Geometry Modeling for Unstructured Mesh Adaptation

Michael A. Park,* Bil Kleb,† and William T. Jones‡
NASA Langley Research Center, Hampton, VA 23681, USA

Joshua A. Krakos§ and Todd Michal¶
The Boeing Company, St. Louis, MO, 63166, USA

Adrien Loseille‖
INRIA Paris-Saclay, Alan Turing Building, 91120 Palaiseau, France

Robert Haimes**
Massachusetts Institute of Technology, Cambridge, MA, 02139, USA

John F. Dannenhoffer, III††
Syracuse University, Syracuse, NY, 13244, USA

The quantification and control of discretization error is critical to obtaining reliable simulation results. Adaptive mesh techniques have the potential to automate discretization error control, but have made limited impact on production analysis workflow. Recent progress has matured a number of independent implementations of flow solvers, error estimation methods, and anisotropic mesh adaptation mechanics. However, the poor integration of initial mesh generation and adaptive mesh mechanics to typical sources of geometry has hindered adoption of adaptive mesh techniques, where these geometries are often created in Mechanical Computer-Aided Design (MCAD) systems. The difficulty of this coupling is compounded by two factors: the inherent complexity of the model (e.g., large range of scales, bodies in proximity, details not required for analysis) and unintended geometry construction artifacts (e.g., translation, uneven parameterization, degeneracy, self-intersection, sliver faces, gaps, large tolerances between topological elements, local high curvature to enforce continuity). Manual preparation of geometry is commonly employed to enable fixed-grid and adaptive-grid workflows by reducing the severity and negative impacts of these construction artifacts, but manual process interaction inhibits workflow automation. Techniques to permit the use of complex geometry models and reduce the impact of geometry construction artifacts on unstructured grid workflows are presented. Two complex MCAD models from the AIAA Sonic Boom and High Lift Prediction Workshop are shown to demonstrate the utility of the current approach.

I. Introduction

The use of Reynolds-averaged Navier–Stokes (RANS) equations with a turbulence model has become a critical tool for the design of aerospace vehicles. However, the RANS-based Computational Fluid Dynamics (CFD) analysis and design process has not reached the level of automation desired by practitioners. Alauzet and Loseille [1] documented the dramatic progress made in the last decade for solution-adaptive methods that includes the anisotropy to resolve simulations with shocks and boundary layers, and they identify where continued investment is necessary for complex simulations. Park et al. [2] documented the state of solution-based anisotropic mesh adaptation and motivated further development with the impacts that improved capability would have on aerospace analysis and design in a context broader

---

*Research Scientist, Computational AeroSciences Branch, AIAA Associate Fellow.

†NASA Senior Researcher for Computational Aerothermodynamics, AIAA Associate Fellow.

‡Computer Engineer, Computational Aerosciences Branch, AIAA Associate Fellow.

§Engineer, AIAA Senior Member.

¶Technical Fellow, AIAA Senior Member.

‖Researcher, GAMMA3 Team, AIAA Member.

**Principal Research Engineer, Aerospace Computational Design Laboratory, Department of Aeronautics and Astronautics, AIAA Member.

††Associate Professor, Aerospace Computational Methods Laboratory, Mechanical and Aerospace Engineering, AIAA Associate Fellow.

than that articulated in the CFD Vision 2030 Study by Slotnick et al. [3]. The Vision Study provides a number of case studies to illustrate the current state of CFD capability and capacity. The Study identifies mesh generation and adaptivity as continuing bottlenecks in the CFD workflow.

The components of unstructured mesh adaptation are shown on Fig. 1. Starting with an initial mesh, a flow solution is computed. The information from the flow solution are used to estimate error and specify a new mesh resolution and orientation request (metric field). If the estimated errors (based on an objective function) are larger than limits specified by the practitioner, the current mesh system is modified by mesh mechanics to adhere to the mesh resolution request. Once the adapted mesh is available, the previous flow solution is interpolated to the new mesh to provide an initial condition for the flow solver that approximates the converged solution. This improved initial condition may decrease the execution time and improve the robustness of the flow solution calculation. The process is repeated until exit criteria are met (e.g., accuracy requirement, resource limit). This process interacts with the geometry at two stages: the initial mesh generation and adaptive mesh mechanics.
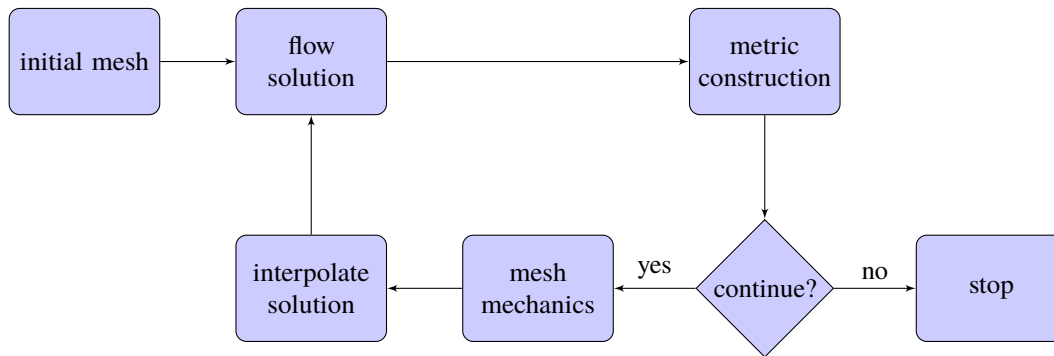
Fig. 1   Solution-based mesh adaptation process.

An informal Unstructured Grid Adaptation Working Group (UGAWG) has been formed to mature mesh adaptation technology as described in their first benchmark [4], which focused on evaluating adaptive mesh mechanics for analytic metric fields on planar and simple curved domains. The UGAWG verified the adaptive process by documenting design-order convergence rates of scalar function interpolation error and accurate results for a simple wing configuration [5, 6]. The first benchmark contains a list of future directions, which includes the focus of this paper: Mechanical Computer-Aided Design (MCAD) integration. These previous UGAWG publications restricted models to simple geometry to allow a focus on developing mesh mechanics and documenting metric conformity (how well the adapted grid satisfies the metric size request). The current effort details the complexities encountered in supporting complex, realistic geometry encountered in typical use cases.

Taylor and Haimes [7] provide an overview of geometry modeling concepts and requirements for computational simulation, which was motivated by the First AIAA Geometry and Mesh Generation Workshop (GMGW-1) [8]. Here we focus on Boundary REPresentation (BREP) geometry modeling techniques. While there are alternative techniques, Taylor and Haimes share this focus on BREP because it underpins the geometry modeling kernels embedded in all industrial MCAD platforms. Details of the BREP can have direct impacts on the reliability, robustness, and repeatability of rapid mesh generation processes as reviewed by Gammon, Bucklow, and Fairey [9]. Diagrams of the relationship between topology (face, edge, and node) and geometry (surface, curve, and point) in a BREP solid are provided in [9] and [7]. The elements of a discrete mesh are referred to as triangle, segment, and vertex.

topology	geometry	discrete	
node	point	vertex	·
edge	curve	segment	−
face	surface	triangle	△

GMGW-1 highlighted a number of features of the High Lift Common Research Model geometry description [10] that complicate the interaction of the BREP with discrete mesh generation and adaptation. Some of these features were inherited from the cruise version of the Common Research Model used in the Drag Prediction Workshops [11]. Ideally, a BREP should be free of these complications, but unfortunately they are quite common. These problematic BREP artifacts are often produced or exacerbated by typical geometry handling processes for aerodynamics as described by Taylor [12]. Examples of these issues are small gaps, missing faces, duplicate surfaces, overlapping surfaces, self-intersecting surfaces, inadequate tolerances, discontinuous parameterization, degeneracy, small edges, sliver faces,

narrow regions, cusps, incorrect edge orientation, intersecting edges, voids, excessive detail, or inconvenient topology [9, 13]. Taylor [13] made the following observations based on a survey of GMGW-1 participants,

> Of the various actions that were required to "repair" the IGES or STEP models (which included filling small gaps at trailing edges, replacing missing faces, removing duplicate surfaces, and addressing overlapping surface patches and/or self-intersecting faces), no two responses were the same. In other words, each respondent appeared to be faced with a slightly different set of problems to resolve. . . . Some reported joining short edges and/or splitting surface patches possessing high local curvature along iso-parametric lines in order to aid the subsequent meshing. Again, a diverse range of actions were reported, with no evidence of a consistent approach being adopted by any two respondents.

The lack of consistent "repair" mechanics may simply be a measure of the geometry tools available to each group.

A follow-on survey of commercial mesh generation software users* indicated the same pattern of eliminating small gaps and unnecessary features with improvements to tolerances and alignment of surfaces. IGES and STEP are the most common geometry file formats. Propriety, native MCAD formats were used by a minority of the respondents. These observations indicate that the vast majority of existing mesh generation processes require manual intervention at the geometry import and processing phase to enable the success of subsequent meshing operations.

While some classes of manual geometry repair may be necessary, the goal is to harden the initial generation and adaptive mesh procedure for many common issues. In some cases, robustness to geometry issues may come with a reduction in efficiency (i.e., additional refinement required by geometry topology, not required by the error estimate). A longer execution time may be preferred over mandatory manual intervention, especially when the problem is detected downstream in a High Performance Computing (HPC) environment during adaptation. In the cases where efficiency loss is detected or manual intervention is required, the location and type of problem should be reported to provide the practitioner the ability to quickly triage, and optionally repair. Repairs can be made directly to the geometry model or an abstraction layer can be created to map the existing BREP topology to a more desirable topology for meshing.

Improvements to face and edge topology can reduce constraints on mesh generation [14] to create elements with improved shapes and sizes consistent with the spacing request. Virtual topology operators [15] or surface quilts and edge chains [16] collect or split [17] the low level topological faces and edges into virtual faces and edges. A quilt can have a global parameterization [18–20], but constructing a well-behaved (continuous and orthogonal) parameterization can be difficult. Quilts have the potential for preventing topology changes to virtual topology during model deformation [17, 21], but only to a certain point.

Virtual BREP topology modification is a powerful tool to provide an improved geometry model to fixed-mesh and adapted-mesh workflows, but these improvements remain locked in the system that defined the improved topology. An open, geometry kernel neutral, schema to persist mesh-geometry association, including virtual topology and attribution, is under development, see Karman and Wyman [22]. This schema includes the discrete mesh association to the topology and parameters of the BREP. This association can be reconstructed during mesh adaptation, but this recovery process is error prone. The information describing this association is typically available during initial mesh generation and can be persisted to eliminate BREP association error. The mesh generation and adaptation community has not adopted a standard, but defining a common standard to persist virtual BREP topology and the association of a discrete mesh vertices, segments, and triangles to BREP nodes, edges, and faces would benefit many parties in the mesh generation and adaptation process.

Many surface mesh generation packages perform internal BREP modifications to increase robustness and improve element shape. These changes are undocumented and a BREP that matches the resulting surface mesh is not provided. Adopting a standard to persist virtual topology and mesh-geometry association would allow for communication of these BREP changes and allow more initial mesh generators to support adaptive meshing. An example of discrete mesh to BREP association persistence via mesh file records adopted by the UGAWG is described in Section II.B to illustrate the information required by adaptive meshing techniques. An alternative BREP surrogate geometry that can implicitly account for virtual topology is described in Section II.C.

Both Taylor and Haimes [7] and Gammon, Bucklow, and Fairey [9] discuss the fit between topological entities or model tolerancing. Discrete mesh tools must be aware of this property because a BREP is not closed at machine precision. Adaptive meshes generated on MCAD geometry for the Sixth AIAA Drag Prediction, Third AIAA High Lift Prediction and GMGW-1 workshops by Michal, et al. [23–25] highlighted that the acceptable fit between geometry surfaces (model tolerance) is often much tighter for adaptive meshes compared to fixed-mesh applications. As local

---

*Survey Results – Mesh Generation and CAD Interoperability `https://blog.pointwise.com/2018/06/13/survey-results-mesh-generation-and-cad-interoperability/` [retrieved 18-MAR-2019].

mesh resolution approaches the resolution of model tolerance between geometry surfaces, large angles or folds can develop in the surface mesh. The required level of model tolerance is therefore directly driven by the local mesh resolution requirements.

For fixed-mesh processes, the geometry is only used during the preprocessing phase of mesh generation. Before the analysis begins, geometry tolerance issues can be identified and corrected (as the adaptation mechanics require) based on the mesh resolution choices the user controls. With adaptive meshing, geometry is queried throughout the solution process. Geometry requirements become coupled to the flow solution and mesh resolution specified by the error estimate, both of which are constantly changing throughout the analysis. Assessing the adequacy of a geometry model before mesh adaptation becomes as challenging as predicting future mesh requirements. In addition, the surface resolution of adapted meshes is generally much finer than fixed meshes due to the interaction between highly anisotropic sizing metrics and geometry curvature as shown later in Section V.B. The mesh adaptation process must be tolerant of "loose" model tolerances, because rarely are the tolerances in the geometry definition tight enough to satisfy the smallest mesh size that the adaptive mesh may request.

The interaction of BREP and initial/adaptive mesh generation is examined for a few classes of common intended and unintended geometry model artifacts (e.g., high curvature, small feature size, degeneracy). Methodologies are described to evaluate BREP geometry models or construct and evaluate a discrete surrogate of the geometry model. An approach to persist the association of the discrete mesh and the BREP is detailed. The combination of BREP evaluation (with intended and unintended artifacts), discrete mesh to BREP association persistence, and interpolation error control allow for CFD solutions to be generated from MCAD models without tedious geometry preparation and manual mesh sizing specification. Examples are provided from two AIAA prediction workshops: The NASA Concept 25D with Flow-Through Nacelle (C25F) used in the Second AIAA Sonic Boom Prediction Workshop (SBPW-2) [26] and the Japan Aerospace Exploration Agency Standard Model (JSM) with nacelle used as Case 2d of the Third AIAA High Lift Prediction Workshop (HiLiftPW-3) [27].

II. Mesh Mechanics

The interaction of BREP geometry with the mesh mechanics step of the solution-based mesh adaptation process (Fig. 1) is the focus of the method description. Three tools are described, where each is designed to output a unit mesh [28] in a provided metric field from an input mesh that conforms to the geometry description. After a brief introduction, details of how each of these tools interact with BREP and BREP construction artifacts are provided. There are a number of similarities between these three tools, but some BREP construction artifacts are only directly accommodated by a subset of the three tools or a tool may have alternative methods to accommodate a specific BREP artifact.

The *refine* open source mesh adaptation mechanics package fulfills the error estimation and mesh mechanics components. It is available via `https://github.com/NASA/refine` under the Apache License, Version 2.0. The current version under development uses the combination of split, collapse, and element swap operations [29]. Some of these classic operators are undergoing replacement with cavity operators [30]. Vertex relocation is performed to improve adjacent element shape with a convex combination of ideal vertex locations [31] or nonsmooth optimization based on Freitag and Ollivier-Gooch [32]. *refine* requires the domain to be manifold and a one-to-one correspondence of discrete vertex, segment elements, and triangle elements to geometry node, geometry edge, and geometry face entities, i.e., virtual topology, chains, and quilts are not supported.

The EPIC anisotropic mesh adaptation package developed at Boeing provides a modular framework for anisotropic mesh adaptation that can be linked with external flow solvers [33]. EPIC relies on repeated application of split, collapse, element reconnection, and vertex movement operations to modify a mesh such that element segment lengths match a given anisotropic metric tensor field. EPIC can support virtual topology indirectly through a discrete surface surrogate.

FEFLO.A is a 2*D*, 3*D*, and surface mesh adaptation tool. It uses a combination of generalized standard operators (e.g., insertion, collapse, element swap). The generalized operators are based on recasting the standard operators in a cavity framework [30, 34]. The cavity operator allows a simultaneous application of multiple standard operator combinations. Quality improvements are attained with the cavity operator that are not possible through a sequential application of standard operators. FEFLO.A can support virtual topology indirectly through a discrete surface surrogate.

A. Geometry Interrogation

refine and FEFLO.A access the topology and evaluate geometry via the Electronic Geometry Aircraft Design System (EGADS) [35]. EGADS is part of the Engineering Sketch Pad (ESP) [36], which includes the Open-source

Constructive Solid Modeler (OpenCSM) [37] for initial geometry import and setup. High-performance computing equipment, including graphic processing units (GPUs), are supported by EGADSlite [38], a lightweight ANSI-C version of the EGADS functions required by the initial and adaptive mesh process. *refine* utilizes EGADSlite for efficient parallel execution and geometry evaluation.

EPIC uses its own lightweight, natively-implemented geometry kernel to access geometry and provide basic interrogation functions such as point projections, curvature, and normal evaluations. The geometry model can be defined by a topologically connected set of trimmed surface faces, or by a discrete collection of triangle or quadrilateral boundary elements. The results in this paper were obtained using trimmed Nonuniform Rational Basis Spline (NURBS) surface faces.

B. BREP, Mesh, and Mesh-to-BREP Persistence

The mesh-to-BREP association information cached in the Gamma Mesh Format[†] is used to populate a *refine* data structure. This data structure contains a list of geometric associations at each discrete vertex, where a vertex can have zero or more associations. The associations can be to geometry nodes, edges, or faces. These associations record the type of parent geometry entity, an index to the instance of the entity, and the parameters of a curve or surface location on the entity.

The Gamma Mesh Format is also used for mesh, solution, and metric interchange in addition to geometry association. *refine* can read and write this format natively to interact with a single file for the whole domain during parallel execution. The mesh-to-BREP association persistence keywords are listed in [6] and the EGADSlite geometry model data are stored as a `ByteFlow` keyword.

EPIC associates mesh vertex, segment and boundary triangle with geometric entities. Each association may have multiple instances (e.g., a vertex between two surfaces will be associated with its two neighboring geometry faces). Associations are stored as parametric locations in the case of NURBS geometry and as references to the geometry boundary triangles in the case of discrete geometry. The geometry associations are saved with the initial mesh and tracked on all subsequent adapted meshes in the EPIC native mesh file format.

C. Higher-Order Surface Surrogate

Allowing the discrete grid topology to be different than the BREP topology has the possibility of alleviating the challenges of problematic BREP features discussed in the introduction. There are also situations where a geometry model may not be available (e.g., adaptation based on an existing mesh without underlying geometry support). The input mesh may have been constructed with an external tool that made BREP topology modifications internally without providing a consistent geometry model that matches the surface mesh. For these situations EPIC and FEFLO.A support a high-order discrete surface grid as a surrogate geometry model. EPIC constructs quadratic Nagata patches representing each discrete triangle [39]. FEFLO.A constructs cubic triangular surfaces.

FEFLO.A uses a two-step procedure to recover the BREP-mesh association and increase the robustness of geometry evaluation during the remeshing procedure. Recovering the BREP-mesh associativity is based on a fast inverse projection process. For each entity on edge curves or face surfaces, given a point P, a nonlinear search is performed to find $\sigma(u, v)$ minimizing $\|P - \sigma(u, v)\|$, where σ is the parametric function. This problem consists of finding roots of the following (f, g) functions:

$$\begin{cases} g(u, v) = (\sigma(u, v) - P) \cdot \sigma_u(u, v) = 0 \\ f(u, v) = (\sigma(u, v) - P) \cdot \sigma_v(u, v) = 0. \end{cases}$$

As for as any nonlinear problem, the initial guess is critical to ensure validity and fast convergence. Octree and neighboring vertex associations are used for robust initial guesses. The result of this optimization is a match between each surface vertex and the underlying geometry model. The association recovery process can be applied to any initial surface mesh, where most of the topology, gaps, and tolerances issues have been resolved. Initializing the mesh adaptation process with an association to a valid surface mesh implicitly recovers these geometry repairs with the association. To increase evaluation robustness, a 3rd-order (P_3) mesh is built as a surrogate geometry model. A P_3 surface ensures a $\mathcal{G}_1$ continuity at mesh vertices and exact imposition of tangent planes. The P_3 mesh is used as an intermediate level between the linear discrete mesh and the fully continuous geometry model. When a new surface vertex is inserted, a hierarchical projection is used. The vertex is projected to the BREP (through geometry model

---

[†]a keyword-based file format with reference implementation and description available at `https://github.com/LoicMarechal/libMeshb` [retrieved 16-APR-2019]

interaction) and the P_3 mesh. P_3 projection can replace a BREP evaluation failure, provide a guess to BREP inverse evaluation, or help detect BREP construction artifacts in the case of a mismatch. The P_3 surface mesh can be used in isolation when no geometry is provided. The JSM nacelle geometry is shown in Fig. 2 for linear and P_3 meshes.
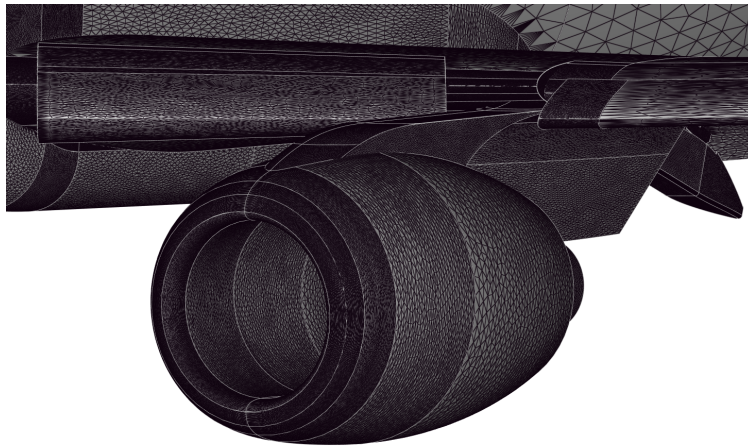

(a) Linear surface mesh.
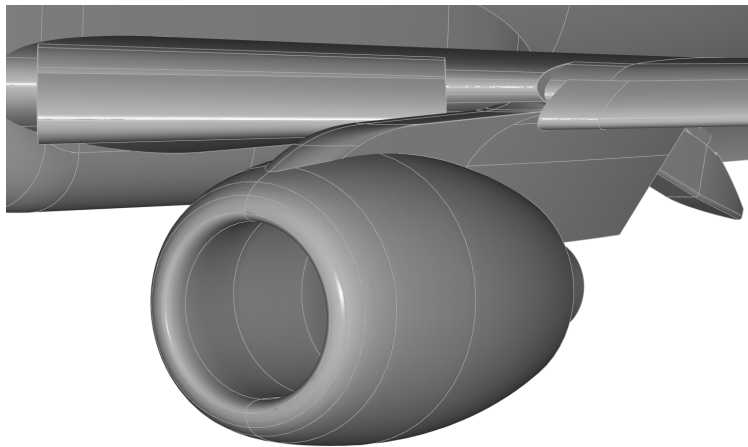
(b) P_3 surface.

Fig. 2   Illustration of the (a) linear surface mesh and (b) surrogate P_3 mesh model on the JSM nacelle geometry.

D. Parameterization Periodicity and Degeneracy

In *refine*, the topology exposed by EGADS is traversed to mark discrete vertices with parametric periodicity or degeneracy. This gives *refine* awareness of parametric discontinuities visualized in Fig. 3 and allows for correct evaluation (and inverse evaluation) of existing and new discrete vertices. Taylor and Haimes [7] describe the topology of a cylinder with a periodic edge and cone with periodic and degenerate edges. Aubry et al. [40] describe the impact of these features on mesh generation. Periodicity is when a face uses an edge twice, which results in two values of the surface parameter u, v at an edge t (u at the top of the cylinder Fig. 3(a)). A degeneracy is where a parameter range evaluates to a single x, y, z location (u at the center of the circular face of the cylinder Fig. 3(b)). Topologically, there is an edge with one node at this degeneracy. The circular face has a four edge topology: one at the center with one node, a radial periodic edge used twice, and the edge along the circumference. EPIC does not support periodic geometry surfaces. Prior to running EPIC, a preprocessing step is performed to identify periodic faces and split them along an

interior iso-parametric line of the underlying surface creating two nonperiodic faces. Degenerate edges are supported by EPIC provided a vertex in the initial mesh exists on the degenerate edge. In this case, the vertex is marked as frozen (i.e., the vertex cannot be deleted or moved in the adapted mesh).

(a) u parameter. (b) v parameter.
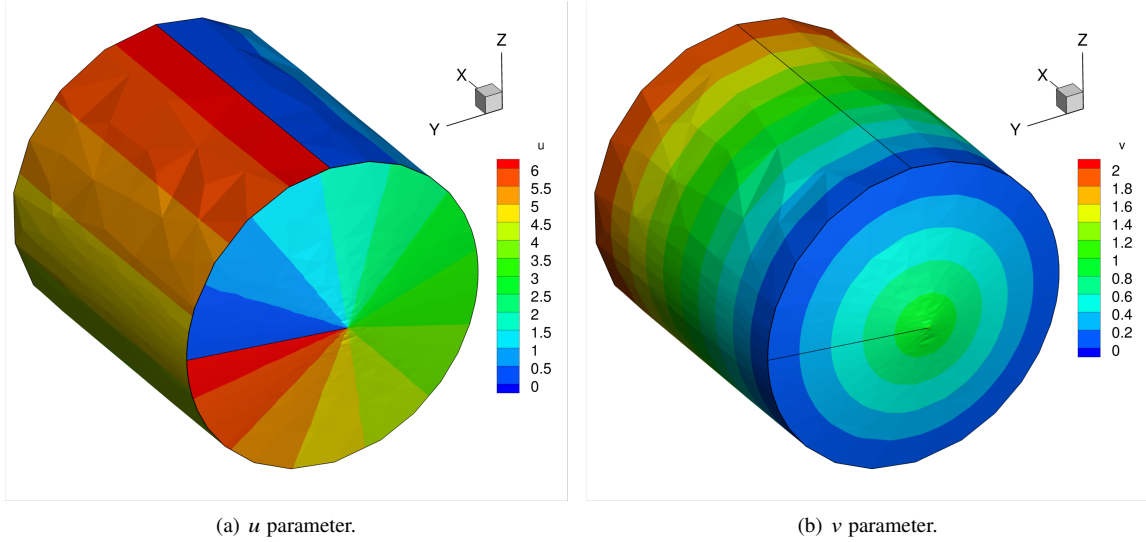
Fig. 3 Illustration of cylinder geometry with u-parameter periodicity and degeneracy.

E. Geometry Feature Size and Curvature Constraints

Unresolved geometry features (short edges and narrow faces) and high curvature can create situations where subsequent adaptive modifications are difficult, especially where anisotropic metric sizing requests are incompatible with the geometry.

refine estimates a feature size on each mesh vertex associated with a geometry edge. This size is computed by querying the (typically two) faces that use this edge. For each of these faces, the vertex is inversely evaluated to all edge curves of the face that do not share a geometry node with the edge associated with the vertex. The feature size is computed as the distance between the vertex and the location on the edge curve identified via inverse projection. If a vertex is associated with a geometry node, the edge feature size process is repeated for each incident edge. An isotropic metric is formed at the vertex with an eigenvalue that matches the geometry feature size. While the current implementation produces an isotropic metric, an anisotropic metric could be formed from edge curve normal and tangent proximity queries.

The curvature k is evaluated for the curve that supports the geometry edge associated with a vertex. The curvature is converted into an isotropic spacing constraint,

$$h_{\text{constraint}} = \frac{1}{S_{\text{radian}} |k|}, \tag{1}$$

where S_{radian} is the number of segments used to discretize one radian of a $1/k$ radius circle. The *refine* default is $S_{\text{radian}} = 2$.

The curvature k is zero for a straight line. To prevent a division by zero, the constraint is not computed if $|k|$ is smaller than $10/(dS_{\text{radian}})$ where d is the diagonal of the domain bounding box. The local tolerance of the EGADS entity is queried and the constraint is ignored if $h_{\text{constraint}} < 100\,t$, where t is the edge tolerance. This tolerance test is a crude filter to omit constraints that are unlikely to be the intent of the model creator and are hard to consistently represent where discrete mesh size approaches the model tolerance. These locally high curvatures are often the result of enforcing the continuity of one or more derivatives between geometry entities in model construction. A more sophisticated filter that examines the neighboring geometry entities is beyond the scope of the current work, but may be required for increasing robustness to pathological geometry construction issues. If the $h_{\text{constraint}}$ is accepted by this filter, an isotropic metric is formed at the vertex with an eigenvalue that matches $h_{\text{constraint}}$ and this metric is intersected with other geometry metric constraints at this vertex.

EGADS provides the principle surface curvatures k_r, k_s in an orthogonal set of vectors tangent to the surface $\hat{r}, \hat{s}$. A vector normal to the surface is formed with $\hat{n} = \hat{r} \times \hat{s}$. For locations where the surface parameterization is degenerate, the curvature is evaluated 1% of the min-max parameter range of the face toward the center of the face, away from the nondegenerate parameter (the parameter with the larger x, y, z parameter derivative). Each curvature is converted to a spacing with Eq. (1). The normal spacing is set to $h_n = 0.1d$ based on the domain bounding box d. Aspect ratio of the metric constraint is limited to $h_{\text{ratio}} = 10$ by setting

$$h_r = \min\left(h_r, h_{\text{ratio}} h_s\right) \tag{2}$$

$$h_s = \min\left(h_{\text{ratio}} h_r, h_s\right) \tag{3}$$

$$h_n = \min\left(h_n, h_r, h_s\right). \tag{4}$$

Limiting the aspect ratio of the surface mesh prevents incompatibilities with isotropic initial volume mesh generation methods (e.g., Delaunay). The surface curvature constraint is ignored (in the same way as edge constraints) if $h_r < 100\,t$ or $h_s < 100\,t$, where t is the face tolerance. A metric is formed at the vertex with the eigenvalues of $1/h_n^2, 1/h_r^2, 1/h_s^2$ and eigenvectors of $\hat{n}, \hat{r}, \hat{s}$. The surface curvature metric is intersected with other geometry metric constraints at this vertex.

An example from the Japan Aerospace Exploration Agency Standard Model (JSM) with nacelle is shown in Fig. 4 to illustrate the necessity of curvature limits. A view forward of the configuration looking aft is shown in Fig. 4(a,b). A detail of the lower leading edge of the inner and outer flap blend region is shown in Fig. 4(c), where $\max(k_r, k_s)$ is shown with an exponential color scale. A thin blue horizontal streak is seen in the center of Fig. 4(c), where the streak corresponds to a small region of the surface with $O(1000)$ curvature surrounded by a region of $O(10)$ curvature. The region of $O(10)$ curvature is likely the design intent of the model, but the $O(1000)$ curvature is likely a result of imposing a tangency constraint between the adjacent lower and upper flap nose blend faces. Deactivating the face curvature constraint on the thin blue streak of Fig. 4(c) and $v = 0$ of Fig. 4(d) prevents refinement of the unintended construction artifact that can fail when the refinement approaches the tolerance of the geometry model.
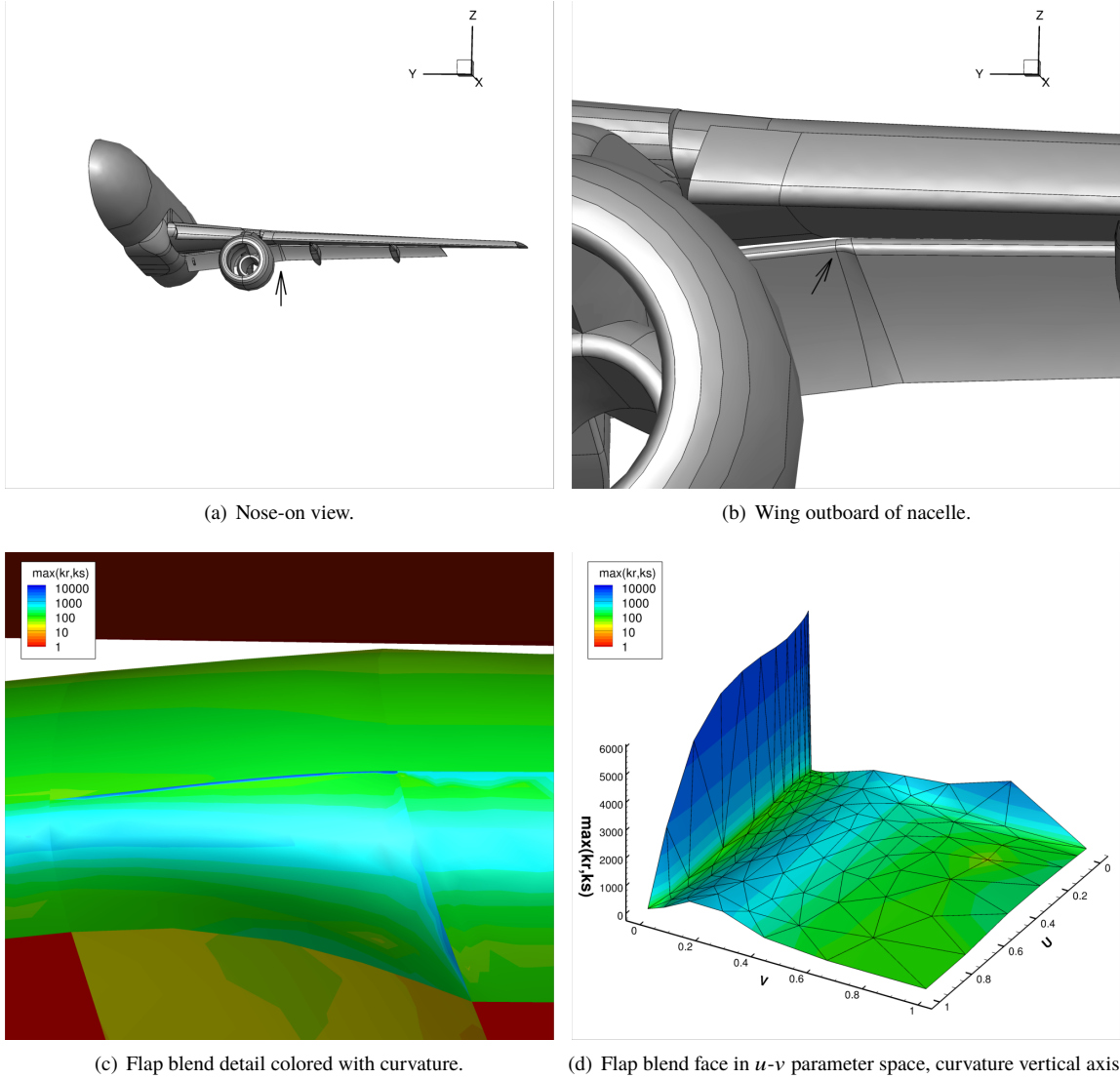
The constraint metric on surface vertices is interpolated across the surface and through the volume with "mixed-space-gradation" of [41]. This smooth constraint metric is used for adaptation of the initial surface mesh and volume mesh before adaptation begins. This ensures that the mesh satisfies the geometry constraint metric during the entire mesh adaptation processes. Once the solution and solution-based metric are available, the solution-based metric is constrained by the geometry metric. After the input metric is interpolated from the background mesh to the current mesh, the interpolated metric is intersected with the constraint metric. These constraints are active on the initial mesh and then may become inactive as the solution-based metric requests finer spacing.

EPIC constructs an anisotropic sizing metric based on the principal curvatures of geometry curves and surfaces. At each surface vertex, a curvature-based sizing metric is formed for each geometry surface or curve associated with the vertex. A sizing metric for a single surface or curve is created from the magnitude and direction of the surface principal curvature vectors evaluated at the vertex. A constraint size in each principal direction is computed using Eq. (1). An ellipsoid is then constructed that covers the principal direction vectors and computed sizes. At vertices with multiple geometry surface or curve associations, the curvature-based metrics from each associated entity are intersected to form a single geometry metric. In addition to curvature-based sizing, discontinuities between adjacent geometry curves or surfaces can be detected and the sizing metric normal to the discontinuity can be further refined. EPIC also detects narrow geometry faces and constructs a limit metric at each vertex based on the distance in the narrow direction and orientation. The resulting geometry curvature and thin face metrics are then intersected with the sizing metric derived from the solution error estimate. An example of where thin face limiting can be beneficial is shown in Fig. 5. This close up view of the JSM nacelle trailing edge consists of a thin planar surface trimmed by two concentric curves. These curves are too close together to distinguish at this scale. The geometry curvature metric at this location will limit the circumferential size to prevent points on the inner and trim outer curves from crossing in Fig. 5(b). *refine* constrains the metric on the nacelle trailing edge with a feature size isotropic constraint based on the width of the trailing edge face in Fig. 5(c).

F. Surface Normal Deviation and Area Constraints

refine has a triangle surface normal deviation constraint, where the deviation $n_d = \hat{n} \cdot \bar{n}$. Here $\hat{n}$ is the normal of the geometry surface at the triangle u, v parameter centroid, and $\bar{n}$ is the normal of the discrete triangle. Ideal conformity of the triangle normal to the surface normal is $n_d = 1$, and a completely inverted triangle has a deviation of $n_d = -1$. Mesh modification operations are rejected unless $n_d \geq 0.1$. While *refine* is typically able to produce surface meshes with deviation of 0.8 or greater, this lower limit prevents tangling of the surface during metric conformity optimization.

(a) Nose-on view.

(b) Wing outboard of nacelle.

(c) Flap blend detail colored with curvature.

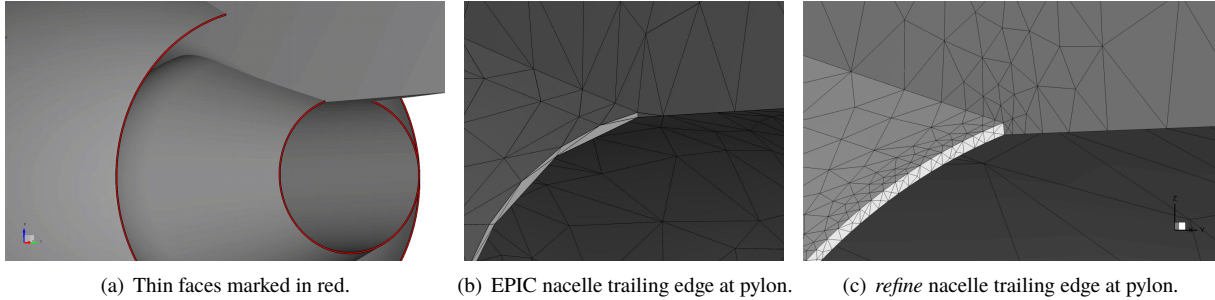(d) Flap blend face in u-v parameter space, curvature vertical axis.

Fig. 4   JSM blend leading edge between inner and outer flap sections.

The $\hat{n}$ surface normal is computed by a cross product of the u and v derivative. This cross product may point into or out of the domain. The EGADS model is queried to determine the "sense" of a face's surface and the direction of $\hat{n}$. The area of the triangle in u, v is also constrained to be positive (or negative based on the "sense" of a face's surface). EPIC and FEFLO.A also compare the normal vector deviation between mesh boundary triangles and geometry surfaces to prevent any mesh operations that cause a deviation beyond a specified tolerance. In FEFLO.A, the minimum dot product between the normal of the triangle and the normal of the surface at the vertices is constrained to be greater than 0.8 or the current dot product.

G. Initial Mesh

The initial mesh (Fig. 1) in a *refine*-based process is produced in four steps. First, the initial surface mesh is provided by an EGADS tessellation object. Default global parameters are provided to the EGADS specifying $0.25d$ for the maximum length of a triangle side, $0.001d$ for the maximum deviation between the centroid of the discrete object and the underlying geometry, and $15°$ for the maximum interior dihedral angle, where d is the diagonal of the domain bounding box. The user can override these defaults by setting these tessellation parameters as attributes either with the EGADS API or via OpenCSM. A smaller maximum length or deviation parameter may be required to force a face tessellation to

9

(a) Thin faces marked in red.   (b) EPIC nacelle trailing edge at pylon.   (c) *refine* nacelle trailing edge at pylon.

Fig. 5   JSM nacelle trailing edge geometry constraint.

contain a triangle, where the defaults produce zero triangles on a face. *refine*'s assumption of one-to-one geometry and discrete mesh topology requires that all faces have one or more triangles (and edges have one or more discrete segment elements). Second, the EGADS surface tessellation is adapted by *refine* to satisfy the geometry metric constraint with the surface mesh. Third, the volume of the adapted surface mesh is filled by Advancing-Front/Local-Reconnection (AFLR) [42] or TetGen [43]. The surface (with EGADS geometry associativity via EGADS tessellation object) and volume meshes are merged. Fourth, this initial volume mesh is adapted so that the tetrahedra also satisfy the geometry metric constraint. The volume adapted to geometry constraints provides a start to the adaptive procedure.

The initial mesh for EPIC is generated by first creating a coarse isotropic triangular mesh based on surface curvature using the Boeing MADCAP‡ mesh generation tool. The isotropic surface mesh is then adapted based on the geometry metric to form a coarse anisotropic mesh. Anisotropy of the adapted surface mesh is limited to increase the robustness of the volume mesh boundary recovery. An initial volume mesh for EPIC is then constructed with AFLR.

The initial mesh for FEFLO.A is built in two stages. The initial surface mesh is constructed from the geometry using the `mg_cadsurf` module. The initial volume mesh is constructed from the surface mesh with the `mg_tetra` module. These modules are distributed by Distene as part of the MeshGems suite based on the INRIA surface [44] and the volume [45] mesh generation research.

III. Flow Solvers

Two flow solvers are employed to compute the flow solution in Fig. 1. The focus of this work is on accommodating complex geometry through mesh mechanics. The flow solvers are used to provide relevant inviscid and RANS solutions to drive solution-based adaptation and evaluate the mesh mechanics. Both of these solvers are capable of using mixed-element meshes, but only purely-tetrahedral meshes are considered here.

A. FUN3D

FUN3D-FV [46, 47] is a finite-volume Navier-Stokes solver in which the flow variables are stored at the vertices or nodes of the mesh. At interfaces between neighboring control volumes, the inviscid fluxes are computed using the Roe approximate Riemann solver [48] based on the values on either side of the interface. For second-order accuracy, interface values are extrapolated from the vertices with gradients computed at the mesh vertices. These gradients are reconstructed with an unweighted least-squares technique [46]. The interface values are limited for the C25F with the van Leer limiter [49].

The full viscous fluxes are discretized using a finite-volume formulation in which the required velocity gradients on the dual faces are computed using the Green-Gauss theorem. On tetrahedral meshes, this is equivalent to a Galerkin type approximation. The solution at each time step is updated with a backward Euler time-integration scheme. At each time step, the linear system of equations is approximately solved with a multicolor point-implicit procedure [50]. Local time-step scaling is employed to accelerate convergence to steady state. The Spalart-Allmaras (SA) turbulence model [51] is loosely-coupled to the meanflow equations, where the meanflow and turbulence model equations are relaxed in an alternating sequence.

---

‡MADCAP User's Guide is available `https://www.grc.nasa.gov/www/winddocs/windus3.0/madcap/index.html` [accessed 06-APR-2019].

The SA turbulence model requires the distance from every node to the nearest noslip boundary condition. The standard wall distance calculation in FUN3D-FV finds the nearest surface node and then searches adjacent triangles to see if they are closer than the closest surface node. The standard wall distance method is inaccurate if the closest triangle is not adjacent to the closest surface node. To provide an accurate wall distance, which is critical to the SA model, an alternative method is used on adapted meshes. The alternative method encloses each surface triangle in a bounding box. These bounding boxes are stored in an Alternating Digital Tree (ADT) [52] for fast searches. The alternative wall distance method finds the closest surface triangle for adapted unstructured meshes.

B. GGNS

GGNS (General Geometry Navier-Stokes) is a Boeing-developed flow solver built upon the SUPG finite-element discretization. The code uses piecewise linear finite elements resulting in a 2nd-order accurate discretization. Additional 1st-order artificial viscosity built upon the DG discretization is added for shock capturing. The indicator triggering this additional stabilization is based on the oscillation of the Mach number across a cell. The number of degrees of freedom for the 2nd-order SUPG scheme is equal to the number of vertices in the computational mesh. The discretization is vertex-based in the sense that it is conservative over the dual volumes of an unstructured mesh. More details on discretization used in the GGNS solver, including the particular choices of discretization variables and special treatment of the essential boundary conditions via the Lagrange-multiplier based technique [53], can be found in Kamenetskiy et al. [54].

The discrete nonlinear solver in the GGNS code implements a variant of the Newton-Krylov-Schwarz algorithm. On the code level, this is accomplished using the Portable, Extensible Toolkit for Scientific Computation (PETSc) [55–57] framework. Time stepping is employed to drive to the steady state solution. On each time step, an exact Jacobian matrix for the discretization is formed by an automatic differentiation technique. The linear system arising from the Newton's method is approximately solved using GMRES with a drop-tolerance-based block-ILU preconditioner (locally on subdomains) implemented in the context of the additive Schwarz method with minimal overlap [58]. Right preconditioning is employed to maintain consistency between the nonlinear and linear residuals. The compact stencil property of the SUPG scheme helps to reduce the fill-in levels in the approximate factorization, thereby reducing the memory footprint.

A line search is applied along the direction provided by the approximate solution of the linear system. Residual decrease and physical realizability of the updated state are tracked during the line search. A heuristic feedback algorithm is implemented to communicate failure of the line search back to the time-stepping algorithm, so that the CFL number can be increased or decreased as necessary. There is no upper preset limit for the CFL number in the time-marching algorithm; so Newton-type quadratic convergence (or, at least, superlinear, due to inexact linear solves) is routinely achieved at steady state.

IV. Multiscale Metric Construction

The metric construction step of the solution-based mesh adaptation process (Fig. 1) specifies the new mesh resolution and orientation request to control estimated error in the solution. Classic Hessian-based adaptation methods [59, 60] control quadratic interpolation error estimates in the L^∞-norm, which results in excessive refinement of nonsmooth regions. The multiscale metric $\mathcal{M}_{L^p}$ controls the L^p-norm of the interpolation error of a scalar field [61], which balances refinement at multiple rapidly and smoothly varying spatial scales. The key ingredient of the multiscale metric is the local scaling by the (reconstructed) Hessian $\mathcal{H}$, determinant,

$$\mathcal{M}_{L^p} = D_{L^p} \det(\mathcal{H})^{\frac{-1}{2p+d}} |\mathcal{H}|, \tag{5}$$

where a global scaling D_{L^p},

$$D_{L^p} = \left(\frac{C_t}{C \left(\det(\mathcal{H})^{\frac{-1}{2p+d}} |\mathcal{H}| \right)} \right)^{2/d}, \tag{6}$$

corrects the complexity of the locally scaled Hessian to produce $\mathcal{M}_{L^p}$ with specified target complexity C_t. Complexity is a measure of density of the spacing request and can be interpreted as the continuous counterpart of the spacing used

for uniform mesh refinement. The complexity, C, of a continuous metric field, $\mathcal{M}$, is defined as the integral,

$$C(\mathcal{M}) = \int_\Omega \sqrt{\det(\mathcal{M}(x))}\, dx, \qquad (7)$$

and is evaluated on the discrete mesh and metric. Both scaling operations depend on the dimensionality of the domain, which is $d = 3$ in this case. A mesh conforming to $\mathcal{M}_{L^p}$ provides optimal control of the scalar field interpolation error in the p-norm. A lower p-norm targets weaker variations of the scalar field and a larger p-norm targets rapid variations of the scalar field.

A. *refine* Multiscale Metric

To form the metric, a Hessian is reconstructed by recursive application of the L^2-projection gradient reconstruction scheme. The gradient is computed in each element and a volume-weighted average is collected at each vertex [61]. The 2nd-derivative Hessian terms are formed by computing the reconstructed gradients of these gradients formed in the first pass. The mixed derivative terms of the Hessian are averaged. A special boundary treatment is employed, where the reconstructed Hessian on the boundary is replaced with an extrapolation from neighboring interior vertices, which have a well-formed stencil.

The reconstructed Hessian is then diagonalized into eigenvalues and eigenvectors. The absolute value of the Hessian is formed by recombining the absolute value of the eigenvalues with eigenvectors to ensure the Hessian is symmetric positive definite. The Hessian at each vertex is scaled to control the L^p norm [61] with Eq. (5). The gradation of the metric field is limited with the "mixed-space-gradation" of [41]. The complexity is computed, and the metric is globally scaled to set its complexity to a specified value. The complexity is evaluated discretely by assuming it is piecewise constant in each median dual.

B. GGNS+EPIC Multiscale Metric

The Mach Hessian for each element is evaluated from the flow solution by using a least-squares approach on an extended stencil in GGNS. GGNS then passes the Hessian at each element to EPIC, which converts it to adaptation metrics via an element-centered modification of Alauzet and Loseille [61], which minimizes the L^p norm of interpolation error of the scalar field for a given grid complexity. In this modification, each elemental Hessian is scaled to control the L^p norm. The complexity of the resulting elemental adaptive metric is computed, and the global scale factor, D_{L^p}, is adjusted to better match the requested value. The metric is then iteratively recomputed until the computed complexity is within a specified tolerance of the requested value. A continuous metric field is generated by Log-Euclidean [62] interpolation of the elemental metrics to the grid vertices.

V. Mesh-Adapted Results on MCAD Models

The mesh-adaptation process (Fig. 1) is performed on example supersonic and high-lift configurations from two AIAA prediction workshops. The focus is the accommodation of complex geometry models with intended and unintended BREP features described in the Section I with mitigations described in Section II. A cursory presentation of workshop-relevant outputs show the utility of the current approaches without distracting from the focus on the key geometric features of the MCAD models.

A. C25F Configuration

The C25F is a notional configuration created to represent a sonic boom demonstrator class vehicle. The model includes wing, body, tail, nacelle, and flow-through engine path. The C25F was designed primarily for a low ground loudness level near the centerline of the flightpath [63], with some reduction in the noise level over the entire boom carpet [64]. The inviscid solver Cart3D [65] was used to design the configuration.

The geometry for C25F was created by an OpenCSM script that translated the Jaguar [66] description used to develop the model. Jaguar lacks direct geometry export to a geometry interchange format. The OpenCSM model is not a perfect recreation of the Jaguar model. Due to an oversight, the OpenCSM nacelle outer mold line was linearly interpolated between fuselage stations where the Jaguar model is splined. There are other smaller known differences that result in different loudness levels for the workshop configuration than AIAA papers describing the design. A STEP geometry file was exported from OpenCSM. The geometry was provided with a nose-up orientation of $3.375°$ to include the design incidence so that the CFD calculation is performed at $0°$ angle of attack. The freestream Mach number is 1.6.

There are complex shock and expansion interactions present in the C25F nearfield. Derlaga, Park, and Rallabhandi [67] provide details on how the inlet shock is reflected from the upper wing surface and lower horizontal tail surface to impact the nearfield and propagated ground signatures. This observation appears to be supported by the relatively large discretization error estimates reported in this region by Park and Nemec [26].

The initial EGADS coarse tessellation has high-valence vertices with large gradation (e.g., vertices connected to many triangles on the leading edge of the wing in Figs. 6(a) and 7(a)). These high degree vertices would be difficult to remove from a volume mesh, but much easier to remove via edge swaps of the surface mesh. Surface modifications are easier without the requirement of maintaining valid volume element topology. Surface adaptation before attempting to fill the volume reduces the execution time of the overall process and produces a surface mesh that is more metric conforming to the geometry constraints. Examples of surface meshes with controlled gradation to geometry constraints are shown in Figs. 6(b) and 7(b). Conformity to a low-gradation metric implicitly reduces vertex-triangle valance. This conformity to a set of constraints with controlled gradation dramatically increases the likelihood of initial mesh generation success and subsequent mesh adaptation operation success. The initial surface mesh constructed with MADCAP and adapted with EPIC is show in Fig. 8.

The initial and adapted surface meshes of the flow-through inlet are shown in Fig. 9. Fifty loops of the flow chart in Fig. 1 are performed with FUN3D-FV, *refine*, and multiscale metric to control interpolation error in Mach number with inviscid flow. The final inlet surface mesh is shown in Fig. 9(b) for the final volume mesh of 53M vertices. The adapted surface of the GGNS+EPIC final volume mesh of 22M vertices is shown in Fig. 9(d). A complex set of shock reflections, initiated by flow-through inlet spillage, is resolved by the surface grid. This interaction of inlet spillage, wing, and horizontal stabilizer is seen in Fig. 10. The fixed meshes provided by the SBPW-2 committee lacked adequate resolution to capture this interaction on coarser members of a uniformly-refined mesh family [26, 67].

Details of the wing tip and leading edge break are shown in Fig. 11 for the *refine* and EGADS workflow. The initial coarse EGADS tessellation is shown in Figs. 11(a) and 11(b). The final surface mesh for the volume mesh of 53M vertices is shown in Figs. 11(c) and 11(d). An EGADS tessellation has the key elements of being watertight, while coarse to reduce the initial grid generation time. The robustness of the combined surface and volume generation is indicated in the resolution of the high curvature convex leading edge and the concave leading edge break. An initial mesh that satisfies the geometry constraint metric is a key contributor to robustness. Imprints of the upper wing shock on the surface are seen in both adapted surface grids.

Descriptive statistics were gathered from participants of the Second AIAA Sonic Boom Prediction Workshop as described by Park and Nemec [26]. The pointwise mean (line) and standard deviation (error bar vertical extent) of the centerline pressure signature at five body lengths below the vehicle is shown in Fig. 12 for the ensemble of fine-grid participant submissions. Shock and expansion positions and levels of the FUN3D-FV+*refine* adapted solution are within one standard deviation of participant fine-mesh submission. The $X = [234 - 238]$ m region is a result of the critical shock reflection pattern highlighted in Fig. 10 that is captured by multiscale adaptation.
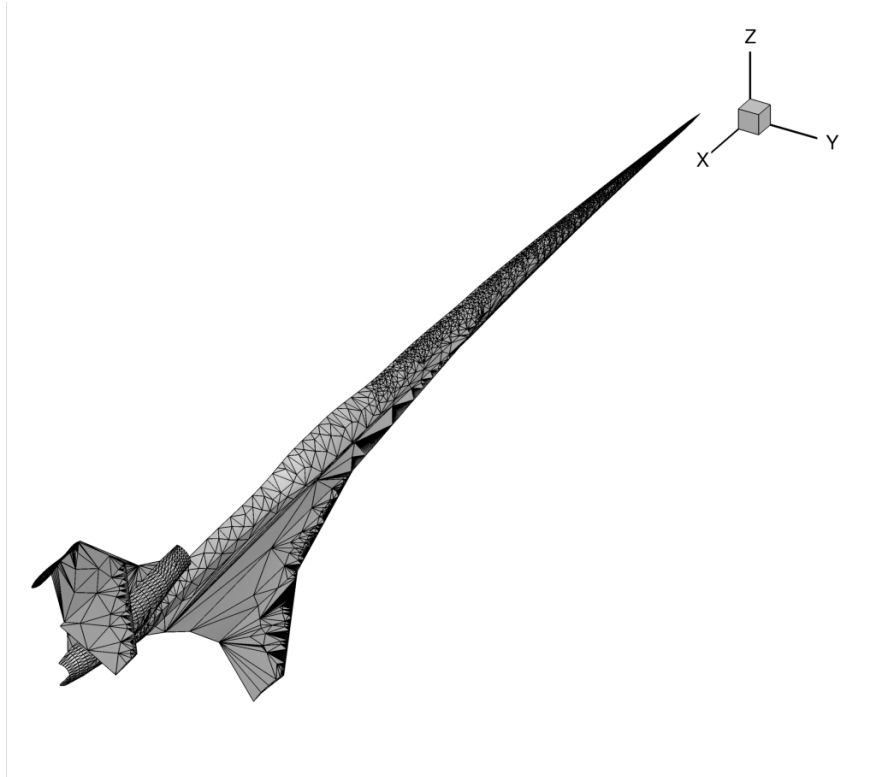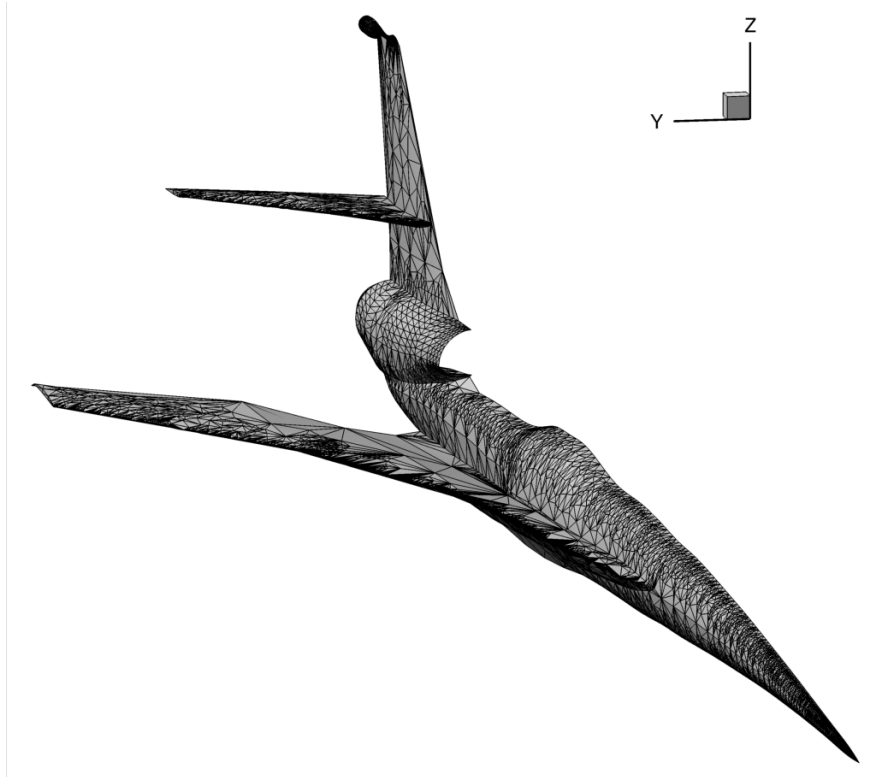
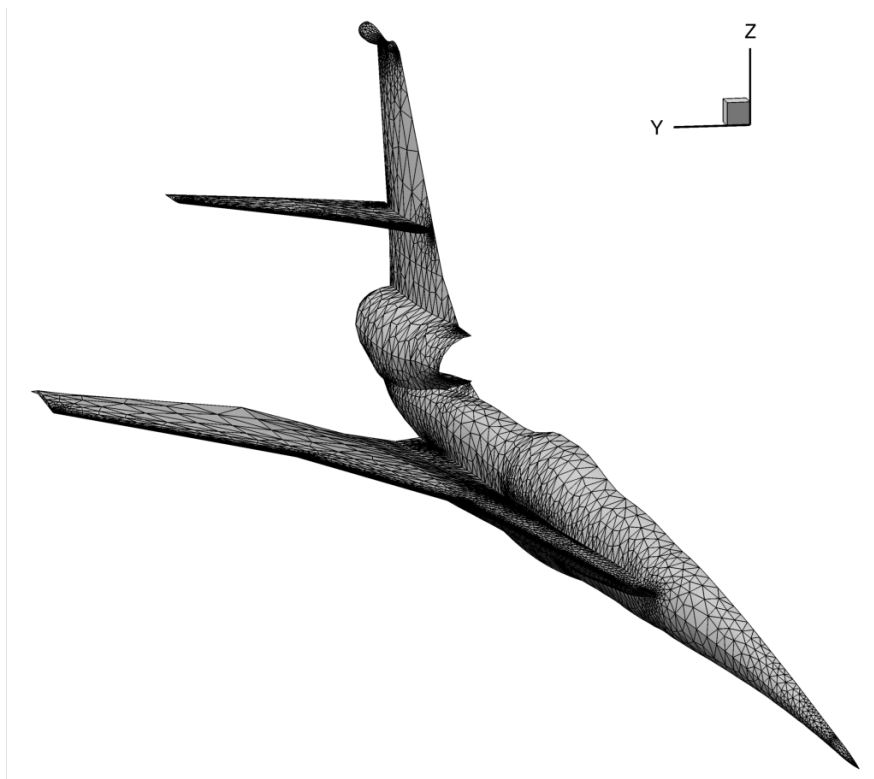(a) Initial coarse tessellation provided by EGADS.
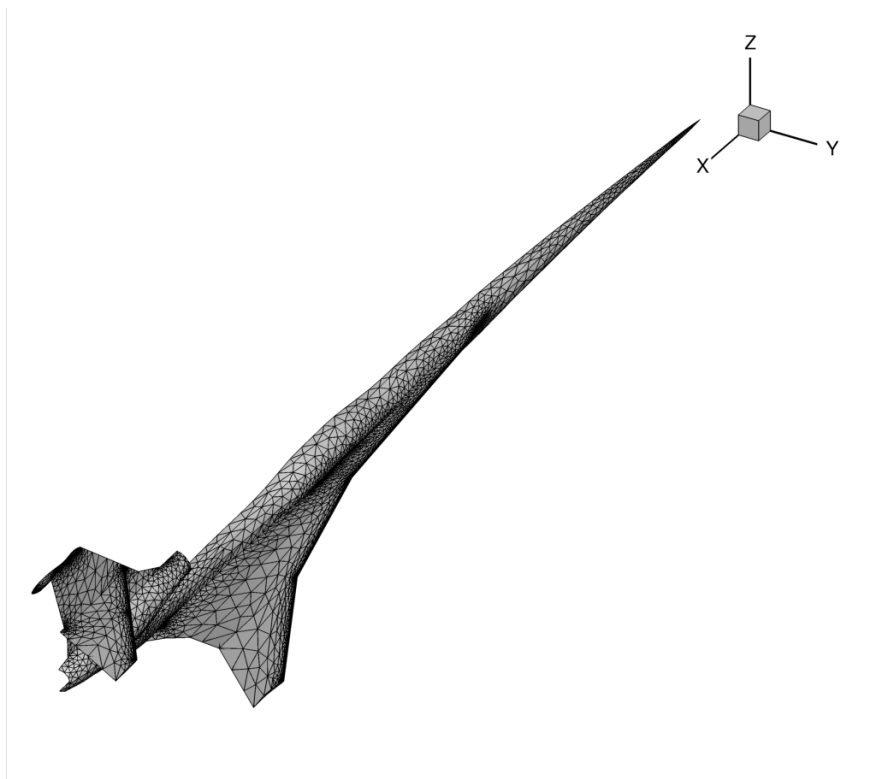
(b) Surface mesh adapted to *refine* geometry constraints.

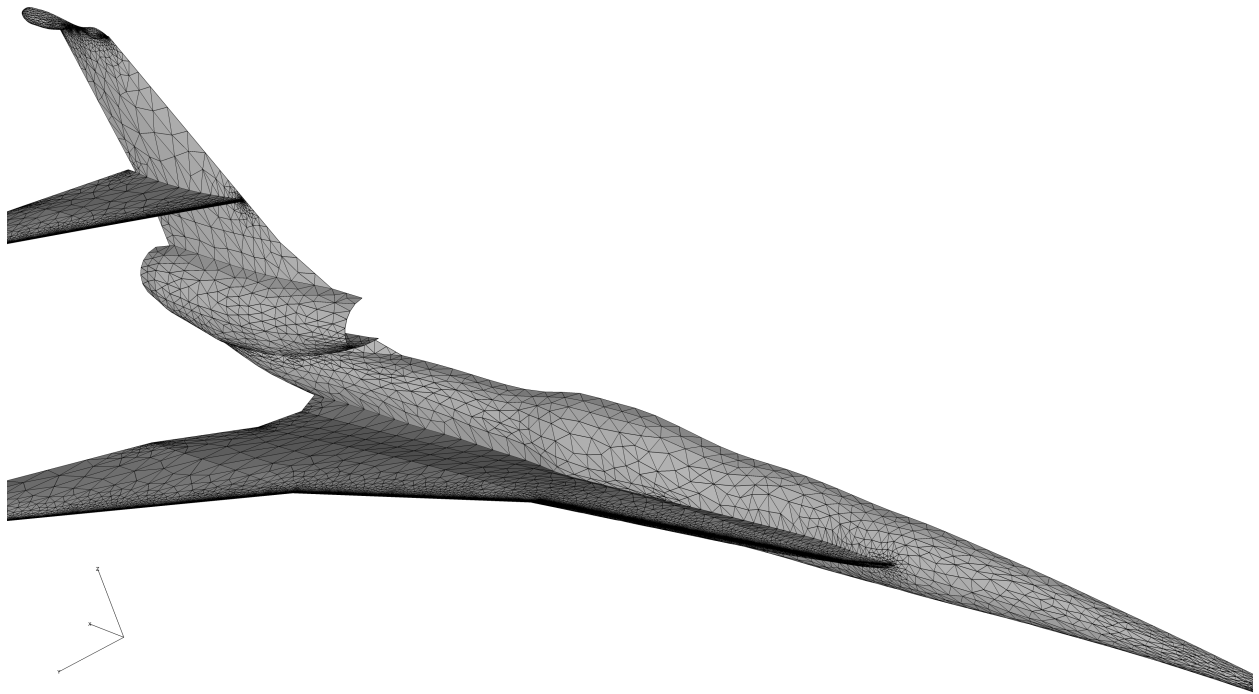**Fig. 6 Isometric view of the surface mesh.**
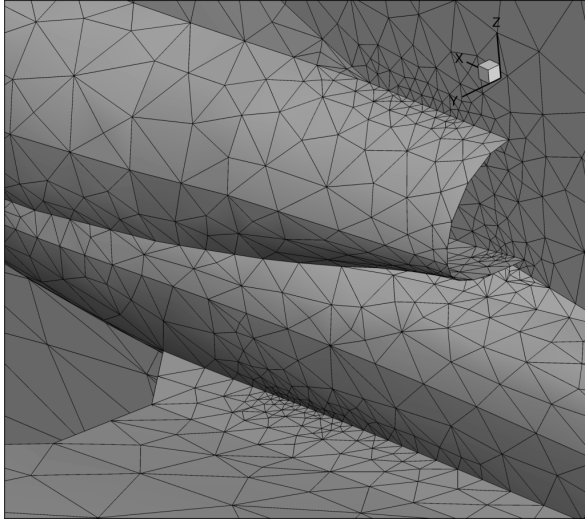
(a) Initial coarse tessellation provided by EGADS.

(b) Surface mesh adapted to *refine* geometry constraints.

Fig. 7   Surface meshes.

Fig. 8   Initial surface mesh with MADCAP geometry constraints.
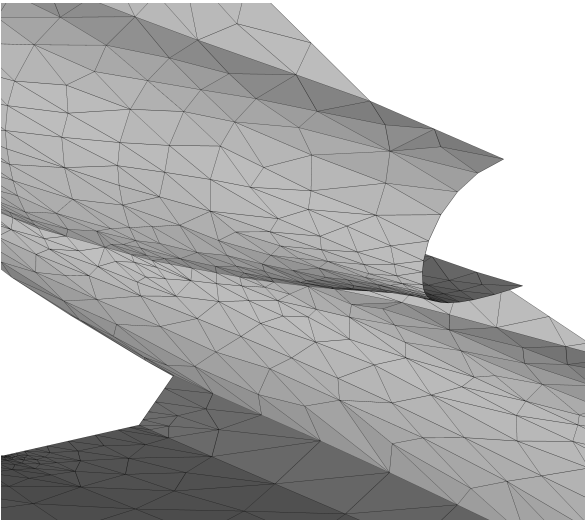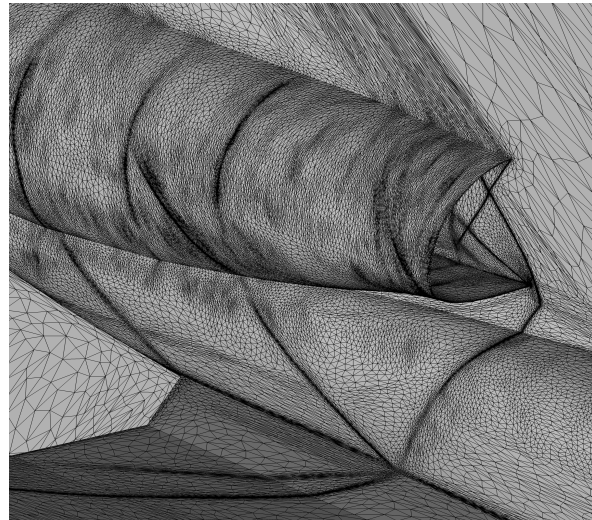
(a) Initial *refine* mesh.

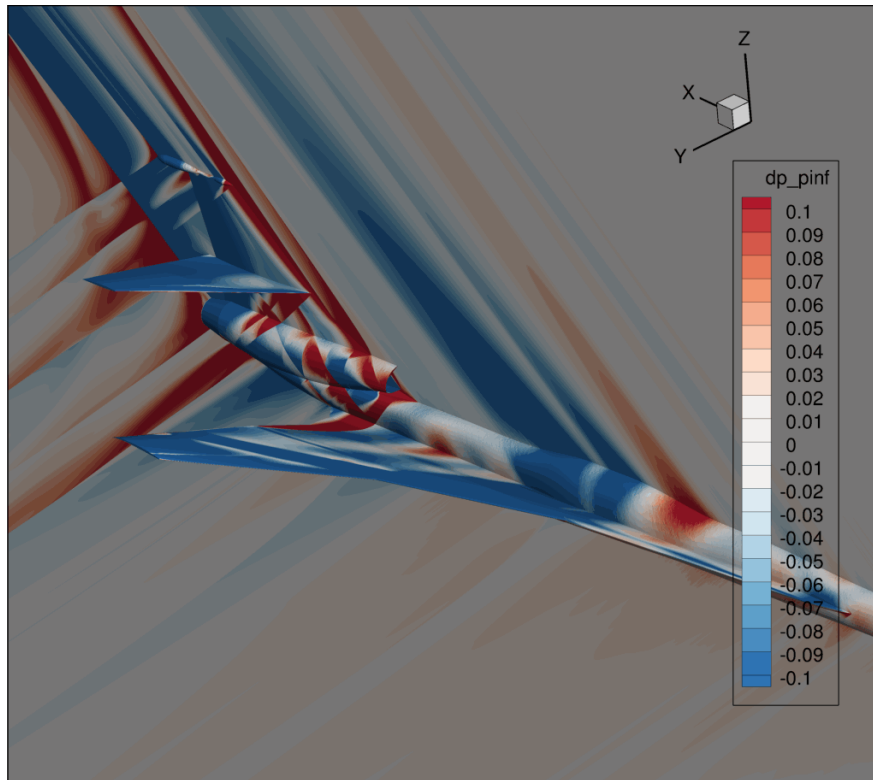
(b) 53M vertices, FUN3D-FV+*refine* multiscale adaptation.

(c) Initial MADCAP mesh.

(d) 22M vertices, multiscale GGNS+EPIC adaptation.

Fig. 9   Inlet surface meshes.

(a) Pressure difference, 53M vertices, FUN3D-FV+*refine* multiscale adaptation.
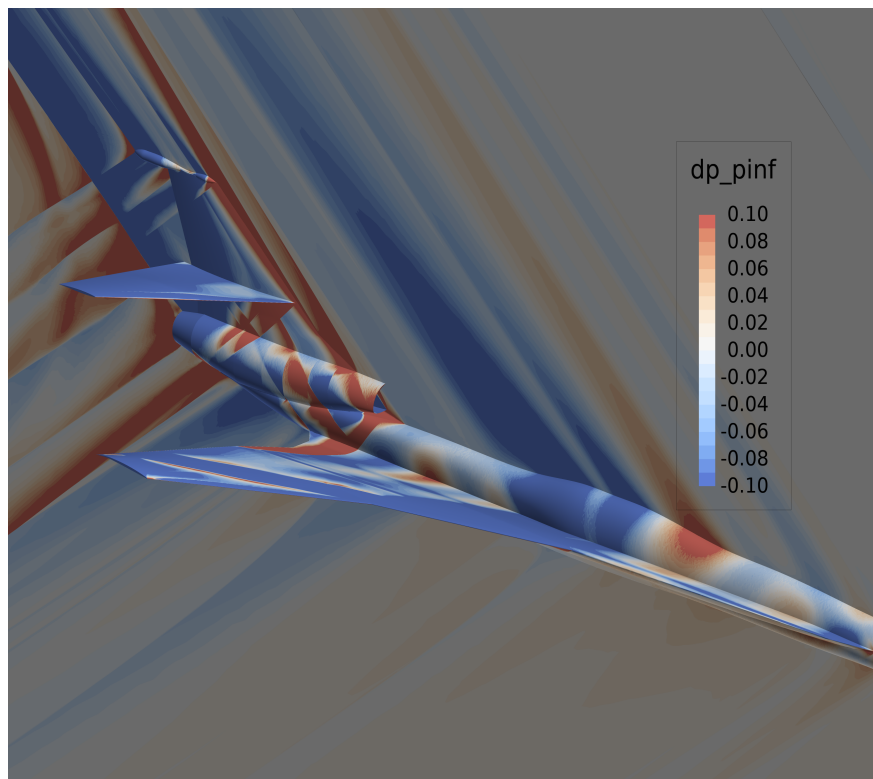
(b) Pressure difference, 22M vertices, GGNS+EPIC multiscale adaptation.

Fig. 10   Pressure difference on C25F and symmetry plane.
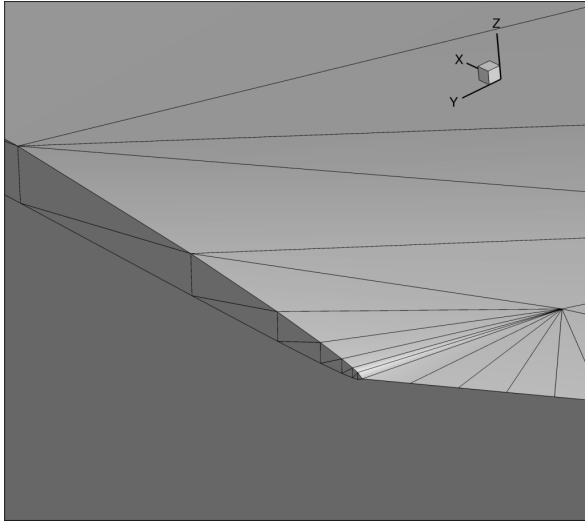
(a) Upper wing tip detail, EGADS tessellation.
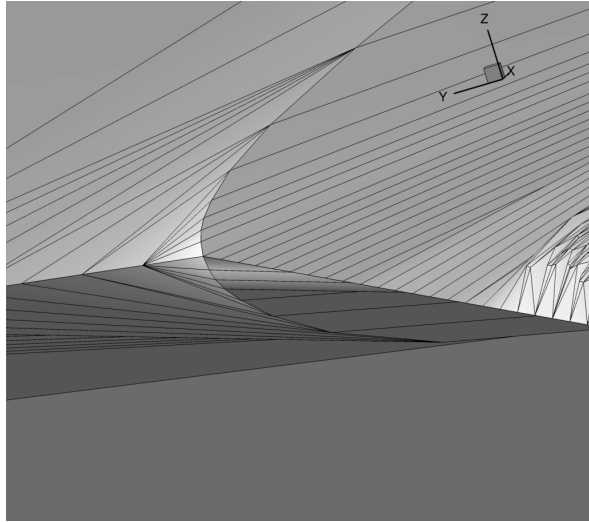
(b) Leading edge break detail, EGADS tessellation.
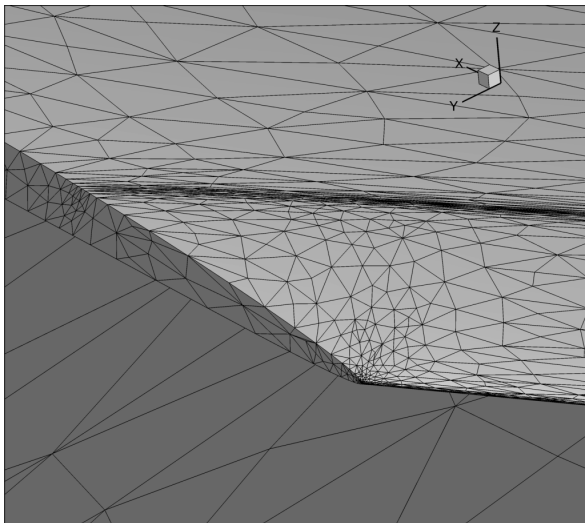
(c) Upper wing tip detail with footprint of wing shock, multiscale adaptation.
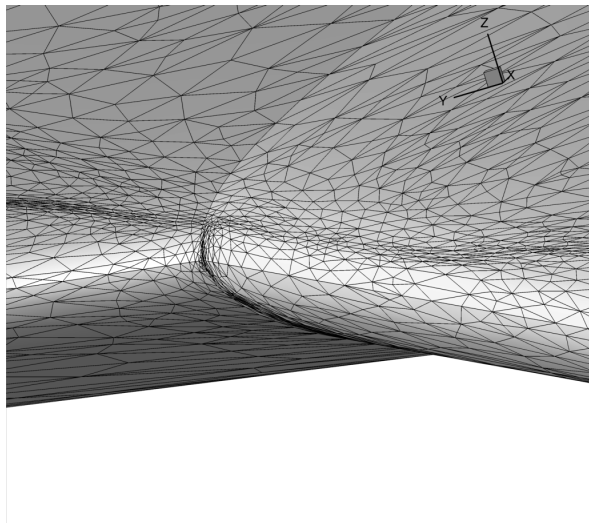
(d) Leading edge break detail, multiscale adaptation.

Fig. 11   Wing surface mesh details for EGADS and FUN3D-FV+*refine*.
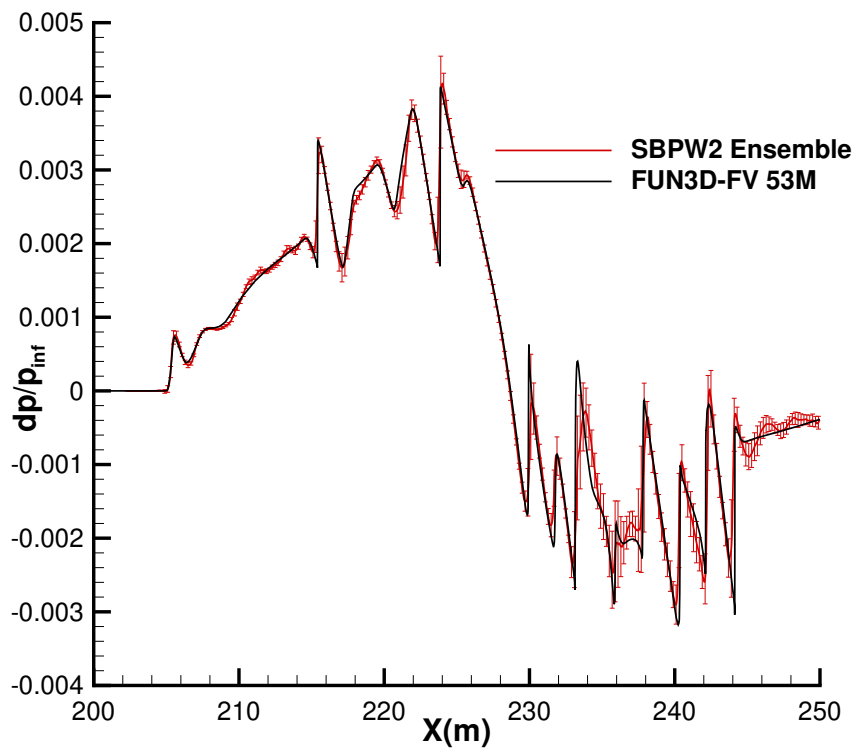
Fig. 12   Centerline pressure signature at five body lengths below the vehicle.

B. JSM Configuration

Rumsey, Slotnick, and Sclafani [27] describe the use of the JSM as part of HiLiftPW-3. Case 2 of HiLiftPW-3 was a nacelle installation study for the JSM, which included all slat and flap support hardware. The nacelle-on geometry with mesh adaptation Case 2d is the focus of this section.

refine makes a one-to-one assumption about the topology of the BREP model and the discrete mesh. The nacelle-on STEP file from the HiLiftPW-3 website has a zero area face where the inner nacelle and pylon intersect. With this zero area face, *refine* was unable to create a valid surface mesh that met the normal deviation constraint and was free of self-intersecting triangles. This zero-area face had to be removed manually and a new STEP file created for *refine* initial grid generation. This is a clear example where implementing virtual topology to suppress this zero area face and supporting edges may avoid a manual topology repair operation.

Details of the JSM nacelle initial EGADS tessellation are shown in Fig. 13. The lower surface of the inboard slat intersects the upper surface of the slat for the initial coarse EGADS tessellation. The surface meshes adapted to *refine* and EPIC surface constraints are shown in Fig. 14. The curvature-based adaptation resolves the lower and upper slat surfaces to alleviate this self-intersection that would prevent successful volume mesh generation. The dark spots in Fig. 14(a) are locations where *refine* feature size or curvature constraints imposed a small mesh spacing as compared to other surface locations. Resolving these constraints increases the execution time of *refine* and FUN3D-FV, but improves the robustness of subsequent mesh adaptation operations. This additional cost could be reduced by repairing the underlying features in the STEP file that trigger refinement (e.g., unintended curvature, short edges, narrow faces). Introducing virtual topology and improving curvature filtering could reduce the cost incurred to resolve the unintended STEP features, but is considered a topic for future work. The EPIC surface constraint in Fig. 14(b) did not resolve the construction artifacts shown in Fig. 14(a). Away from these artifacts, both *refine* and EPIC produced similar triangle size with some differences in size and aspect ratio on thin faces (nacelle trailing edge) and high curvature (slat and nacelle leading edges).
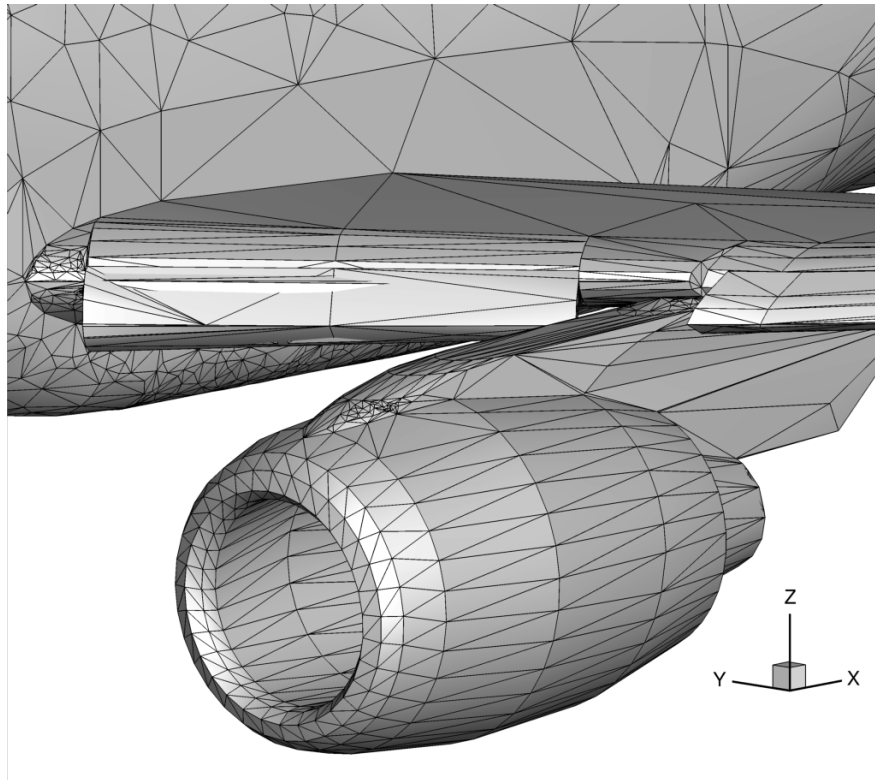
Fig. 13   Nacelle, wing, and fuselage detail of JSM initial coarse EGADS tessellation.

The Mach number is 0.172 for Case 2d of HiLiftPW-3. Reynolds Number based on mean aerodynamic chord is 1.93 million and the static temperature is 551.79 °R. The 10.47° angle of attack case of the drag polar is selected. FUN3D-FV and GGNS use the SA turbulence model. The multiscale metric is used to control estimated interpolation error in

Mach. An intermediate adapted mesh is shown for FUN3D-FV+*refine* (Fig. 15(a)) and GGNS+EPIC (Fig. 15(b)). The multiscale metric based on these turbulent calculations reduced the size of the triangles where the surface is curved to control interpolation error in the boundary layer that follows the curved surface. The surface grid of low-curvature regions (e.g., side of pylon) have larger triangles.

The final adapted meshes in Fig. 16 of the nacelle are highly resolved. The imprint of the slat wake is seen on the side of the fuselage and the imprint of the nacelle wake is seen on the pylon. The mesh is fine enough over most of the curved surfaces to saturate the images. This is an indication that tight model tolerances are required to support this degree of adaptive refinement.

The underside of the wing, pylon, and flap fairings are shown in Fig. 17 for the intermediate adapted mesh and Fig. 18 for the final adapted mesh. As for the nacelle grids, the surface resolution and anisotropy is very similar for the two integrated adaptation processes. This similarity has previously been confirmed in verification studies by Park et al. [5]. An imprint of vortices along the sides of the flap fairings is seen clearly in the GGNS+EPIC adapted surface meshes but not as clearly in the FUN3D-FV+*refine* surface meshes. This may be due to differences in the flow solver discretization or metric construction.

The final adapted meshes for the upper surface of the wing is shown in Fig. 19 for the intermediate adapted mesh and Fig. 20 for the final adapted mesh. Imprints of wakes trailing the slat attachment hardware are seen on the upper surface of the main element. Imprints of wakes trailing the flap attachment hardware are seen on the upper surface of the flap. The slat wakes are more clearly seen in the FUN3D-FV+*refine* intermediate surface meshes than GGNS+EPIC intermediate surface meshes, Fig. 19. The wake imprint topology is different for the inboard flap attachment hardware and the outboard slat attachment hardware in Fig. 20.

Force trajectories are shown in Fig. 21. GGNS+EPIC is shown with circle symbols and FUN3D-FV+*refine* is shown with squares. These trajectories consist of a number of grid adaptations at fixed complexity (a small variation in h approximated by number of the vertices to the minus one-third power). The complexity is increased in a number of steps, which result in step decreases in h. The adapted grid forces are within the scatter of HiLiftPW-3 core participant submissions [27] shown as error bars where lift coefficient submissions were 2.24 ± 0.08 and drag coefficient submissions were 0.273 ± 0.011. The conclusions of the HiLiftPW-3 summary and the range of values across the participant submissions indicate that the community has not obtained solutions and grids in the asymptotic region for this geometry and physical model. Accurate results for the JSM may require larger grids and/or output-based/goal-oriented metrics. However, stalled convergence of the mean flow equations and turbulence model of FUN3D-FV shown in Fig. 22 may impede the convergence of the adjoint or dual system for use in error estimation and metric construction.
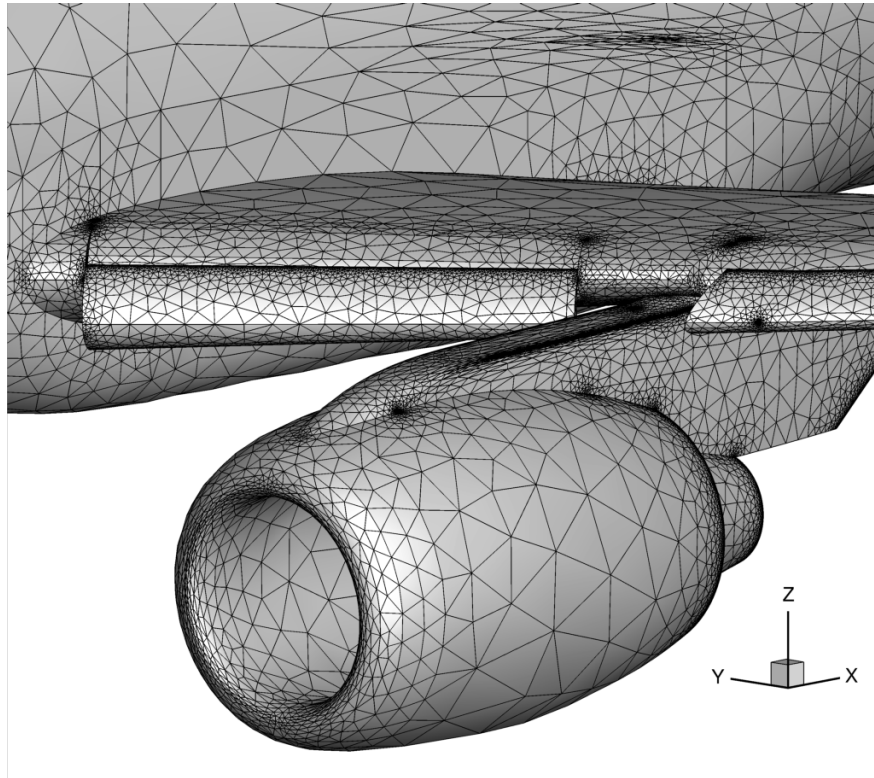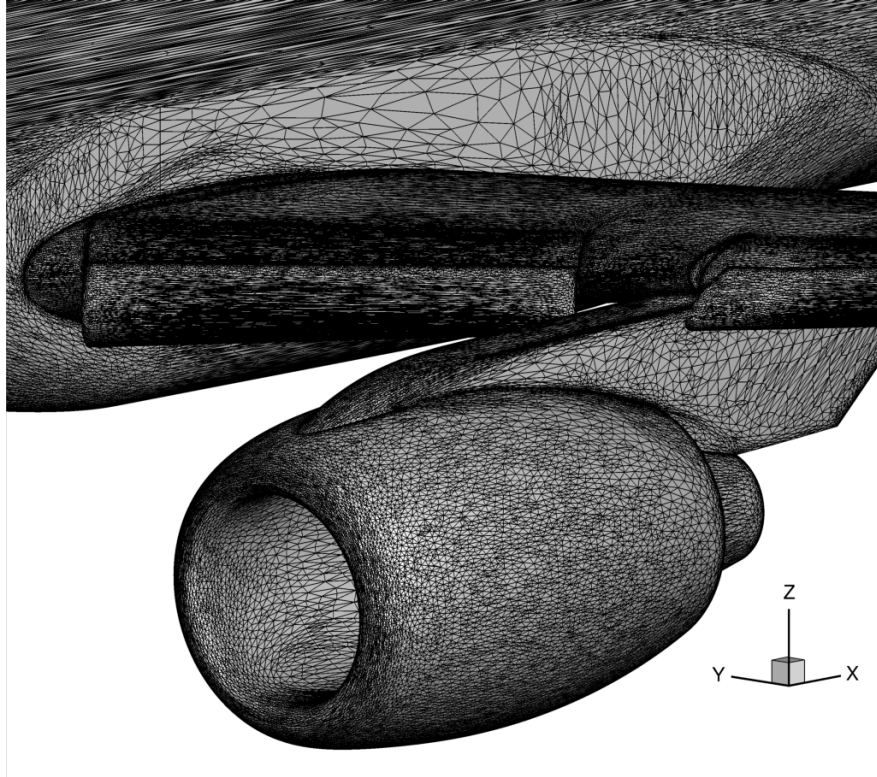
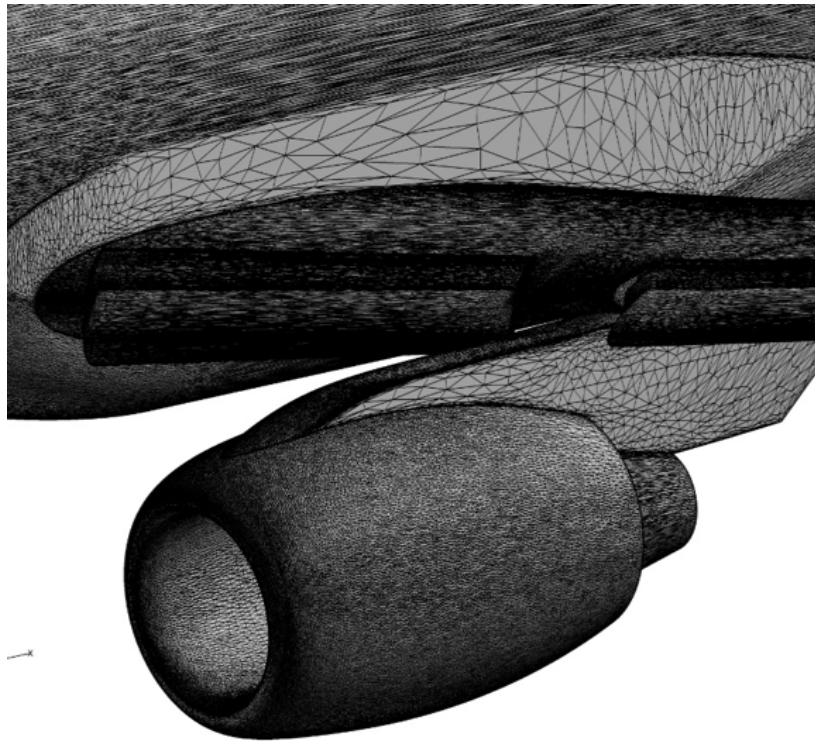(a) Surface mesh adapted to *refine* geometry constraints.
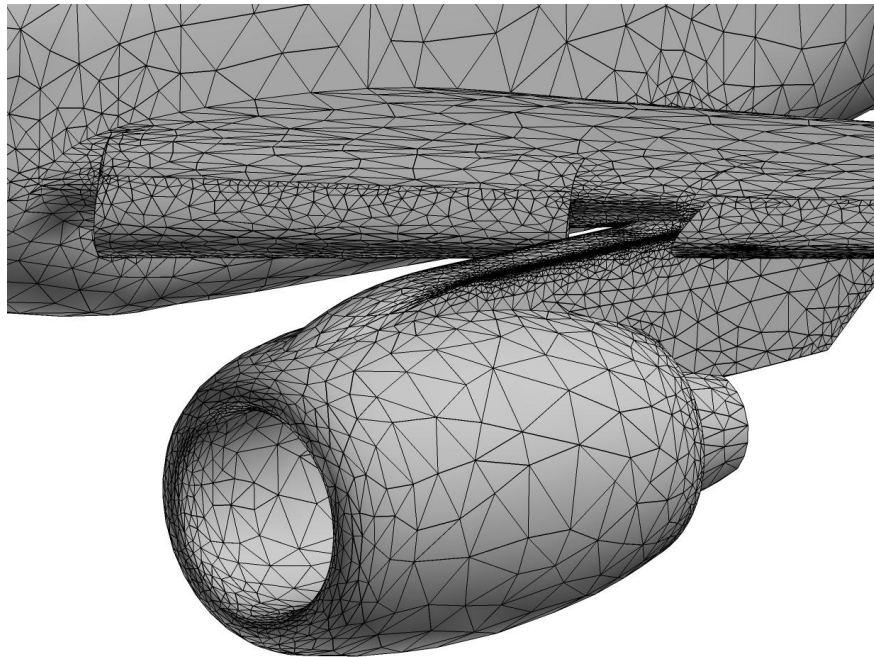
(b) Surface mesh adapted to EPIC geometry constraints

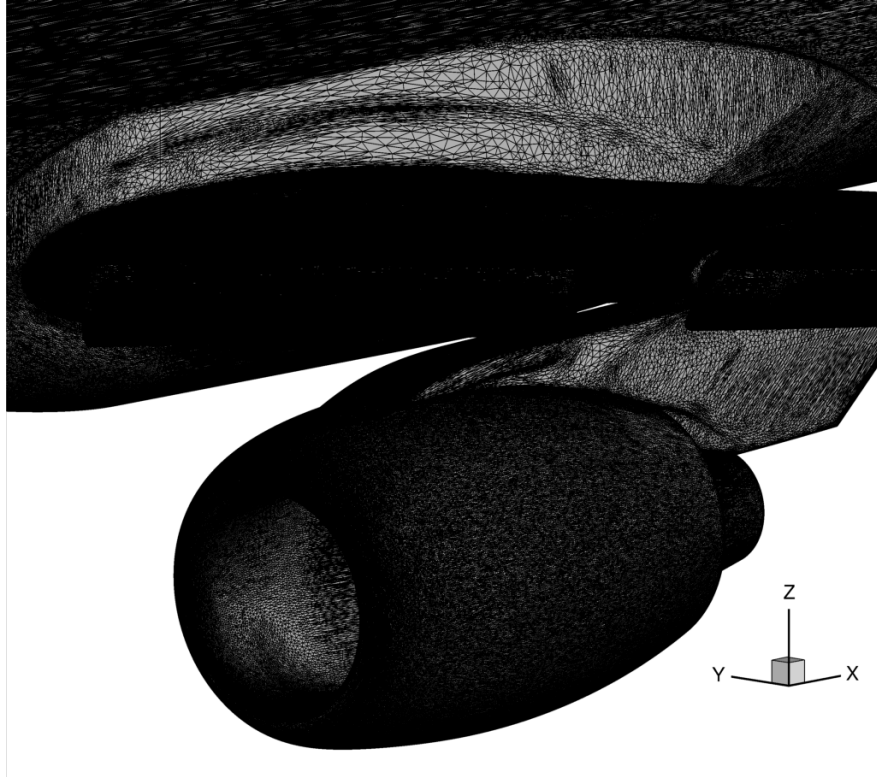**Fig. 14   Nacelle, wing, and fuselage detail of JSM adapted surface meshes.**
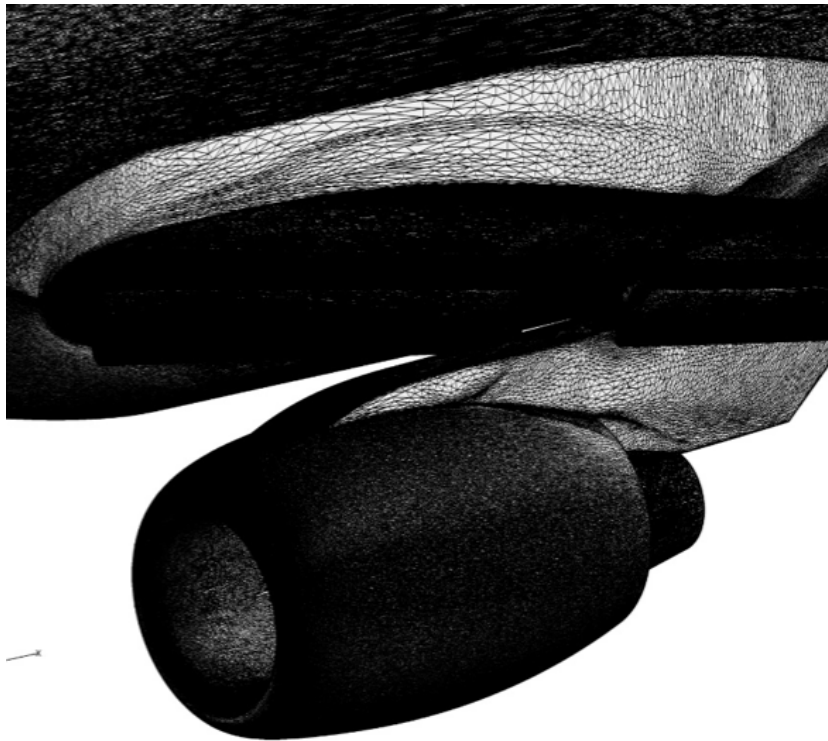
(a) FUN3D-FV+*refine* 4.5M vertices.

(b) GGNS+EPIC 7.4M vertices.

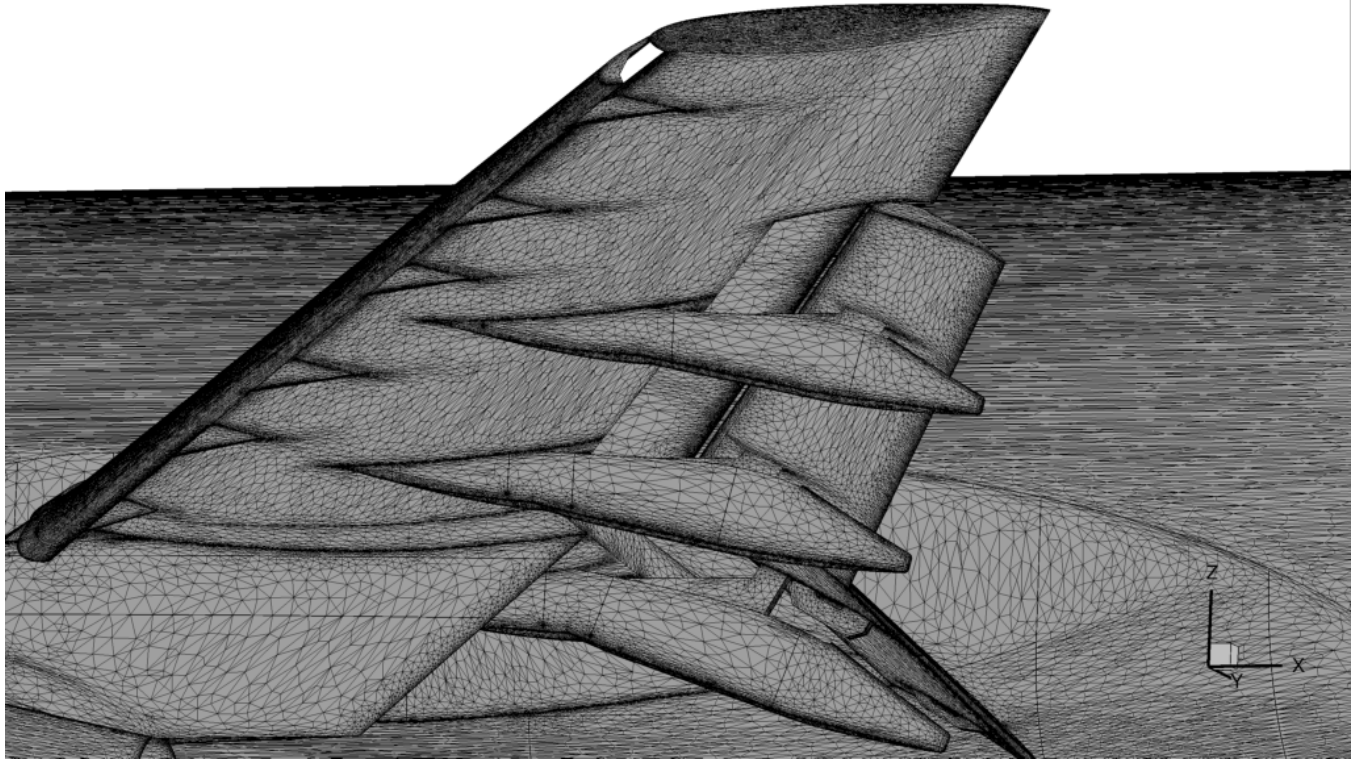**Fig. 15   Nacelle, wing, and fuselage detail of JSM intermediate adapted meshes.**
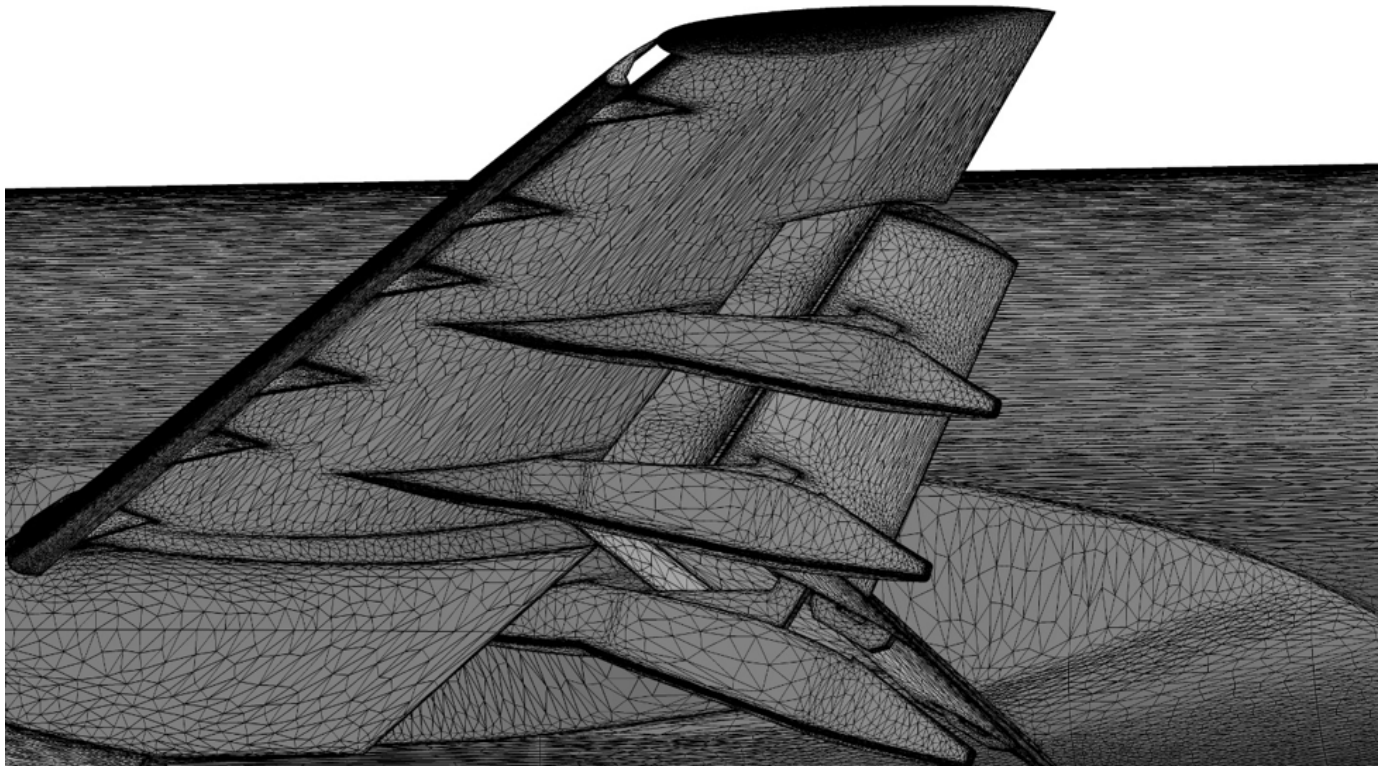
(a) FUN3D-FV+*refine* 32M vertices.

(b) GGNS+EPIC 37M vertices.

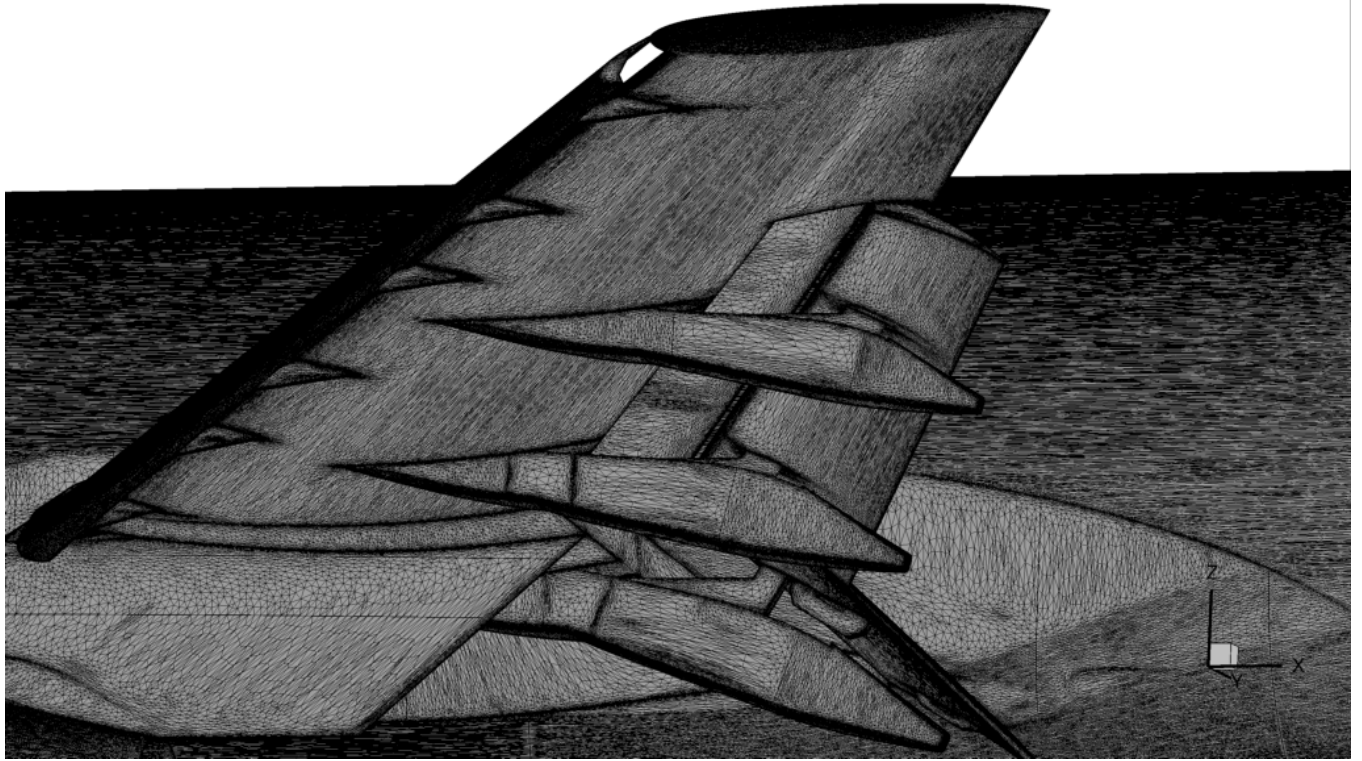**Fig. 16   Nacelle, wing, and fuselage detail of JSM final adapted meshes.**
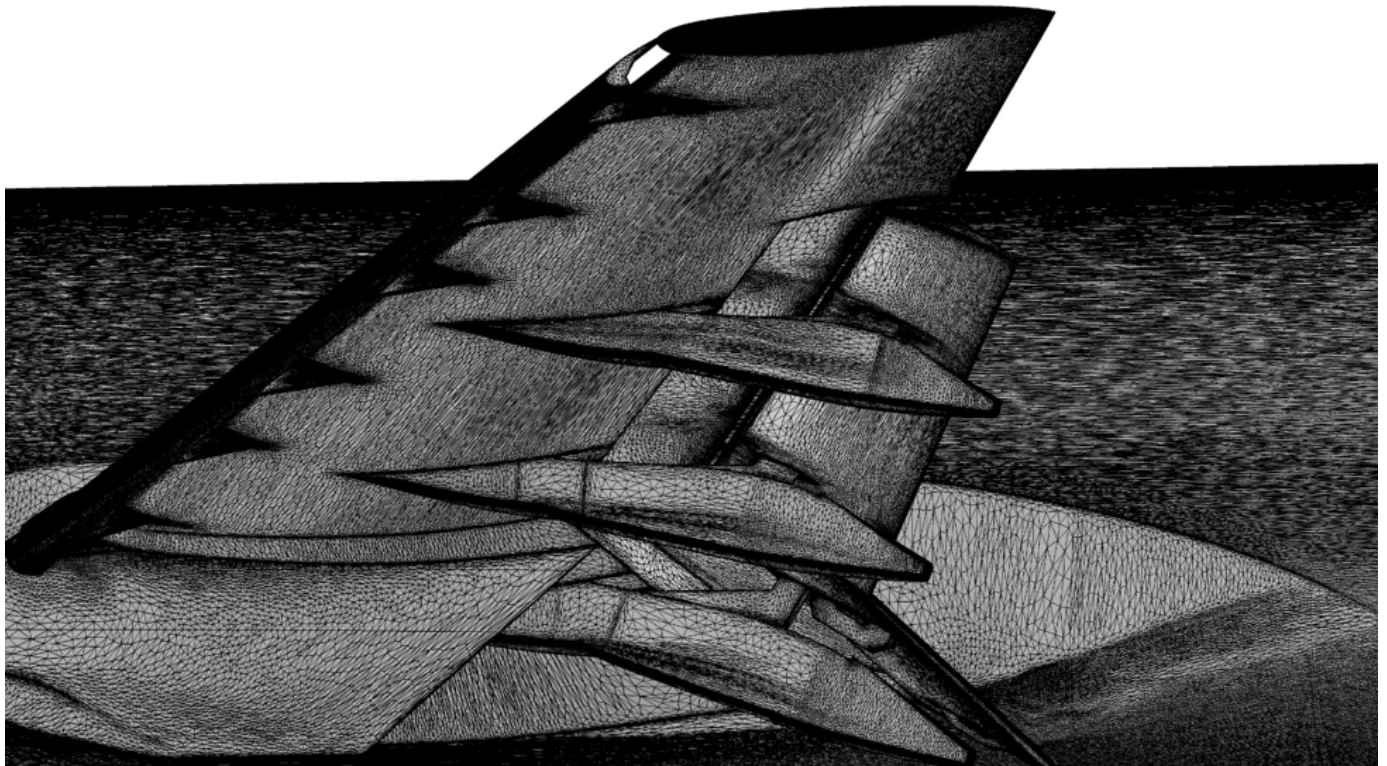
(a) FUN3D-FV+*refine* 4.5M vertices.

(b) GGNS+EPIC 7.4M vertices.

Fig. 17   Lower wing detail of JSM intermediate adapted meshes.
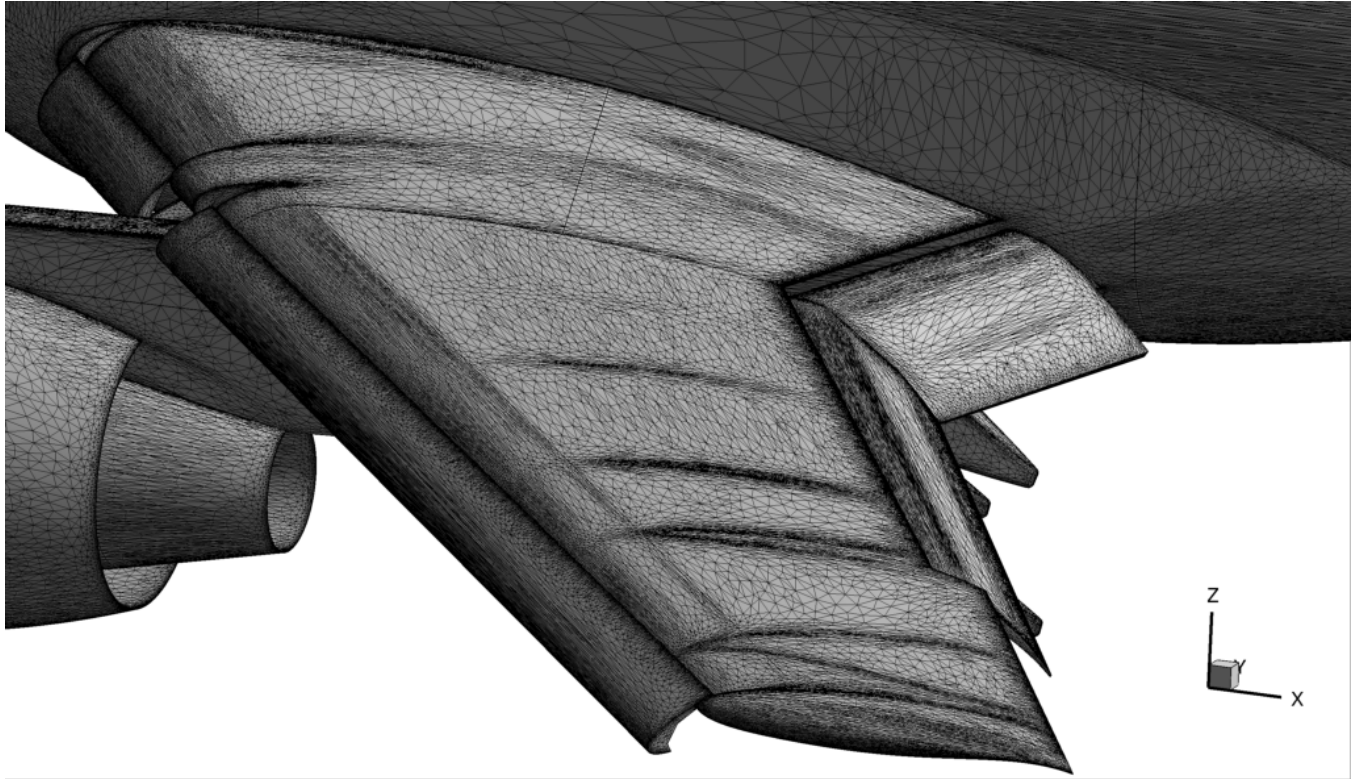
26

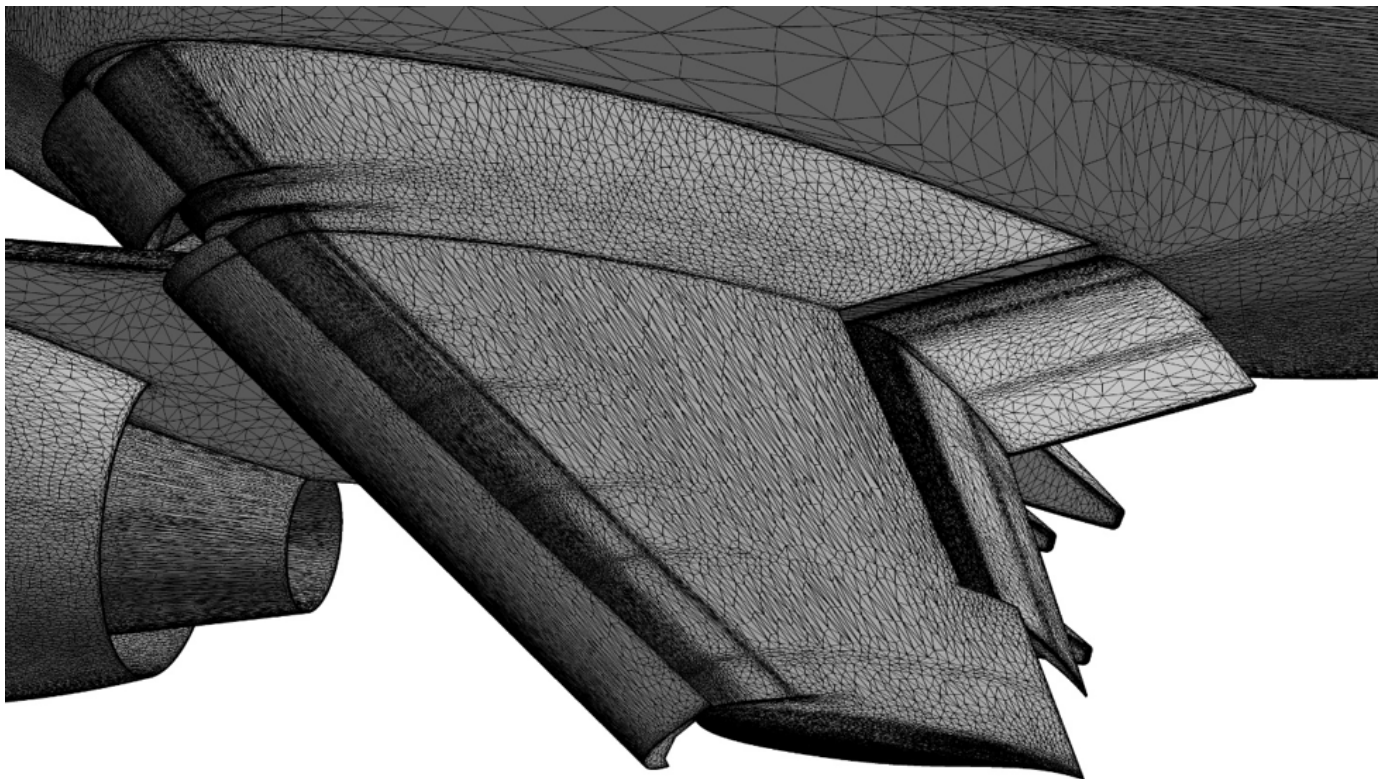(a) FUN3D-FV+*refine* 32M vertices.

(b) GGNS+EPIC 37M vertices.

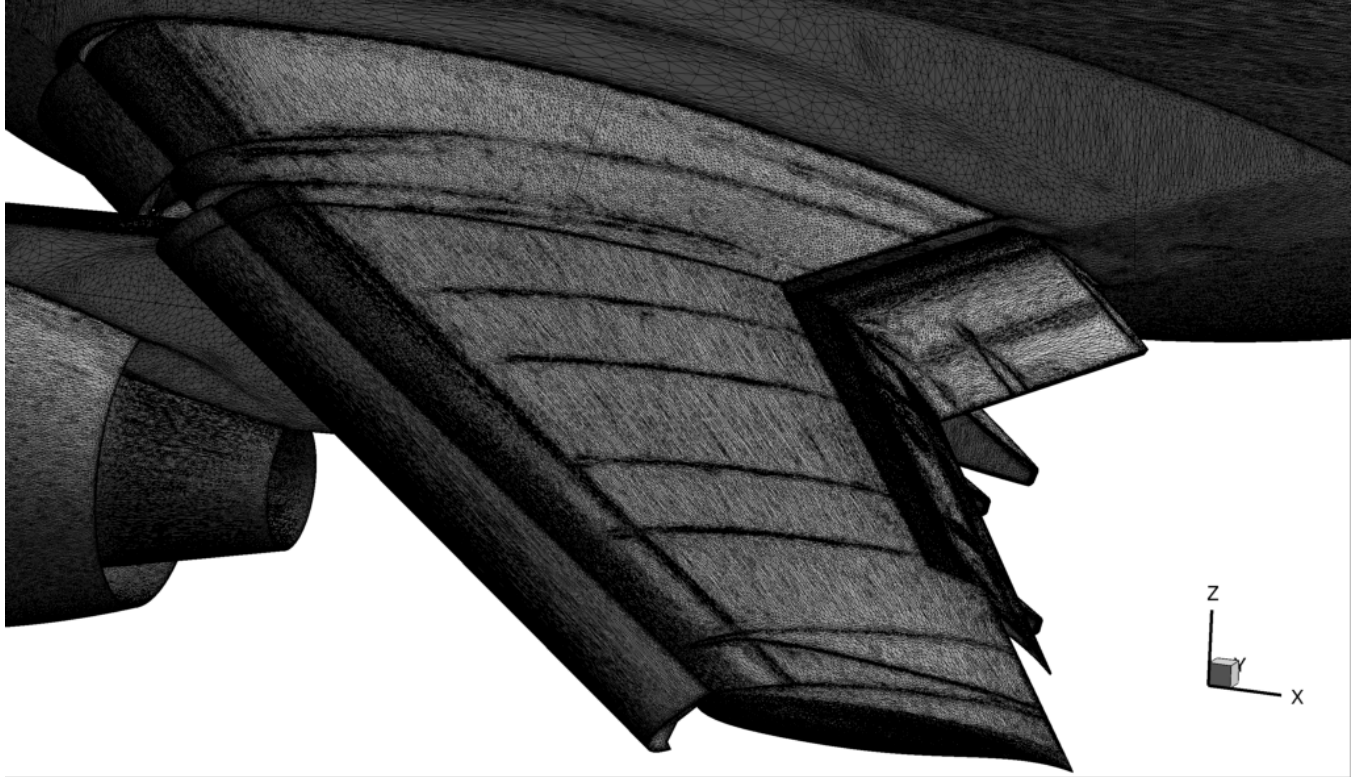**Fig. 18   Lower wing detail of JSM final adapted meshes.**
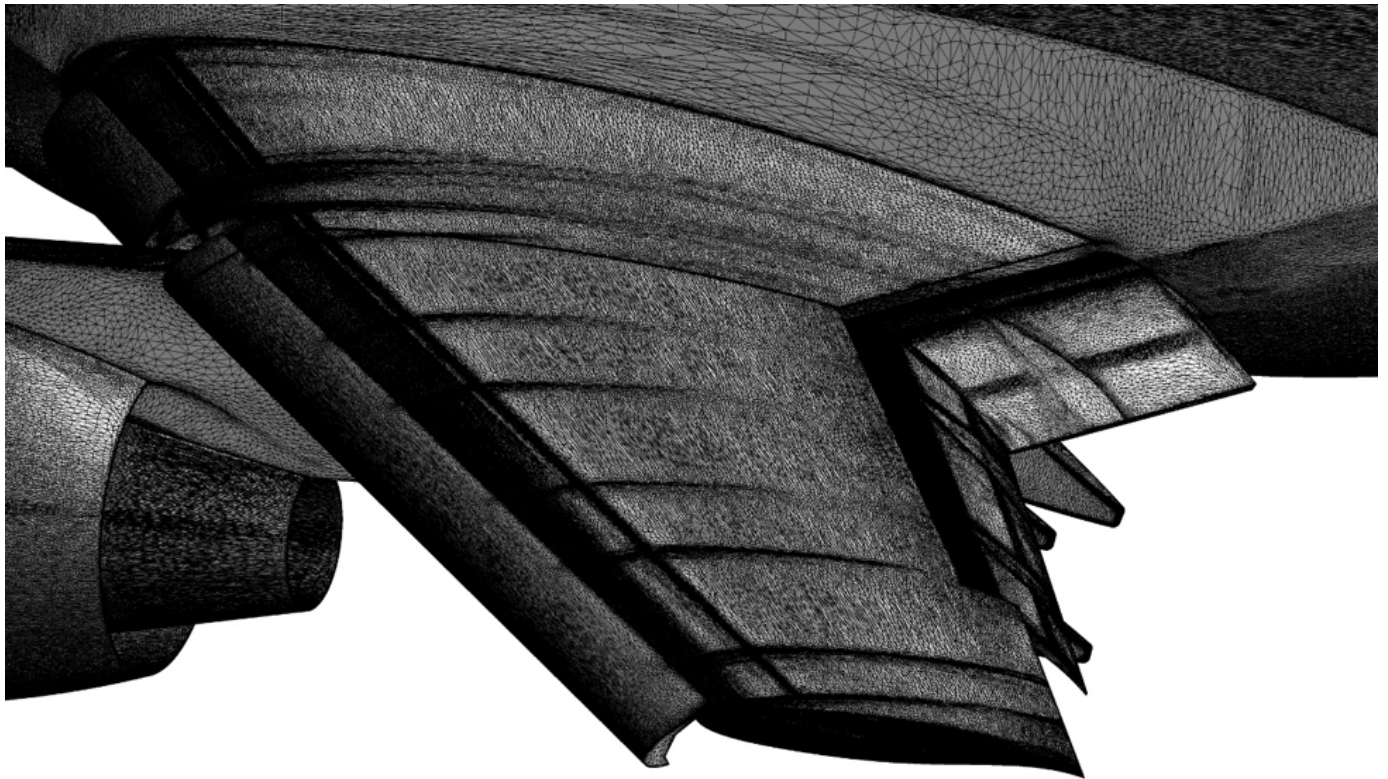
(a) FUN3D-FV+*refine* 4.5M vertices.

(b) GGNS+EPIC 7.4M vertices.

Fig. 19   Upper wing detail of JSM intermediate adapted meshes.
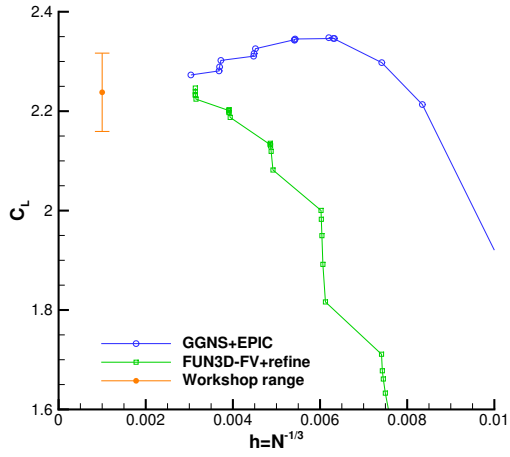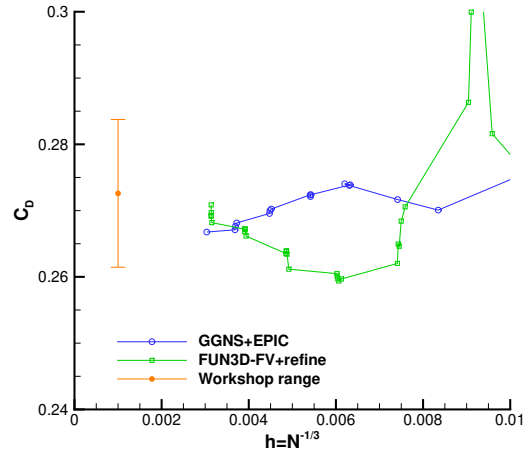
(a) FUN3D-FV+*refine* 32M vertices.

(b) GGNS+EPIC 37M vertices.

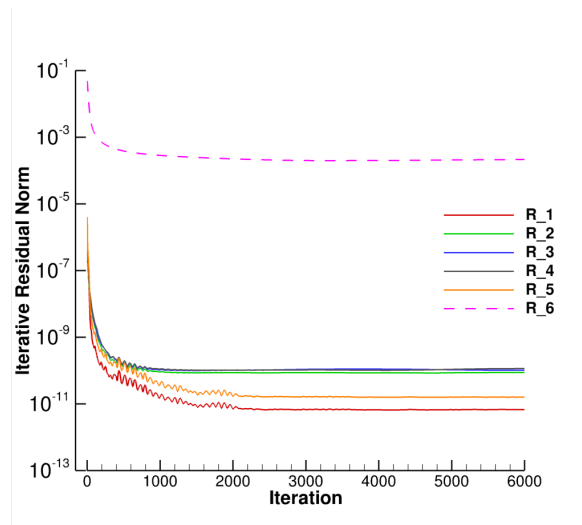**Fig. 20 Upper wing detail of JSM final adapted meshes.**

(a) Lift coefficient convergence.

(b) Drag coefficient convergence.

Fig. 21   JSM forces.

Fig. 22   FUN3D-FV iterative convergence, 32M-vertex adapted grid, initial condition is interpolated solution from previous grid.

VI. Conclusions

A review of unintended geometry construction artifacts that plague Boundary REPresentation (BREP) commonly manipulated through Mechanical Computer-Aided Design (MCAD) systems was presented. Examples of these artifacts [9, 13] included translation, uneven parameterization, degeneracy, self-intersection, sliver faces, gaps, large tolerances between topological elements, and local high curvature to enforce continuity. Surveys of the AIAA Geometry and Mesh Generation Workshop (GMGW-1) participants and users of commercial mesh generation software indicated that these issues are widespread and were routinely addressed through manual processing of input geometry prior to fixed (nonadaptive) meshing. This manual processing entailed eliminating small gaps and unnecessary features with improvements to tolerances and alignment of surfaces. A survey of GMGW-1 participants [13] indicate, "a diverse range of actions were reported, with no evidence of a consistent approach being adopted by any two respondents." Without consistency in repair type or methods, a pessimistic outlook remains for automating this repair task in current workflows. If these unintended geometry construction artifacts can not be repaired in an automated manner, then workflows must be hardened to accept them.

The goal of unstructured grid adaptation is the automation of the labor-intensive fixed-grid generation process. Methods were described that alleviate these classes of unintended geometry construction artifacts. Examples were provided from two AIAA prediction workshops. The NASA Concept 25D with Flow-Through Nacelle (C25F) was used in the Second AIAA Sonic Boom Prediction Workshop (SBPW-2). The Japan Aerospace Exploration Agency Standard Model (JSM) with nacelle was used from Case 2d of the Third AIAA High Lift Prediction Workshop (HiLiftPW-3). Both of these models contain intended and unintended artifacts that challenge fixed and adaptive mesh generation.

Multiscale adaptation proceeded from initial grid generation based on these geometry sources to final adapted grids using two tool chains. The first was OpenCSM, EGADS, *refine*, and FUN3D-FV. The second tool chain was MADCAP, EPIC, and GGNS. Details of the initial grid generation process and constraints imposed to increase the likelihood of success were described. Virtual topology was discussed, but not directly implemented. Surrogate geometry sources that implicitly handle topology was discussed for FEFLO.A and EPIC. The implementation of virtual topology or surrogate geometry into the *refine* tool chain could have eliminated the one point of remaining human interaction, removing a zero area face from the intersection of the JSM nacelle and pylon.

Construction details of initial, intermediate, and final adapted surface grids were presented. Details of the surface grid resolution required by the high-curvature C25F leading edge and JSM slat and flap support hardware was seen. The JSM turbulent flow solution required more surface resolution than the C25F inviscid solution to resolve the interpolation error of the boundary layer attached to curved geometry, and therefore, placed greater demands on geometry model tolerances.

To help focus on the geometry aspects, the study was limited to the multiscale metric to control interpolation error of Mach number. Results presented were within the core submissions for both workshops. Extending the current study to output-based or goal-oriented metrics may improve the consistency of the results by lowering discretization error of workshop requested outputs.

Acknowledgments

References

[1] Alauzet, F., and Loseille, A., "A Decade of Progress on Anisotropic Mesh Adaptation for Computational Fluid Dynamics," *Computer-Aided Design*, Vol. 72, 2016, pp. 13–39. doi:10.1016/j.cad.2015.09.005, 23rd International Meshing Roundtable Special Issue: Advances in Mesh Generation.

[2] Park, M. A., Krakos, J. A., Michal, T., Loseille, A., and Alonso, J. J., "Unstructured Grid Adaptation: Status, Potential Impacts, and Recommended Investments Toward CFD Vision 2030," AIAA Paper 2016–3323, 2016.

[3] Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D., "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences," NASA CR-2014-218178, Langley Research Center, Mar. 2014. doi:2060/20140003093.

[4] Ibanez, D., Barral, N., Krakos, J., Loseille, A., Michal, T., and Park, M., "First Benchmark of the Unstructured Grid Adaptation Working Group," *Procedia Engineering*, Vol. 203, 2017, pp. 154–166. doi:10.1016/j.proeng.2017.09.800, 26th International Meshing Roundtable, IMR26, 18-21 Sept. 2017, Barcelona, Spain.

[5] Park, M. A., Balan, A., Anderson, W. K., Galbraith, M. C., Caplan, P. C., Carson, H. A., Michal, T., Krakos, J. A., Kamenetskiy, D. S., Loseille, A., Alauzet, F., Frazza, L., and Barral, N., "Verification of Unstructured Grid Adaptation Components," AIAA Paper 2019–1723, 2019.

[6] Park, M. A., Barral, N., Ibanez, D., Kamenetskiy, D. S., Krakos, J., Michal, T., and Loseille, A., "Unstructured Grid Adaptation and Solver Technology for Turbulent Flows," AIAA Paper 2018–1103, 2018.

[7] Taylor, N. J., and Haimes, R., "Geometry Modelling: Underlying Concepts and Requirements for Computational Simulation," AIAA Paper 2018–3402, 2018.

[8] Chawner, J. R., Michal, T., Slotnick, J. P., and Rumsey, C. L., "Summary of the 1st AIAA Geometry and Mesh Generation Workshop (GMGW-1) and Future Plans," AIAA Paper 2018–128, 2018.

[9] Gammon, M., Bucklow, H., and Fairey, R., "A Review of Common Geometry Issues Affecting Mesh Generation," AIAA Paper 2018–1402, 2018.

[10] Taylor, N. J., Jones, W. T., and Gammon, M., "Preparation and Analysis of the Geometry Models used in the 1st AIAA Geometry and Mesh Generation Workshop," AIAA Paper 2018–130, 2018.

[11] Taylor, N. J., Gammon, M., and Vassberg, J. C., "The NASA Common Research Model: a Geometry-Handling Perspective," AIAA Paper 2016–3486, 2016.

[12] Taylor, N. J., "Industrial Perspectives on Geometry-Handling for Aerodynamics," AIAA Paper 2015–3408, 2015.

[13] Taylor, N. J., "Analysis of Participant Questionnaires submitted to the 1st AIAA Geometry and Mesh Generation Workshop," AIAA Paper 2018–129, 2018.

[14] Dannenhoffer, J. F., III, and Haimes, R., "Using Quilts and Chains to Improve Structured and Unstructured Surface Grids," AIAA Paper 2004–610, 2004.

[15] Sheffer, A., Bercovier, M., Blacker, T., and Clements, J., "Virtual Topology Operators for Meshing," *International Journal of Computational Geometry & Applications*, Vol. 10, No. 03, 2000, pp. 309–331. doi:10.1142/S0218195900000188.

[16] Dannenhoffer, J. F., III, and Haimes, R., "Quilts: A Technique for Improving Boundary Representations for CFD," AIAA Paper 2003–4132, 2003.

[17] Haimes, R., and Dannenhoffer, J. F., III, "Control of Boundary Representation Topology in Multidisciplinary Analysis and Design," AIAA Paper 2010–1504, 2010.

[18] Jones, W. T., "Toward a Global Parameterization for Quilted CAD Entities," AIAA Paper 2004–611, 2004.

[19] Dannenhoffer, J. F., III, and Haimes, R., "Surface Parameterization of 3D Configurations via Quilts," AIAA Paper 2005–5238, 2005.

[20] Gammon, M. R., Fellows, C. D., and Whyman, S. H., "Enabling Virtual Topology for High Quality CFD Surface Meshing of Complex CAD Geometry," AIAA Paper 2018–3721, 2018.

[21] Sun, L., Robinson, T. T., Armstrong, C. G., Marques, S., and Ya, W., "Surface Mesh Deformation in CAD-based Shape Optimization," AIAA Paper 2019–2360, 2019.

[22] Karman, S. L., and Wyman, N., "Automatic Unstructured Mesh Generation with Geometry Attribution," AIAA Paper 2017–1721, 2019.

[23] Michal, T. R., Kamenetskiy, D. S., Krakos, J., Mani, M., Glasby, R. S., Erwin, T., and Stefanski, D., "Comparison of Fixed and Adaptive Unstructured Grid Results for Drag Prediction Workshop 6," *AIAA Journal of Aircraft*, Vol. 55, No. 4, 2018, pp. 1420–1432. doi:10.2514/1.C034491.

[24] Michal, T., Krakos, J., and Kamenetskiy, D., "Generation of Anisotropic Adaptive Meshes for the First AIAA Geometry and Mesh Generation Workshop," AIAA Paper 2018–658, 2018.

[25] Michal, T., Kamenetskiy, D., and Krakos, J., "Anisotropic Adaptive Mesh Results for the Third High Lift Prediction Workshop (HiLiftPW-3)," AIAA Paper 2018–1257, 2018.

[26] Park, M. A., and Nemec, M., "Near Field Summary and Statistical Analysis of the Second AIAA Sonic Boom Prediction Workshop," *AIAA Journal of Aircraft*, 2018. doi:10.2514/1.C034866.

[27] Rumsey, C. L., Slotnick, J. P., and Sclafani, A. J., "Overview and Summary of the Third AIAA High Lift Prediction Workshop," *AIAA Journal of Aircraft*, Vol. 56, No. 2, 2019, pp. 621–644. doi:10.2514/1.C034940.

[28] Loseille, A., and Alauzet, F., "Continuous Mesh Framework Part I: Well-Posed Continuous Interpolation Error," *SIAM Journal on Numerical Analysis*, Vol. 49, No. 1, 2011, pp. 38–60. doi:10.1137/090754078.

[29] Park, M. A., "Anisotropic Output-Based Adaptation with Tetrahedral Cut Cells for Compressible Flows," Ph.D. thesis, Massachusetts Institute of Technology, Sep. 2008. doi:1721.1/46363.

[30] Loseille, A., and Löhner, R., "Cavity-Based Operators for Mesh Adaptation," AIAA Paper 2013–152, 2013.

[31] Alauzet, F., "A Changing-Topology Moving Mesh Technique for Large Displacements," *Engineering with Computers*, Vol. 30, No. 2, 2014, pp. 175–200. doi:10.1007/s00366-013-0340-z.

[32] Freitag, L. A., and Ollivier-Gooch, C., "Tetrahedral Mesh Improvement Using Swapping and Smoothing," *International Journal for Numerical Methods in Engineering*, Vol. 40, 1997, pp. 3979–4002. doi:10.1002/(SICI)1097-0207(19971115)40:21<3979::AID-NME251>3.0.CO;2-9.

[33] Michal, T., and Krakos, J., "Anisotropic Mesh Adaptation Through Edge Primitive Operations," AIAA Paper 2012–159, 2012.

[34] Loseille, A., Alauzet, F., and Menier, V., "Unique Cavity-Based Operator and Hierarchical Domain Partitioning for Fast Parallel Generation of Anisotropic Meshes," *Computer-Aided Design*, Vol. 85, 2017, pp. 53–67. doi:10.1016/j.cad.2016.09.008, 24th International Meshing Roundtable Special Issue: Advances in Mesh Generation.

[35] Haimes, R., and Drela, M., "On The Construction of Aircraft Conceptual Geometry for High-Fidelity Analysis and Design," AIAA Paper 2012–683, 2012.

[36] Haimes, R., and Dannenhoffer, J. F., III, "The Engineering Sketch Pad:' A Solid-Modeling, Feature-Based, Web-Enabled System for Building Parametric Geometry," AIAA Paper 2013–3073, 2013.

[37] Dannenhoffer, J. F., III, "OpenCSM: An Open-Source Constructive Solid Modeler for MDAO," AIAA Paper 2013–701, 2013.

[38] Haimes, R., and Dannenhoffer, J. F., III, "EGADSlite: A Lightweight Geometry Kernel for HPC," AIAA Paper 2018–1401, 2018.

[39] Nagata, T., "Simple Local Interpolation of Surfaces using Normal Vectors," *Computer Aided Geometric Design*, Vol. 22, No. 4, 2005, pp. 327–347. doi:10.1016/j.cagd.2005.01.004.

[40] Aubry, R., Karamete, K., Mestreau, E., and Dey, S., "Singularities in Parametric Meshing," *Proceedings of the 21st International Meshing Roundtable*, edited by X. Jiao and J.-C. Weill, Sandia National Laboratories, Springer Berlin Heidelberg, 2013, pp. 225–241. doi:10.1007/978-3-642-33573-0_14.

[41] Alauzet, F., "Size Gradation Control of Anisotropic Meshes," *Finite Elements in Analysis and Design*, Vol. 46, No. 1–2, 2010, pp. 181–202. doi:10.1016/j.finel.2009.06.028.

[42] Marcum, D. L., and Weatherill, N. P., "Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection," *AIAA Journal*, Vol. 33, No. 9, 1995, pp. 1619–1625. doi:10.2514/3.12701.

[43] Si, H., "TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 41, No. 2, 2015, pp. 11:1–11:36. doi:10.1145/2629697.

[44] Laug, P., and Borouchaki, H., "BLSURF - Mesh Generator for Composite Parametric Surfaces - User's Manual," INRIA Report RT-0215, May 2006.

[45] George, P. L., Borouchaki, H., and Saltel, E., "'Ultimate' Robustness in Meshing an Arbitrary Polyhedron," *International Journal for Numerical Methods in Engineering*, Vol. 58, No. 7, 2003, pp. 1061–1089. doi:10.1002/nme.808.

[46] Anderson, W. K., and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1–22. doi:10.1016/0045-7930(94)90023-X.

[47] Biedron, R. T., Carlson, J.-R., Derlaga, J. M., Gnoffo, P. A., Hammond, D. P., Jones, W. T., Kleb, B., Lee-Rausch, E. M., Nielsen, E. J., Park, M. A., Rumsey, C. L., Thomas, J. L., and Wood, W. A., "FUN3D Manual: 13.3," NASA TM-2018-219808, Langley Research Center, Feb. 2018. doi:2060/20180001971.

[48] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372. doi:10.1016/0021-9991(81)90128-5.

[49] van Leer, B., "Towards the Ultimate Conservative Difference Scheme. II. Monotonicity and Conservation Combined in a Second-Order Scheme," *Journal of Computational Physics*, Vol. 14, No. 4, 1974, pp. 361–370. doi:10.1016/0021-9991(74)90019-9.

[50] Nielsen, E. J., Lu, J. C.-C., Park, M. A., and Darmofal, D. L., "An Implicit, Exact Dual Adjoint Solution Method for Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 33, No. 9, 2004, pp. 1131–1155.

[51] Spalart, P. R., and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," *La Recherche Aérospatiale*, Vol. 1, 1994, pp. 5–21.

[52] Bonet, J., and Peraire, J., "An Alternating Digital Tree (ADT) Algorithm for 3D Geometric Searching and Intersection Problems," *International Journal for Numerical Methods in Engineering*, Vol. 31, No. 1, 1991, pp. 1–17.

[53] Allmaras, S. R., "Lagrange Multiplier Implementation of Dirichlet Boundary Conditions in Compressible Navier-Stokes Finite Element Methods," AIAA Paper 2005-4714, 2005.

[54] Kamenetskiy, D. S., Bussoletti, J. E., Hilmes, C. L., Venkatakrishnan, V., and Wigton, L. B., "Numerical Evidence of Multiple Solutions for the Reynolds-Averaged Navier-Stokes Equations," *AIAA Journal*, Vol. 52, No. 8, 2014, pp. 1686–1698. doi:10.2514/1.J052676.

[55] Abhyankar, S., Brown, J., Constantinescu, E. M., Ghosh, D., Smith, B. F., and Zhang, H., "PETSc/TS: A Modern Scalable ODE/DAE Solver Library," *Computing Research Repository (CoRR)*, Vol. Numerical Analysis (math.NA), No. arXiv:1806.01437, 2018. URL http://arxiv.org/abs/1302.6066.

[56] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Rupp, K., Smith, B. F., Zampini, S., Zhang, H., and Zhang, H., "PETSc Users Manual," Tech. Rep. ANL-95/11 - Revision 3.8, Argonne National Laboratory, 2017.

[57] Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F., "Efficient Management of Parallelism in Object Oriented Numerical Software Libraries," *Modern Software Tools in Scientific Computing*, edited by E. Arge, A. M. Bruaset, and H. P. Langtangen, Birkhäuser Press, 1997, pp. 163–202.

[58] Saad, Y., *Iterative Methods for Sparse Linear Systems*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

[59] Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C., "Adaptive Remeshing for Compressible Flow Computations," *Journal of Computational Physics*, Vol. 72, No. 2, 1987, pp. 449–466. doi:10.1016/0021-9991(87)90093-3.

[60] Peraire, J., Peirò, J., and Morgan, K., "Adaptive Remeshing for Three-Dimensional Compressible Flow Computations," *Journal of Computational Physics*, Vol. 103, No. 2, 1992, pp. 269–285. doi:10.1016/0021-9991(92)90401-J.

[61] Alauzet, F., and Loseille, A., "High-Order Sonic Boom Modeling Based on Adaptive Methods," *Journal of Computational Physics*, Vol. 229, No. 3, 2010, pp. 561–593. doi:10.1016/j.jcp.2009.09.020.

[62] Arsigny, V., Fillard, P., Pennec, X., and Ayache, N., "Log-Euclidean Metrics for Fast and Simple Calculus on Diffusion Tensors," *Magnetic Resonance in Medicine*, Vol. 56, No. 2, 2006, pp. 411–421. doi:10.1002/mrm.20965.

[63] Wintzer, M., Ordaz, I., and Fenbert, J. W., "Under-Track CFD-Based Shape Optimization for a Low-Boom Demonstrator Concept," AIAA Paper 2015–2260, 2015.

[64] Ordaz, I., Wintzer, M., and Rallabhandi, S. K., "Full-Carpet Design of a Low-Boom Demonstrator Concept," AIAA Paper 2015–2261, 2015.

[65] Anderson, G. R., Aftosmis, M. J., and Nemec, M., "Cart3D Simulations for the Second AIAA Sonic Boom Prediction Workshop," *AIAA Journal of Aircraft*, 2018. doi:10.2514/1.C034842.

[66] Wintzer, M., "Optimization and Adjoint-Based CFD for the Conceptual Design of Low Sonic Boom Aircraft," Ph.D. thesis, Stanford University, Aug. 2012.

[67] Derlaga, J. M., Park, M. A., and Rallabhandi, S. K., "Application of Exactly Linearized Error Transport Equations to Sonic Boom Prediction Workshop," *AIAA Journal of Aircraft*, 2018. doi:10.2514/1.C034841.