INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
*Int. J. Numer. Meth. Fluids* 2011; **66**:1274–1294
Published online 23 March 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/fld.2312

# Feature-based adaptive mesh refinement for wingtip vortices

N. Kasmai[1], D. Thompson[2,*,†], E. Luke[3], M. Jankun-Kelly[1] and R. Machiraju[4]

[1]*Center for Advanced Vehicular Systems-SimCenter, Mississippi State University, Mississippi State, MS 39762, U.S.A.*
[2]*Department of Aerospace Engineering, Mississippi State University, P.O. Box A, Mississippi State, MS 39762, U.S.A.*
[3]*Department of Computer Science and Engineering, Mississippi State University, Mississippi State, MS 39762, U.S.A.*
[4]*Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210, U.S.A.*

## SUMMARY

Feature-based solution-adaptive mesh refinement is an attractive strategy when it is known *a priori* that the resolution of certain key features is critical to achieving the objectives of a simulation. In this paper, we apply vortex characterization techniques, which are typically employed to visualize vortices, to identify regions of the computational domain for mesh refinement. We investigate different refinement strategies that are facilitated by these vortex characterization techniques to simulate the flow past a wing in a wind tunnel. Our results, which we compare with experimental data, indicate that it is necessary to refine the region within and near the vortex extent surface to obtain an accurate prediction. Application of the identified mesh refinement strategy also produced observed improvement in the results predicted for a spinning missile with deflected canards. Copyright © 2010 John Wiley & Sons, Ltd.

Received 10 August 2009; Revised 27 January 2010; Accepted 29 January 2010

KEY WORDS: adaptive mesh refinement; vortices; feature detection

## 1. INTRODUCTION

There has been much work reported in the last decade on the development of algorithms for extracting and characterizing vortices in computational fluid dynamics (CFD) simulations. The impetus to innovate has been driven by the need to develop quantitative as well as qualitative descriptions of the flow and to create effective visualizations of complex flow phenomena. However, the use of vortex characterization methods beyond visualization is still not widespread. Solution-adaptive meshing is one candidate application of this technology.

Solution-adaptive mesh refinement for vortex-dominated flows can be challenging. There are two distinct physical phenomena that must be resolved, vortex formation and downstream vortex convection. If vortex formation is not adequately resolved, the downstream evolution of the vortex will not be predicted accurately. Likewise, if there is inadequate resolution of the vortex downstream of its formation, the evolution will be poorly predicted. The primary culprit in both cases is the artificial dissipation that is inherent to numerical solution techniques, which tends to reduce the

*Correspondence to: D. Thompson, Department of Aerospace Engineering, Mississippi State University, P.O. Box A, Mississippi State, MS 39762, U.S.A.
†E-mail: dst@ae.msstate.edu

Contract/grant sponsor: NSF; contract/grant number: ITR/NGS-0326386
Contract/grant sponsor: NASA
Contract/grant sponsor: DoD

velocity gradients. The peak velocities in the vortex tend to be reduced and the vortex decays more rapidly than is physically observed. Since enhanced mesh resolution decreases the effects of the artificial dissipation, it is advantageous to increase the nodal density in regions near the vortex. Although this approach relies on an indirect indicator of error, i.e. the presence of a feature, it is useful because there are relatively few, if any, rigorous error estimators for finite-volume schemes of the type employed for most CFD codes [1].

The primary contributions of this paper are (1) application of feature-based visualization strategies to facilitate adaptive mesh refinement and (2) identification of effective mesh refinement strategies for vortices by comparison of predicted results with experimental data. We demonstrate an adaptive meshing strategy that employs feature-level descriptions of vortices to identify regions of the computational domain in which the mesh should be refined. An approach of this type is attractive when it is known *a priori* that the resolution of certain key features is critical to achieving the objectives of the simulation. Although other efforts to apply techniques traditionally employed for visualization to solution adaptive meshing have been reported in the literature [2–6], this strategy has not been fully exploited as the methods have become more sophisticated. Current vortex characterization algorithms provide us with tools to investigate which mesh refinement strategies are most effective for vortical flows.

We employ the approach described by Jankun-Kelly *et al.* [7] to extract the core line and extent surface (as defined by the surface of maximum tangential velocity [8]) for each vortex. In essence, we create geometric descriptions of intrinsic feature properties and employ them to identify regions of the computational domain for refinement. In the current effort, regions near the core line, near the extent surface, or inside the extent surface can be identified and employed to mark mesh nodes located in these regions. Once the nodes are marked, the strategy described in Chalasani *et al.* [9] is employed to refine the mesh in the specified regions. The flow solver we use is the Loci/CHEM code [10–12]. We demonstrate the efficacy of our approach by applying it to the prediction of the flow fields around two different configurations, a wing in a wind tunnel at 10° angle of attack [13] and a missile configuration [14] at 0° angle of attack with canards deflected 15° spinning at 30 and 60 Hz. We use the wing results to evaluate the effectiveness of different refinement strategies by comparing the numerically predicted results with experimental data [13].

## *1.1. Related work*

In contrast to finite-element techniques [15], the finite-volume techniques commonly employed for CFD simulations do not have a direct mechanism for error estimation [1]. Local error approximation techniques for finite-volume techniques include a divided difference [16], the difference between the solution and a local reconstruction of the solution [17], and the eigenvalues of a symmetric Hessian matrix for a derived field quantity [18]. Gradient-based weight functions [19, 20] can also be grouped into this category. These techniques can be considered to be feature-based approaches with the approximated error being the feature of interest. A slightly different strategy is based on the premise that a flow field is well resolved if its features are well resolved. In this case, the features of interest are fluid dynamics features, such as shock waves and expansions [21].

Several categories of vortex detection algorithms have been reported in the literature. In field-type methods, a scalar field is defined that can be visualized as regions using techniques such as isosurfacing [22–27]. Topology-based methods exploit the fact that there is a critical point at the vortex core in the plane containing the swirling motion. By their very nature, these methods provide a description of a vortex in terms of its core line [28–32] or core region [33]. There are also techniques that produce feature-level descriptions of a vortex using ridge/valley-line extraction [7, 34–36]. Each of these methods has advantages and disadvantages that have been documented in the literature. Hybrid methods, which exploit multiple detection methods, have also been developed [37–39].

Several feature-based adaptive mesh refinement strategies tailored to vortices have appeared in the literature. Dindar *et al.* [2] and Kenwright and Haimes [3] employed mesh refinement at the vortex core line, identified using the eigenvector-based approach of Sujudi and Haimes [28], to predict the flow for a UH-60A rotor blade in hover. The radius of the refined region was taken to be the distance at which the local vorticity magnitude was 10% of the core value. Dindar *et al.* reported that the feature-based refinement resolved small-scale features around the vortex core while an error-based refinement did not.

Murayama *et al.* [4] also employed mesh refinement at the vortex core line, this time identified using the critical point-based approach of Sawada [40], to enhance predictions of vortex breakdown on a delta wing. They observed that results computed on meshes with refinement along the core line showed improved agreement with experimental data relative to results computed on meshes without refinement. They attributed this improvement, in part, to the enhanced resolution of the streamwise pressure gradient along the core of the vortex.

Kim and Rhee [5] computed the solution for the Dacles-Mariani wing [13] on a single-block, structured grid. They employed the second invariant of the rate of deformation tensor as the refinement sensor and studied the effects of various turbulence models.

Turnock *et al.* [6] employed the statistical vortex detection method described in Pemberton *et al.* [41] to identify vortex core regions and then semi-automatically refine the mesh in these regions for a marine propeller application. A prescribed mesh density was employed across the vortex core. An annular region of lower density cells was used in the high-tangential velocity zone immediately outside the viscous core. Their results showed significant improvement relative to computations on the unrefined mesh.

With a somewhat different philosophy, Ito *et al.* [42] employed extent surfaces computed using vortex characterization [7] as a component of a *mesh regeneration* strategy for the Dacles-Mariani wing [13]. The vortex extent surface was triangulated and used as an embedded surface that was maintained as the volume mesh was generated. After each computation, a new embedded surface and volume mesh was generated. While promising, this approach is limited because an embedded surface cannot lie too closely to other surfaces because of the difficulties associated with generating meshes in small gaps.

The adjoint-based approach, which is a relatively recent development, has shown much promise for performing adaptation that impacts the accuracy of selected functional outputs [43–47]. This is accomplished by solving an additional equation that relates the sensitivity of the output function to the linearized flow residuals. The resulting adjoint variables can then be used as weight functions for the adaptive meshing procedure. Recently, Fidkowski and Roe [48] established a connection between the entropy-based refinement and the adjoint-based approach. The entropy variables employed Fidkowski and Roe are shown to satisfy an adjoint equation suggesting that adjoint-like results may be obtained without the additional expense of solving the adjoint equations.

## 2. APPROACH: FEATURE-BASED MESH REFINEMENT

The motivation for our work comes, in part, from the results of Murayama *et al.* [4] who attributed the observed improvement in results predicted for a delta wing experiencing vortex breakdown to improved resolution of the streamwise pressure gradient in the vortex core, and from the observation that there is significant curvature in the velocity field near the surface of maximum tangential velocity [13]. In our approach, we employ vortex characterization, which has been primarily used for visualization applications, to identify regions of the computational domain near the vortex. Specifically, we employ the vortex characterization techniques described in detail in Jankun-Kelly *et al.* [7] and briefly described in Section 3 to generate a high-level description of a vortex that consists of its core line position and an extent surface, which can be defined in an unambiguous manner by lofting between curves that describe the location of the maximum tangential velocity in the swirl plane [8], i.e. the plane normal to the vortex core line that contains the swirling motion.

The adaptive mesh refinement strategy is a multi-step process that is illustrated in Figure 1. The flow solver computes a solution on a baseline mesh. The solution and mesh are input to the vortex characterization method. The resulting abstract vortex characterizations are input to a sensor module that marks those nodes in the mesh that are near the core line, near the extent surface, or inside the extent surface. Note that, since the vortex characteristics are mesh independent, the

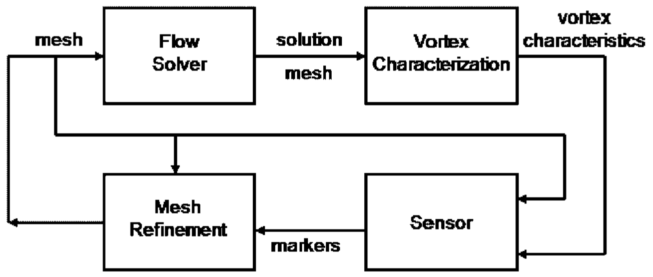

Figure 1. Schematic of adaptive mesh refinement process.

mesh to be marked does not have to be the same as the mesh from which the vortex characteristics were computed. These markers are input to the mesh refinement procedure, to generate a refined mesh. The refined mesh is then passed back to the flow solver for a new simulation and the process is repeated.

## 3. VORTEX CHARACTERIZATION

The basic function of vortex characterization is to describe a vortex in terms of high-level descriptors. In deference to its central role in our mesh refinement strategy, we briefly describe the vortex characterization method described by Jankun-Kelly *et al.* [7] to obtain the vortex core line and the extent surface. We also discuss how this vortex characterization algorithm lends itself to feature-based mesh refinement. Please refer to Figure 2 where we illustrate the results obtained from each step of the method using the serrated wing data set [49].

### *3.1. Identify candidate cells and aggregate*

The local extrema method [7] exploits the fact that, for certain scalar fields, a vortex core line coincides with a line-type extremum, or ridge/valley line, in the field variable. Therefore, a vortex core line can be extracted from an appropriate scalar field $f$ by locating local extrema in a series of properly oriented planes and connecting the location of the extreme values with line segments to form a curve. In this case, we use the swirl parameter [25] as the scalar field. The normal to the swirl plane, i.e. the plane containing the swirling motion, is taken to be parallel to the real eigenvector of the velocity gradient tensor. The cells containing local extrema are marked as candidate cells and contiguous candidate cells are grouped into aggregates. The result for this step is shown in Figure 2(a).

### *3.2. Identify aggregate topology and segment*

One of the problems with existing vortex detection algorithms is that they have difficulty distinguishing vortices that are merging. In our algorithm, only one vortex core line is found per aggregate. Therefore, branching aggregates must be subdivided into component and non-branching aggregates before core line extraction can be applied. We employ a variation of the $k$-means clustering algorithm [50, 51] designed specifically for unstructured meshes to identify where a branch occurs. The idea is to decompose the aggregate into small segments (clusters) and determine the topology of the aggregate based on cluster neighbor counts. A *simple* cluster should have at most two neighboring clusters while a *terminal* cluster has only one neighbor cluster. A *non-simple*, or branching, cluster will have more than two neighboring clusters. These branching clusters are identified and merged separately. Figure 2(b) shows the result from the clustering and merging step of our algorithm. Notice that we have correctly distinguished the merging vortices from the merged vortex.

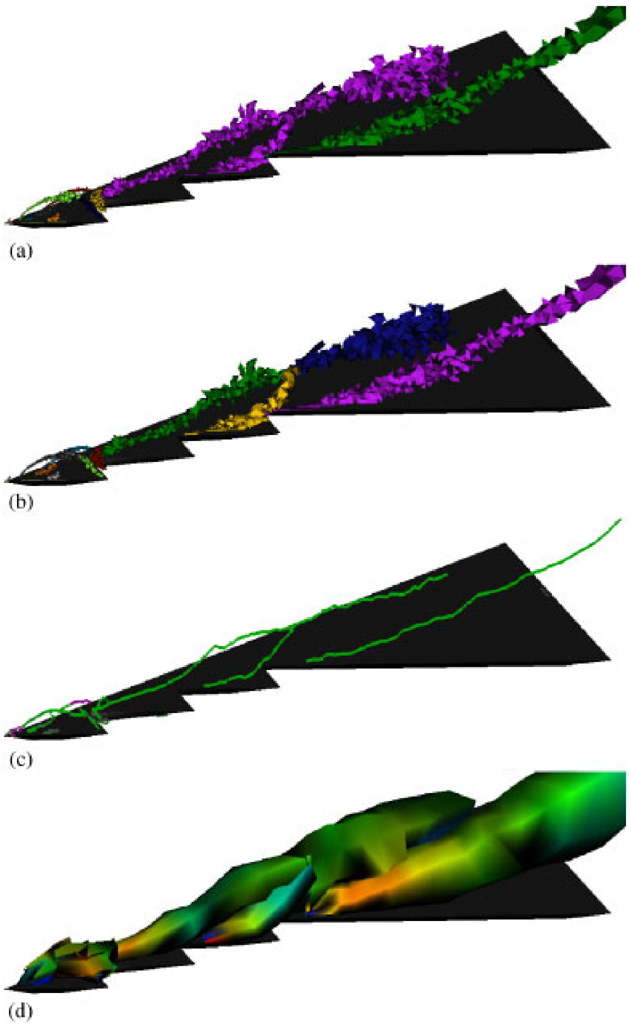

Figure 2. Schematic of vortex characterization algorithm from Jankun-Kelly *et al.* [7]; @ 2006 IEEE. The flow about the serrated wing [49] is employed to illustrate the results of each step: (a) candidate cells are identified and aggregated. Aggregates with less than a specified number of cells are removed; (b) distinct aggregates identified by clustering algorithm. Notice that the single large aggregate has been divided into three constituent aggregates; (c) vortex core lines extracted, one line per aggregate; and (d) vortex visualization with extent surface shaded by local tangential velocity.

### *3.3. Extract vortex core line*

The seed point for the core line extraction in each aggregate is estimated to be the point on the aggregate's bisecting plane with the largest swirl value. Its position is corrected using a function fitting procedure that produces subcell resolution. A conical variation of the scalar field is assumed and the location of the core line is taken to be the point in the plane that produces the best fit of the surrounding scalar data. A prediction of the location of the next point on the core line is made using the swirl vector, which is an approximate tangent to the core line. The correction algorithm is then applied and the process is repeated. The core line extraction algorithm terminates when it hits an aggregate other than the one in which it started, when it hits a domain boundary, or when it has traversed more than a specified number of contiguous unmarked cells. The results from this step are illustrated in Figure 2(c) for the serrated wing.

### 3.4. Extract vortex extent

Vortex extent is a nebulous concept [52]. The maximum tangential velocity definition [8] is appealing because of its simplicity and its lack of ambiguity. Following Banks and Singer [37], the two-dimensional extent curve is found by marching radially outward, independently in each of several directions in the swirl plane, until a local maximum in tangential velocity is found. In some cases, such as a vortex near a no-slip boundary, a plateau in the maximum tangential velocity may exist rather than an extreme value [53]. Detection of these plateaus is based on identifying small rates of change in the tangential velocity. The vortex extent surface is created by repeatedly finding two-dimensional extent curves and lofting between them. The extent surfaces are shown in Figure 2(d).

### 3.5. Impact on adaptive mesh refinement

The vortex characterization process outputs a compact description of a vortex by means of a set of characteristics that may include geometric, kinematic, and dynamic properties of the vortex. The primary geometrical characteristics of a vortex are the position of the vortex core line and a description of the vortex extent. Here, the core line is represented as a series of points while the extent is represented as a series of connected curves. Each closed curve lies in a plane perpendicular to the core line.

There are two attributes of our algorithm that make it attractive for feature-based adaptive mesh refinement. First, because of our aggregate topology and segmentation algorithm, the method can extract core lines for complex vortex topologies. Second, the extracted extent surface is an unambiguous descriptor that captures the region of high curvature in the tangential velocity profile. Other extent surfaces could be derived but, in general, they depend on problem-specific, user-defined thresholds rather than intrinsic vortex characteristics.

## 4. IDENTIFICATION OF REGIONS FOR REFINEMENT

We now describe, how the vortex descriptors are employed to mark nodes in the mesh for refinement. The sensor module (Figure 1) marks nodes in regions defined by the vortex characterization module. The vortices employed to mark regions for refinement may be selected using a variety of criteria such as length, strength, etc. In addition, different regions within the extracted vortex geometry may be selected for refinement as shown in Figure 3.

The region containing the vortex is defined by the volume that is enclosed by the extent surface. The sensor module determines whether or not a node should be marked based on a series of inside–outside tests. To perform these tests, the volume is subdivided into wedges. If a node is

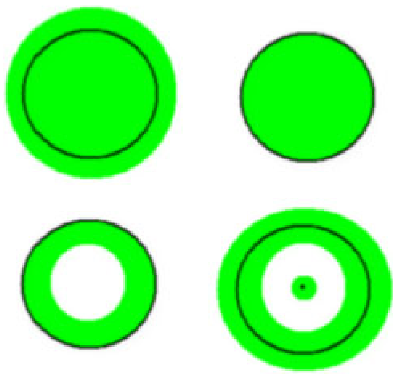

Figure 3. The circle represents vortex extent. The dot in the middle is the core line. The shaded areas show possible regions that can be marked. From left to right, top to bottom: inside extent+near extent, inside extent, inside extent−core, and near extent+near core.

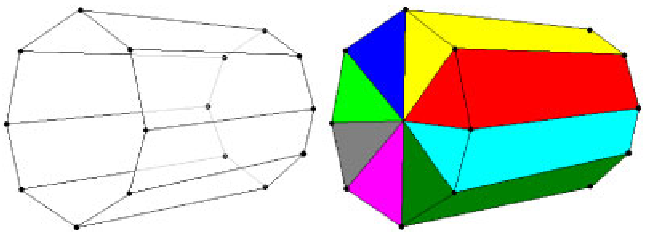

Figure 4. The extent surface is described by lofting between two closed curves on planes perpendicular to the vortex axis (left). This region is decomposed into wedges to determine if a node is inside the extent surface (right).

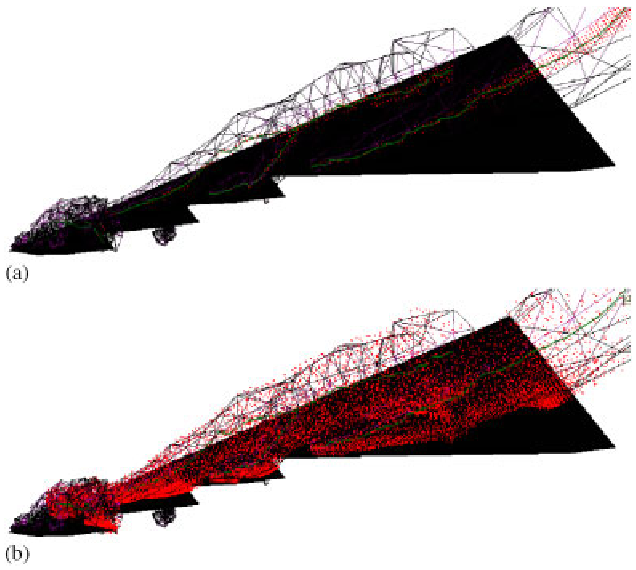

Figure 5. The sensor module can mark nodes near the core line, near the extent surface, and inside the extent surface: (a) nodes near core line marked for refinement and (b) nodes near extent surface marked for refinement.

'inside' each of the five faces of a wedge, it is inside the wedge, and thus, inside the extent surface (Figure 4). As noted above, the points on the extent surface are defined in terms of a distance from the core line measured along a ray extending from the core line. Additional surfaces can be developed by scaling the distance to the extent surface by the desired factor. Figures 5(a) and (b) show nodes marked near the core line and nodes marked inside the extent surface, respectively, for the serrated wing data set [49].

## 5. ADAPTIVE MESH REFINEMENT

We employ the isotropic mesh refinement strategy described in Chalasani *et al.* [9]. In this approach, an *n-sided* face is subdivided into *n* faces by inserting a node at the centroid of the face and connecting it to the mid point of each edge. A new node is then inserted at the centroid of the cell and edges that connect the cell centroid to the face centroids are created. These newly created edges form faces that partition the interior of the cell. Although most of the refined cells are of standard type—prisms, tetrahedrons, and hexahedrons are refined to hexahedrons as shown in Figure 6—cells with four-valence nodes, such as pyramids, are refined to hexahedrons and a non-standard polyhedral cell that contains the four-valence node as shown in Figure 7(a). Further, unrefined cells that neighbor refined cells become non-standard cell types as shown in Figure 7(b).

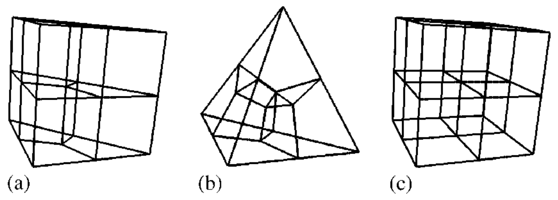

Figure 6. Canonical subdivisions for face-based isotropic refinement: (a) refined prism; (b) refined tetrahedron; and (c) refined hexahedron.

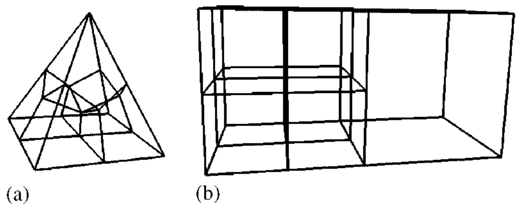

Figure 7. General elements can be created when refinement is applied to a cell with a four-valence node (left) or when a refined cell is a neighbor to an unrefined cell (right): (a) refined pyramid and (b) refined/unrefined cell neighbors.

Here, the cell on the left has been refined into eight hexahedrons while the cell on the right has not been refined. The cell on the right now has nine faces. Although this refinement strategy can be employed with the Loci/CHEM flow solver [10–12], it cannot be employed with all flow solvers. It should be noted that our feature-based approach is not limited to use with the refinement strategy described here and can be used in conjunction with conventional approaches.

To ensure that the volumes of neighboring cells do not vary greatly, which can degrade the accuracy of the flow solver, a cell-balancing strategy [54] is employed to ensure that there is a smooth spatial variation in cell volume. If a neighboring cell is refined more than one additional level beyond the cell under consideration, the current cell is refined until the difference in refinement between the two cells is a single level. The cells shown in Figure 7(b) are said to be balanced because they differ by only one level of refinement. Some care must also be taken during refinement since almost all non-simplex faces are non-planar to some degree, which, in some cases, may result in the creation of invalid cells. To combat this problem, each non-simplex face is decomposed into triangles. A face is marked as folded if the dihedral angle between the normals of two triangles is larger than a specified threshold. In many cases, a face that is marked as folded has a short edge. A minimum length edge criterion, which is typically imposed to help reduce the size of the mesh and, therefore, the computational cost, naturally eliminates many of theses cases. More generally, the constraint that the two elements on both sides of a folded face must be treated identically, either both refined or both not refined, can be imposed to eliminate this problem.

## 6. RESULTS

In this section, we present solution-adaptive mesh refinement results for two configurations, a wing at 10° angle of attack in a wind tunnel [13] and a missile configuration [14] at 0° angle of attack with canards deflected 15° spinning at 30 and 60 Hz. Several different feature-based refinement strategies for the wing case, which focus on refining different regions around the vortex, are evaluated by comparison of predicted results with experimental data [13]. The selected refinement strategy is then applied to the spinning missile configuration.

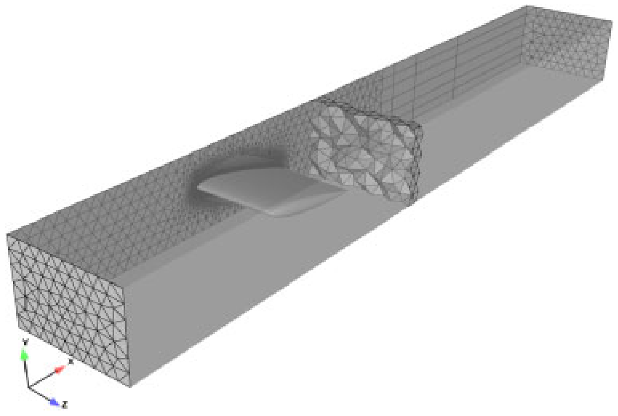

Figure 8. Computational domain employed for wing-wind tunnel computation [13]. The crinkle-cut section illustrates where comparisons are made with experimental data.

### *6.1. Results for wing in wind tunnel*

The first configuration considered here is the low-speed, turbulent flow past a rectangular planform half-wing described by Dacles-Mariani *et al.* [13] who report both experimental data and results from CFD simulations. The half-wing has an aspect ratio of 0.75 (3 ft span and 4 ft chord) and a constant NACA0012 cross section. Although the wing tip is rounded in both the experiment and the simulation, the precise geometry is not known, which does introduce uncertainty in the simulation. The model is mounted in a rectangular cross-section wind tunnel (4 ft wide by 2.667 ft tall) at an angle of 10° with respect to the horizontal. The reference velocity is 170ft/s, which, at the specified conditions, results in a Reynolds number of $4.6\times 10^6$ and a Mach number of 0.1986.

The primary motivation for investigating this problem is to determine the most effective meshing strategy, in terms of solution accuracy, for resolving a wingtip vortex, i.e. a vortex with its axis of rotation parallel to the bulk fluid motion. While other strategies have focused on refinement in the region near the vortex core line [2–4, 6], it is not clear what region of the computational domain needs to be refined in order to capture the vortex downstream of the wing. Our vortex characterization software, described in Section 3, provides us with a tool to investigate this question.

The computational domain employed for our simulations is shown in Figure 8. The inflow and outflow boundaries are located approximately 6.5 ft upstream of the wing leading edge and approximately 22.5 ft downstream of the wing leading edge, respectively. The origin of the coordinate system was located on the inboard tunnel wall, midway between the upper and lower walls. The quarter chord of the wing, about which the wing was rotated to the 10° angle of attack, was located at $x/c=0.25$. Our domain was significantly larger than the one used in the simulations reported by Dacles-Mariani *et al.* [13]. They used experimentally defined inflow and outflow boundary conditions that were not available to us. In lieu of these known boundary conditions, we extended the computational domain and applied a fixed-mass-flow boundary condition at the inflow and a farfield-type (characteristic based) boundary condition at the outflow. Since detailed flow field data were not available to specify the inflow boundary conditions, i.e. the incoming boundary layer, the tunnel walls were modeled using slip boundary conditions, which also reduced the computational cost. The surface meshes on the inboard tunnel wall, the inflow boundary, and the outflow boundary are shown in Figure 8 to provide some indication of the mesh spacing employed in the domain. The highly stretched prismatic mesh, evident in the figure in the downstream third of the domain, was employed to help dissipate the vortex before it reached the downstream boundary. Ironically, adaptive mesh refinement introduced problems at the downstream boundary due to a pressure mismatch between the vortex core and the specified back pressure. We tried several different approaches to solve this problem and this one produced the best results. Shown for reference is a crinkle-cut section located at $x/c=1.4$, which is the location at which comparisons are made with experimental data.

Flow simulations were performed using the Loci/CHEM flow solver [10–12], which was originally developed for simulating non-equilibrium, chemically reacting flow but can also be used to simulate non-reacting flow. Loci/CHEM is an application built using the Loci framework [55] that provides the infrastructure for facilitating intra-application coordination of fine-grained numerical kernals and methods. Loci/CHEM employs an upwind finite-volume algorithm that allows for arbitrary element types.

The turbulence model employed for these simulations is a version of the Spalart–Allmaras model [56] that is modified as suggested by Nichols *et al.* [57]. It is based on the modification to the Baldwin–Barth [58] reported by Dacles-Mariani *et al.* [13]. The production term in both the Baldwin–Barth and Spalart–Allmaras models is given by

$$P = C_1 v R_\tau S, \tag{1}$$

where $C_1$ is a constant, $v$ is the laminar kinematic viscosity, $R_\tau$ is the turbulent Reynolds number, and $S$ is a scalar quantity representative of the rate of deformation tensor. The suggested modification

$$P = C_1 v R_\tau (|\omega| + 2\min(0, |s| - |\omega|)) \tag{2}$$

effectively reduces the eddy viscosity in regions where the vorticity $|\omega|$ exceeds the strain rate, such as occurs in a vortex core, and is negligible in thin-shear layers for which the baseline model has been validated.

Adaptive mesh refinement for this problem is challenging, in part, because the mesh downstream of the wing is quite coarse. The downstream mesh was designed in this manner to test the ability to initiate a feature-based refinement from a relatively poor representation of the solution. In contrast, the mesh on the wing surface, as shown in Figure 9, was relatively fine. This was done to ensure that the flow over the wing tip developed properly and is not considered a serious limitation because it is known *a priori* that the mesh needs to be refined in this region.

We now consider results computed on the baseline mesh. Figures 10(a) and (b) show the mesh and crossflow velocity in a plane located at $x/c = 1.4$. The wing is visible in the lower foreground. The streamwise location of this plane is shown in Figure 8. Note that the vortex is quite weak and diffuse. This is not surprising considering the coarseness of the mesh at this location. Figure 10(c) shows a comparison between the predicted crossflow velocity and the experimental data along four rays emanating from the vortex core at $\theta = 0$, 90, 180, and 270°. The 0° ray is parallel to the span and points in the outboard direction while the 90° ray points vertically. As expected, there is poor agreement between the simulation and experiments.

The discrepancy between the predicted crossflow velocity and the experimental data shown in Figure 10(c) is due to the artificial dissipation that is inherent in the coarse mesh solution. It would be unwise to refine the mesh multiple times using the vortex characteristics extracted from the coarse mesh solution because an excessive number of cells would be refined. One possible strategy would be to refine all elements marked for refinement once, compute a new solution, extract a

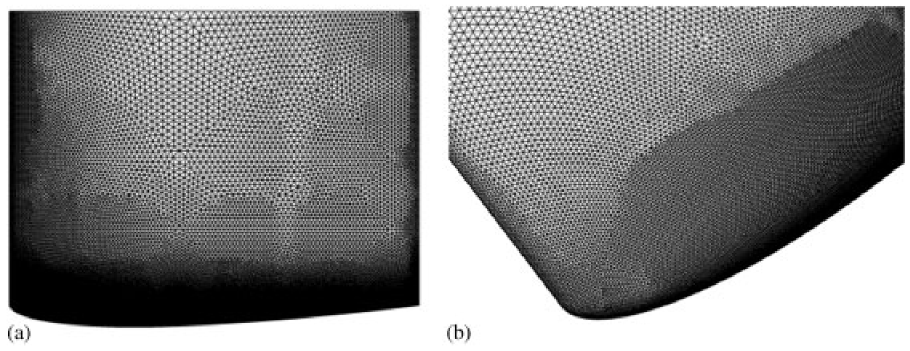

Figure 9. Wing surface mesh: (a) top view and (b) surface mesh near wing tip.

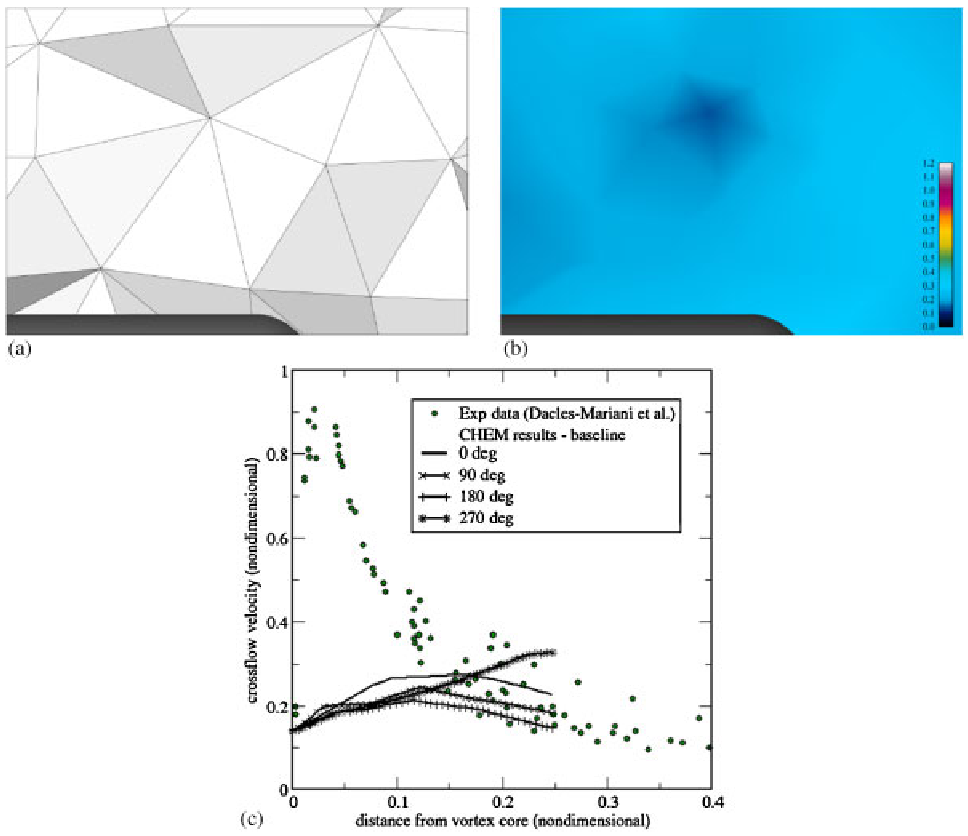

Figure 10. Cutting plane in baseline at $x/c=1.4$ for wing in wind tunnel: (a) baseline mesh; (b) non-dimensional crossflow velocity; and (c) comparison with experimental data.

new set of descriptors, and repeat until some criteria were satisfied. However, this can be very expensive since the numerical solution is computationally expensive.

The compromise strategy we employed was to perform a refinement cycle consisting of two refinements on the region inside of the extracted vortex extent surface. This process was repeated until a reasonable approximation to the vortex was obtained. In this context, vortex quality is based on the diameter of the extent surface. Figures 11(a) and (b) show the mesh in the cutting plane after one and two refinement cycles, respectively. The region containing the vortex is apparent in the mesh refinement. It should be noted that, due to the limitations inherent to the visualization software, the hanging nodes introduced by mesh refinement are not visible in the image. Figure 12 shows the crossflow velocity in the cutting plane. Note that the structure of the vortex is becoming more evident as the mesh is refined. Figure 13 shows comparisons of the predicted crossflow velocities with experimental data. The vertical dashed line shows the approximate region of the mesh that was refined in the current refinement cycle. Note that as the mesh is refined, the region of the mesh that is refined during the next cycle decreases. Also note that the mesh refinement significantly improves the agreement between the predicted crossflow velocity and the experimental data (Figure 13). The vortex extent surfaces computed on the baseline, cycle #1, and cycle #2 meshes are shown in Figure 14. The extent surface diameter decreases with refinement as the artificial dissipation is decreased.

The vortex extent obtained after cycle #2 was deemed to be adequate and this mesh was employed as the initial mesh to investigate four different refinement strategies

1. Refine in the region near the vortex core line with the radius of the refinement being limited to 25% of the local distance to the extent surface. This is denoted as refinement cycle #3(a).

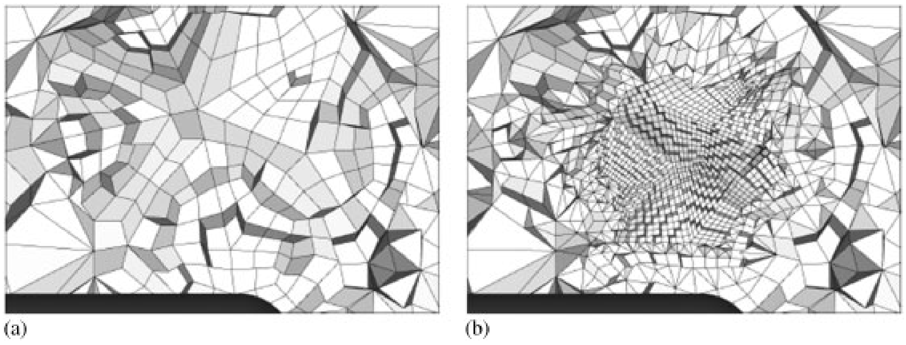

Figure 11. Mesh refinement in cutting plane at $x/c = 1.4$ for wing in wind tunnel: (a) cycle #1 and (b) cycle #2.

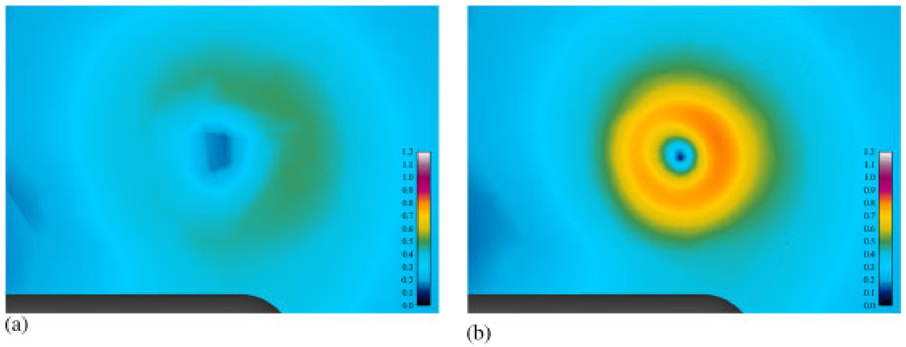

Figure 12. Non-dimensional crossflow velocity in cutting plane at $x/c = 1.4$ for wing in wind tunnel: (a) cycle #1 and (b) cycle #2.

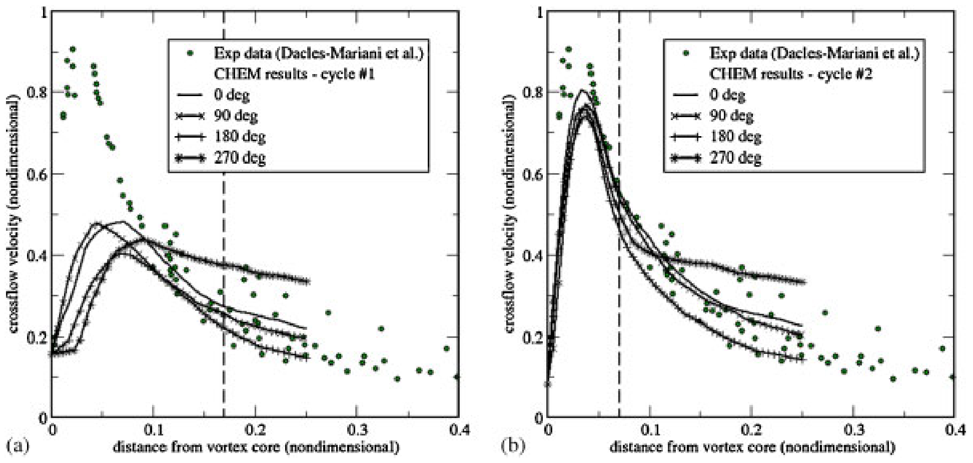

Figure 13. Extracted non-dimensional crossflow velocity along rays emanating from vortex core in plane at $x/c = 1.4$: (a) cycle #1 and (b) cycle #2.

2. Refine in the region near the extent surface with the refinement limited to 25% of the local distance from the core line to the extent surface on either side of the extent surface. This is denoted as refinement cycle #3(b).

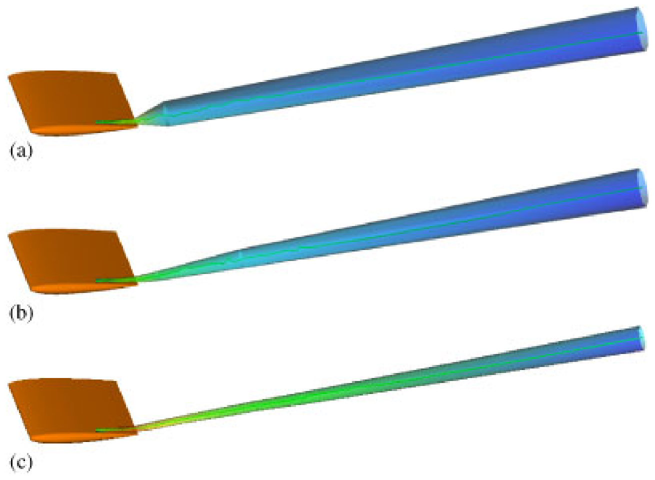

Figure 14. Vortex extent surfaces extracted from flow solutions. Note that the radius of the vortex extent decreases with refinement: (a) wingtip vortex extracted from baseline mesh; (b) wingtip vortex extracted from mesh after first refinement (cycle #1); and (c) wingtip vortex extracted from mesh after second refinement (cycle #2).

3. Refine inside the extent surface. Since the vortex extent naturally contracts as the mesh is refined, this is a fairly conservative strategy that ensures that the vortex is captured within the refined mesh. This is denoted as refinement cycle #3(c).
4. Refine inside the extent surface and a distance outside of the extent surface equal to 25% of the local distance from the core line to the extent surface. This strategy, denoted as refinement cycle #3(d), is equivalent to the refining in a region that is the union of the regions employed in cycles #3(b) and #3(c).

We performed one additional refinement cycle using one of the four approaches described above, this time refining each marked element four times. This strategy is based on the assumption that the region containing the vortex has been identified reasonably well on the cycle #2 mesh. To limit the computational cost, an edge tolerance was specified that ensured that no cell with an edge of length less than 0.205% of the chord was refined.

Figure 15 shows the mesh in the plane located at $x/c = 1.4$. As the mesh is refined beyond cycle #2, the different refinement strategies become evident. For example, in Figure 15(b), which shows refinement near the extent surface, the region near the vortex core is not refined. There is some irregularity in the mesh refinement due to the fact that multiple refinements were performed on an initially tetrahedral mesh. Figure 16 shows the crossflow velocity in the downstream plane. In this case, it is apparent that the strategy that produces the best results is cycle #3(d), i.e. Figure 16(d), in which the mesh is refined inside and just outside the extent surface. A well-defined region of high-tangential velocity surrounds the clearly visible low-velocity core. In contrast, Figure 16(a), in which there is refinement only in the core region, shows only marginally better resolution of the maximum velocity region than cycle #2 (see Figure 12(b)). Also, neither Figure 16(b), in which the mesh is refined near the extent surface only, or Figure 16(c), in which the mesh is refined inside the extent surface, capture the region of maximum crossflow velocity as well as the results shown in Figure 16(d). Also, the vortex core is better resolved for the cases in which there is refinement in the core—Figures 16(a), (c) and (d), —relative to the case for which there is no refinement in the core—Figure 16(b).

Figure 17 shows plots of the tangential velocity versus distance from the vortex core corresponding to the solutions shown in Figure 16. Again, the vertical dashed line shows the approximate region of the mesh that was refined in the current refinement cycle. Note first that the tangential flow outside of the core is relatively unchanged relative to the cycle #2 results (Figure 13(b)). This suggests that the bulk swirling motion is well established by the second refinement cycle.

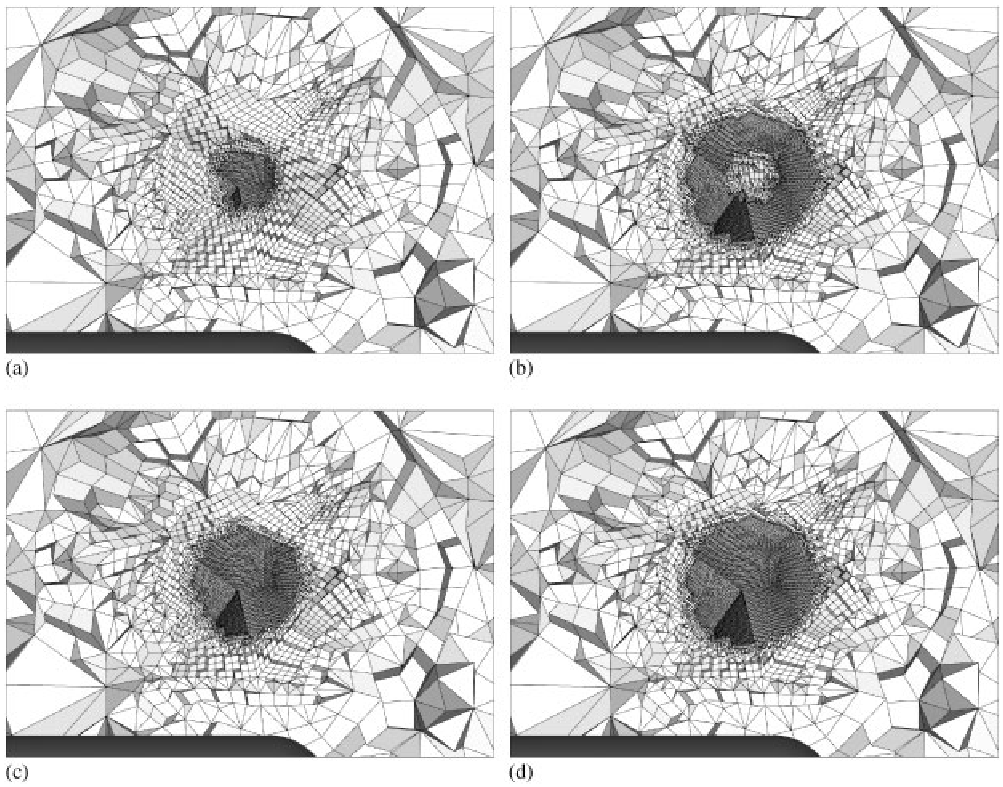

Figure 15. Mesh refinement in cutting plane at $x/c = 1.4$ for wing in wind tunnel: (a) cycle #3(a) core refinement only; (b) cycle #3(b) refinement near extent; (c) cycle #3(c) refinement inside extent; and (d) cycle #3(d) refinement inside and near extent.

In the cases in which the mesh is refined near the coreline, i.e. Figures 17(a), (c) and (d), the velocity variation near the coreline is well predicted while in Figure 17(b), in which there is no refinement in the core, the velocity gradient in the core region is significantly underpredicted. These results indicate that it is necessary to refine in the core region. Figure 17(a) shows the effects of no refinement near the region of maximum curvature in the velocity field and offers no significant improvement relative to Figure 13(b) near the peak velocity. Figure 17(d) shows the best agreement with the experimental data.

Table I shows the non-dimensional location of the vortex core, as defined by the location at which the crossflow velocity is a minimum, for selected refinement cycles and from other simulation results [42, 59]. Here, $\Delta/c$ is the non-dimensional nominal mesh spacing in the region near the vortex core. Ito *et al.* [42] employed an adaptive mesh regeneration strategy that used embedded surfaces to control the point spacing in the interior of the domain while Luke *et al.* [59] used *a priori* mesh refinement in the region near the wing in which it was anticipated that the wingtip vortex would occur. In both cases, the vortex core position was extracted from the simulation data. Note that as the mesh is refined, the vortex core position converges. Additionally, the final location of the vortex computed on the cycle #3(d) mesh compares favorably with the results from the other simulations.

Mesh statistics shown in Table II illustrate how the mesh size increases with refinement. Because this is a volume mesh refinement, there is a significant increase in the number of cells for the cases that are refined near the extent and inside the extent. Many more cells were required relative to the results reported by Dacles-Mariani *et al.* because they employed a higher-order solution method as well as a structured grid on which numerical methods typically exhibit less artificial dissipation.

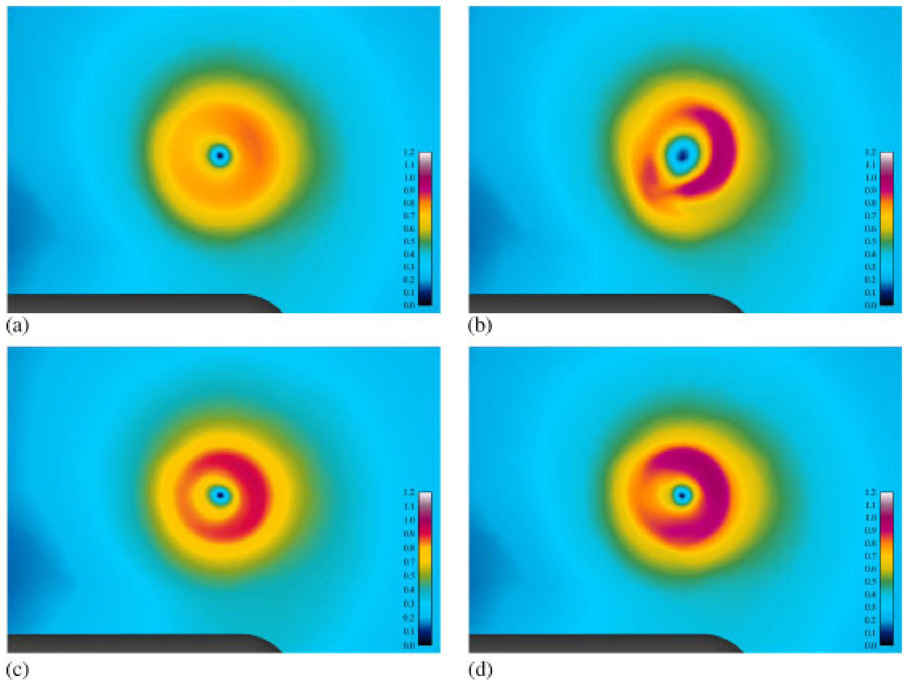

Figure 16. Non-dimensional crossflow velocity in cutting plane at $x/c$=1.4 for wing in wind tunnel: (a) cycle #3(a) core refinement only; (b) cycle #3(b) refinement near extent; (c) cycle #3(c) refinement inside extent; and (d) cycle #3(d) refinement inside and near extent.

In summary, these results demonstrate that refining the mesh both inside and just outside of the extent surface is an effective solution-adaptive mesh refinement strategy for a vortex with its axis of rotation parallel to the primary flow direction. Refinement near the core line or refinement near the extent surface with no additional refinement near the core line was demonstrated to be less effective. Further, we demonstrated that it is possible to initiate the computation from a very coarse downstream mesh.

### 6.2. *Results for spinning missile with canards*

The second configuration under consideration is a missile with canards deflected 15° [14]. The flow is a supersonic with a Mach number of 1.6, turbulent with a Reynolds number of $41.3\times10^6$ based on body length, and the angle of attack is 0°. Two different missile rotation rates, 30 and 60 Hz, are considered. In both cases, the flow is steady in the frame rotating with the missile. In response to the canard deflection (and missile rotation), a pressure difference is formed and a counter-rotating vortex pair is generated from the inboard and outboard tips of each canard. Figures 18(a) and (b) show the extent surfaces extracted from computations performed on the baseline mesh for the 30 and 60 Hz cases, respectively. In these images, vehicle rotation is about the vehicle axis with the right-handed rotation vector pointing toward the nose of the vehicle. As expected, both cases show vortex patterns with significant asymmetry and curvature. In the 60 Hz case, the vortex emanating from the inboard tip of the port canard impinges onto the body of the missile. Although there are other vortices present in the flow, we focus on resolving these four vortices in the results presented below. This case illustrates the complexity of the vortices that can be generated and why it would be difficult to generate a mesh that captures the vortices without prior knowledge of the vortex trajectory. The figures also illustrate one of the shortcomings of our vortex detection algorithm, i.e. the extracted corelines can become irregular as the vortex weakens or the cell size increases (see uppermost vortices in Figures 18(a) and (b)).

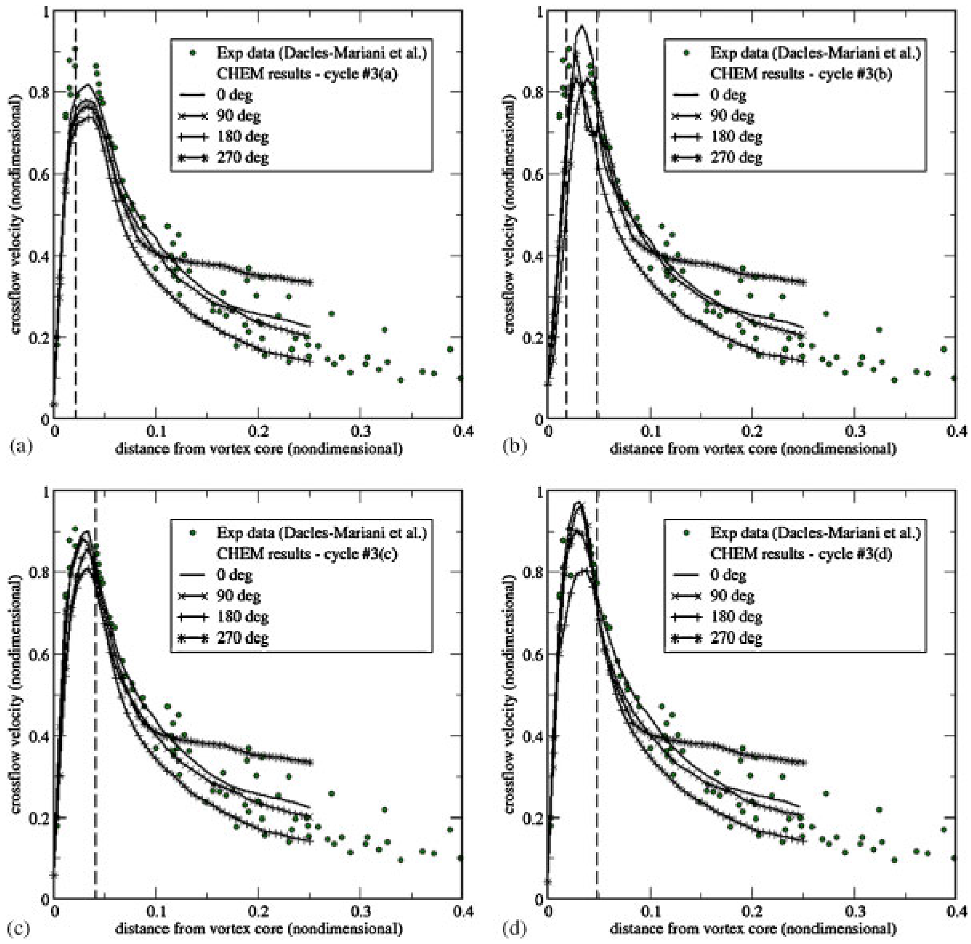

Figure 17. Extracted non-dimensional crossflow velocity for wing in wind tunnel along rays emanating from vortex core in plane at $x/c = 1.4$: (a) cycle #3(a) core refinement only; (b) cycle #3(b) refinement near extent; (c) cycle #3(c) refinement inside extent; and (d) cycle #3(d) refinement inside and near extent.

Table I. Non-dimensional position of vortex core at $x/c = 1.4$.

| Simulation | $y/c$ (vertical) | $z/c$ (spanwise) | $\Delta/c$ |
|---|---|---|---|
| Baseline | −0.0738 | 0.6609 | 0.064 |
| Cycle #1 | −0.0467 | 0.6729 | 0.016 |
| Cycle #2 | −0.0654 | 0.6670 | 0.004 |
| Cycle #3(d) | −0.0634 | 0.6677 | 0.001 |
| Ito *et al.* [42] | −0.0622 | 0.6654 | 0.0018 |
| Luke *et al.* [59] | −0.0620 | 0.6610 | 0.002 |

Based on the results obtained for the wingtip vortex, the mesh was refined inside and near the extent surface. It should be noted that the baseline mesh was already somewhat refined in the region surrounding the missile and can be considered to be the equivalent of the cycle #2 mesh for the wing case. Again, to limit computational cost, the mesh was refined three levels with an edge length tolerance of 0.1085% of the body length. Table III shows the mesh statistics for the baseline mesh, which was used for both spinning cases, and the two refined meshes. The only significant difference between the two refined meshes is that one of the vortices impinges on the body and terminates in the 60 Hz case; therefore, somewhat fewer nodes are marked for refinement in this case.

Table II. Statistics of meshes for wing in wind tunnel.

| | # Nodes | # Faces | # Cells |
|---|---|---|---|
| Baseline | 6 130 629 | 33 424 764 | 13 831 237 |
| Cycle #1 | 6 237 640 | 33 690 109 | 13 912 495 |
| Cycle #2 | 6 741 269 | 35 106 020 | 14 370 064 |
| Cycle #3(a) (near core line) | 9 288 182 | 42 292 565 | 16 694 027 |
| Cycle #3(b) (near extent) | 15 662 728 | 60 612 441 | 22 671 680 |
| Cycle #3(c) (inside extent) | 14 050 903 | 56 218 050 | 21 279 807 |
| Cycle #3(d) | 17 854 792 | 67 374 656 | 24 959 518 |

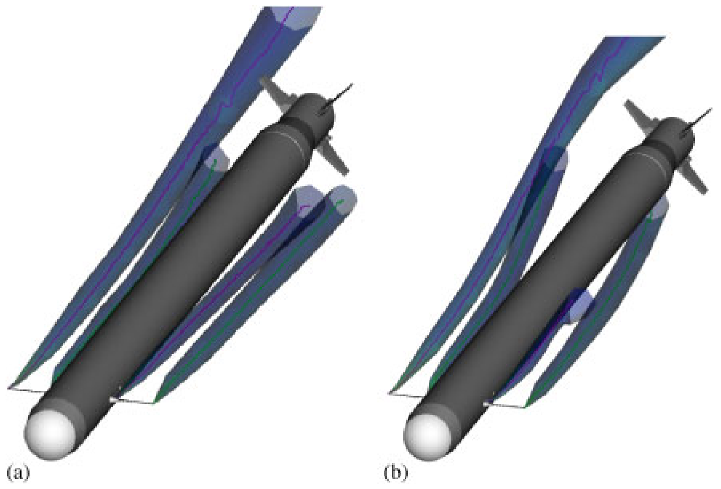

Figure 18. Extent surfaces extracted from baseline mesh solutions for missile spinning at 30 and 60 Hz: (a) baseline mesh 30 Hz and (b) baseline mesh 60 Hz.

Table III. Mesh statistics for non-spinning and spinning missile.

| | # Nodes | # Faces | # Cells |
|---|---|---|---|
| Baseline | 6 143 000 | 47 724 225 | 21 984 916 |
| Refined—cycle #1 (spinning—30 Hz) | 16 222 586 | 75 526 411 | 30 921 952 |
| Refined—cycle #1 (spinning—60 Hz) | 15 054 869 | 72 329 706 | 29 896 644 |

Figures 19 and 20 show comparisons between contours of stagnation pressure computed on the baseline mesh and the refined mesh for the missile spinning at 30 and 60 Hz, respectively. The vortices on the refined mesh show much better definition over the length of the missile than on the baseline mesh for both the 30 and 60 Hz rotation rates. Additional refinement was performed for both cases and no significant improvements were observed. As shown in Figure 19, the resolution of the vortices is markedly improved for the 30 Hz case on the refined mesh. The vortex cores are much better resolved for the two outboard vortices. Less improvement is observed in the two inboard vortices. This can be attributed to enhanced mesh resolution in this region in the baseline mesh. Significant asymmetry is present in the flow field for the 60 Hz rotation rate as shown in Figure 20. In addition to the inboard port vortex impinging on the missile body, the outboard port vortex is 'wrapping around' the underside of the missile and the two starboard vortices are

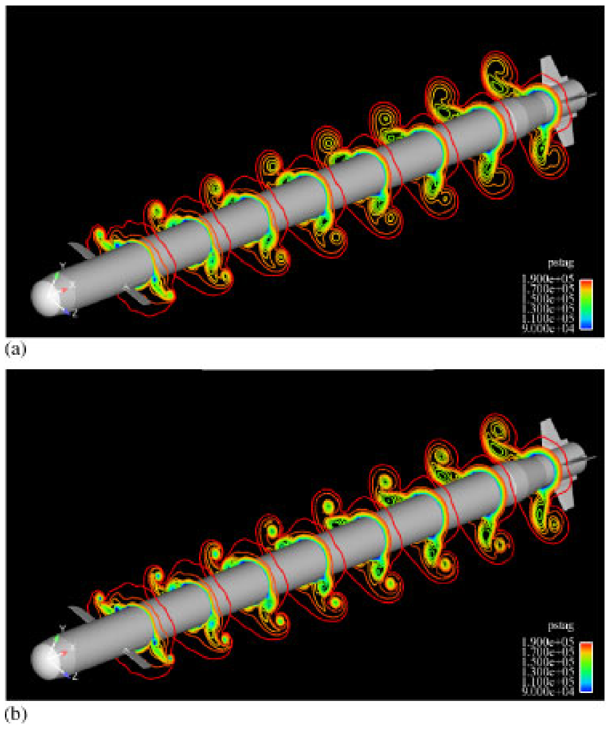

Figure 19. Stagnation pressure contours computed for missile spinning at 30 Hz. Note that the vortices predicted on the baseline mesh are diffused much more rapidly in comparison to those computed on the refined mesh: (a) baseline mesh and (b) refined mesh (cycle 1).

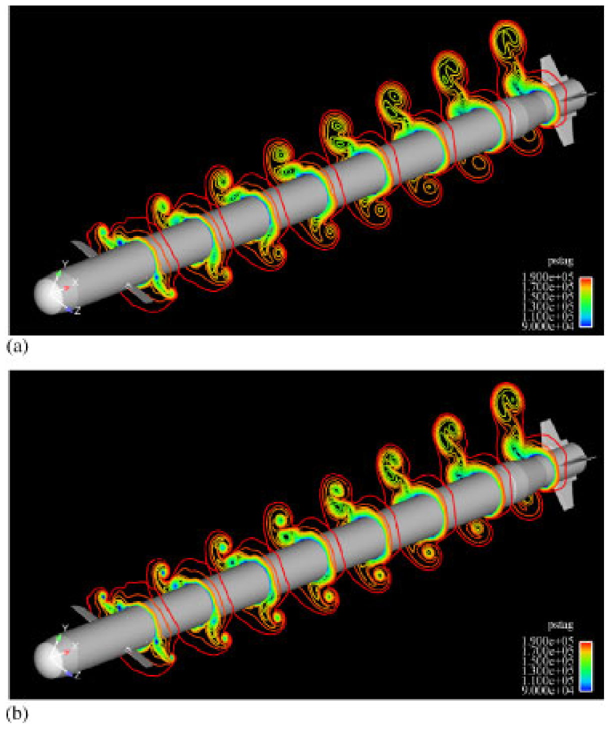

Figure 20. Stagnation pressure contours computed for missile spinning at 60 Hz. Again, the vortices are more persistent on the refined mesh than on the baseline mesh: (a) baseline mesh and (b) refined mesh (cycle 1).

interacting with one another and are in the process of merging. There is significant improvement in the resolution of the outboard port vortex over the length of the missile. Improvement in resolution for the starboard vortices is noticeable over the first three-quarters of the body length.

## 7. CONCLUSION

We have demonstrated the effectiveness of vortex-based adaptive mesh refinement for two different configurations: a wing at 10° angle of attack in a wind tunnel and a missile at 0° angle of attack with canards deflected 15° for two spinning cases—30 and 60 Hz. An existing vortex characterization algorithm [7] was employed as an enabling technology for solution-adaptive mesh refinement by using high-level vortex descriptors to mark nodes in regions of the computational domain where a vortex is located. We evaluated several different mesh refinement strategies by comparison of results computed for the wing with experimental data and determined that the most effective strategy was to refine the region inside and just outside of the extent surface. Results were significantly improved relative to results obtained on a coarse baseline mesh for the wing case. For the missile cases, a baseline mesh that had already been refined in a region surrounding the body was employed. The results were significantly improved based on an observed enhanced coherence in the vortex cores over the length of the vortex relative to the baseline mesh. The rotation-induced asymmetry in the vortex patterns illustrates the need to be able to refine the mesh based on the solution rather than on an anticipated vortex position.

ACKNOWLEDGEMENTS

The authors to thank NSF (ITR/NGS-0326386), NASA, and DoD for supporting this research. The authors also to thank the reviewers for their helpful comments and suggestions.

## REFERENCES

1. Knupp P. Remarks on mesh quality. *AIAA 46th Aerospace Sciences Meeting and Exhibition*, Reno, NV, 2008. *Paper 2008-0933*.
2. Dindar M, Lemnios A, Shephard M, Jansen K, Kenwright D. Effect of tip vortex resolution on UH-60A rotor-blade hover performance calculations. *Proceedings of the 54th AHS International Annual Forum and Technology Display* Washington, DC, vol. 1, 1998; 45–57.
3. Kenwright D, Haimes R. Automatic vortex core detection. *IEEE Computer Graphics and Applications* 1998; **18**(4):70–74.
4. Murayama M, Nakahashi K, Sawada K. Simulation of vortex breakdown using adaptive grid refinement with vortex-center identification. *AIAA Journal* 2001; **39**(7):1305–1312.
5. Kim SE, Rhee S. Toward high-fidelity prediction of tip-vortex around lifting surfaces—what does it take? *Proceedings of the 25th Symposium on Naval Hydrodynamics*, St. Johns, NL, Canada, 2004; 62–70.
6. Turnock S, Pashias C, Rogers E. Flow feature identification for capture of propeller tip vortex evolution. *Proceedings of the 26th Symposium on Naval Hydrodynamics*, Rome, Italy, 2006; 223–240.
7. Jankun-Kelly M, Jiang M, Thompson D, Machiraju R. Vortex visualization for practical engineering applications. *IEEE Transactions on Visualization and Computer Graphics* 2006; **12**(5):957–964.
8. Garth C, Tricoche X, Salzbrunn T, Scheuermann G. Surface techniques for vortex visualization. *Proceedings of Eurographics/IEEE VGTC Symposium on Visualization*, Konstanz, Germany, 2004; 155–164.
9. Chalasani S, Senguttuvan V, Thompson D, Luke E. On the use of general elements in fluid dynamics simulations. *Communications in Numerical Methods in Engineering* 2008; **24**(6):435–448.
10. Luke E, Tang L, Tong X-L, Wu J, Cinnella P. A step towards 'shape-shifting' algorithms: Reactive flow simulations using generalized grids. *AIAA 39th Aerospace Sciences Meeting and Exhibition*, Reno, NV, 2001. *Paper 2001-0745*.
11. Luke E. On robust and accurate arbitrary polytope cfd solvers (invited). *AIAA 18th Computational Fluid Dynamics Conference*, Miami, FL, 2007. *Paper 2007-3956*.
12. Luke E, Cinnella P. Numerical simulations of mixtures of fluids using upwind algorithms. *Computers and Fluids* 2007; **36**(10):1547–1566.
13. Dacles-Mariani J, Zilliac G, Chow J, Bradshaw P. Numerical/experimental study of a wingtip vortex in the near field. *AIAA Journal* 1995; **33**(9):1561–1568.
14. Blades E, Marcum D. Numerical simulation of a spinning missile with dithering canards using unstructured grids. *Journal of Spacecraft and Rockets* 2004; **41**(2):248–256.

15. Ainsworth M, Oden J. *A Posteriori Error Estimation in Finite Element Analysis*. Wiley: Hoboken, NJ, 2000.
16. Mavriplis D. Adaptive meshing techniques for viscous flow calculations on mixed element unstructured meshes. *International Journal for Numerical Methods in Engineering* 2000; **34**(2):93–111.
17. Cavallo P, Sinha M, Feldman G. Parallel unstructured mesh adaptation method for moving body applications. *AIAA Journal* 2005; **43**(9):1937–1945.
18. Peraire J, Vahdati M, Morgan K, Zienkiewicz O. Adaptive remeshing for compressible flow problems. *Journal of Computational Physics* 1987; **72**(2):449–466.
19. Anderson D. Adaptive grid methods for partial differential equations. *Advances in Grid Generation*; *ASME Applied Mechanics*, *Bioengineering*, *and Fluids Engineering Conference*, Houston, TX, 1983; 1–15.
20. Dwyer H. Grid adaptation for problems in fluid dynamics. *AIAA Journal* 1983; **22**(12):1705–1712.
21. Pirzadeh S. An adaptive unstructured grid method by grid subdivision, local remeshing, and grid movement. *AIAA 14th Computational Fluid Dynamics Conference*, Norfolk, VA, 1999. *Paper 1999-3255*.
22. Hunt J. Vorticity and vortex dynamics in complex turbulent flows. *Transactions of the Canadian Society for Mechanical Engineering* 1987; **11**(1):21–35.
23. Perry A, Chong M. A description of eddying motions and flow patterns using critical point concepts. *Annual Review of Fluid Mechanics* 1987; **19**:125–155.
24. Levy Y, Degani D, Seginer A. Graphical visualization of vortical flows by means of helicity. *AIAA Journal* 1990; **28**(8):1347–1352.
25. Berdahl C, Thompson D. Eduction of swirling structure using the velocity gradient tensor. *AIAA Journal* 1993; **31**(1):97–103.
26. Jeong J, Hussain F. On the identification of a vortex. *Journal of Fluid Mechanics* 1995; **285**:69–94.
27. Strawn R, Kenwright D, Ahmad J. Computer visualization of vortex wake systems. *AIAA Journal* 1999; **37**(4):511–512.
28. Sujudi D, Haimes R. Identification of swirling flow in 3d vector fields. *AIAA 12th Computational Fluid Dynamics Conference*, San Diego, CA, 1995. *Paper 1995-1715*.
29. Roth M, Peikert R. A higher-order method for finding vortex core lines. *Proceedings of IEEE Visualization '98*, Research Triangle Park, NC, 1998; 143–150.
30. Peikert R, Roth M. The 'parallel vectors' operator—a vector field visualization primitive. *Proceedings of IEEE Visualization '99*, San Francisco, CA, 1999; 263–270.
31. Reinders F, Sadarjoen A, Vrolijk B, Post F. Vortex tracking and visualization in a flow past a tapered cylinder. *Computer Graphics Forum* 2002; **21**(4):675–682.
32. Weinkauf T, Sahner J, Theisel H, Hege HC. Cores of swirling particle motion in unsteady flows. *IEEE Transactions on Visualization and Computer Graphics* 2007; **13**(6):1759–1766.
33. Jiang M, Machiraju R, Thompson D. A novel approach to vortex core region detection. *Proceedings of Eurographics/IEEE VGTC Symposium on Visualization*, Barcelona, Spain, 2002; 217–225.
34. Sahner J, Weinkauf T, Hege H-C. Galilean invariant extraction and iconic representation of vortex core lines. *Proceedings of the Joint Eurographics–IEEE VGTC Symposium on Visualization*, Leeds, U.K., 2005; 151–160.
35. Stegmaier S, Rist U, Ertl T. Opening the can of worms: an exploration tool for vortical flows. *Proceedings of IEEE Visualization '05*, Minneapolis, MN, 2005; 463–470.
36. Sahner J, Weinkauf T, Teuber N, Hege H-C. Vortex and strain skeletons in eulerian and lagrangian frames. *IEEE Transactions on Visualizations and Computer Graphics* 2007; **13**(5):980–990.
37. Banks D, Singer B. A predictor-corrector technique for visualizing unsteady flow. *IEEE Transactions on Visualization and Computer Graphics* 1995; **1**(2):151–163.
38. Tricoche X, Garth C, Kindlmann G, Deines E, Scheuermann G, Ruetten M, Hansen C. Visualization of intricate flow structures for vortex breakdown analysis. *Proceedings of IEEE Visualization '04*, Minneapolis, MN, 2004; 187–194.
39. Burger R, Muigg P, Ilcik M, Doleisch H, Hauser H. Integrating local feature detectors in the interactive visual analysis of flow simulation data. *Proceedings of Eurographics/IEEE VGTC Symposium on Visualization*, Norrkoping, Sweden, 2007; 171–178.
40. Sawada K. A convenient visualization method for identifying vortex centers. *Transactions of the Japan Society of Aeronautical and Space Sciences* 1995; **38**(120):102–116.
41. Pemberton R, Turnock S, Dodd T, Rogers E. A novel method for identifying vortical structures. *Journal of Fluids and Structures* 2002; **16**(8):1051–1057.
42. Ito Y, Shih A, Koomullil R, Kasmai N, Jankun-Kelly M, Thompson D. Solution adaptive mesh generation using feature-aligned embedded surface meshes. *AIAA Journal* 2009; **47**(8):1879–1888.
43. Giles M. On adjoint equations for error analysis and optimal grid adaptation in cfd. *Technical Report 97/11*, Oxford University Computing Laboratory, Numerical Analysis Group, Oxford, England, 1997.
44. Venditti D, Darmofal D. Grid adaptation for functional outputs: application to two-dimensional inviscid flow. *Journal of Computational Physics* 2002; **176**(1):40–69.
45. Barth T. *Numerical Methods and Error Estimation for Conservation Laws on Structured and Unstructured Meshes*. Lecture Notes, Series: 2003–04. von Karman Institute for Fluid Dynamics, 2003.
46. Balasubramanian R, Newmann III J. Comparison of adjoint-based and feature-based grid adaptation for functional outputs. *International Journal for Numerical Methods in Fluids* 2007; **53**(10):1541–1569.
47. Nemec M, Aftosmis M, Wintzer M. Adjoint-based adaptive mesh refinement for complex geometries. *AIAA 46th Aerospace Sciences Meeting and Exhibition*, Reno, NV, 2008. *Paper 2008-0725*.

48. Fidkowski K, Roe P. Entropy-based mesh refinement, I: the entropy adjoint approach. *AIAA 19th Computational Fluid Dynamics Conference*, San Antonio, TX, 2009. *Paper 2009-3790*.
49. Hammons C, Thompson D. A numerical investigation of novel planforms for micro UAVs. *AIAA 44th Aerospace Sciences Meeting and Exhibition*, Reno, NV, 2006. *Paper 2006-1265*.
50. MacQueen J. Some methods for classification and analysis of multivariate observations. *Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, CA, vol. 1, 1967; 281–296.
51. Kanungo T, Mount D, Netanyahu N, Piatko C, Silverman R, Wu A. An efficient k-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2002; **24**(7): 881–892.
52. Lugt H. *Vortex Flow in Nature and Technology*. Wiley: Hoboken, NJ, 1972.
53. Jiang M. A feature-based approach to visualizing and mining simulation data. *Ph.D. Thesis*, The Ohio State University, 2005.
54. Aftosmis M, Berger M, Melton J. Melton and surface modeling for cartesian mesh methods. *AIAA 12th Computational Fluid Dynamics Conference*, Sand Diego, CA, 1995. *Paper 1995-1725*.
55. Luke E, George T. Loci: A rule-based framework for parallel multidisciplinary simulation synthesis. *Journal of Functional Programming*, *Special Issue on Functional Approaches to High-Performance Parallel Programming* 2005; **15**(3):477–502.
56. Spalart P, Allmaras S. A one-equation model for aerodynamic flows. *La Recherche Aerospatiale* 2005; **1**5–21.
57. Nichols R, Tramel R, Buning P. Evaluation of two high order weno schemes. *AIAA 25th Applied Aerodynamics Conference*, Miami, FL, 2007. *Paper 2007-3920*.
58. Baldwin B, Barth T. A one-equation transport model for high Reynolds number wall-bounded flows. *Technical Report TM 102857*, NASA, 1990.
59. Luke E, Hebert S, Thompson D. Theoretical and practical evaluation of solver-specific mesh quality (invited). *AIAA 46th Aerospace Sciences Meeting and Exhibition*, Reno, NV, 2008. *Paper 2008-0934*.