed using cycle 3 mesh in the plane at $x/c = 1.462$ (see Fig. 4).



a) Baseline all-tetrahedral mesh in symmetry plane

b) Detected shock surfaces – embedded surfaces and refined regions

c) Mesh P_0

d) Mesh P_1

Fig. 8 Illustration of solution adaptive meshing with embedded surfaces and refinement for the notional X-38/booster configuration.

1) To initiate the process, a computation was performed on a baseline all-tetrahedral mesh, shown in Fig. 8a, in which the orbiter, booster, and symmetry plane surfaces are shaded by pressure.

2) The solution is characterized by two primary bow shocks, one each for the booster and the orbiter, and numerous secondary shocks. Figure 8b shows the result of applying the normalized Mach-number-based shock detector. The two primary shocks are represented as embedded surfaces, whereas the other shocks are resolved with the isotropic mesh refinement.

3) The surfaces for the two primary bow shocks were extracted and represented as cones using least-squares fitting. A triangular surface mesh was generated on each embedded surface.

4) A volume mesh was generated with those embedded surface meshes. Figure 8c shows a cross section of the mesh near the symmetry plane. Prismatic layers are placed on the embedded surfaces (referred to as mesh P_0).

5) A new solution was computed on the mesh.

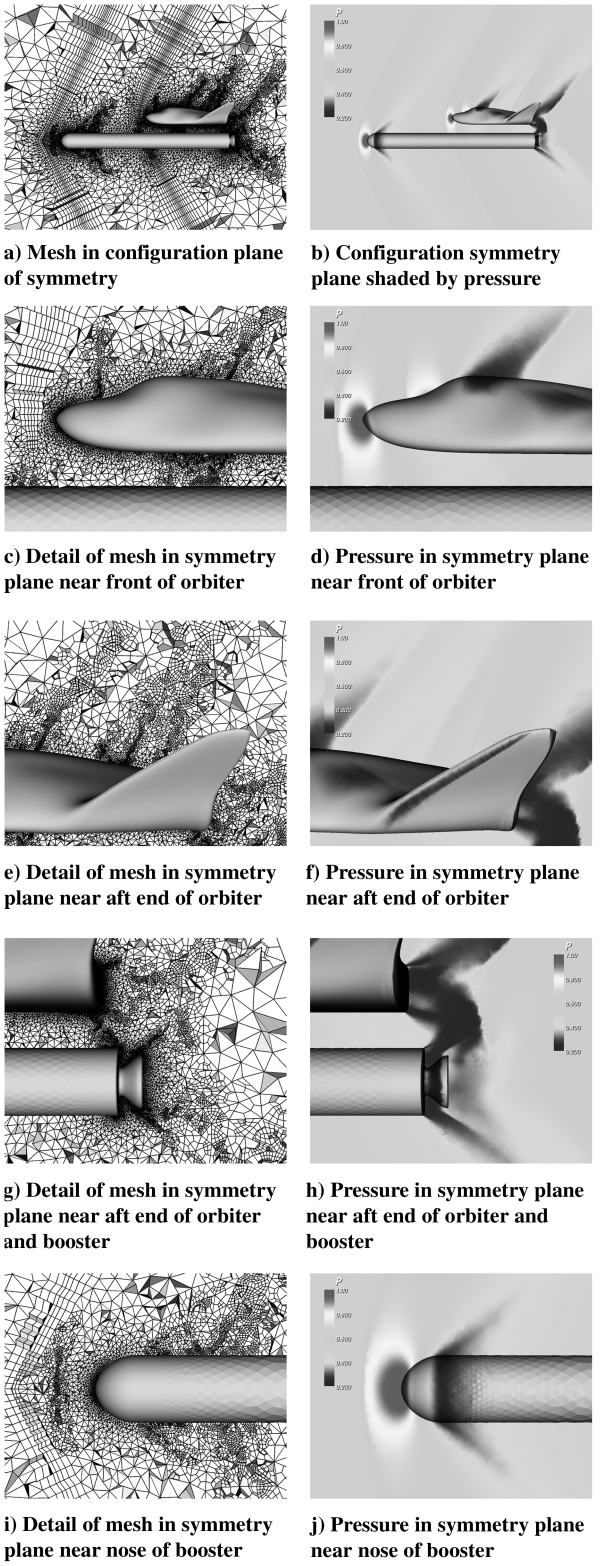
6) The process of extracting the bow shocks, embedding the surface meshes, etc., was repeated and the solution was interpolated onto the new mesh.

7) To avoid refining elements in the near field of each embedded mesh, all nodes within a prescribed distance from the embedded surfaces were excluded from the refinement operation. Because the mesh in the region near an embedded surface is anisotropic, in most cases it would not be desirable to refine these elements isotropically because that would eliminate the benefit of employing anisotropic elements. Figure 8d shows the mesh in the symmetry plane after one refinement cycle (referred to as mesh P_1).

8) A new solution was then computed on the refined mesh, and the procedure was repeated.

Figure 9 shows the mesh and the pressure on the booster, orbiter, and symmetry plane after two mesh refinement cycles (referred to as mesh P_2). Figure 10 plots the convergence history of the L2 norm of normalized residuals in log scale and the force coefficients in the x (longitudinal) and y (vertical) directions for the notional X-38 model (F_x and F_y, respectively). The main reason for the slight unsteadiness in the solution shown in Fig. 10 is probably due to the base flow behind the notional X-38 and the rocket booster. In Figs. 9a and 9b, notice that both bow shocks show good resolution with the feature-aligned embedded surfaces. Figures 9c and 9d show a well-resolved windshield shock that is clearly captured in the mesh. A shock on the top centerline of the orbiter that occurs just after acceleration over the "canopy" is also well resolved in both the mesh and the pressure plot. The shock that occurs between the orbiter and the booster is less well resolved, as the solution in this region of the domain appears somewhat noisy. In Figs. 9e and 9f, there are two shocks near the aft end of the orbiter that are well resolved. Figures 9g and 9h show a detailed view of the aft end of the booster near the nozzle. The expansion of the flow around the base of the booster, which is terminated by oblique shocks, is clearly evident in both images. A shock that is only barely evident in the flow is indicated by the presence of a pressure rise on the base of the orbiter. Additionally, there is a base shock present on the lower centerline of the orbiter after the expansion. Finally, Figs. 9i and 9j show a detailed view of the nose of the booster. In this case, the embedded surface did not follow the bow shock well near the nose of the booster. Isotropic refinement automatically refines the mesh in this region. However, there is no benefit to be derived from feature/face alignment for the isotropic refinement strategy. Additionally, there are oscillations in the surface pressures that appear in the region just downstream of the nose. We attribute these results to inadequate mesh resolution in this region. We have seen similar behavior in previous computations for a case in which coarse mesh was not aligned with the shock [33]. In a later example with better initial resolution, we will show significantly reduced oscillations. This issue highlights one of the problems associated with feature-based mesh adaptation, namely, that the initial mesh must be able to at least partially resolve the feature of interest. It also illustrates the importance of face/feature alignment.

We now consider the issue of mesh quality in more detail. Figure 11 shows the effect of isotropic refinement on the quality of the mesh with the embedded, feature-oriented prismatic mesh. Again, we use the included face angles to illustrate mesh quality. The quality plot for the hybrid mesh with the embedded surface meshes (P_0; Fig. 8c) is shown in Fig. 11a. In this case, there are peaks near 60 deg (for the triangular faces) and 90 deg (for the quadrilateral faces). Figure 11b shows the quality plot for the refined mesh after the first refinement cycle (P_1; Fig. 8d). Notice that there are relative peaks at 60, 90, and 180 deg. Figure 11c shows the quality plot for the



a) Mesh in configuration plane of symmetry

b) Configuration symmetry plane shaded by pressure

c) Detail of mesh in symmetry plane near front of orbiter

d) Pressure in symmetry plane near front of orbiter

e) Detail of mesh in symmetry plane near aft end of orbiter

f) Pressure in symmetry plane near aft end of orbiter

g) Detail of mesh in symmetry plane near aft end of orbiter and booster

h) Pressure in symmetry plane near aft end of orbiter and booster

i) Detail of mesh in symmetry plane near nose of booster

j) Pressure in symmetry plane near nose of booster

Fig. 9 Mesh P_2 and pressure field at various locations in the notional X-38/booster configuration symmetry plane.

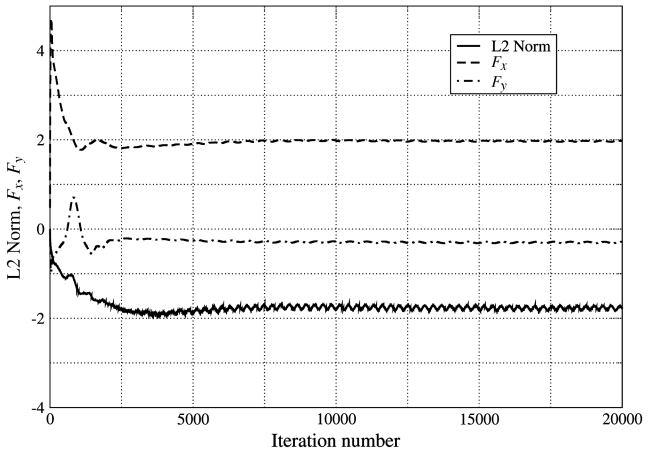
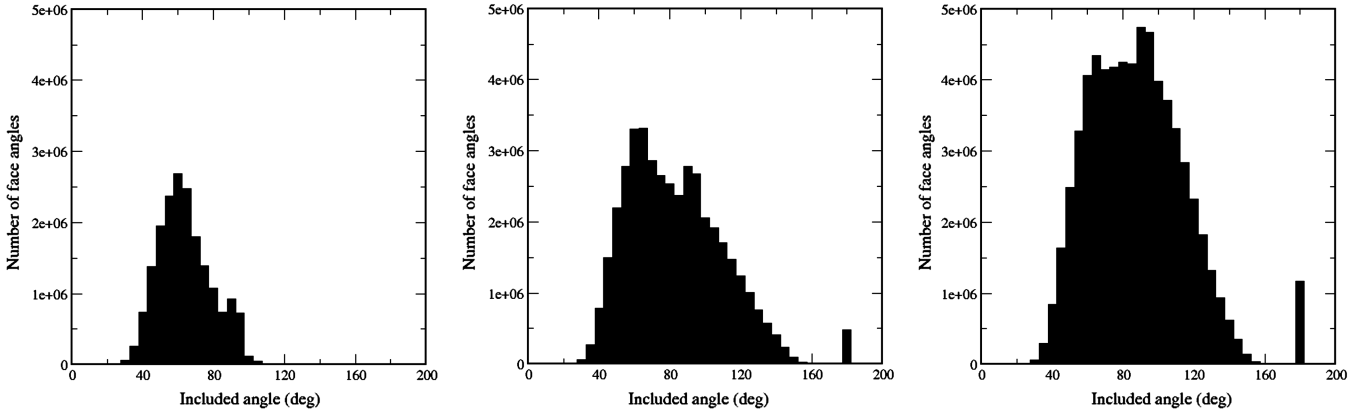


Fig. 10 Convergence history of mesh P_2 in terms of the L2 norm of normalized residuals in log scale and the force coefficients in the x and y directions for the notional X-38 model (F_x and F_y, respectively).

refined mesh after the second refinement cycle (P_2; Fig. 9). Notice the continued growth of angles near 90 deg, which indicates refinement of quadrilateral faces, and near 180 deg, which indicates that the number of general elements is growing. Figure 12 shows how the changes in included face angle are distributed for different face types for the first refinement. As anticipated, there is a net decrease in the number of triangular faces and the included angles for these faces center around 60 deg. Of course, this distribution reflects the underlying distribution of included face angles in a given mesh. We observe that a significant number of quadrilateral elements are added to the mesh. This occurs because of the method used to refine the faces. The peak near 180 deg for the quadrilateral faces occurs because of the refinement of an edge in a triangular face. The number of general polyhedral elements that are added to the mesh is relatively small in comparison with the overall number of elements.

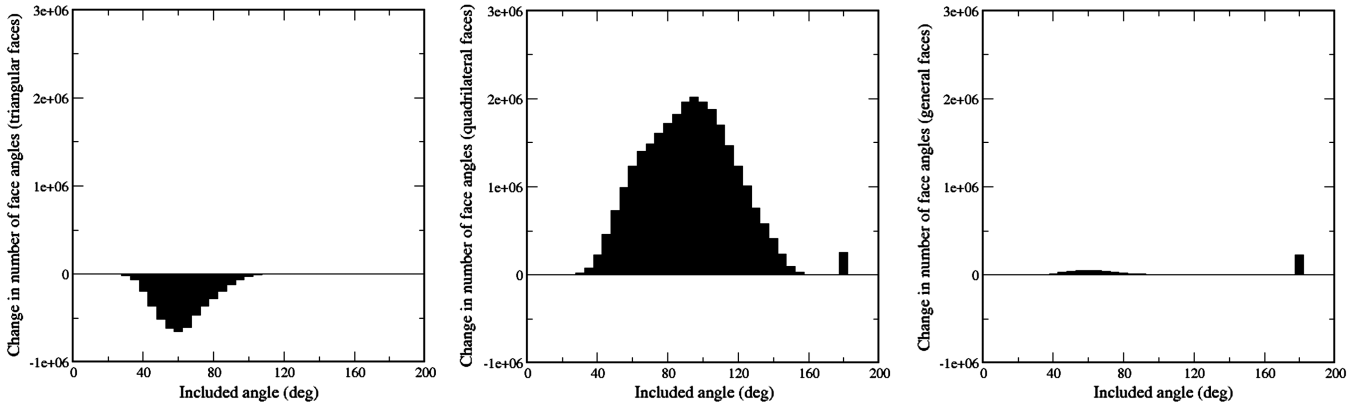
We also considered a case in which the regions near the embedded surfaces were discretized using isotropic tetrahedral elements. Meshes with tetrahedra around the embedded surfaces are referred as T_0 (without refinement), T_1 (after one refinement cycle), and T_2 (after two refinement cycles). Figure 13 shows mesh T_2 and pressure distribution on the geometry and the symmetry plane after two refinement cycles. Qualitatively, the pressure results look very similar to those shown in Fig. 9b. The primary exception is in the two bow shocks, for which, upon close examination, the solution on the prismatic mesh shows slightly better resolution. This can probably be attributed to the better face alignment in the layered prismatic mesh. The refined mesh looks somewhat less coherent when compared with Fig. 9a. This may be attributed to improved bow shock resolution because the quality of the feature detection is higher in the mesh with the embedded prismatic regions. Figure 14 shows the quality of the mesh and again shows the general trend of increasing the relative population of the obtuse face angles. In this case, there is no initial peak near 90 deg because the elements in the mesh before refinement are all tetrahedra.

For comparison purposes, mesh refinement was performed on a baseline all-tetrahedral mesh, B_0, using the results of the shock detection. In this case, no embedded surfaces were employed. The



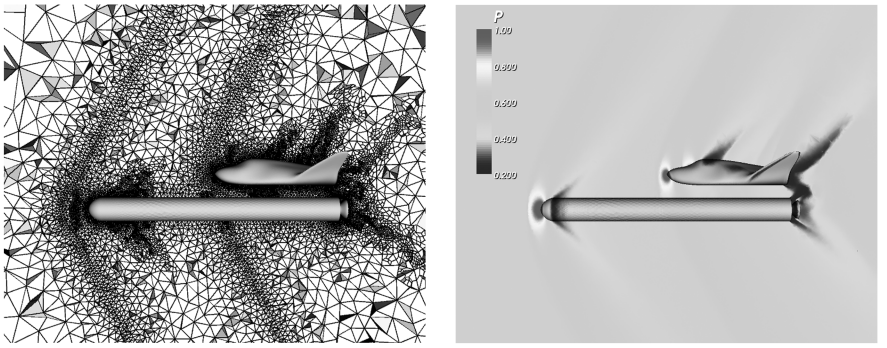
a) Face quality for mesh P_0 **b) Face quality for mesh P_1** **c) Face quality for mesh P_2**

Fig. 11 Effects of isotropic refinement on mesh quality for the notional X-38/booster configuration, meshes with embedded surface (prismatic). In these figures, face quality is measured in terms of the included angles of a face.



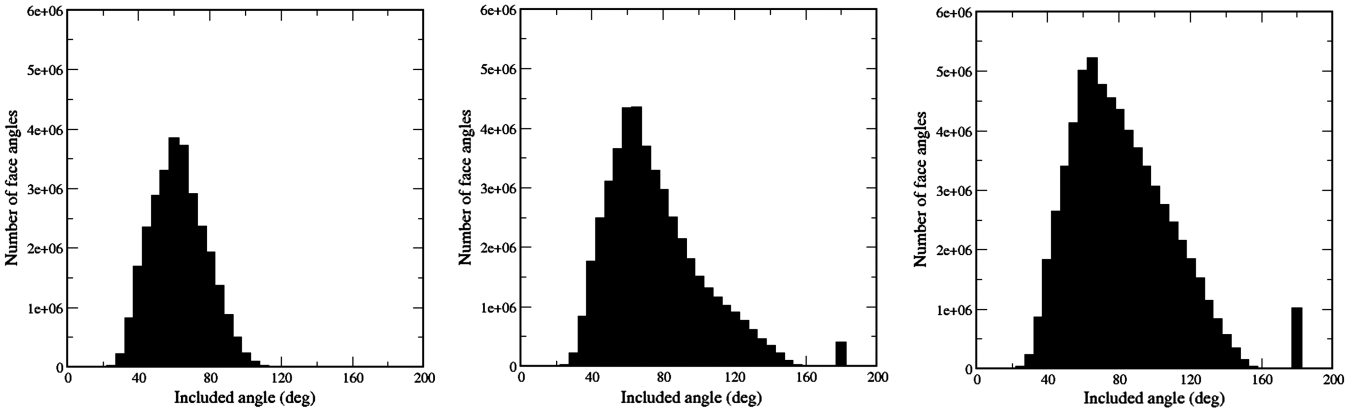
a) Change in face quality (triangular faces) **b) Change in face quality (quadrilateral faces)** **c) Change in face quality (general faces)**

Fig. 12 Change in mesh quality due to one cycle of isotropic refinement for different face types, in this case, the first refinement cycle. In these figures, face quality is measured in terms of the included angles of a face.



a) Mesh in configuration plane of symmetry b) Pressure in configuration plane of symmetry

Fig. 13 Mesh T_2 and pressure in configuration symmetry plane for the notional X-38/booster configuration; initial embedded mesh consists only of tetrahedra.



a) Face quality for mesh T_0 b) Face quality for mesh T_1 c) Face quality for mesh T_2

Fig. 14 Effects of isotropic refinement on mesh quality for the notional X-38/booster configuration, meshes with embedded surface (tetrahedral). In these figures, face quality is measured in terms of the included angles of a face.

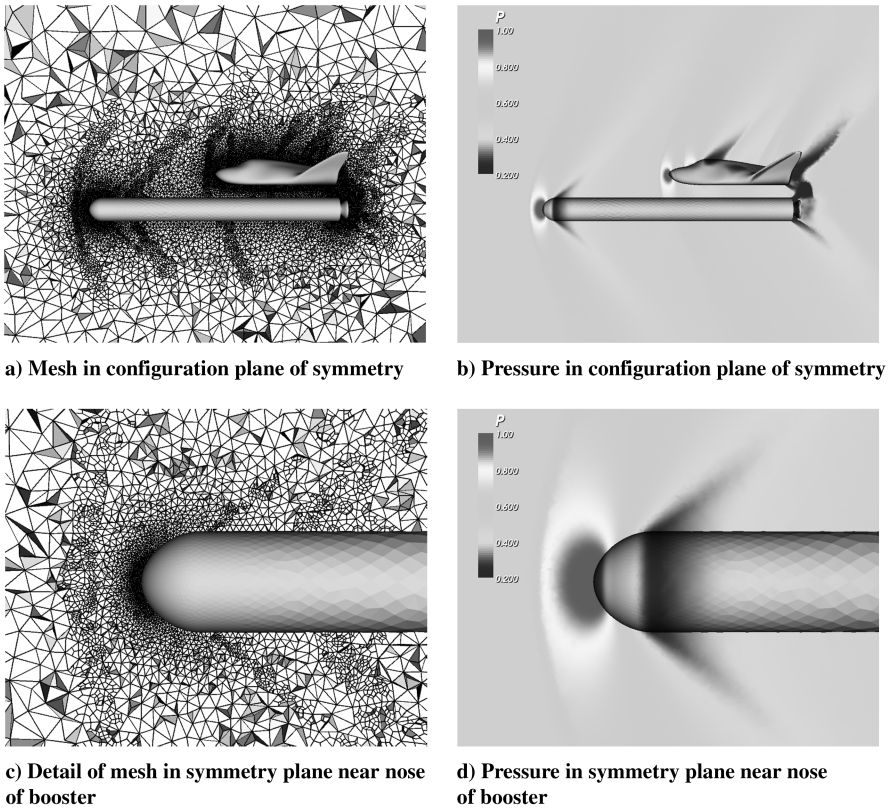
pressure and refined mesh, B_1, in the symmetry plane are shown in Fig. 15. We note that the baseline mesh was finer near the booster and the orbiter than in the previous two cases to capture flow features better without embedded surfaces. Figures 15a and 15b show that the two bow shocks of interest were not as well resolved as the two previous cases. This was expected because no attempt was made to align faces with either shock. In addition, Figs. 15c and 15d show that, with better resolution, the pressure oscillations on the nose of the booster are significantly reduced. Figure 15d shows oscillations, but they occur where the mesh is once again coarse, which supports our assertion made regarding the oscillations observed for the coarser mesh without feature alignment.

An element-based description of each mesh is presented in Table 2. In general, the percentage of general elements increases as the mesh is refined, reaching a maximum value of approximately 9% in the cases considered here. The baseline mesh with the tetrahedral embedded region, T_0, contains approximately 62% more elements than the baseline mesh with the prismatic embedded region, P_0. This is the advantage of using the prismatic embedded mesh. The bow shocks are highly anisotropic features. Therefore, it is very efficient to discretize regions containing these features using a similarly anisotropic mesh. However, the growth in the number of elements for the mesh with the prismatic embedded regions was larger for each refinement cycle than for the mesh with the tetrahedral embedded regions. This is related to the apparent improved coherence in the aforementioned feature detection, that is, a larger region was refined because the features were more coherent. Because fewer elements are required to obtain a solution that appears improved, at least indirectly, the prismatic approach may be preferred. It is difficult to draw any conclusions about the relative efficiency of the finer mesh with no surface embedding. However, the finer initial mesh did facilitate the prediction of better-quality surface pressures, at least in terms of reducing the oscillations, on the nose of the booster.

Figure 16a shows the pressure coefficient (C_p) distribution of three meshes, P_2, T_2, and B_1, on the x axis shown in Fig. 16b. The origin is on the nose of the notional X-38. The C_p jump in the plot is due to the bow shock. The smooth rise after that is due to compression in the region between the shock and the nose. The mesh with an embedded prismatic region after two refinement cycles, P_2, predicts the shock well.

Table 2 Statistics for notional X-38/booster meshes

Meshes	Tetrahedra	Hexahedra	Prisms	Pyramids	General	Total
Mesh with embedded prismatic region (P_0)	2,808,214	2336	166,748	3269	0	2,980,567
One refinement cycle (P_1)	2,001,732	2,145,677	166,592	3257	226,136	4,543,394
Two refinement cycles (P_2)	1,888,467	3,649,538	164,092	3183	555,050	6,260,330
Mesh with embedded tetrahedral region (T_0)	4,817,941	0	0	0	0	4,817,941
One refinement cycle (T_1)	4,191,087	1,544,474	0	0	199,337	5,934,898
Two refinement cycles (T_2)	4,059,574	2,824,811	0	0	488,968	7,373,353
Baseline tetrahedral mesh (B_0)	5,329,333	0	0	0	0	5,329,333
One refinement cycle (B_1)	4,037,631	3,265,307	0	0	397,346	7,700,284



a) Mesh in configuration plane of symmetry

b) Pressure in configuration plane of symmetry

c) Detail of mesh in symmetry plane near nose of booster

d) Pressure in symmetry plane near nose of booster

Fig. 15 Mesh B_1 and pressure in configuration symmetry plane for notional X-38/booster configuration with refinement only. These results were obtained after one refinement cycle.

VI. Conclusions

A solution adaptive mesh generation approach using feature-aligned embedded surfaces was presented. This approach exploits the capability of feature detection algorithms to geometrically characterize certain types of features, in this case shocks and vortices, as discrete surfaces that are used in the mesh generation process. For features that are geometrically or topologically complex, isotropic mesh refinement is offered as a supplemental approach. The results presented here suggest that using an embedded feature-aligned anisotropic mesh near these appropriate features may be an effective strategy. This approach could be coupled with an anisotropic refinement method [9,33] for further refinement in the embedded regions, if needed. Issues associated with the stability and accuracy of the flow solvers that employ meshes with general elements need additional research to be resolved. The algorithm to extract complex embedded surfaces needs to be further enhanced to be more effective and robust.

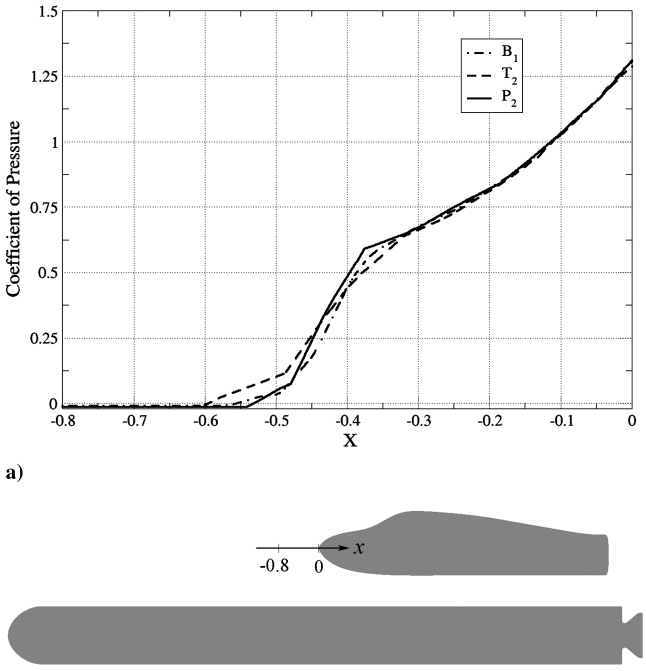


Fig. 16 Pressure coefficient distribution on the x axis: a) pressure coefficient distribution from three meshes, P_2, T_2, and B_1; and b) coordinate system.

Acknowledgments

This work was partially supported by the NASA Constellation University Institutes Project (NCC3-994 and NNC06GA29G), the National Science Foundation (ACS-0085969, EEC-0121807), and the Army Research Office (DAA D19-00-1-0155). The assistance of Edward Luke, Department of Computer Science and Engineering at Mississippi State University, is gratefully acknowledged.

References

[1] Thompson, J. F., Soni, B. K., and Weatherill, N. J. (eds.), *Handbook of Grid Generation*, CRC Press, Boca Raton, FL, 1998.

[2] Thompson, J. F., "A Reflection on Grid Generation in the 90s: Trends, Needs, and Influences," *Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Field Simulations*, edited by B. K. Soni, Mississippi State Univ., Mississippi State, MS, 1996, pp. 1029–1110.

[3] Amato, M., Iaccarino, G., and Paparone, L., "Automatic Local Grid Refinement for Multiblock Solvers," *Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Field Simulations*, edited by B. K. Soni, Mississippi State Univ., Mississippi State, MS, 1996, pp. 77–86.

[4] 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, No. 4, 2000, pp. 1183–1204. doi:10.1016/S0045-7825(99)00373-4

[5] Yang, J. C., and Soni, B. K., "Structured Adaptive Grid Generation," *Applied Mathematics and Computation*, Vol. 65, No. 1–3, 1994, pp. 256–279.
doi:10.1016/0096-3003(94)90182-1.
[6] Carpenter, J. G., and McRae, D. S., "Adaption of Unstructured Meshes using Node Movement," *Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Field Simulations*, edited by B. K. Soni, Mississippi State Univ., Mississippi State, MS, 1996, pp. 269–278.
[7] Benson, R. A., McRae, D. S., and Edwards, J. R., "Numerical Simulations Using a Dynamic Solution-Adaptive Grid Algorithm, with Applications to Unsteady Internal Flows," AIAA Paper 92-2719-CP, 1992.
[8] Benson, R. A., and McRae, D. S., "A Three-Dimensional Dynamic Solution-Adaptive Mesh Algorithm," AIAA Paper 90-1566, 1990.
[9] Senguttuvan, V., Chalasani, S., Luke, E. A., and Thompson, D. S., "Adaptive Mesh Refinement Using General Elements," AIAA Paper 2005-0927, 2005.
[10] Chalasani, S., Senguttuvan, V., Thompson, D., and Luke, E., "On the Use of General Elements in Fluid Dynamics Simulations," *Communications in Numerical Methods in Engineering*, Vol. 24, No. 6, 2008, pp. 435–448.
doi:10.1002/cnm.1017
[11] Murayama, M., Nakahashi, K., and Sawada, K., "Simulation of Vortex Breakdown Using Adaptive Grid Refinement with Vortex-Center Identification," *AIAA Journal*, Vol. 39, No. 7, 2001, pp. 1305–1312.
doi:10.2514/2.1448
[12] Rivara, M. C., "Selective Refinement/Derefinement Algorithms for Sequences of Nested Triangulations," *International Journal for Numerical Methods in Engineering*, Vol. 28, 1989, pp. 2889–2906.
doi:10.1002/nme.1620281212
[13] Spragle, G. S., Smith, W. A., and Weiss, J. M., "Hanging Node Solution Adaption on Unstructured Grids," AIAA Paper 95-0216, 1995.
[14] Zhang, S. J., Liu, J., Chen, Y.-S., and Wang, T. S., "Adaptation for Hybrid Unstructured Grid with Hanging Node Method," AIAA Paper 2001-265, 2001.
[15] Cavallo, P. A., and Baker, T., "Efficient Delaunay-Based Solution Adaptation for Three-Dimensional Unstructured Meshes," AIAA Paper 2000-0809, 2000.
[16] Schneiders, R., "Refining Quadrilateral and Hexahedral Element Meshes," *Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Field Simulations*, edited by B. K. Soni, Mississippi State Univ., Mississippi State, MS, 1996, pp. 679–688.
[17] Laflin, K. R., and McRae, D. S., "Solution-Dependent Grid-Quality Assessment and Enhancement," *Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Field Simulations*, edited by B. K. Soni, Mississippi State Univ., Mississippi State, MS, 1996, pp. 3–12.
[18] Yang, J. C., and Soni, B. K., "Algebraic Grid Adaptation Method using Non-Uniform Rational *B*-Spline Surface Modeling," *Proceedings of Fluid and Thermal Workshop*, NASA Langley Research Center, Hampton, VA, 1992, pp. 379–399.
[19] Yang, J. C., and Lie, C. P., "The Full-Dimensional Adaptive Grid Generation Using Two-Dimensional Equidistribution Statement," *Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Field Simulations*, edited by B. K. Soni, Mississippi State Univ., Mississippi State, MS, 1996, pp. 147–156.
[20] Mavriplis, D. J., "Grid Resolution Study of a Drag Prediction Workshop Configuration Using the NSU3D Unstructured Mesh Solver," AIAA Paper 2005-4729, 2005.
[21] Ito, Y., Shih, A. M., Koomullil, R. P., and Soni, B. K., "A Solution-Based Adaptive Redistribution Method for Unstructured Meshes," *Proceedings of the 15th International Meshing Roundtable*, Springer, Berlin/Heidelberg, Sept. 2006, pp. 147–161.
doi:10.1007/978-3-540-34958-7.
[22] Lovely, D., and Haimes, R., "Shock Detection from Computational Fluid Dynamics Results," AIAA Paper 1999-3285, 1999.
[23] Jankun-Kelly, M., Thompson, D. S., Brewer, W., Jiang, M., and Machiraju, R., "A Multistage Vortex Visualization Method," AIAA Paper 2006-0946, 2006.
[24] Jankun-Kelly, M., Jiang, M., Thompson, D., and Machiraju, R., "Vortex Visualization for Practical Engineering Applications," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 12, No. 5, 2006, pp. 957–964.
doi:10.1109/TVCG.2006.201
[25] Berdahl, C., and Thompson, D. S., "Eduction of Swirling Structured Using the Velocity Gradient Tensor," *AIAA Journal*, Vol. 31, No. 1, 1993, pp. 97–103.
doi:10.2514/3.11324
[26] Garth, C., Tricoche, X., Salzbrunn, T., and Scheuermann, G., "Surface Techniques for Vortex Visualization," *Proceedings of Joint Eurographics-IEEE TCVG Symposium on Visualization*, A. K. Peters, Wellesley, MA, 2004, pp. 155–164.
[27] Ito, Y., and Nakahashi, K., "Direct Surface Triangulation Using Stereolithography Data," *AIAA Journal*, Vol. 40, No. 3, 2002, pp. 490–496.
doi:10.2514/2.1672
[28] Ito, Y., and Nakahashi, K., "Surface Triangulation for Polygonal Models Based on CAD Data," *International Journal for Numerical Methods in Fluids*, Vol. 39, No. 1, 2002, pp. 75–96.
doi:10.1002/fld.281
[29] Ito, Y., Shih, A. M., Soni, B. K., and Nakahashi, K., "Multiple Marching Direction Approach to Generate High Quality Hybrid Meshes," *AIAA Journal*, Vol. 45, No. 1, 2007, pp. 162–167.
doi:10.2514/1.23260
[30] Ito, Y., Shih, A. M., and Soni, B. K., "Hybrid Mesh Generation with Embedded Surfaces Using a Multiple Marching Direction Approach," *International Journal for Numerical Methods in Fluids* (to be published).
doi:10.1002/fld.1962
[31] Chalasani, S., and Thompson, D., "Quality Improvements in Extruded Meshes Using Topologically Adaptive Generalized Elements," *International Journal for Numerical Methods in Engineering*, Vol. 60, No. 6, 2004, pp. 1139–1159.
doi:10.1002/nme.1000
[32] Knupp, P., "On the Invertibility of the Isoparametric Map," *Computer Methods in Applied Mechanics and Engineering*, Vol. 78, No. 3, 1990, pp. 313–329.
doi:10.1016/0045-7825(90)90004-6
[33] Chalasani, S., Luke, E., Senguttuvan, V., and Thompson, D., "Assessing Generalized Mesh Quality via CFD Solution Validation," AIAA Paper 2005-0687, 2005.
[34] Dacles-Mariani, J., Zilliac, G. G., Chow, J. S., and Bradshaw, P., "Numerical/Experimental Study of a Wingtip Vortex in the Near Field," *AIAA Journal*, Vol. 33, No. 9, 1995, pp. 1561–1568.
doi:10.2514/3.12826
[35] Wu, J., Tang, L., Luke, E. A., Tong, X.-L., and Cinnella, P., "Comprehensive Numerical Study of Jet-Flow Impingement over Flat Plates," *Journal of Spacecraft and Rockets*, Vol. 39, No. 3, 2002, pp. 337–366.
[36] Spalart, P. R., and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 1992-0439, 1992.
[37] Gaither, J. A., Marcum, D. L., and Mitchell, B., "SolidMesh: A Solid Modeling Approach to Unstructured Grid Generation," *Proceedings of 7th International Conference on Numerical Grid Generation in Computational Field Simulation*, International Society of Grid Generation, Mississippi State, MS, 2000, pp. 829–838.
[38] 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.
doi:10.2514/2.655
[39] Koomullil, R. P., Soni, B. K., and Singh, R. K., "A Comprehensive Generalized Mesh System for CFD Applications," *Mathematics and Computers in Simulation*, Vol. 78, Nos. 5–6, 2008, pp. 605–617.
doi:10.1016/j.matcom.2008.04.005.
[40] Koomullil, R. P., Cheng, G. C., Soni, B. K., Noack, R. W., and Prewitt, N. C., "Moving-Body Simulations Using Overset Framework with Rigid Body Dynamics," *Mathematics and Computers in Simulation*, Vol. 78, Nos. 5–6, 2008, pp. 618–626.
doi:10.1016/j.matcom.2008.04.009.

Z. Wang
Associate Editor