

# An Automated Marching Scheme for Overset Structured Surface Mesh Generation

Shishir A. Pandya\*, William M. Chan\* and Robert Haines\*\*

Corresponding author: Shishir.Pandya@nasa.gov

\* NASA Ames Research Center, USA; \*\* Massachusetts Institute of Technology, USA.

**Abstract:** Starting with a Boundary Representation (BRep) of the geometry of an aerospace vehicle, an automated marching scheme is presented for generation of structured overset surface meshes. First, a pre-processing step automatically generates discrete representations of the BRep faces and BRep edges by tessellating in parameter space. Topological connectivity between the discretized BRep edges is then established, followed by automatic grid point distribution on these edges based on local turning angle, proximity to sharp geometric features, and prescribed maximum stretching ratio and grid spacing. A set of initial curves for algebraic or hyperbolic marching on a surface is then derived from the redistributed edge curves. A spatially-variable marching distance together with a grid point distribution in the marching direction are automatically determined for each initial curve. A set of overset surface meshes that covers the entire geometry is then obtained by combining the surface meshes around the BRep edges, and the structured meshes derived from the discretized BRep faces.

*Keywords:* Overset Grids, Surface Grid Generation.

## 1 Introduction

Structured overset grid technology has been successfully applied to a wide range of complex geometry configurations [1, 2, 3, 4, 5, 6, 7]. The flow solution procedure [8, 9] is highly efficient compared to other grid strategies especially in moving-body and rotational settings. However, the grid generation step typically requires expert knowledge and significant manual effort to produce good quality grids. The overset grid generation step consists of three parts: surface grid generation, volume grid generation, and domain connectivity. It is widely recognized that surface grid generation is by far the most difficult task requiring laborious human interaction and expertise [10, 11].

Overset structured surface mesh generation typically involves the following steps: (1) The surface domain is decomposed into four-sided regions bounded by one, two, three, or four surface features such as sharp edges, leading edge curves, and open boundaries. (2) Curves following these surface features are discretized with a grid point distribution governed by multiple constraints. These include appropriate resolution of high curvature regions and sharp convex edges, maximum prescribed grid spacings, and low stretching ratios in each direction to keep truncation error in check. (3) Surface meshes are typically created using a combination of algebraic (transfinite interpolation) or hyperbolic methods depending on the number of initial curves on each surface domain.

Early efforts to reduce the manual work in generating overset structured surface meshes involves a scripting approach[12] which has several advantages. Geometry and mesh inputs such as maximum stretching ratio and grid spacings can be easily parameterized. The entire process is recorded in a script, and is thus repeatable. Mesh modifications are simply a minor edit of the script provided there are no topological changes in the geometry. However, scripting also has disadvantages. Each new geometry requires significant development time to build the initial script, and topological geometric changes also demand non-trivial extra effort in script modifications. In order to utilize overset grid technology in increasingly complex and rapidly changing geometries as in a design environment, it is clear that an automated surface mesh generation scheme

that is capable of quickly creating surface meshes on topologically distinct new geometries is needed.

The scheme described in this paper is based on earlier ideas on automation of structured overset surface mesh generation around surface features or seam curves[13]. There are multiple improvements in the current scheme over the earlier work. The current scheme begins with an analytic Boundary Representation (BRep)[14] geometry definition while the previous work begins with a discrete unstructured surface triangulation. The marching scheme in the current work is more flexible in conforming to geometric features since it allows either algebraic or hyperbolic methods, while the previous scheme only allows hyperbolic marching. For example, when marching from the root of a swept wing towards the wing tip, it is more appropriate to use algebraic marching which follows the swept isoparametric lines which are typically also constant curvature lines of the wing. In contrast, a standard hyperbolic method would generate grid lines orthogonal to the wing root and not follow the local surface curvature. Grid point distribution in the current scheme is automatically determined while it has to be manually prescribed in the previous scheme. The marching distance is spatially variable and automatically determined in the current scheme while it is constant and manually specified in the previous scheme. In the previous method, a singular point topology is applied at a seam point, a vertex where multiple seam curves meet. In the current procedure, a non-singular patch grid is constructed around a seam point to take solver limitations into consideration. A patch topology allows an implicit solution procedure with a vertex-based scheme through the seam point rather than an explicit boundary condition update as is imposed if a singular point topology is used.

The current work begins with a pre-processing step that converts the analytic BRep solids from a STEP or IGES file into discrete representations for the BRep faces and edges. Details of this process are described in Section 2. Extraction of an initial curve set from the BRep edges for use as starting curves for either an algebraic or hyperbolic marching scheme is presented in Section 3. The algorithm used to determine the choice of algebraic versus hyperbolic marching for a given initial curve is described in Section 4. Details of the scheme for computing the variable marching distance, and grid point distribution in the marching direction are provided in Section 5. The surface mesh generation procedure is presented in Section 6. This paper will focus primarily on the decomposition of the surface geometry into appropriate overlapping quadrilateral domains. A more sophisticated grid point distribution scheme with improved overlap between neighboring meshes is deferred to a future paper. Test cases on application of the current scheme to various aerospace conceptual design geometries are described in Section 7, followed by a summary and concluding remarks in Section 8. For the remainder of this paper, BRep faces and BRep edges will be referred to simply as Edges and Faces.

## 2 Geometry Definition Pre-processing

The current scheme assumes that the geometry definition is available as a BRep solid model described in a STEP or IGES standard file. Access to the model topology and entities is accomplished through the EGADS API which is a foundational component of the Engineering Sketch Pad (ESP) [15]. The EGADS library contains a rich set of functions for interrogation and evaluation of the BRep geometry model which has a dependency on OpenCASCADE for construction and standard file IO. Moreover, it is freely available and downloadable under an open source license.

As a pre-processor to the automation scheme, discrete representations of various geometric entities are generated from each solid. This part of the process is currently performed by a tool called *egads2srf* which accepts three simple input control parameters for the grid cell attributes: the maximum dihedral angle between adjacent cells, the maximum edge length of a cell, and the maximum deviation between the cell center and the analytic geometry. Structured grid and function files generated by *egads2srf* are in PLOT3D[16] format, while unstructured triangulation grid and function files are in Cart3D[17, 18] format.

These files include the following:

1. A structured surface grid file containing an untrimmed structured patch for each Face based on tessellation of the Face parameter space (directions indicated by  $u$  and  $v$ ). An integer tag (iblack) at each vertex is used to identify whether the vertex lies inside (iblack=0) or outside (iblack=1) of the trimmed region (.uvx file, Fig.1a,b)
2. A curve grid file containing tessellated Edges (.cur file, Fig. 1c).

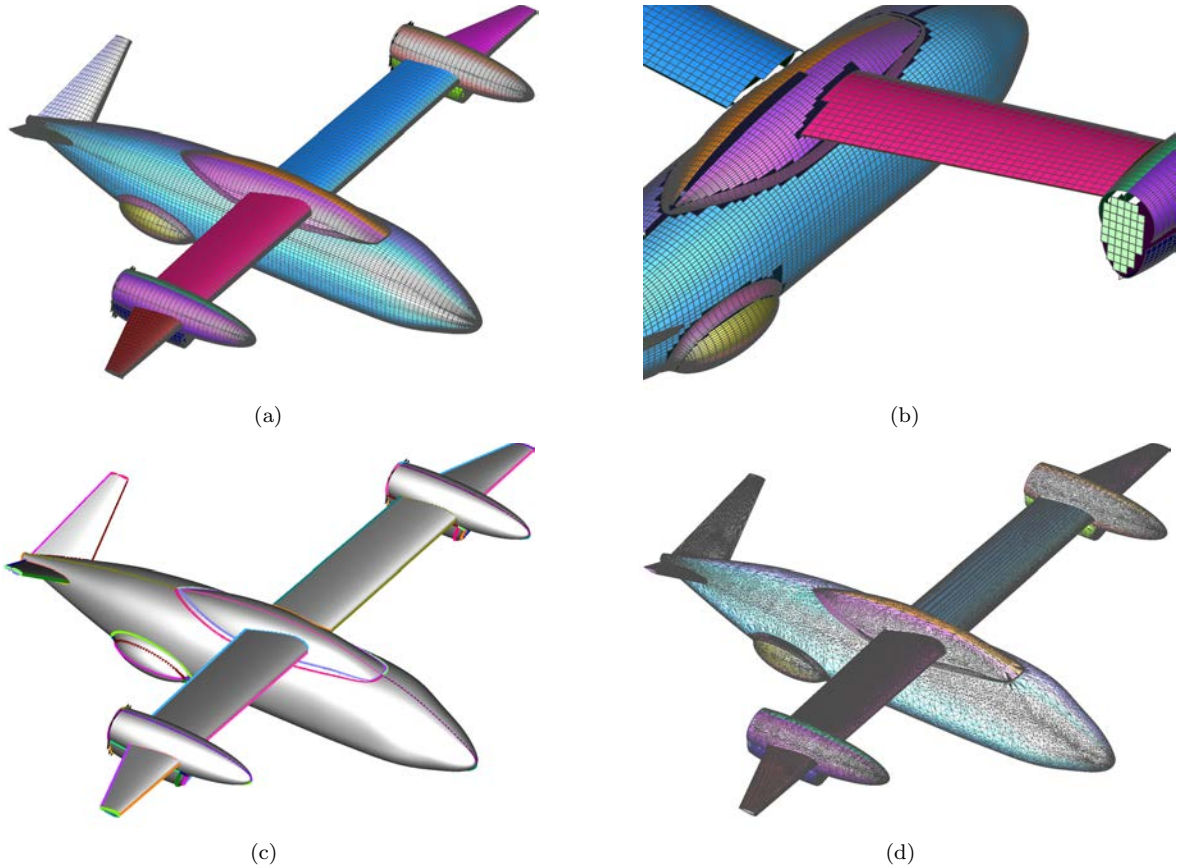


Figure 1: Discrete geometry representation from pre-processing of solids from STEP files. (a) Structured surface mesh patches from tessellation of  $u, v$  parameter space in global view. (b) Structured surface mesh patches from tessellation of  $u, v$  parameter space showing blanking of vertices in trimmed regions. (c) Structured curves representing tessellation of the Edges. (d) Unstructured surface triangulation where each trimmed Face is shown in a different color.

3. A structured grid function file containing  $u, v$  parameter space values for each vertex in the `.uvx` file (`.uv` file).
4. A structured grid function file containing the  $u, v$  parameter space values for each vertex in the `.cur` file (`.cuv` file).
5. An unstructured surface triangulation containing the trimmed Faces (`.i.tri` file, Fig. 1d).
6. A topology file that contains the identification indices of the Faces adjacent to each Edge, as well as the directional sense of the Edge on each associated Face. Under the BRep solid model framework, each interior Edge is guaranteed to have exactly two adjacent Faces, one on each side of the Edge. The directional sense of an Edge on a Face is the direction along an Edge where untrimmed geometry is identified on the left side (Fig. 2).

### 3 Initial Curve Set Preparation for Grid Marching

The surface mesh generation automation scheme proposed for the current work involves algebraic or hyperbolic marching. For both cases, a mesh is created by marching from an initial curve. The first step is to establish an initial curve set by determining the topological connectivity between the curves, followed by placing an appropriate grid point distribution on these curves. These point-redistributed curves are then trimmed and concatenated to form the initial curve set.

On processing a BRep solid, EGADS is able to extract information on the topological connectivity between the Edges. However, an independent topological connectivity between the Edges stored in the `.cur` file is established in the current work. This independent step allows the possibility of starting from a set of discrete geometry representation files (listed at the end of the previous section) that is not necessarily created by ESP. Each end point of each curve is coincident with one or more other end points from other curves. The vertex where multiple curve end points meet is called a seam point (BRep node). The valence of a seam point is defined to be the number of curve end points that meet there. For each seam point, the valence and the list of curve end points that meet at the seam point are automatically determined. For each curve end point, the identity of the associated seam point is then found. Fig. 3 shows several valence 3 and 4 seam points.

Heuristic rules based on the attributes of the Edge curves are used to determine the grid point distribution on these curves. Attributes considered include the total curve arc length relative to the characteristic length of the solid, total and local curve turning angles. The grid spacing at a curve end point is determined by whether it falls on a concave or convex sharp edge of the geometry, as well as the concave/convex angle of the sharp edge. The resulting grid point distribution needs to provide sufficient resolution in steep flow gradient regions which include high curvature locations such as wing leading edges, as well as sharp convex features such as wing trailing edges. Details of the grid point redistribution scheme are described in Ref. [19].

One of the most difficult steps in structured surface mesh generation is surface domain decomposition, the task of partitioning the surface domain into quadrilateral regions for the creation of structured meshes. Sharp geometric features defined by some of the Edges need to be captured by the surface meshes. For the overset grid approach, constraints on the surface meshes are not as severe as the abutting grid approach since neighboring meshes are allowed to overlap arbitrarily. However, sharp surface features should still be resolved well by appropriate grid lines from neighboring meshes. The current work follows a similar automated surface domain decomposition strategy outlined in Ref. [13] except at the seam points.

Two groups of curves contribute to the initial curve set for algebraic or hyperbolic marching. Curves in the two groups are constructed to guarantee  $2 \times N_f + 1$  points overlap between neighboring meshes where  $N_f$  is the number of fringe points at the outer and hole boundaries of a grid where interpolation of dependent variables of the flow solution occurs. For example,  $N_f$  is set to 2 for a flow solver with a five-point differencing stencil which guarantees that the first interior solution point is supported by such a stencil. The first group is created by trimming each of the original redistributed Edge curves (Fig. 4a) away from the seam points (Fig. 4b). The second group is formed by constructing an initial curve that straddles each seam point. Let  $N_c$  be the number of curves that meet at the seam point. From this set of curves, two are selected to be concatenated to serve as the initial curve for constructing a marching grid from the seam point.

For  $N_c = 3$ , a special test is designed to ensure appropriate grid topology near wing trailing edges in the root and tip regions, as well as in other regions away from sharp corners. The angle between each pair of curves ending at the seam point is computed. If the largest angle is greater than 120 degrees, the two curves adjacent to that angle are concatenated (Fig. 5a). Otherwise, the two curves adjacent to the smallest angle are concatenated (Fig. 5b).

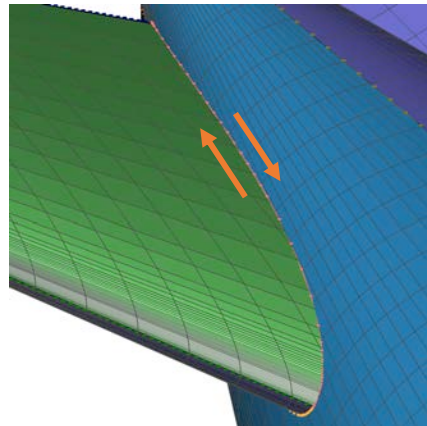


Figure 2: Directional sense of an Edge relative to its two adjacent Faces with the untrimmed geometry on the left side of the sense arrow.

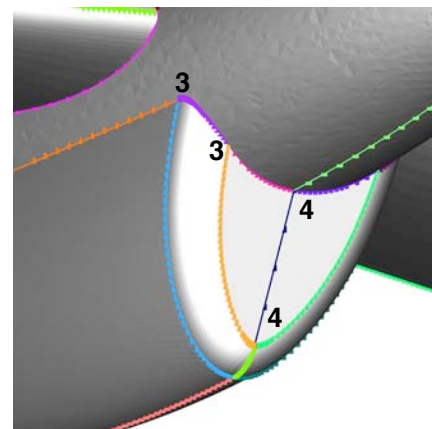


Figure 3: Seam points and their associated curves with valence marked by integer label.

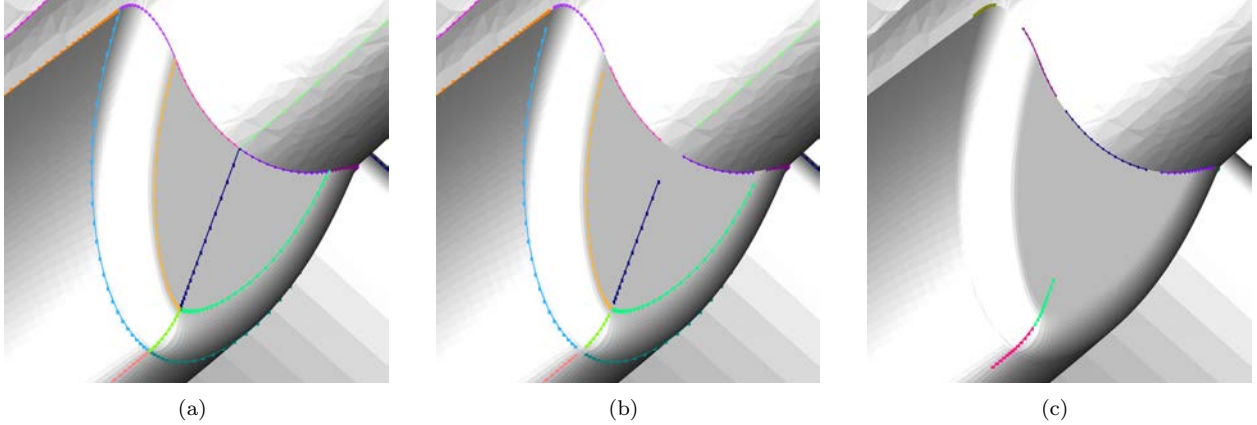


Figure 4: Construction of initial curve set for algebraic or hyperbolic marching. (a) Original curves. (b) Curves derived from trimmed Edge curves. (c) Curves derived from concatenation of two appropriate curves that meet at seam points.

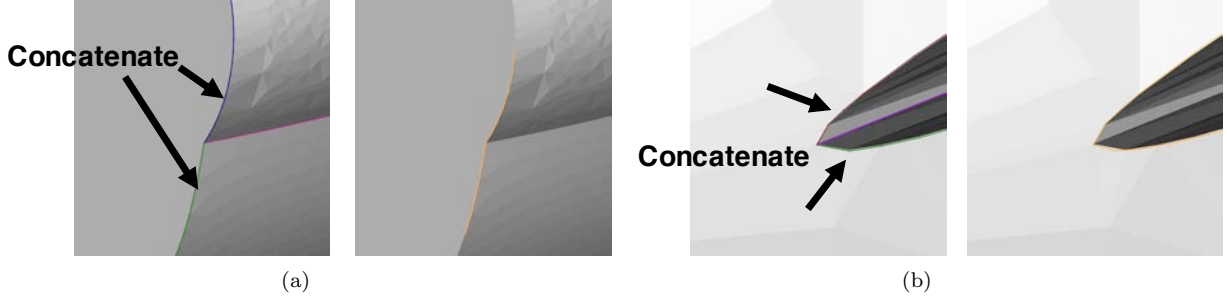


Figure 5: Concatenation of Edge curves for valence 3 seam points. (a) Largest angle  $\geq 120$  degrees. (b) Largest angle  $< 120$  degrees.

For  $N_c = 4$ , the two that meet at closest to 180 degrees are selected to be concatenated to form the initial curve at the seam point (Fig. 4c). The concatenation procedure also gives priority to curves that represent sharp edges of the geometry. Note that at a typical seam point, not all curves meeting at the point belong to sharp features. However, if they do, the resulting surface mesh marched from a seam point with  $N_c \leq 4$  could still capture all sharp features meeting at the point. It is recognized that this no longer holds true when  $N_c > 4$ . However, the current approach is automated and should produce mesh topologies that are no worse than ones that are manually generated for these cases. We note further here that  $N_c \leq 4$  holds for the vast majority of seam points encountered in typical aerospace vehicle geometries. The above construction process implies that the total number of initial curves is the sum of the number of original Edge curves and the number of seam points.

## 4 Determination of Marching Type and Direction

Starting with the initial curves, structured patches that follow constant  $u$  and constant  $v$  lines, and a reference triangulation of the surface, we have two choices for marching from the curves to generate surface meshes in the vicinity of the curves. The first follows the constant  $u$  or  $v$  lines of a Face adjacent to the initial curve, and interpolates along the curves defined by the constant- $uv$  surface. We term this method algebraic. The other method is to use the curve as an initial front and solve the hyperbolic surface mesh generation equations [20] to march the curve along the reference triangulation. Both methods produce a proper mesh, but the algebraic method requires less computation. When the initial curve is lined up with either a constant- $u$  or constant- $v$  contour, it is preferable to generate an algebraic mesh to maximize alignment of the computational mesh

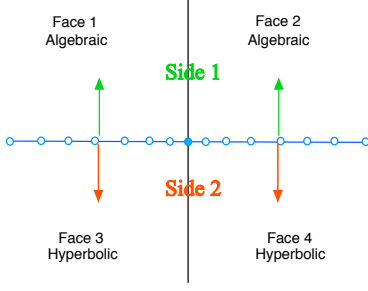


Figure 6: Marching from a curve attached to a seam point.

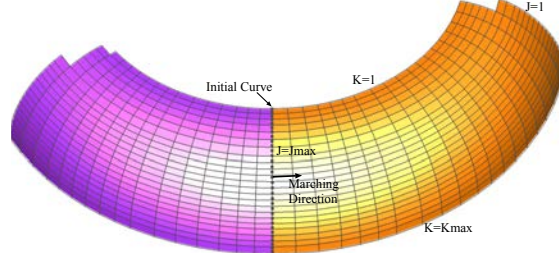


Figure 7: An initial curve and adjacent Faces. The curve is coincident with patch boundaries.

with constant parameter space lines which are also typically constant curvature lines. When the initial curve cuts through both  $u$  and  $v$  contours, a special tie-break test is needed to determine whether a hyperbolic or algebraic method is more appropriate.

It is important to note that this determination must be done for each curve/Face pair. It is possible (and often necessary) for a curve to march algebraically onto a Face on one side and hyperbolically on the Face on the other side. For trimmed edge curves, there are only two neighboring Faces and the determination is simply independent for each curve/Face pair. However, for curves emanating from a seam point, two, three, four or more neighboring Faces are possible. As noted earlier in Section 3, seam points of valence higher than four are treated like a seam point of valence four. Each curve/Face pair is considered independently. Once this independent determination is complete, each side of the curve is examined. If it is determined that the curve must march onto each of the Faces on one side of the curve algebraically (See Fig. 6, side 1), the determination is complete. However, if it is determined that hyperbolic marching is to be used for one of the two Faces on the same side of the curve, then the entire curve is flagged for hyperbolic marching on that side of the curve. A characteristic of this process is that marching onto these Faces must result in a single grid emanating from the seam point curve. A consequence of this is the requirement that the resulting meshes must match the number of points in the marching direction on each side of the curve.

To decide whether it is preferable to use the algebraic method from a given initial curve onto a neighboring reference Face, the variation in the  $u$  and  $v$  values is computed. Let  $j$  and  $k$  be the index space representation of the discretized  $u$  and  $v$  values on the Face. Here, it is of note that this variation is different for each neighboring Face for a given curve. First, the minimum and maximum values of  $u$  and  $v$  are computed for each curve/Face pair. If the variation in  $v$  values is large across the curve, but the variation in  $u$  is small (see Fig. 7), the curve is closely aligned with a  $j$  index of the curve and is flagged to be marched in the  $u$  or  $j$  direction. However, two possibilities arise for growing the mesh. The mesh can be marched in the  $+j$  or  $-j$  direction of the reference patch. To decide which of these is the correct direction, the curve sense information from the *.topo* file is used, where untrimmed geometry lies on the left side of a directed Edge given by the sense in the *.topo* file. Thus, an examination of the direction associated with the Edge/Face (curve/patch) pair provides enough information to make this choice. Similar logic is applied when the variation in  $u$  is large, but the variation in  $v$  is small.

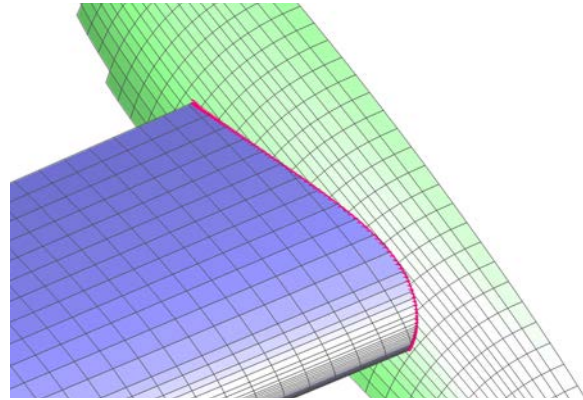


Figure 8: An initial curve at a wing/body intersection. Curve is not aligned with constant  $u$  or  $v$  lines on the wing and the body.

If the variation of  $u$  and  $v$  on the curve is large compared to an adjacent Face's  $u$  and  $v$  values, then there are two possibilities. One is that the curve is making a large turn through the  $u, v$  representation (green fuselage Face on Fig. 8). In this case, the curve is flagged for marching hyperbolically onto this patch. The other possibility is that the curve must be going through the patch at an angle to the  $u, v$  contours (blue wing Face on Fig. 8).

To march onto the adjacent blue Face, a further determination must be made to see if algebraic meshing is preferable. To do this, we first determine the marching direction on the Face using the Edge sense. The result of this step is  $+u$ ,  $-u$ ,  $+v$ , or  $-v$ . The angle between the curve direction, and the marching direction on each parent cell is then computed at every point on the curve. If the variation in this angle across the entire curve is less than  $60^\circ$ , the algebraic method is selected (wing side). For larger variations, the hyperbolic method is more appropriate (fuselage side). The result of these choices is shown in a grid in Fig. 9.

The above procedure is used for all trimmed curves (group 1). Once each trimmed curve is assigned algebraic or hyperbolic type along with marching direction ( $+u$ ,  $-u$ ,  $+v$ , or  $-v$  for algebraic and  $+$  or  $-$  direction relative to curve direction for hyperbolic), this information can be directly used to assign marching type and direction of the seam point curves in group 2.

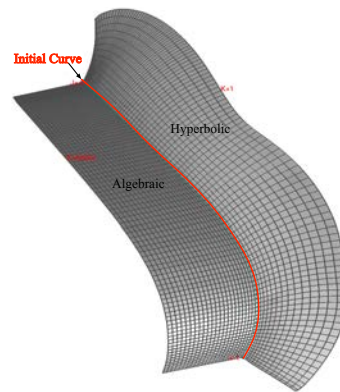


Figure 9: Mesh from an initial curve at a wing/body intersection.

## 5 Determination of Marching Scheme Inputs

Automation of a marching scheme, whether algebraic or hyperbolic, requires the determination of an initial spacing, marching distance, and a grid point distribution in the marching direction for each point on the initial curve. For a hyperbolic method, automatic selection of boundary conditions and smoothing parameters is needed. This section describes the methods used to determine these inputs automatically.

### 5.1 Initial Spacing Determination

The point distribution on the initial set of reference curves has been assigned as described in Section 3. For a given curve in this set, each end point is connected to at least one other curve since the valence of any seam point is at least two or higher. The initial spacing for marching away from a curve is then guided by the average of the end point grid spacings of connected curves. Fig. 10 shows four initial curves. For curve 2, the initial spacing in the marching direction is taken from the average of the end point spacing at 1b from curve 1 and 3b from curve 3. This initial estimate is then corrected to assure that a user-specified minimum number of points can fit within the minimum marching distance over all points on the curve (see next subsection). Where a minimum number of points does not fit, the initial spacing is reduced.

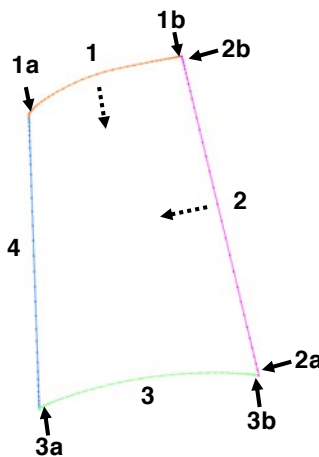


Figure 10: Determination of marching direction initial spacing from end point spacing of connected curves. Marching direction is indicated by dashed lines.

### 5.2 Marching Distance Determination

Marching distance determination is dependent on the type of marching due to the fact that algebraic marching is limited to stay on the constant- $u$  or  $v$  lines of the single Face that the initial curve is associated with in the marching direction. Hyperbolic marching is free to grow onto multiple neighboring Faces in any direction.

#### 5.2.1 Algebraic

For algebraically marched curves, the distance from the curve point to the end of the associated Face in the marching direction is considered to be a good starting estimate of the maximum possible marching distance. To do this, a parent cell on each Face is found, and interpolation coefficients within that cell are determined to identify the value of the constant  $u$  or  $v$  iso-parametric line to follow. This continues in the marching

direction (Fig. 11) until either a blanked cell is encountered or the end of the Face is reached. A blanked cell may be present due to an inner loop cutting a hole in the Face.

As an example, Fig. 11 depicts a curve/patch pair. A point ( $J_c$ ) along the initial curve belongs to the parent cell highlighted by its indices. To avoid large errors due to large reference cells, coefficients describing the location of the point in this cell are computed. These coefficients are used to compute the distance along the green arc to the cell boundary. Marching distance estimate is then accumulated cell by cell in the  $+v$  marching direction. In this example, the red inner loop is encountered before the end of the Face. For other points on the curve, the marching distance is accumulated to the end of the Face.

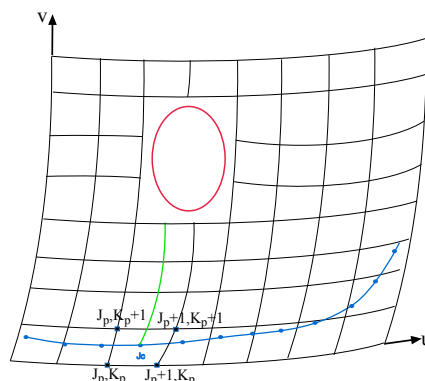


Figure 11: Computing marching distance on a Face along the  $+v$  direction from the blue initial curve.

### 5.2.2 Hyperbolic

For a hyperbolically marched curve, the reference surface triangulation is employed for the marching distance estimate since the mesh is free to grow over any number of Faces on the surface geometry. From a starting point on the initial curve, the parent triangle on the neighboring Face in the marching direction (found when determining the marching direction) is used as the starting point. From this triangle, a vector in the marching direction is used to walk from triangle to triangle accumulating the distance along this arc. An example of this procedure is shown in Fig. 12 where Fig. 12(a) shows the starting curve in green. A point on this initial curve is shown to be “walking” along the triangulation in blue. Fig. 12(b) shows the resulting tracks from all points along the initial curve. Note that when “walking” along the triangulation, one must consider the possibility that the path may cross an edge or a vertex. Also note that the marching direction must be updated along the path through each encountered triangle using the local surface normal, allowing the path to traverse through a curved surface.

The “walking” procedure has three possible termination criteria. The first is that there is an end to the triangulation. This should never be encountered for a closed solid. The second criterion is that a sharp edge is encountered. All sharp edges are marked a-priori if the angle between the normals to the triangles on each side exceeds a user specified threshold. It is recommended that this threshold be set to indicate large or sudden changes in slope along the geometry. The third termination criterion is when the angle between the initial marching direction and the final marching direction exceeds  $90^\circ$ . In other words, the track has undergone a large change in direction.

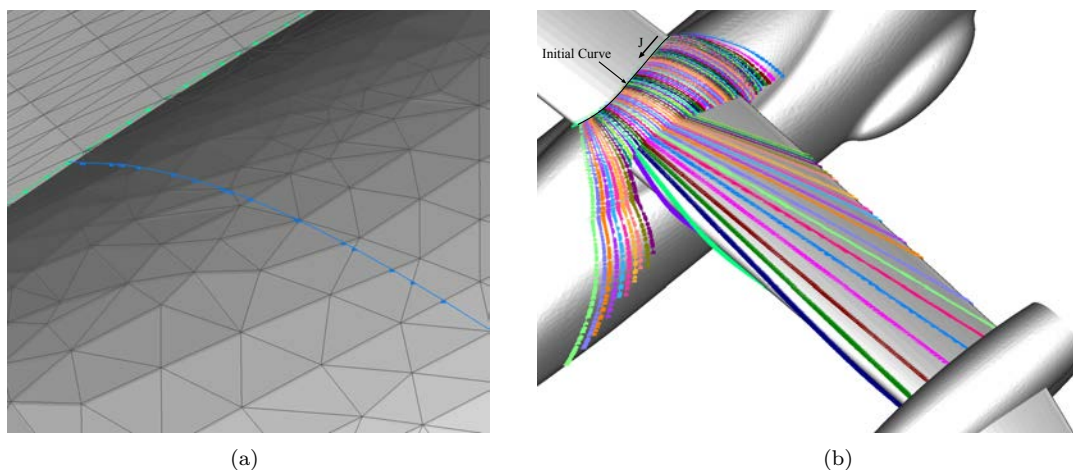


Figure 12: Computing marching distance along a triangulation. (a) Blue curve indicates path taken by marching distance estimator along a triangulation from a point on the green curve. (b) Curves derived from marching on a triangulation.



Fig. 12b shows the marching tracks for all points along an initial curve. The initial curve is the intersection between the upper surface of the starboard wing and the fuselage. Due to the fuselage upper bulge, a large number of points at the beginning of the curve (near the top of the figure) terminate from having undergone a large change in marching direction. The same is true at the other end of the curve, but this time it is because the curve has gone from the top of the fuselage to the other side of the fuselage. A number of tracks terminate at the intersection of the fuselage and the rear portion of the port wing, while a number of tracks closer to the leading edge portion of the wing march right onto the wing. This is due to a larger discontinuity in slope between the fuselage and the rear portion of the wing as compared to the forward portion of the wing and the threshold is set so that only the edges on the rear portion are considered sharp. Note that the tracks that continue onto the wing are terminated by either the trailing edge or the boom due to sharp edges.

### 5.3 Limiting and Smoothing the Marching Distance

There are several reasons to not use the full estimated marching distance. One reason is that since every reference patch is surrounded by initial curves derived from Edges, it is not necessary to march any initial curve the entire length computed by the marching distance estimate. Doing so would create meshes with excessively redundant overlap. Thus, this initial distance estimate is first trimmed by a fraction (on the order of 0.4) before other considerations come into play as described later in this section. However, there are places where nearly the entire estimated distance is needed due to the proximity to an opposing boundary. When the opposing boundary is close, the initial spacing needs to be modified as described in the previous subsection. Thus, the 0.4 fraction is only applied to cases where the initial spacing is not modified.

Another limiter is necessitated by the fact that blanking due to inner loops or termination due to different criteria while walking along a triangulation can often produce marching distances with large variations (Fig. 12b). Thus, the ratio of the maximum marching distance to the minimum is limited to a given value. A value of 1.5 to 2 seems reasonable, though a user can specify larger values when deemed necessary. Fig. 13 depicts a case which shows where this kind of limiter is necessary.

Finally, if the variable marching distance is left as a limited estimate computed above, it may contain discontinuities. To address this, a 3-point weighted average smoothing procedure is introduced where the weights on an interior point and its two neighbors are 0.5 and 0.25, respectively. Fig. 13 shows the effect of the smoother in an inset figure.

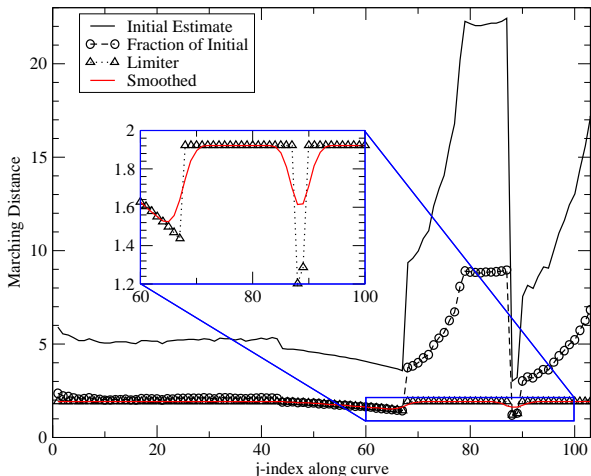


Figure 13: Estimate, limiting, and smoothing of variable marching distance.

### 5.4 Determination of Grid Point Distribution in Marching Direction

The last step before creating surface meshes is to specify the point distribution in the marching direction. This step must be performed for every point along all initial curves since marching distance varies from point to point. To determine a distribution from a given point for a user-specified stretching ratio, pre-computed marching distance, and pre-computed initial spacing, the hyperbolic tangent stretching function [21] is utilized. First, the number of points ( $N_k$ ) that satisfies the constraints for hyperbolic tangent stretching is computed for the point where the marching distance is the largest on each initial curve. Subsequently, the point distribution for every point along that curve is computed to fit  $N_k$  points within its marching distance.

## 6 Creation of Surface Meshes

For the algebraically marched surface meshes, each point on an initial curve is advanced (marched) along the pre-determined direction (e.g.  $+u$ ) along the reference Face. To do this, the cell containing the initial point on the Face is determined and an approximation for the next point is computed in the marching direction using the point distribution computed in the previous section. Linear interpolation within a cell is then used to adjust the exact location of the point from one cell to the next. This procedure is repeated until all  $N_k$  points have been created for all points on all curves. For the hyperbolically marched surface meshes, the procedure presented in Ref. [20] is used to generate a surface grid on the reference surface triangulation using the pre-computed point distributions.

The marching scheme described in the above sections takes care of domain decomposition around the Edge curves resulting in surface meshes in the region around the surface features. One may be tempted to extend the variable marching distance capability of the current scheme to ensure complete coverage of the surface domain with appropriate overlap between neighboring meshes. Further consideration indicates that it is more advantageous to allow the automatically generated surface patches from parameter space tessellation to take over coverage of the region away from the Edge curves. Since the parameter space isolines are typically surface curvature aligned, surface patches created from such isolines are superior in resolving the surface geometry provided appropriate grid spacings are imposed.

As described in Section 2, the surface patches derived from parameter space tessellation carry a tag at each vertex to indicate whether the vertex falls in a trimmed region of the geometry. Such a vertex is labelled a ‘blanked’ vertex and can be disregarded in subsequent flow solver computations. The cells from the parameter space surface patches with non-blanked vertices cover the surface regions not occupied by the meshes created from the marching scheme. One could directly utilize these parameter space surface patches with blanked vertices as surface meshes since the overset mesh approach supports such a blanking scheme for flow solution computation. If a non-trivial fraction of the vertices on these surface patches are indeed blanked, savings in grid points, and hence flow solution computational time, can be achieved by breaking up these surface patches into smaller pieces where most vertices are not blanked. Such a grid point reduction scheme, together with local grid refinement to match grid resolutions of neighboring meshes, are left as tasks for future work.

## 7 Test Cases

The scheme described in the previous sections is applied to a number of aerospace vehicle test cases. At the time of completion of this paper, about 80-90% of the automatically generated meshes for these test cases are satisfactory. The remaining defective meshes are mainly caused by an inappropriate choice between algebraic and hyperbolic marching, or an inappropriate determination of a local marching distance. Since 80-90% of the resulting meshes are usable, this is still a significant improvement in time and labor needed over manually created meshes.

The first test case is a conceptual design rotorcraft called Quad with 30 Faces, 58 Edges, and 42 seam points. Fig. 14a shows 30 surface meshes derived from tessellation of the Faces in the geometry. Vertices that fall inside the trimmed regions of the geometry ( $iblack=0$ ) are not displayed. Fig. 14b shows 100 surface meshes generated from the decomposition and tessellation of the Edges and employment of an algebraic or hyperbolic marching scheme. There are 130 surface meshes with a total of 166923 grid points. The total surface mesh generation time took 46 seconds of wall clock time on a Linux workstation.

The second test case is a conceptual design rotorcraft called Vehicle B HECTR in airplane mode with 49 Faces, 100 Edges, and 63 seam points. Fig. 15a shows 49 surface meshes derived from tessellation of the Faces in the geometry. Fig. 15b shows 163 surface meshes generated from the decomposition and tessellation of the Edges and employment of an algebraic or hyperbolic marching scheme. There are 212 surface meshes with a total of 286623 grid points. The total surface mesh generation time took 35 seconds of wall clock time on a Linux workstation.

The third test case is a conceptual design rotorcraft called Tiltwing in airplane mode with 56 Faces, 133 Edges, and 85 seam points. Fig. 16a shows 56 surface meshes derived from tessellation of the Faces in the geometry. Fig. 16b shows 218 surface meshes generated from the decomposition and tessellation of the Edges and employment of an algebraic or hyperbolic marching scheme. There are 274 surface meshes with a total

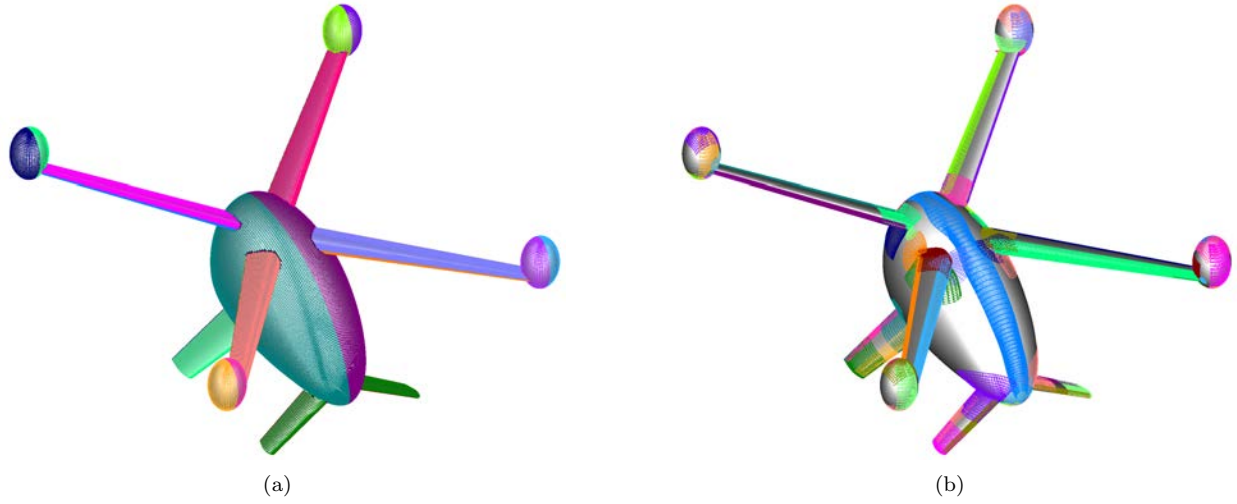


Figure 14: Automatically generated surface meshes for Quad. (a) Structured surface patches from tessellation of Faces in parameter space where vertices in trimmed regions are blanked. (b) Structured surface meshes generated from decomposition and tessellation of Edges.

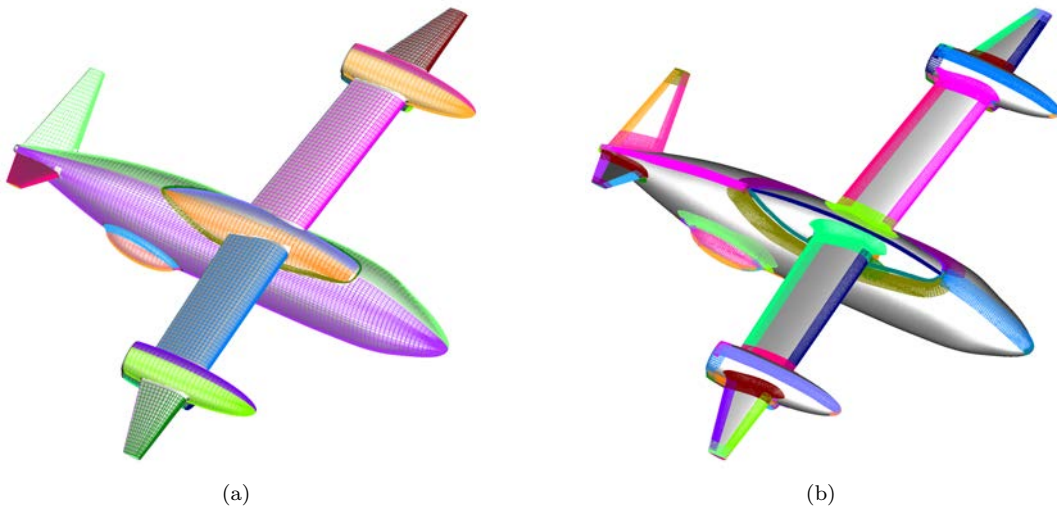


Figure 15: Automatically generated surface meshes for Vehicle B in airplane mode. (a) Structured surface patches from tessellation of Faces in parameter space where vertices in trimmed regions are blanked. (b) Structured surface meshes generated from decomposition and tessellation of Edges.

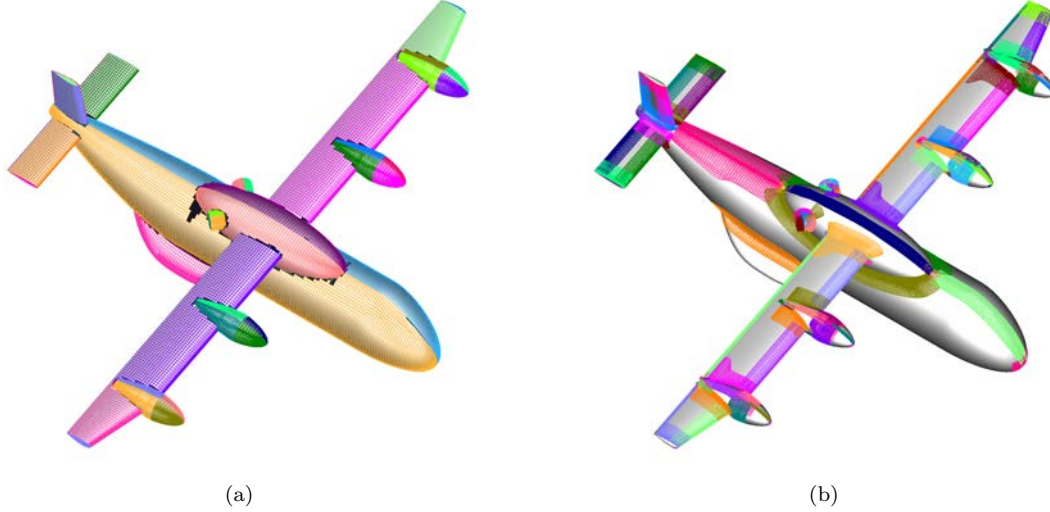


Figure 16: Automatically generated surface meshes for Tiltwing in airplane mode. (a) Structured surface patches from tessellation of Faces in parameter space where vertices in trimmed regions are blanked. (b) Structured surface meshes generated from decomposition and tessellation of Edges.

of 256959 grid points. The total surface mesh generation time took 71 seconds of wall clock time on a Linux workstation.

## 8 Summary and Conclusions

A scheme has been presented for the automation of overset surface mesh generation based on algebraic and hyperbolic marching methods starting from a BRep (solid) model of the geometry. The pre-processing step, using ESP software, creates discrete representations of the Brep Faces and Edges derived from a parameter space tessellation. Attributes of the discrete representations are user-controllable via parameters for maximum dihedral angle, edge arc length, and chordal deviation. The connectivity topology of the Edges is automatically determined by storing Edge curve end points at the seam points. An initial curve set for grid marching is derived from two sources: (1) Edge curves retracted away from the seam points, (2) curves created by concatenation of two appropriate Edge curves at each seam point. Grid point distribution on these curves is automatically performed based on consideration of curve arc length, total and local turning angles, and local geometry behavior at the curve end points.

The marching scheme type (algebraic or hyperbolic) is determined by considering parameter space values of the initial curve vertices, and the relative marching directions of each vertex on the initial curve. A spatially-varying marching distance is computed by determination of surface arc lengths to the closest sharp surface feature for the hyperbolic marching scheme, and by the surface arc lengths of adjacent Faces for the algebraic marching scheme. Surface meshes created from the marching schemes cover the surface domain around the Edge curves, and hence all geometric surface features. The remainder of the surface domain is covered by the original automatically generated parameter space tessellation of the Faces. The combination of these two sets of meshes forms an overset mesh system that covers the entire surface domain.

A number of test cases demonstrating the capability of the current scheme are presented. The surface domain of each case is successfully decomposed into structured mesh regions with a grid point distribution that is mostly satisfactory. Although the automatically generated surface domain decomposition is typically different from and contains more meshes than that created manually by an expert, the automated process is highly efficient. All test cases presented here are completed in 0.5 to 1.25 minutes on a desktop workstation, compared to about one to two days of wall clock time if these meshes are generated manually.

A number of improvements are still needed before the proposed scheme could be used as a production tool. As indicated earlier, about 10-20% of the automatically generated meshes are currently unsatisfactory. Robustness in the choice of algebraic versus hyperbolic marching could be further enhanced. The

automatically determined spatially variable marching distance may be inappropriate in some cases. A more sophisticated estimate based on concavity/convexity of the initial curve in the marching direction would help improve the distance estimate. The grid point distribution between neighboring meshes could be made more compatible in some situations. The current scheme made no attempt to optimize the set of meshes obtained by Face tessellation, resulting in excessive overlap and un-used grid points in some places. A simple projection and repartitioning step could provide further blanking or removal of unneeded grid points from these meshes. Faces with very small Edges, such as those on finite thickness trailing edges, would result in mesh overlap in the middle of the thin Face under the marching scheme approach described in this paper. By combining with an algebraic transfinite interpolation scheme approach as described in Ref. [19], such overlaps can be avoided, allowing a more implicit solution computation process. All of these concerns are planned to be addressed in a future paper.

### Acknowledgements

This work is partially funded by NASA's Transformational Tools, and Technology (TTT), Revolutionary Vertical Lift Technology (RVLT), and Advanced Air Transport Technology (AATT) projects.

### References

- [1] Slotnick, J. P., Kandula, M. and Buning, P. G. Navier-Stokes Simulation of the Space Shuttle Launch Vehicle Flight Transonic Flowfield Using a Large Scale Chimera Grid System. AIAA Paper 1994-1860, 1994.
- [2] Kiris, C. C., Kwak, D., Chan, W. M., Housman, J. A. High-Fidelity Simulations of Unsteady Flow Through Turbopumps and Flowliners. *Computers & Fluids*, Vol. 37, pp. 536-546, 2008.
- [3] Bhagwat, M., Dimanlig, A., Saberi H., Meadowcroft, E., Panda, B. and Strawn, R. CFD/CSD Coupled Trim Solution of the Dual-Rotor CH-47 Helicopter Including Fuselage Modeling. Proceedings of the American Helicopter Society Aeromechanics Specialist's Conference, San Francisco, 2008.
- [4] Kiris, C., Housman, J., Gusman, M., Chan, W. and Kwak, D. . Time-Accurate Computational Analysis of Ignition Overpressure in the Flame Trench. *Computational Fluid Dynamics Review*, Eds. Hafez, Oshima, Kwak, Publisher: World Scientific, 2010.
- [5] N. M. Chaderjian. Navier-Stokes Simulation of UH-60A Rotor/Wake Interaction Using Adaptive Mesh Refinement. 73rd AHS Forum, 2017.
- [6] Chan, W .M. Overset Grid Technology Development at NASA Ames Research Center. *Computers & Fluids*, Vol. 38, No. 3, pp. 496-503, 2009.
- [7] Rogers, S. E., Dalle, D. J. and Chan, W. M. CFD Simulations of the Space Launch System Ascent Aerodynamics and Booster Separation. AIAA Paper 2015-0778, 2015.
- [8] Nichols, R. H., Tramel, R. W. and Buning, P. G. Solver and Turbulence Model Upgrades to OVERFLOW 2 for Unsteady and High-Speed Applications. AIAA Paper 2006-2824, 2006.
- [9] Kiris, C. C., Housman, J. A., Barad, M. F., Brehm, C., Sozer, E., Moini-Yekta, S. . Computational Framework for Launch, Ascent, and Vehicle Aerodynamics (LAVA). *Aerospace Science and Technology*, Vol. 55, pp. 189-219, 2016.
- [10] W. M. Chan, R. J. Gomez, S. E. Rogers, and P. G. Buning. Best Practices in Overset Grid Generation. AIAA Paper 2002-3191, 2002.
- [11] Chan, W. M. Best Practices on Overset Structured Mesh Generation for the High-Lift CRM Geometry. AIAA Paper 2017-0362, 2017.
- [12] Rogers, S. E., Roth, K., Cao, H. V., Slotnick, J. P., Whitlock, M., Nash, S. M. and Baker, M. D. Computation of Viscous Flow for a Boeing 777 Aircraft in Landing Configuration. *J. of Aircraft*, Vol. 38, No. 6, pp. 1060-1068, December 2001.
- [13] Chan, W. M. and Gomez, R. J. Advances in Automatic Overset Grid Generation Around Surface Discontinuities. AIAA Paper 1999-3303, 1999.
- [14] Gammon, M., Bucklow, H., and Fairey, R. A Review of Common Geometry Issues Affecting Mesh Generation. AIAA Paper 2018-1402, 2018.
- [15] Haimes, R. and Dannenhoffer, J. F. The Engineering Sketch Pad: A Solid-Modeling, Feature-Based, Web-Enabled System for Building Parametric Geometry. AIAA Paper 2013-3073, 2013.

- [16] Walatka, P .P., Buning, P. G., Pierce, L., and Elson, P.A. PLOT3D User's Manual. NASA TM 101067, 1990.
- [17] Aftosmis, M .J., Berger, M. J., and Melton, J. E. Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry. *AIAA J.*, Vol. 36, No. 6, pp. 952–960, 1998.
- [18] Aftosmis, M. J., Berger, M. J., and Adomavicius, G. A Parallel Multilevel Method for Adaptively Refine Cartesian Grids with Embedded Boundaries. AIAA Paper 2000-0808, 2000.
- [19] Chan, W. M. Strategies Toward Automation of Overset Structured Surface Grid Generation. AIAA Paper 2017-3451, 2017.
- [20] Chan, W .M. and Buning, P. G. Surface Grid Generation Methods for Overset Grids. *Computers & Fluids*, Vol. 24, No. 5, pp. 509–522, 1995.
- [21] Vinokur, M. On One-dimensional Stretching Functions for Finite-Difference Calculations. *J. Comp. Phys.*, Vol. 50, Issue 2, pp. 215–234, 1983.