AIAA SciTech Forum
7-11 January 2019, San Diego, California
AIAA Scitech 2019 Forum

10.2514/6.2019-1721

Automatic Unstructured Mesh Generation with Geometry Attribution

Steve Karman*
Nick Wyman*
Pointwise, Inc.

A process is described that automatically creates unstructured meshes from water-tight geometry input. The mesh is constructed using Pointwise via the Glyph scripting language. Without any instructions from a user, isotropic tetrahedral meshes are produced. With instructions provided via geometry attribution, the meshing process can be guided to produce a mesh more aligned with the user's intent. Examples are included that demonstrate automatic meshing for geometry with and without attribution.

I. Introduction

Fully automated mesh generation has been a goal in the Computational Fluid Dynamics (CFD) community for decades. Initial mesh generation methods include conformal mapping techniques and general structured mesh methods. For complex geometries multi-block topologies were typically employed. These can complicate the process and create challenges to producing smooth spacing distributions. [1, 2, 3] Many of the multiple block topology methods are tailored to a specific class of geometries and are not fully automatic. Some attempts at automating the blocking process involve the use of medial-axis techniques, sometimes in a hybrid grid approach. [4] Grid point distribution techniques range in complexity from Transfinite Interpolation to non-linear Partial Differential Equation based smoothing. Overset meshing techniques can be extremely helpful by providing a process for meshing individual geometry components that are assembled into the final mesh. [5] Geometry is the starting point for any of these mesh generation schemes and is key to automation. [6]

Unstructured mesh generation was thought to be the panacea to the difficulties in meshing complex geometries. Blocking topologies are no longer a concern. Hierarchical Cartesian methods are one form of unstructured mesh methods that can produce high quality hexahedral inviscid meshes for extremely complex geometries but are unsuitable for viscous simulations. [7, 8, 9] Delaunay and advancing front techniques have evolved and provide a high degree of robustness for tessellating complex geometries using mostly isotropic tetrahedral elements. Automation is increased by defining background spacing functions derived from surface mesh characteristics. [10] Meshes suitable for viscous simulation require specialized node distributions or the addition of extruded meshes with clustering to the viscous surfaces to resolve the boundary layer flowfield. This hybrid approach typically involves some user interaction to define where these extruded meshes are constructed. Again, geometry is the starting point for these mesh generation schemes and is key to automation.

Scripting languages are frequently available in mesh generation applications to automate many of the repetitive tasks. Typically, all meshing operations available in an interactive setting are also available through scripting. This paper describes the use of Pointwise and the Glyph scripting

* AIAA Associate Fellow

language to automatically generate unstructured meshes for complex geometries. [11] Pointwise generally follows a "bottom-up" mesh generation process, meaning lower order entities (points and edges) are created first followed by higher order entities (surface meshes then volume meshes). The Glyph script, named AutoMesh, alters that paradigm slightly by taking advantage of existing Pointwise capabilities that key off of geometric trimmed surfaces to automatically create surface meshes. The boundary curves (edges) of the surface meshes are a byproduct of this operation. An extensive amount of time is then spent on adjusting dimension and point distributions of the bounding curves. This greatly impacts the resulting surface meshes and the volume mesh. For water tight models a completed volume mesh is created. Input geometry formats include IGES, STEP, EGADS and NMB (Pointwise proprietary). Geometry attribution is currently possible in the EGADS and NMB formats. Without attributes the script produces a tetrahedral mesh that resolves features in the geometry. For this application, attributes are assigned during the geometry creation in Engineering Sketch Pad (ESP). [12, 13] EGADS files are the native format for ESP. With attributes assigned to various geometry entities the Glyph script modifies the mesh generation process to produce a mesh with enhanced characteristics, like viscous clustering to defined boundaries.

The Glyph script follows a set of procedures that have incorporated some best practices, at least from the authors' perspectives. These best practices are followed completely by the script which results in a high-quality final mesh. For fully closed, water tight geometry a completed mesh is often the outcome. When the geometry is not closed or water tight the script cannot complete the volume mesh, but still produces a result that is very far along in the meshing process. Examples of fully and partially completed volume meshes will demonstrate the utility of the Glyph script at automating much of the mesh generation process.

II. Geometry Definition and Attribution

The starting point for any mesh generation endeavor is geometry. And in this article geometry is also the key to automation. Pointwise can import geometry in many formats. For this project it was restricted to IGES, STEP, EGADS and NMB. IGES and STEP are recognized industry standards that Pointwise supports. EGADS is the native file format produced by the ESP software that supports attribution. The NMB format is the native Pointwise format which can also hold attribution data on geometry models, faces, edges and vertices. A closed geometry model is one that contains defined surfaces for all boundaries of the expected computational domain. A water tight model is one in which trimmed surfaces meet along edges within accepted tolerances in the model. When a water-tight, closed model is used a completed volume mesh is usually produced. Nomenclature can vary for different mesh generation applications. Table 1 defines the various mesh entities used by Pointwise for each associated geometry entity.

Table 1. Mapping of geometry name to Pointwise mesh name.

Geometry Name	Mesh Name
Model(s)	Block
Face	Domain
Edge	Connector
Node	Point or vertex

Attributes in the EGADS files are assigned to the various entities when the geometry is created in ESP. NMB files can be created by converting EGADS files. They can also be attributed within the Pointwise software graphical user interface (GUI). The attribution scheme allows the user to label geometry pieces with information that can guide the AutoMesh Glyph script to construct the intended mesh. Most of the attributes are provided to allow users to set meshing parameters that are available through the GUI. Table 2 shows the list of attributes recognized by the AutoMesh Glyph script. The first column is the key; all entries have "PW:" as the prefix. ESP can be used to provide attributes for other applications downstream in the CFD analysis process. The "PW:" prefix is defined to represent attributes for Pointwise meshing operations. The second column is the expected value for the key. It can be a string, preceded by the $ sign, or a number. The third column is the geometry entity where the key-value pair would be located. The final column is a description of the attribute.

Table 2. Attributes recognized by Pointwise Glyph script.

Key	Value	Geometry Location	Description
	Preceding $ means it is a character string		
PW:Name		Face	Boundary name for domain or collection of domains.
PW:QuiltName		Face	Name to give one or more quilts that are assembled into a single quilt. No angle test is performed.
PW:Baffle	$Baffle or $Intersect	Face	Either a true baffle surface or a surface intersected by a baffle.
PW:DomainAlgorithm	$Delaunay, $AdvancingFront, $AdvancingFrontOrtho	Face	Surface meshing algorithm.
PW:DomainIsoType	$Triangle, $TriangleQuad	Face	Surface cell type. Global default is Triangle.
PW:DomainMinEdge	$Boundary or > 0.0	Face	Cell Minimum Equilateral Edge Length in domain.
PW:DomainMaxEdge	$Boundary or > 0.0	Face	Cell Maximum Equilateral Edge Length in domain.
PW:DomainMaxAngle	[0, 180)	Face	Cell Maximum Angle in domain (0.0 = NOT APPLIED)
PW:DomainMaxDeviation	[0, infinity)	Face	Cell Maximum Deviation in domain (0.0 = NOT APPLIED)
PW:DomainSwapCells	true or false	Face	Swap cells with no interior points.
PW:DomainQuadMaxAngle	(90, 180)	Face	Quad Maximum Included Angle in domain.
PW:DomainQuadMaxWarp	(0, 90)	Face	Cell Maximum Warp Angle in domain.
PW:DomainDecay	[0, 1]	Face	Boundary decay applied on domain.
PW:DomainMaxLayers	[0, infinity)	Face	Maximum T-Rex layers in domain.
PW:DomainFullLayers	[0, infinity)	Face	Number of full T-Rex layers in domain. (0 allows multi-normals)
PW:DomainTRexGrowthRate	[1, infinity)	Face	T-Rex growth rate in domain.
PW:DomainTRexType	$Triangle, $TriangleQuad	Face	Cell types in T-Rex layers in domain.
PW:DomainTRexIsoHeight	> 0.0	Face	Isotropic height for T-Rex cells in domain. Default is 1.0.
PW:WallSpacing	> 0.0	Face	Viscous normal spacing for T-Rex extrusion.
PW:TRexIsoHeight	> 0.0	Model	Isotropic height for volume T-Rex cells. Default is 1.0.
PW:TRexCollisionBuffer	> 0.0	Model	T-Rex collision buffer. Default is 0.5.
PW:TRexMaxSkewAngle	[0, 180]	Model	T-Rex maximum skew angle. Default 180 (Off)
PW:TRexGrowthRate	[1, infinity)	Model	T-Rex growth rate.
PW:TRexType	$TetPyramid, $TetPyramidPrismHex, or $AllAndConvertWallDoms	Model	T-Rex cell type
PW:BoundaryDecay	[0, 1]	Model	Volumetric boundary decay. Default is 0.5.
PW:EdgeMaxGrowthRate	[1, infinity)	Model	Volumetric edge maximum growth rate. Default is 1.8.
PW:MinEdge	$Boundary or > 0.0	Model	Tetrahedral Minimum Equilateral Edge Length in block.
PW:MaxEdge	$Boundary or > 0.0	Model	Tetrahedral Maximum Equilateral Edge Length in block.
PW:ConnectorMaxEdge	> 0.0	Edge	Maximum Edge Length in connector.
PW:ConnectorEndSpacing	> 0.0	Edge	Specified connector endpoint spacing.
PW:ConnectorDimension	> 0	Edge	Specify connector dimension.
PW:ConnectorAverageDS	> 0.0	Edge	Specified average delta spacing for connector dimension.
PW:ConnectorMaxAngle	[0, 180)	Edge	Connector Maximum Angle. (0.0 = NOT APPLIED)
PW:ConnectorMaxDeviation	[0, infinity)	Edge	Connector Maximum Deviation. (0.0 = NOT APPLIED)
PW:NodeSpacing	> 0.0	Node	Specified connector endpoint spacing for a node.

A. Model Attributes

Attributes assigned to geometry models are global attributes. They pertain to volumetric meshing operations, such as T-Rex extrusion and isotropic tetrahedral meshing. Any model attribute read from the geometry will override the default parameters. These attributes will be loaded after the unstructured block has been created in Pointwise, prior to volume mesh initialization.

B. Face Attributes

Attributes assigned to geometry faces pertain to surface meshing operations. Many of the recognized key-value pairs are mirrored after the user-selectable options in the GUI. Quilt assembly can be controlled by assigning the same quilt name to multiple geometry faces. Pointwise will attempt to combine those trimmed surfaces into a single quilt. This will impact the surface meshing operations used in Pointwise to initiate the automatic meshing process. Boundaries can also be given names via face attribution. The PW:Name value will be defined as the boundary name in Pointwise for the associated surface mesh on the face's quilt. The PW:WallSpacing attribute will identify that surface mesh as a viscous boundary and define the associated spacing value as the distance of the first point off the surface for a subsequent T-Rex extrusion process. The PW:Baffle attribute is a special key-value pair. It is used to identify geometry surfaces that are intended to be baffle surface (defined surface patch internal to a block) in the Pointwise mesh. The surface mesh that will be created on these surfaces in space will be added to the isotropic tetrahedral meshing operation and forced to exist in the final mesh. If one of these baffle surfaces intersects with another geometry surface, then the second surface should be labeled as an "$Intersect" surface. The Glyph script will ensure the intersecting line of the two surfaces will be an edge in the interior of the mesh of the second surface.

C. Edge Attributes

Edge attributes impact dimensioning and node distributions of connectors in Pointwise. The connectors impact the resolution of the surface meshes (domains) which then impacts the volume mesh (block). As each edge attribute is imposed the hyperbolic tangent distribution function is maintained. Connector dimensions can be increased or decreased as appropriate. Minimum and maximum dimensions are defined early in the process and maintained with each step.

D. Node Attributes

The only node attribute is a spacing value. If present, it is applied to each endpoint distribution of the connectors attached to the node (mesh vertex). Users often have specific spacing values in mind for certain locations in the mesh, such as leading edges and trailing edges of wing components.

III. Mesh Generation Process

The goal of the logic incorporated in the Glyph script is to generate a high-quality unstructured mesh suitable for CFD analysis. The dimension of the mesh is not directly controllable, it is an outcome of the procedures. Some limited control is possible through the parameter settings for the connector dimensioning and spacing control and the volume decay of boundary spacing information. Throughout the mesh generation process a set of best practices are strictly enforced.

- Resolve all geometry features to user prescribed levels through parameters that control the minimum edge dimension, curvature and deviation tolerances.
- Maintain smoothness of the mesh resolution through proximity and optional adaptive dimensioning of connectors.
- Maintain smoothness along the connectors by enforcing edge segment ratio limits in a hyperbolic tangent point distribution. Enforce aspect ratio limits in the surface mesh extrusion process normal to the connectors.

- Maintain consistent spacing at endpoints of connectors that directly impact the quality of the surface mesh extrusion process.
- Enforce smoothness in the volume mesh through extrusion growth rates, edge ratio limits and reasonable spacing decay away from surfaces.

A. Global Meshing Parameters

A set of predefined meshing parameters is loaded by the master Glyph script. These parameters control different aspects of the meshing process, such as the connector dimension limits, spacing decay factors, and different meshing algorithms used to construct the mesh. Parameters are defined at the connector level, the domain level, the block level and general parameters. The initial values of these parameters are the default values found in the Pointwise GUI. Users have the option to load their own set of values of these parameters, which will override the default parameters.

B. Meshing Process Flow Chart

A top-level view of the mesh generation flowchart is shown in Figure 1. The process begins at the upper left with "Start", where the global parameters are loaded. Next the geometry is loaded and processed by Pointwise, creating quilts. Quilts are collections of geometry surfaces that will be meshed as a single surface mesh. If no user instructions are provided through attribution this typically results in one surface mesh per geometry face. With attribution some combining can be performed, such as combining multiple upper or lower wing surface geometry faces into a single mesh.

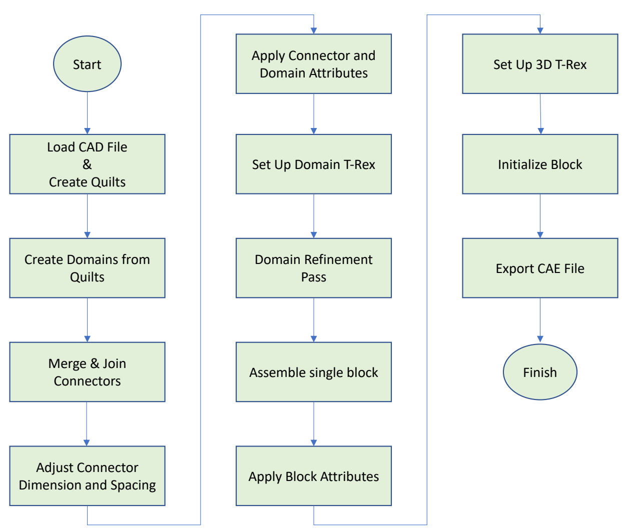

Figure 1. Flowchart for main components of Pointwise auto-mesh Glyph script.

Automation is truly enabled in the second block where domains are created from the assembled quilts. This process is enabled in Pointwise when default connector dimensions are defined, or average connector spacing value is defined. It is impractical to define an average spacing value for an arbitrary geometry, so the connector dimensioning approach is used. An initial dimension of 11 for each connector is defined. If the domain creation fails for any given quilt it is usually because the dimension was too small. The process is restarted with a larger initial dimension. Connectors are then merged and joined once the domain creation is complete. This creates a consistent, minimal set of connectors where no duplicate or overlapping connectors exist. Each connector should be shared by two domains, except for some baffle surface connectors.

A significant amount of time is then spent on adjusting the connector dimensions and point distributions. This incorporates several "best practices" that ensure a quality mesh is produced. This begins by reducing the connector dimensions where possible from the initial dimension used during the domain creation process. Next the dimensions are adjusted based on turning angle in the direction of the connector or deviation from the curve shape. During this step each connector is evaluated as a possible edge for surface TRex meshing. T-Rex is the advancing front technique used by Pointwise to create layers of elements by marching away from initial edges in surface meshing or faces in volume meshing. If the curvature in the direction normal to the edge exceeds a user specified turning angle it is tagged as a curvature-based TRex connector. If the angle between surface elements on opposite sides of the connector exceed a hard angle limit the connector is tagged as a hard-edge TRex connector. Both of these connector categories will result in TRex extrusion along surface in the normal direction from the connector.

The next important step in the adjustment of the connector dimensions is a proximity test where each connector is tested against all other connectors in the mesh. The average spacing of the other connectors is decayed (increased) over the closest distance between the connectors. If the decayed spacing value is smaller than the current average spacing for the connector, the dimension is increased. End spacing of all connectors meeting at a common point is set to the minimum end spacing of the collection of connectors. The distribution along each connector is then controlled using the hyperbolic tangent distribution function and the defined global edge maximum growth rate. The connector dimension can be increased or decreased during this process, depending on the length of the connector and the specified end spacing values. Any connector attributes are applied where appropriate during the connector adjustment process.

The user has the option to perform a curve-based mesh adaptation pass on the connectors. This involves two steps; 1) creation of a target edge length size field and 2) performing the mesh adaptation along each connector.

1. Target Edge Length Definition

The current mesh densities of the connectors are used to define a cloud of points for interpolation of mesh target edge length at any point in the domain. The interpolation of target edge length is termed the size field and it provides two important properties for automated meshing. The first is proximity support where a small length scale is near a large length scale – in this case we wish the smaller length scale to be recovered. The second is mesh smoothness control. For non-trivial problems, the size field will necessarily have a wide range of input target edge lengths distributed non-uniformly across the computational domain. The interpolation scheme must be able to resolve conflicting inputs locally and to smoothly switch definition of the "most important" local length scale depending on query position.

The adaptation process utilizes a network of radial basis functions (RBF) for the size field. The radial basis function is constructed to provide smooth variation of the target length scale locally and is controlled by a decay parameter. Furthermore, the network interpolation scheme uses a compact stencil which recovers the minimum target edge length at the query location.

2. Curved Meshing from Size Field

The dimension and distribution of grid points on a curve to satisfy a target edge length distribution is an over constrained optimization problem. While the grid point distribution is free to adjust to the size field, the dimension variable must be an integer number greater than one which automatically leads to non-optimal solutions to all but the most trivial curve shapes and size fields.

Therefore, the mesh adaptation utilizes a method which searches for a point distribution with the minimum deviation from the size field as defined by Equation (1) which provides a signed deficit of the current edge length from the target edge length locally.

$$\varepsilon_i = (h_-^* - h_-) - (h_+^* - h_+) \tag{1}$$

where, for the trailing interval and likewise for the leading interval,

$$h_- = |X_{i-1} - X_i|$$
$$h_-^* = \frac{T_{i-1} + T_i}{2}$$

Based on the local deficit, an estimate is made to correct the point position on the curve in terms of non-dimensional arc length, *s*, using Equation (2) which assumes locally linear behavior of the curve geometry and target edge length size field.

$$\Delta s_i = \frac{\partial s}{\partial h}(i)\ \varepsilon_i \tag{2}$$

where

$$\frac{\partial s}{\partial h}(i) \cong \frac{s_{i+1} - s_{i-1}}{h_- + h_+}$$

Convergence is declared when each point's correction falls below a prescribed fraction of the local interval size (typically 0.01). In practice, good convergence is observed of this Newton-like algorithm with moderate relaxation of the point correction value.

The above point distribution optimization occurs for a fixed-point dimension. An outer loop is used to determine if and how the point dimension should be modified based on the minimum and maximum value of Equation (3).

$$\min(\rho_i) = min\left[\frac{h_-}{h_-^*}, \frac{h_+}{h_+^*}\right] \quad \text{(likewise, for } \max(\rho_i)\text{)} \tag{3}$$

If $\min(\rho_i) < 0.95$, then point *i* is removed from the curve. Similarly, if $\max(\rho_i) > 1.05$, then a new point is inserted in the leading interval of point *i*. If a point is added or removed (or both) an update of the point distribution is required.

On each pass of the outer loop the maximal error, Equation (4), is computed and recorded as a function of the point dimension. Convergence is declared when either no dimension change is warranted by Equation 3, or when revisiting a previously computed point dimension and distribution would result in a larger maximal error.

$$\varepsilon_{maximal} = |\max(\rho_i) - 1| + |1 - \min(\rho_i)| \tag{4}$$

The connector adaptation process is demonstrated on a simple straight segment in Figure 2 and Figure 3. The size field contains two minima along the length of the connector. The adapted mesh on the right satisfies the size field along the length and clusters toward those locations. The outer loop iterative process of adding and deleting points is required, as nodes on one side of the center peak are prevented from crossing over to the other side.

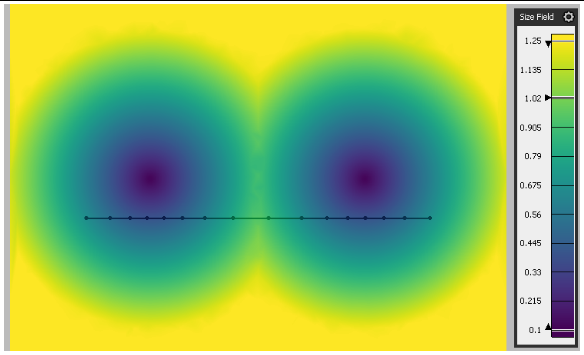

Figure 2. Simple size field for demonstration.

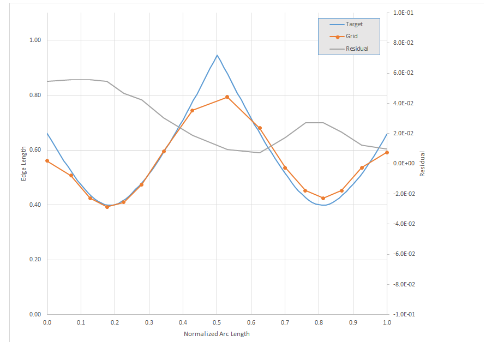

Figure 3. Node distribution for simple size field.

The effect of this adaptation pass is illustrated in Figure 4. The leading-edge point distribution is improved in the vicinity of the pylon. The baseline proximity test with enforcement of the hyperbolic tangent distribution would ignore the finer spacing on the pylon. It only specifies the endpoint spacing and the growth rate along the connector. The mesh adaptation to the size field re-establishes that influence, producing a smooth point distribution in the region.

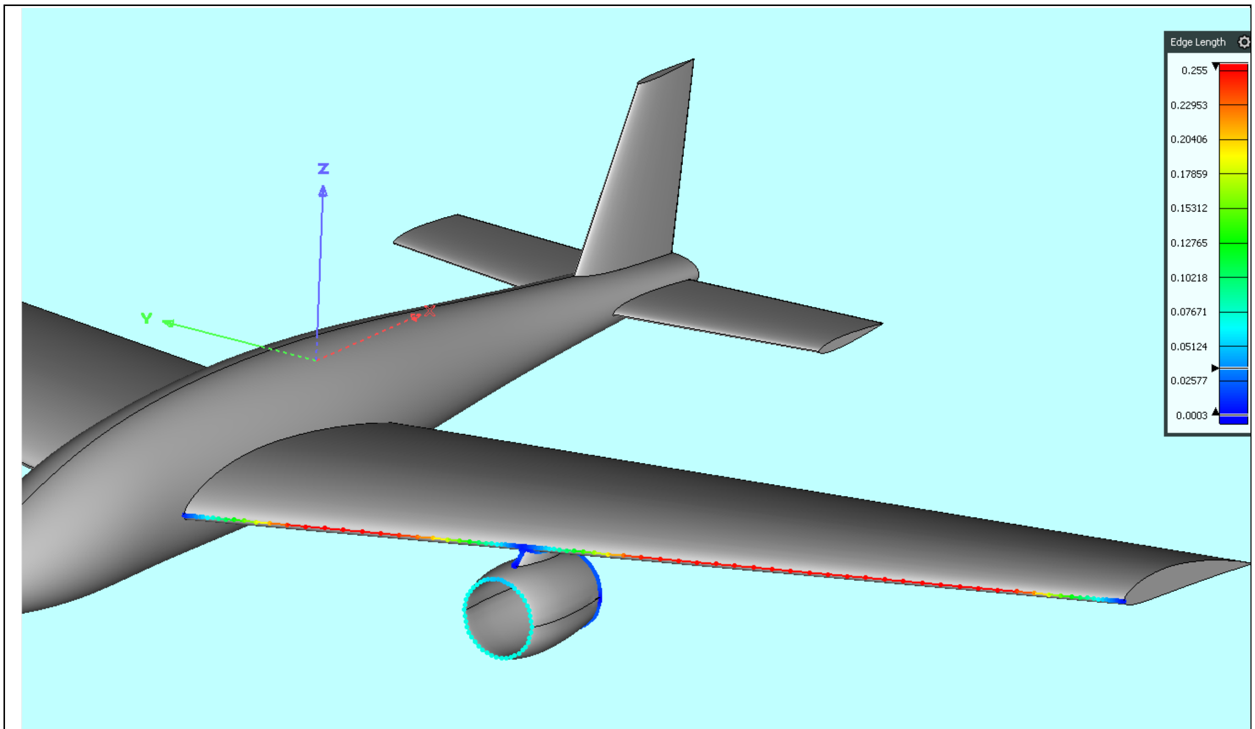

Figure 4. The wing leading edge connector target edge length is a combination of the local wing curvature and the proximity of the engine pylon.

Once all the connector modification is complete the domain attributes are applied where appropriate. The domains are then prepared for any T-Rex surface extrusion. This is a key feature of the Glyph script that automatically identifies connectors of a domain, as described above, as a marching front. The step size in the normal direction of the connector is based on the average of the two end point spacing values. A final refinement pass is performed on all domains to impose all the modifications made by the script.

All domains are then assembled into a single unstructured block. This is only possible when the geometry is closed. For unclosed geometries the meshing process will end with a high-quality set of surface meshes. For closed geometries the meshing process continues with the application of block attributes. Three-dimensional T-Rex properties are set up based on the PW:WallSpacing attributed domains. Domains adjacent to T-Rex boundaries are set up as "match" domains for the extrusion process.

If desired, users can choose to have an adaptation pass performed on the surface meshes during the initialization of the volume mesh. When performed the connectors point distributions are again set up as a size field. Any surface domain identified with viscous wall spacing is marked for adaptation. Then, before the creation of the viscous layers, the domains are adapted to the size field. The effect of this operation is illustrated in Figure 5 and Figure 6. Without adaptation the extra fine spacing on the sphere is ignored on the square face of the block that is in very close proximity to the sphere. With adaptation the size field produced by the connector distribution around the equator of the sphere heavily influences the mesh density in the front of the block. This produces a very smooth point distribution across the region, not widely varying length scales on opposing TRex fronts. This results in better matching of the element sizes in the gap between the edge of the extruded fronts.

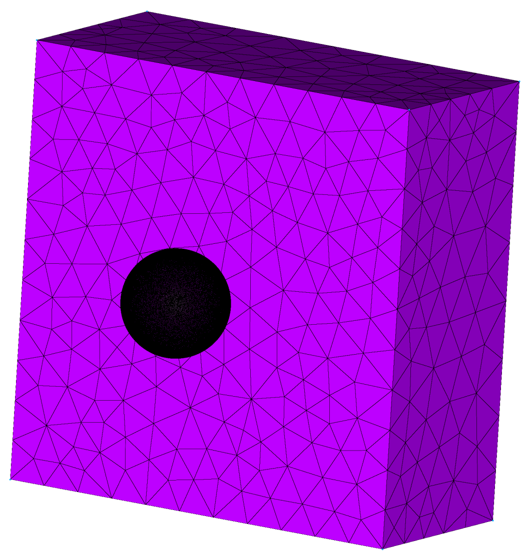

Figure 5. No adaptation pass performed on domains.

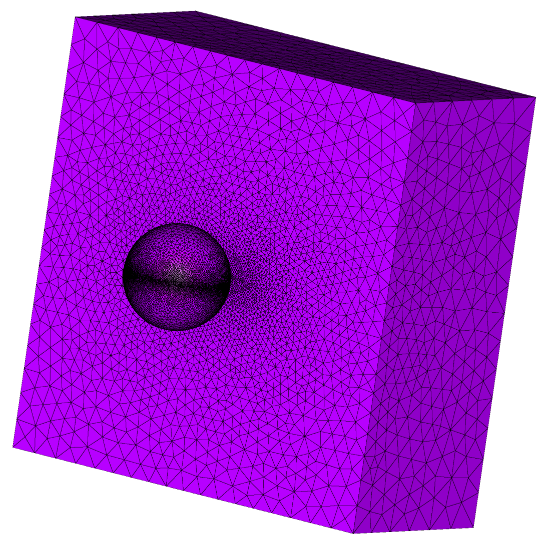

Figure 6. Connector-driven adaptation pass performed on domains.

When all the setup is complete, and the volume mesh is created it is output to the proper file format and a Pointwise save file is written.

IV. Results

The following cases demonstrate the meshing that is possible with AutoMesh on a variety of geometric configurations. No assumption is made by AutoMesh regarding the type of geometry that is supplied. Some cases are created from IGES files that do not have any attribution data.

Other cases are included to illustrate the use of geometry attribution to guide the mesh generation process in Pointwise.

A. Mechanical Part

The first case is a mechanical part contained in an IGES file. The geometry is shown in Figure 7 and two versions of the surface mesh are shown in Figure 8 and Figure 9. The first mesh uses the default input parameters, which results in a Delaunay style surface mesh and no adjustments due to geometry curvature. The second mesh changes the style to advancing front and captures surface curvature exceeding 10 degrees and hard edges exceeding 30 degrees. The edges colored red are geometry curves that exceed the 10-degree curvature threshold in the direction normal to the edge. The yellow colored edges exceed the 30-degree hard edge threshold. Both meshes resolve the different length scales in the geometry. The short edges in the gaps of the structure are resolved and the mesh blends smoothly across the surface mesh. The second mesh enhances the resolution in high curvature regions of the geometry. The internal part of the geometry was meshed with tetrahedra but is not shown.

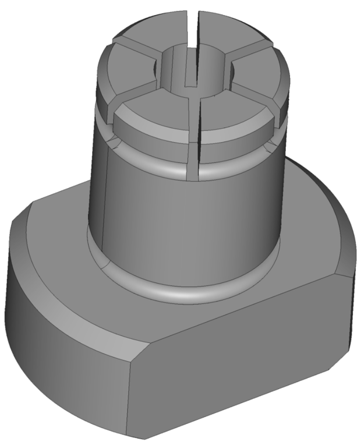

Figure 7. Geometry for simple mechanical part.

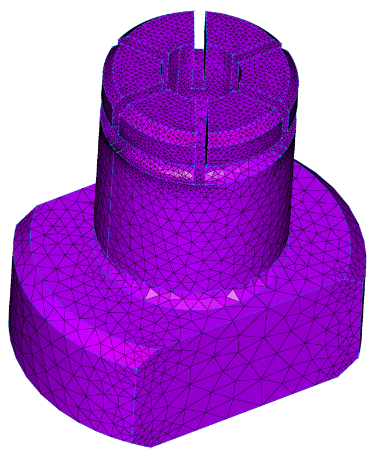

Figure 8. Surface mesh constructed using default parameters.

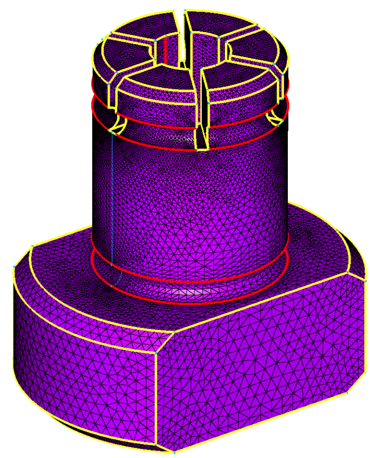

Figure 9. Surface mesh after specifying 10-degree turning angle, 30-degree hard edge angle and advancing front method.

B. High-Order Cowboy Hat

A quadratic high-order mesh was created for a generic cowboy hat. The geometry is shown in Figure 10. The IGES file was imported into Pointwise and all surface patches were attributed with the boundary name “hat” and a viscous wall spacing of 0.001. An outer bounding box 5 body lengths away was constructed during the initial steps of the AutoMesh process, thereby permitting the volume around the hat to be meshed. A curvature turning angle of 20 degrees was specified. The connectors exceeding that threshold are shown in Figure 11. The edges of the brim and the base of the band between the brim and the crown are displayed as red curves.

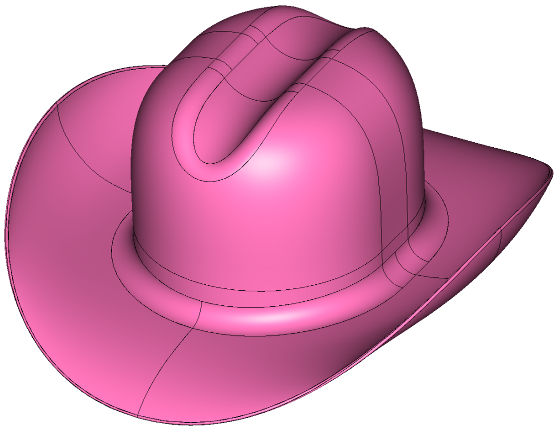

Figure 10. Geometry for generic cowboy hat.

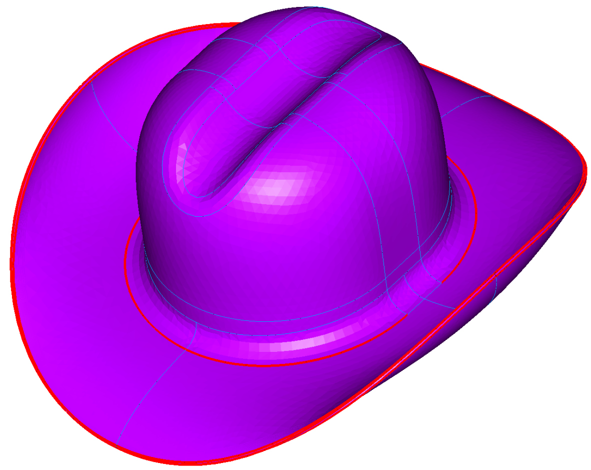

Figure 11. TRex connectors that exceed a 20-degree turning angle are colored red.

AutoMesh instructed Pointwise to create a triangular surface mesh, shown in Figure 12. The specified wall spacing on the geometry initiated a TRex extrusion into the volume, shown in Figure 13. On export Pointwise curved the mesh to quadratic order. The high-order nodes on the hat surface and a crinkle cut through the volume mesh at the tip of the brim are shown in Figure 14. A similar view through a centerline cut is shown in Figure 15.

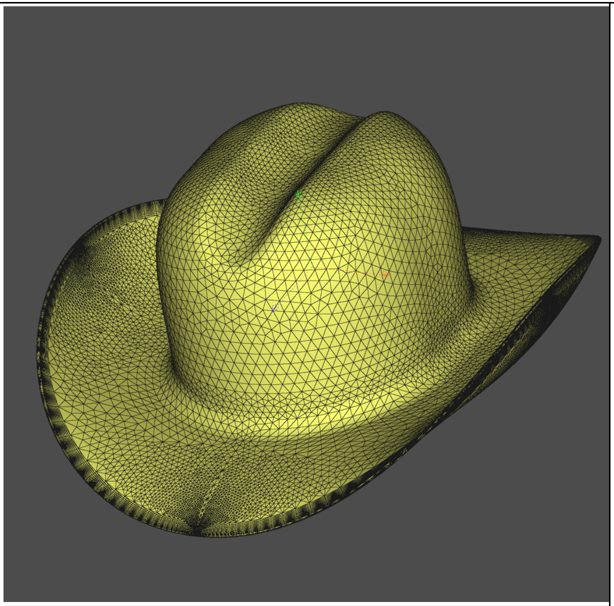
Figure 12. Quadratic surface mesh on cowboy hat.

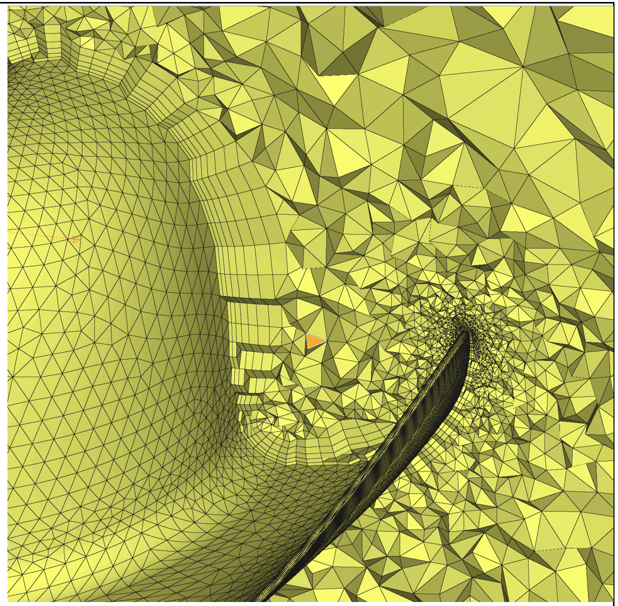
Figure 13. Crinkle cut through quadratic volume mesh.

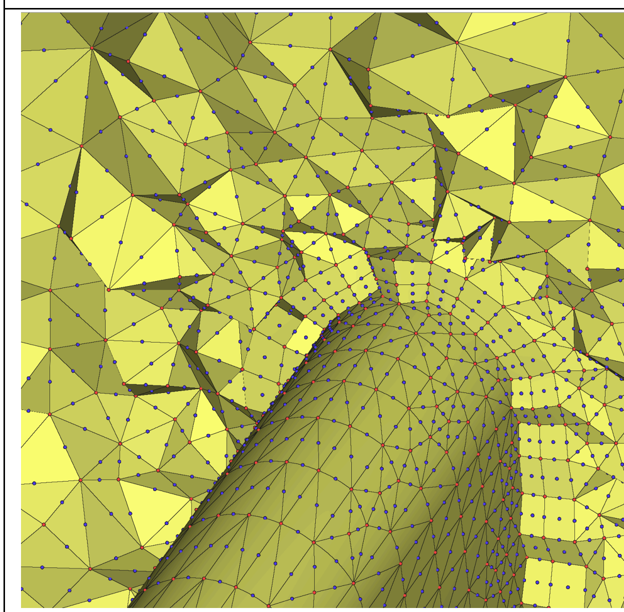
Figure 14. High-order nodes on crinkle cut through volume mesh at rim.

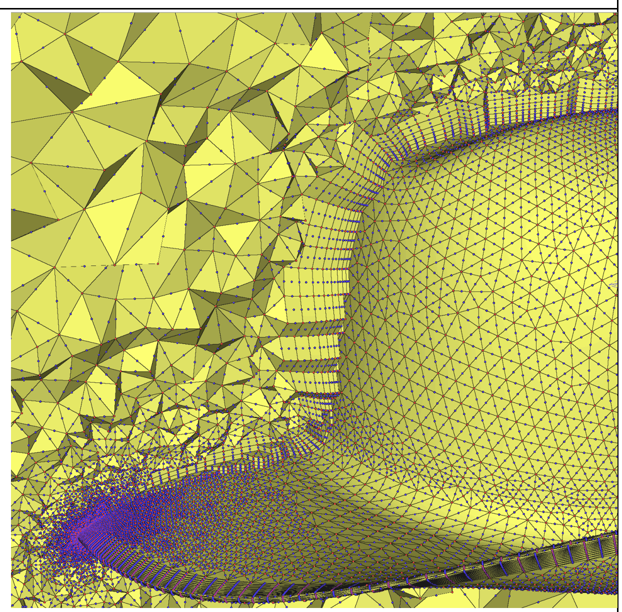
Figure 15. Side view of high-order nodes on crinkle cut through centerline of cowboy hat.

C. Generic Glider

A generic glider configuration was created using ESP. The intent was to analyze the case using a potential flow solver, therefore wake sheets were required off the wing and horizontal tail surfaces. The geometry with the wake sheets is shown in Figure 16. The geometry included the outer boundary as a box. The wake sheets meet at the back-boundary face. The resulting surface mesh and wake baffles are shown in Figure 17. The wake sheet geometry was attributed with

PW:Baffle $Baffle. The back boundary was attributed with PW:Baffle $Intersect. The resulting baffle meshes were properly intersected with the outflow boundary and share the connectors at the back boundary. Other attributes specified in the ESP model were end spacing values for the connectors on the wing and vertical and horizontal tail surfaces. In addition, maximum edge spacing values were defined on the same surface. This forced additional resolution on those lifting surfaces.

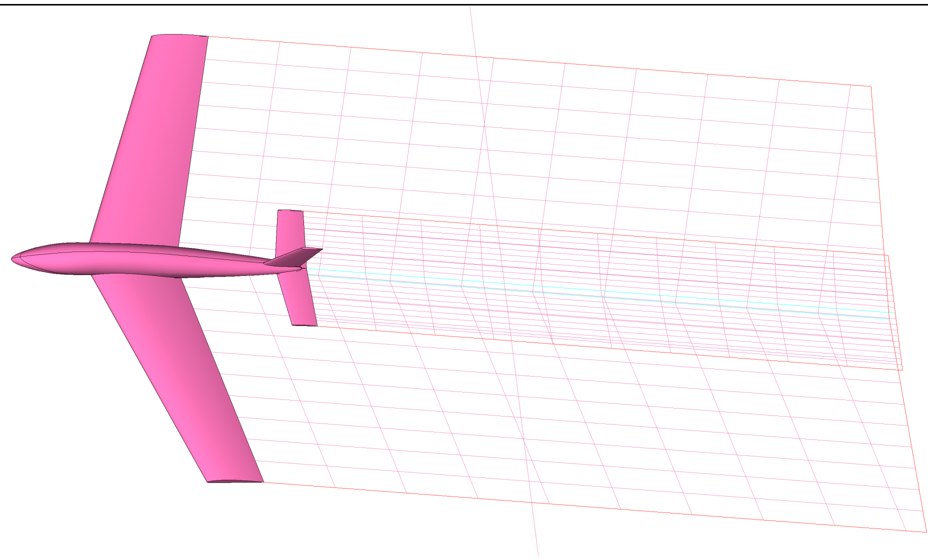

Figure 16. Generic glider configuration with defined wake sheets.

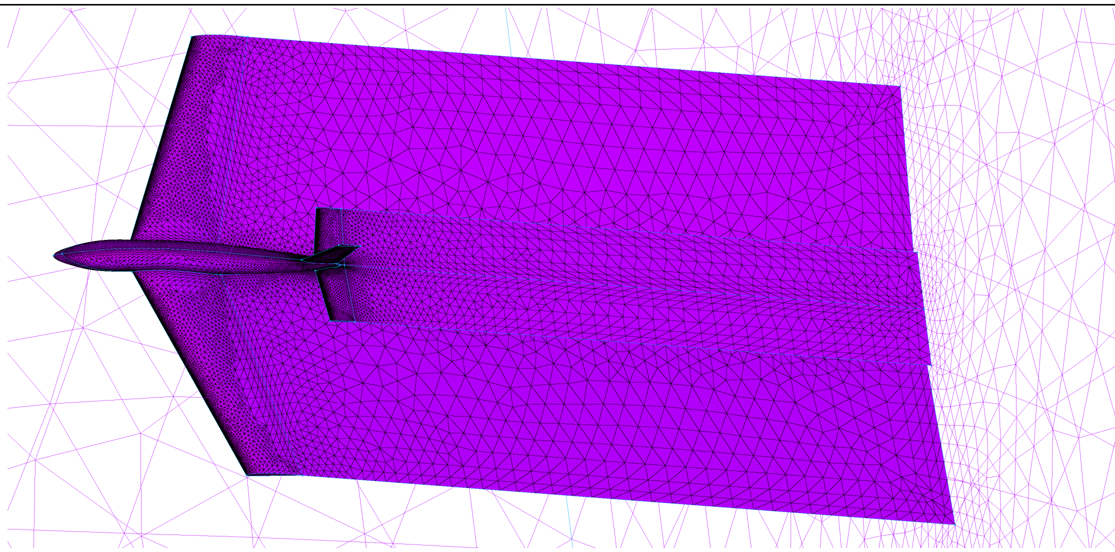

Figure 17. Automatically create glider mesh with baffle surfaces.

D. Generic Assembly of Fan Nozzles

A fan assembly is defined on a circular plate, shown in Figure 18 through Figure 23. Quilting was used to combine the nine individual faces on the top of the assembly into a single quilt. An outer box boundary was automatically made by the AutoMesh script. All boundaries were named.

All surfaces were tagged with a $PW:WallSpacing value that resulted in the viscous layers shown in the images. The resulting mesh contained all tetrahedra.

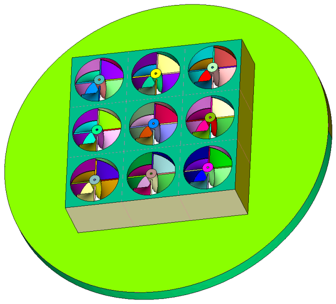

Figure 18. Fan assembly on circular plate.

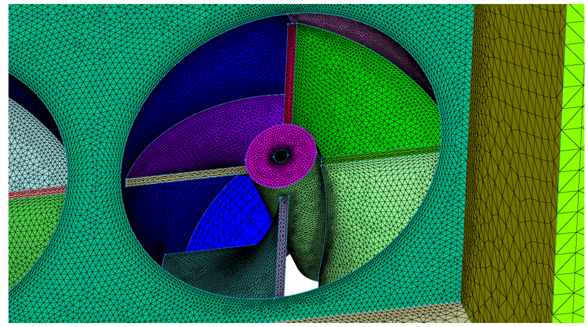

Figure 19. Isolated view of one fan.

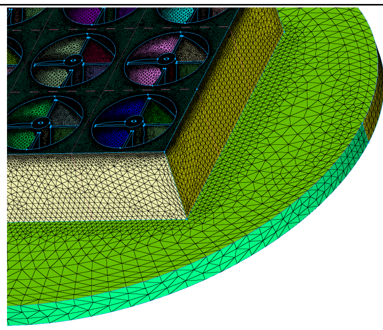

Figure 20. T-Rex meshing on edge of plate.

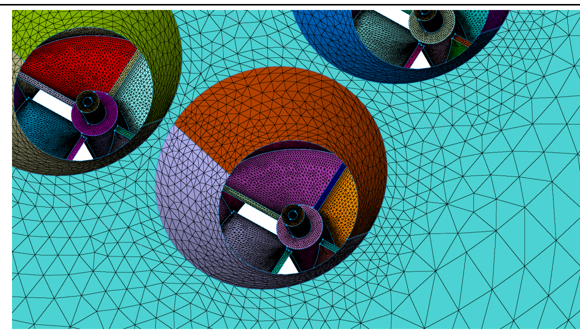

Figure 21. Bottom view of single fan pathway.

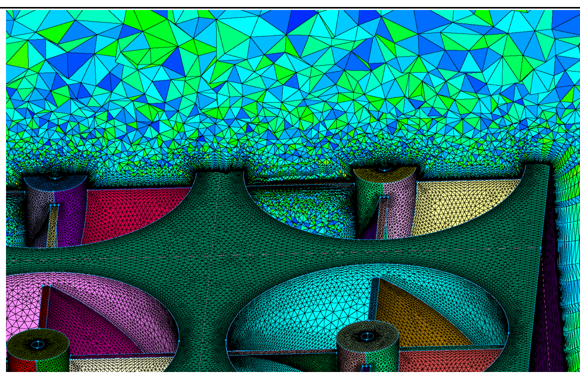

Figure 22. Crinkle surface of volume mesh above fan assembly.

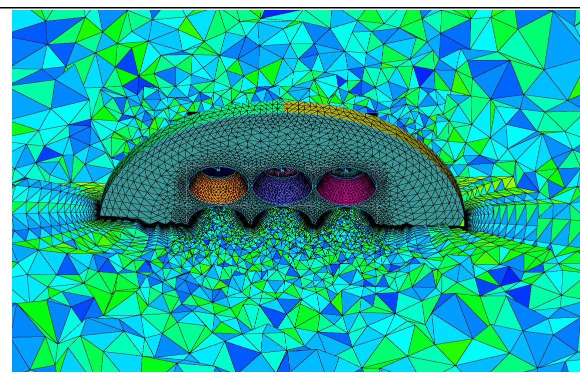

Figure 23. View of crinkle surface from below.

E. John's Made up Rocket 3

A complex geometry from the ESP gallery of cases is the JMR3. The geometry is shown in Figure 24. Various images of the surface mesh details are shown in Figure 25 through Figure 31. All fine length scales in the geometry are captured in the final mesh. The interior of the model was meshed with isotropic tetrahedra (not shown). No surfaces were tagged with a wall spacing value, so the resulting mesh did not contain any TRex extruded cells.

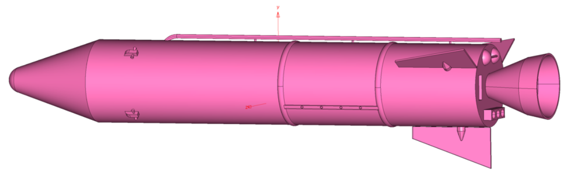

Figure 24. JMR3 geometry with many appendages.

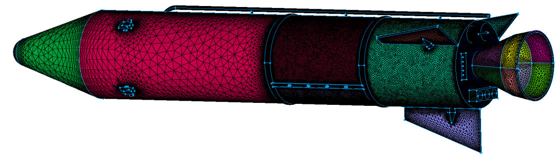

Figure 25. Auto-generated surface meshes on JMR3.

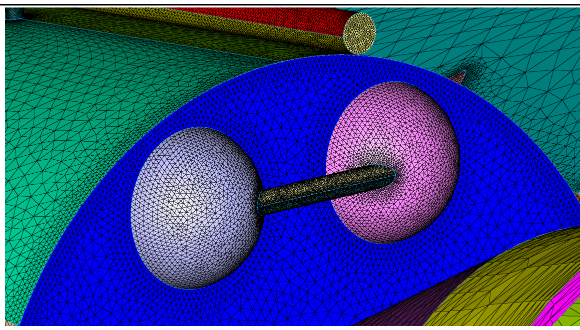

Figure 26. Detailed view of rocket base.

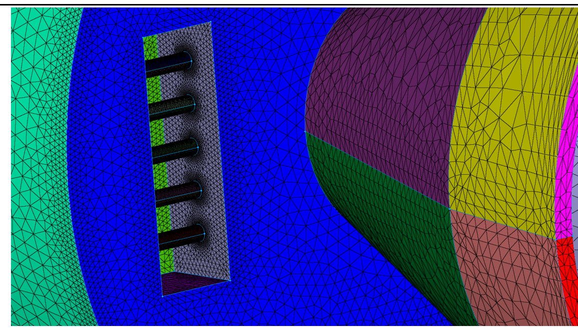

Figure 27. Detailed view of base coolers.

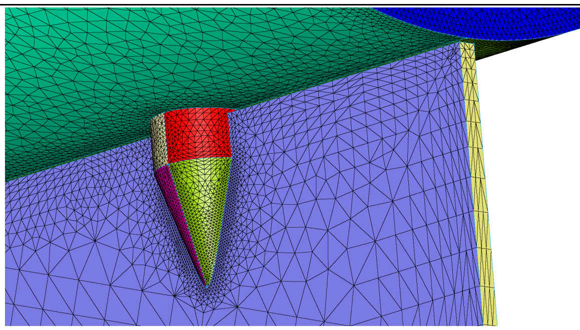

Figure 28. Fin assembly near pivot point.

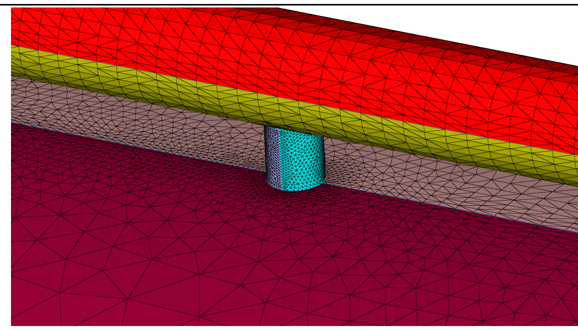

Figure 29. Vent line with supports.

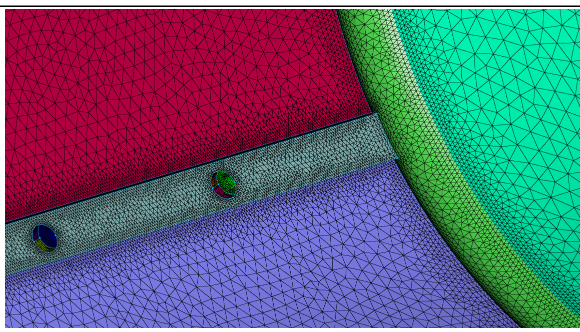

Figure 30. Stiffening ring with holes.

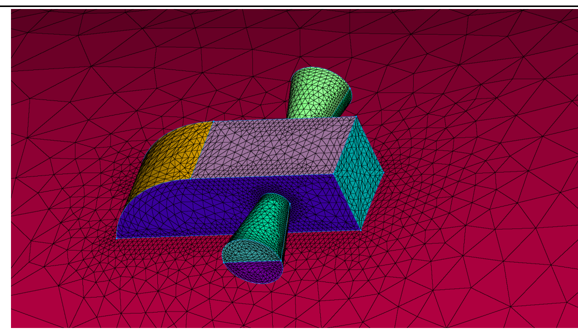

Figure 31. Thrusters on side of rocket.

F. Generic Lander

Another complex case in the ESP gallery is a generic lander. The geometry is shown in Figure 32. The colors indicate the different geometry faces, which will become surface meshes (domains). Surface meshes are shown in Figure 33 through Figure 35. All surfaces were tagged with a $PW:WallSpacing to force viscous layer extrusion in Pointwise. A slice through the resulting volume mesh is shown in Figure 36 and Figure 37. The colors indicate the element type, tetrahedra (red) and triangular prisms (yellow).

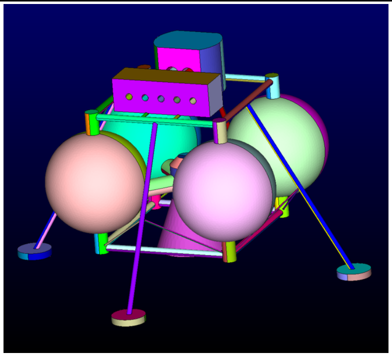

Figure 32. Generic lander geometry.

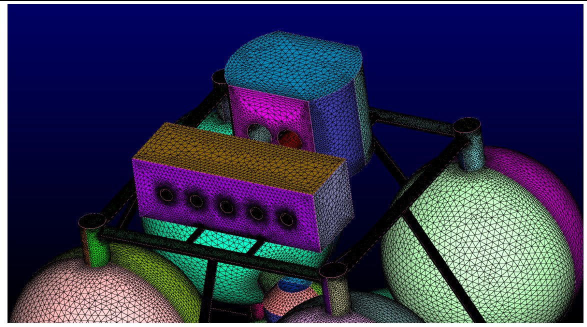

Figure 33. Surface meshes on the top side of the configuration.

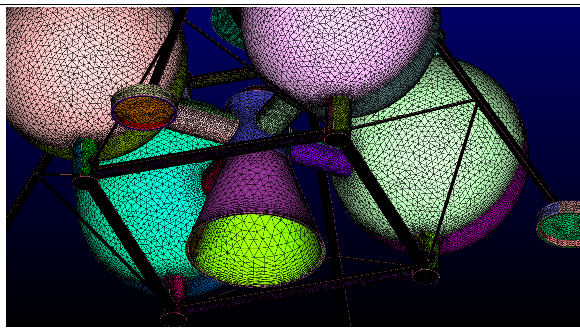

Figure 34. Surface meshes on the bottom side.

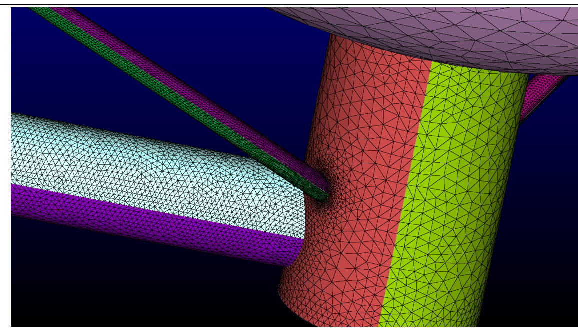

Figure 35. Magnified view of struts connecting to central posts.

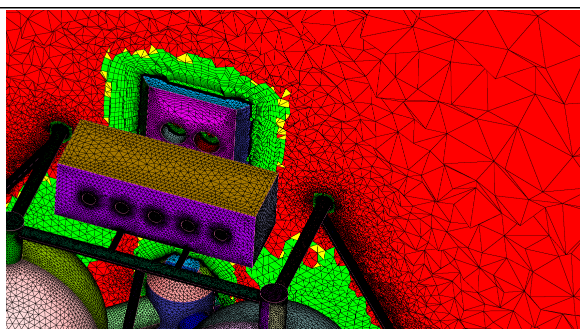

Figure 36. Crinkle surfaces of volume mesh viewed from above.

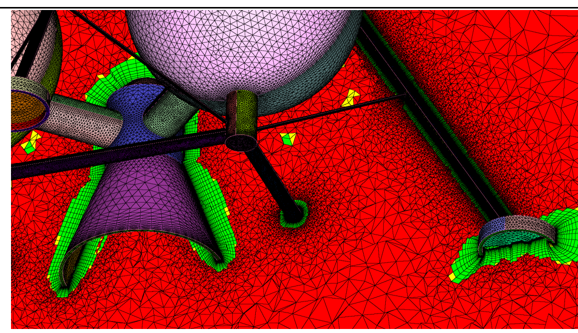

Figure 37. Crinkle surface of volume mesh viewed from below.

G. GMGW-2 Case 3

The final case is a parametric study that was part of the second Geometry and Mesh Generation Workshop held before the AIAA Science and Technology Forum and Exposition 2019. Case 3 of the workshop asked participants to perform a mesh parametric study of a notional commercial transport aircraft, the Open Parametric Aircraft Model #1 (OPAM-1). The baseline configuration, case 3a, is shown in Figure 38. Case 3b widened the fuselage. Case 3c increased the wing sweep. The pylon was moved to the wing break in Case 3d. The final configuration, Case 3e, narrowed the pod and shortened the pylon. The geometry files, created using ESP, have been attributed with boundary names and a specified wall spacing of 1.0e-04.

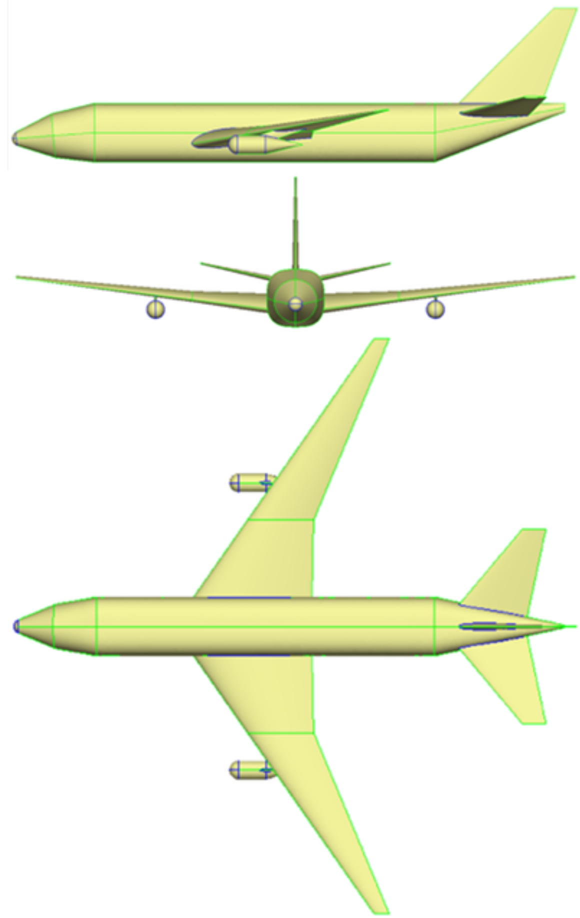

Figure 38. Baseline configuration for Case 3 of GMGW-2 (Case 3a).

The user defined input file specified a geometry curvature turning angle of 6 degrees and a hard edge angle of 30 degrees. The surface meshing algorithm was set to advancing front with a maximum of 10 TRex layers and a maximum aspect ratio limit of 10. The volume isotropic mesh algorithm was set to voxel with 5 transition buffer layers. TRex extruded the triangular surface mesh into prism elements. A source box was specified with a length scale of 5 in the +X direction, a widening angle of 10 degrees and a size growth factor of 20.

The surface mesh for Case 3a is shown in Figure 39. The red colored connectors along the leading edges are the curvature based TRex edges and the yellow colored edges are the hard angle TRex boundaries. All pertinent geometry features are well resolved. A crinkle cut through the centerline of the volume mesh near the vehicle is shown in Figure 40. A side view of the source box for the isotropic portion of the volume mesh is shown in Figure 41. An expanded view of the crinkle cut through the centerline, shown in Figure 42, reveals the effect the sources have on the mesh resolution.

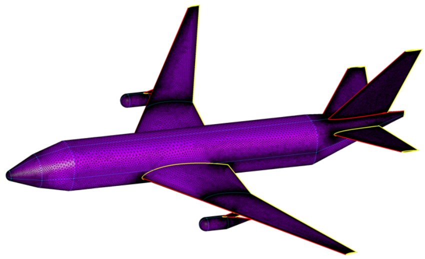

Figure 39. Surface mesh for Case 3a.

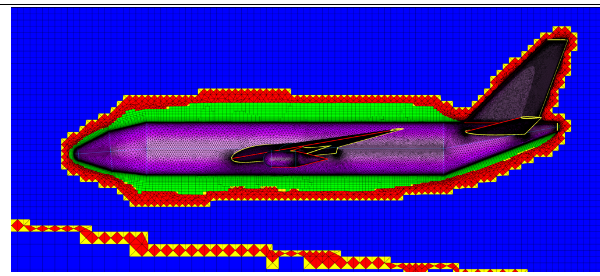

Figure 40. Crinkle cut through centerline of Case 3a.

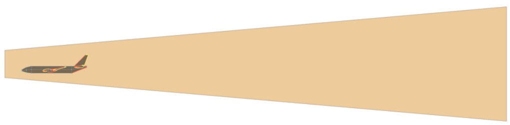

Figure 41. Source box automatically generated for Case 3a.

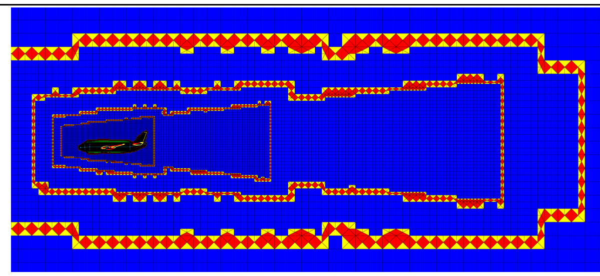

Figure 42. Crinkle cut through centerline showing effect of source box.

The same surface mesh views for Case 3b through Case 3e are shown in Figure 43 through Figure 46. All cases were made using the same user defined file with the appropriate EGADS file. The volume meshes for these cases were similar to the views shown in Figure 40 and Figure 42.

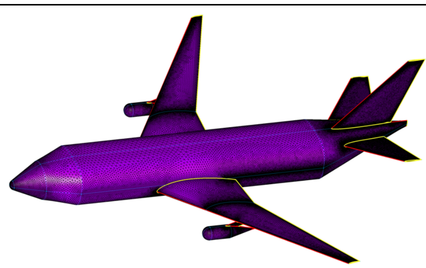

Figure 43. Surface mesh for Case 3b.

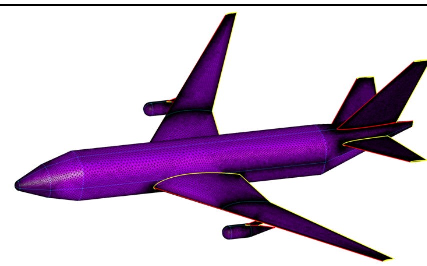

Figure 44. Surface mesh for Case 3c.

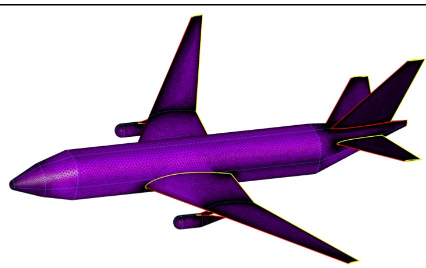

Figure 45. Surface mesh for Case 3d.

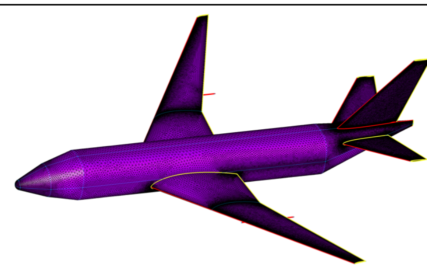

Figure 46. Surface mesh for Case 3e.

Movement of the pylon to the wing break in Case 3d did not present a problem for the AutoMesh scripts, shown in Figure 47. All features are well resolved with smooth spacing variation across the surface mesh. Narrowing of the pod to a slender missile shape also did not present a problem, shown in Figure 48. The pod was very well resolved as the surface curvature forced the scripts to increase the mesh density along the length of the pod.

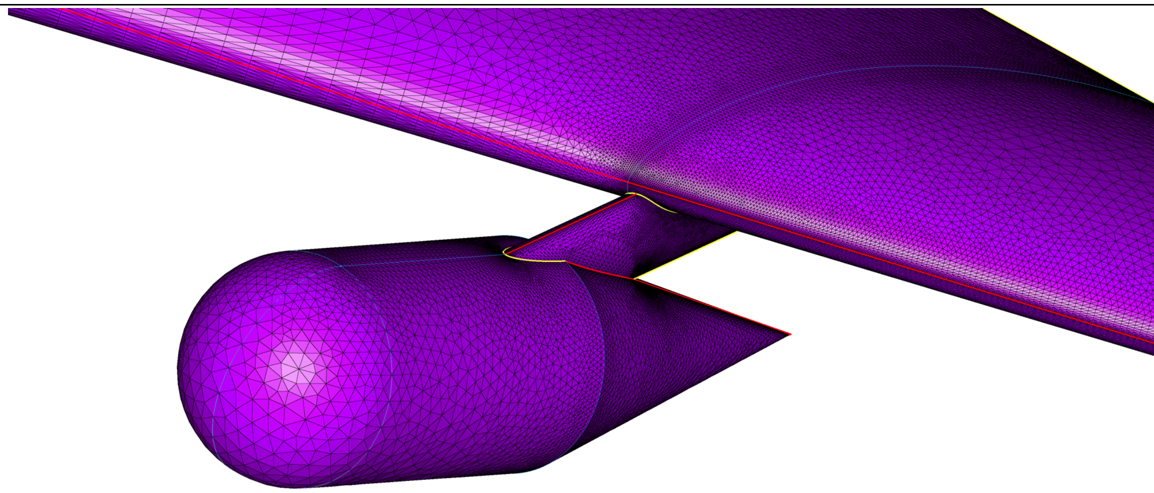

Figure 47. Surface mesh for Case 3d with pylon at wing break.

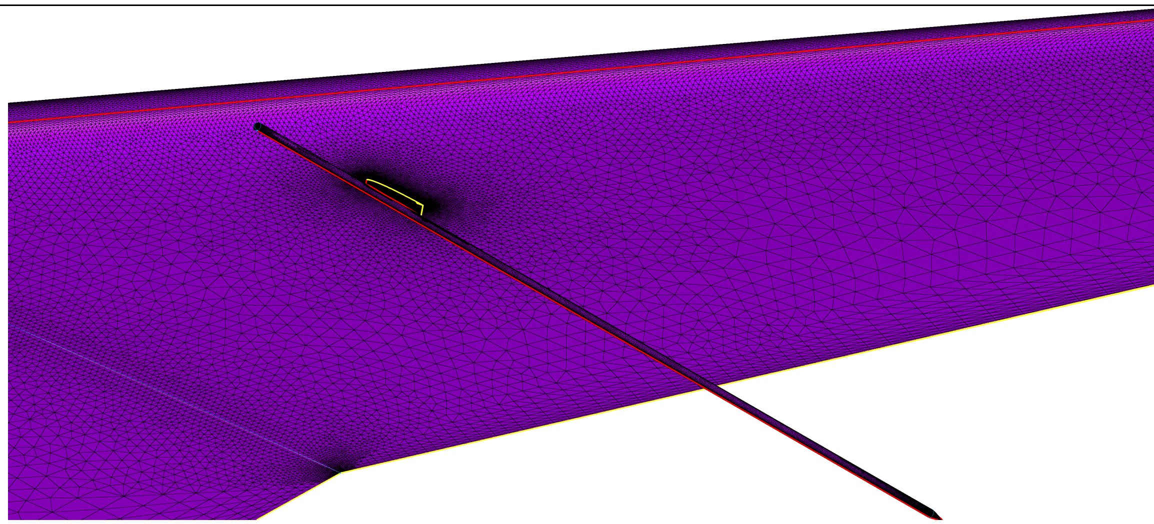

Figure 48. Surface mesh for Case 3e with narrowed pod and shortened pylon.

Statistics for these five OPAM-1 meshes are shown in Table 3. The meshes were created on an Apple iMac workstation with 32 G-Bytes of RAM and a 4 GHz Intel Core i7 processor. Meshes Case 3a through 3d are very similar in size. Case 3e is considerably larger due to the fine spacing on the narrow pod which also increased the number of tetrahedra and prisms. Case 3b had a significant reduction in the number of hexahedra, due to the widened body occupying significantly more volume, filling more space with triangular prisms. All but Case 3e finished in approximately 45 minutes. The increase in computer time for Case 3e is due primarily to the extra time spent in creating the TRex layers.

Table 3. Mesh statics for OPAM-1 cases

Case	Wall Time	# Nodes	# Tetrahedra	# Pyramids	# Prisms	# Hexes
3a	42 Min 13 Sec	13,597,307	5,194,765	428,634	22,497,620	1,212,194
3b	46 Min 2 Sec	13,431,917	5,655,923	385,489	24,841,395	798,666
3c	45 Min 3 Sec	13,814,165	5,311,854	438,012	22,848,728	1,228,503
3d	41 Min 7 Sec	13,706,073	5,404,681	432,068	22,641,490	1,212,480
3e	1 Hr 31 Min 15 Sec	16,462,815	7,372,674	494,774	27,372,696	1,212,879

V. Conclusions

A Glyph script, named AutoMesh, has been developed for the Pointwise mesh generation software. The script attempts to automatically create a volume mesh for geometry input in one of four formats, IGES, STEP, EGADS or NMB. If the geometry is fully closed and water-tight then a volume mesh is constructed. If the geometry is not closed, then just the surfaces are meshed. Attribution is possible using the EGADS and NMB file formats, via tagging performed in the ESP geometry creation software or within the Pointwise GUI. With attribution additional instructions can be provided to Pointwise as the mesh is created. The attributes can include desired connector dimensions, end points spacing, boundary naming, viscous wall spacing and many more. Many of the options available to an interactive Pointwise user are available via attribution.

The mesh generation process follows best practices as the various steps are performed. These best practices are followed completely by the script. An interactive user would probably not be as diligent in adhering to the rules. By following the rules, the resulting connector distributions result in quality surface meshes, which then produce quality volume meshes. Examples were included that demonstrate the utility of AutoMesh at generating high-quality surface meshes for cases with incomplete geometry and high-quality volume meshes for case with complete geometry. Attribution was used in some cases to produce meshes suitable for viscous simulations. The final case demonstrates the utility of AutoMesh at creating five high quality viscous meshes for a parametric geometry study.

VI. Acknowledgement

Funding for development of the automatic mesh generation capability through Glyph scripting in Pointwise was provided by Massachusetts Institute of Technology as a subcontract of the U. S. Air Force Contract FA8650-14-C-2472 “Computational Aircraft Prototype Syntheses (CAPS)”. This support is greatly appreciated.

References

[1] Christian Allen. "Automatic Structured Multiblock Mesh Generation Using Robust Transfinite Interpolation", 18th AIAA Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences, (AIAA 2007-3833) https://doi.org/10.2514/6.2007-3833

[2] Marcello Ferrari. "Automatic Method for Multiblock Structured Grid Generation", 2018 AIAA Aerospace Sciences Meeting, AIAA SciTech Forum, (AIAA 2018-1500) https://doi.org/10.2514/6.2018-1500

[3] Douglas Ross, Yasushi Ito, Fredric Dorothy, Alan Shih, and John Peugeot. "Automatic Mesh Generation of Hybrid Mesh on Valves in Multiple Positions in Feedline Systems", 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, Aerospace Sciences Meetings, (AIAA 2010-163) https://doi.org/10.2514/6.2010-163

[4] John H. Bucklow, Robin Fairey, and Mark R. Gammon. "An automated workflow for high quality CFD meshing using the 3D medial object", 23rd AIAA Computational Fluid Dynamics Conference, AIAA AVIATION Forum, (AIAA 2017-3454) https://doi.org/10.2514/6.2017-3454

[5] Shishir Pandya, William Chan, and James Kless. "Automation of Structured Overset Mesh Generation for Rocket Geometries", 19th AIAA Computational Fluid Dynamics, Fluid Dynamics and Co-located Conferences, (AIAA 2009-3993) https://doi.org/10.2514/6.2009-3993

[6] William M. Chan. "Strategies Toward Automation of Overset Structured Surface Grid Generation", 23rd AIAA Computational Fluid Dynamics Conference, AIAA AVIATION Forum, (AIAA 2017-3451) https://doi.org/10.2514/6.2017-3451

[7] Steve Karman. "Hierarchical Unstructured Mesh Generation", 42nd AIAA Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings, (AIAA 2004-0613) https://doi.org/10.2514/6.2004-613

[8] Steve Karman and Vincent Betro. "Parallel Hierarchical Unstructured Mesh Generation with General Cutting", 46th AIAA Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings, (AIAA 2008-0918) https://doi.org/10.2514/6.2008-918

[9] M. Aftosmis, M. Berger, J. Melton, M. Aftosmis, M. Berger, and J. Melton. "Robust and efficient Cartesian mesh generation for component-based geometry", 35th Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings, (AIAA 1997-0196) https://doi.org/10.2514/6.1997-196

[10] Lee Kania and Shahyar Pirzadeh. "A Geometrically-Derived Background Function for Automated Unstructured Mesh Generation", 17th AIAA Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences, (AIAA 2005-5240) https://doi.org/10.2514/6.2005-5240

[11] Pointwise, Pointwise, Inc., http://www.pointwise.com

[12] Robert Haimes and John Dannenhoffer. "The Engineering Sketch Pad: A Solid-Modeling, Feature-Based, Web-Enabled System for Building Parametric Geometry", 21st AIAA Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences, (AIAA 2013-3073) https://doi.org/10.2514/6.2013-3073

[13] John Dannenhoffer and Robert Haimes. "Generation of Multi-fidelity, Multi-discipline Air Vehicle Models with the Engineering Sketch Pad", 54th AIAA Aerospace Sciences Meeting, AIAA SciTech Forum, (AIAA 2016-1925) https://doi.org/10.2514/6.2016-1925