Analysis of Unstructured Meshes from GMGW-1 / HiLiftPW-3

Carl Ollivier-Gooch*

Advanced Numerical Simulation Laboratory,

The University of British Columbia, Vancouver, BC, Canada

In 2017, AIAA conducted its inaugural Geometry and Mesh Generation Workshop (GMGW-1), in conjunction with its High-Lift Prediction Workshop (HiLiftPW- 3). In preparation for these workshops, several authors submitted candidate meshes for use by the HiLiftPW-3 participants. The GMGW-1 steering committee selected, from amongst the submissions, the baseline meshes that should be used by the HiLiftPW-3 participants. This selection was based upon the meshing guidelines published by the workshop organizers.

This paper describes the techniques developed to assess the degree to which each candidate grid adheres to the guidelines. The assessment includes a quantitative assessment of the distance of the surface mesh points to the actual geometry, statistics about the on-body and off-body spacings, mesh quality, and mesh stretching statistics. Simple checks, such as appropriate mesh connectivity and non-negative cell volumes, are also included. This paper describes these techniques and uses them to analyze both the baseline meshes supplied to HiLiftPW-3 participants and the meshes generated by participants independently.

I. Introduction

The CFD Vision 2030 Study[1] identified several grand challenges in computational fluid dynamics (CFD) that important to NASA's mission. The report also identified deficiencies in current tools and technologies that must be remedied to meet those challenges. Among those deficiencies was mesh generation, which was called out as requiring too much human time and intervention, and not yet being capable of generating the tera-element meshes the authors envision as necessary in the 2030 timeframe.

To establish a clear baseline for the current state of the art in geometry handling and mesh generation, the AIAA Meshing, Visualization, and Computational Environments technical committee (MVCE) held the First Geometry and Mesh Generation Workshop (GMGW-1), in the style of the highly successful Drag Prediction Workshops and High Lift Prediction Workshops, among others. The intention was that participants would generated meshes for a common geometry and report on the issues that arose, as well as providing meshes for simulation and analysis. Accordingly, the organizing committee for the workshop sought a geometry that is reasonably representative of real aircraft geometries and that is publicly accessible.

At about the same time, our colleagues in the Applied Aerodynamics technical committee were beginning preparations for the Third High Lift Prediction Workshop (HiLiftPW-3), and were interested in having meshes to distribute to participants in their workshop that matched their meshing guidelines and were known to be of reasonable quality. The MVCE meshing subcommittee took on this task for the High-Lift Common Research Model (HL-CRM) cases, including both mesh generation and assessment of candidate meshes for suitability. HiLiftPW-3 participants were free to generate their own meshes if they chose, with those meshes becoming part of GMGW-1. The two workshops were held together before AIAA's Aviation Forum in 2017.

To assess the meshes — both those generated by the committee and by the participants — separate tools were developed to measure structured and unstructured mesh quality. Preliminary versions of these tools, and the analysis results for the committee-generated meshes, were described in a previous paper,[2] and a complete analysis of the structured meshes submitted by participants in GMGW-1 and HiLiftPW-3 is given in a companion paper in these proceedings.[3] This paper focuses on assessment of both committee and participant-generated unstructured meshes for these workshops.

[*] Professor, Department of Mechanical Engineering, cfog@mech.ubc.ca. Associate Fellow, AIAA.

II. Assessment Methodology for Unstructured Meshes

The unstructured mesh assessment tool evaluates mesh spacing (both at the wall and on the surface), confirms topological and geometric consistency of the mesh, examines several mesh quality metrics, and determines the distance of surface mesh points from the underlying geometric surface. Mesh quality data is written as binned data and plotted. Spacing data is written to a VTK file containing only the surface mesh for visual inspection; for spacing normal to the wall, some statistics are also generated.

II.A. Preliminaries

The mesh files involved can be quite large. For instance, the committee-generated medium prism-tet mesh has about 26.5M vertices, 65M elements, 153M faces and a (binary) file size of about 1.8 GB; the extra-fine mesh in this sequence is about nine times as large. The memory requirement for analysis is larger, because more connectivity information is needed, and quantities derived from the mesh data must be stored as well. With data of this size, it is essential to minimize memory impact to the extent possible, even at the cost of reading data more than once to avoiding keeping it in memory.

To support this, the first step in analyzing a mesh is to bookmark locations in the file where the coordinate and connectivity data begin. At present, UGRID (binary) and VTK (ASCII) file formats are supported. For .ugrid files, the required locations can be calculated using only the mesh size information in the file header. For .vtk files, the file must be scanned by reading it, because these locations are not computable. In either case, these bookmarks can later be used to move immediately to the desired location in the file to read the necessary data.

As an additional pre-processing step, boundary vertices and faces are also identified. This information is stored for later use.

II.B. Mesh Quality

Geometric mesh quality measures are notoriously hard to interpret in terms of their impact on solution quality; high mesh anisotropy only exacerbates this problem, because high aspect ratios and small angles are quite common by design. We choose to assess the ratio of volumes of adjacent cells, the dihedral angles between faces, the angles between edges, and the distortion of non-planar quadrilateral faces.

II.B.1. Cell Volume Calculation

For tetrahedra, volume calculation is easy, because the volume is proportional to the triple product of the three edge vectors incident on a single vertex. We derive below an exact formula for the volume of a pyramid. To find the volume of prisms and hexahedra, we subdivide them into tetrahedra and pyramids by the addition of a point at the geometric average of the vertices of the cell. Note that this point is not added to the mesh; it is used only as an artifice to reduce all cell volumes to sums of volumes of tetrahedra and pyramids. Each prism is decomposed into two tetrahedra (one connecting each triangular end to the center point) and three pyramids (one connecting each lateral quadrilateral face to the center point). Each hexahedron is similarly divided into six pyramids. The choice of the geometric average for the center introduces the possibility that a highly distorted cell might have one or more subcells with a negative volume. For that matter, a sufficiently distorted cell might have an overall volume that is negative when measured in this way but not when computed in some other way. If any cell volumes are computed to be negative, an additional VTK file is written containing the cells whose volumes are negative for subsequent inspection.

All that remains is to calculate the volume of a pyramid analytically and consistently when its quadrilateral face is non-planar. Dividing each quadrilateral into two triangles by adding a diagonal is not sufficient. Even if the diagonal is chosen consistently, the resulting shape can cause spurious indication of negative cell volume. Instead, we require a curved representation of the quadrilateral that is independent of its orientation (which adjacent cell the quadrilateral is viewed from) and cyclic permutation of vertices (where we begin numbering the vertices of the quadrilateral). We define a mapping from a reference pyramid to a general pyramid as shown in Figure 1, where $\vec{\xi_0}$ maps to $\vec{x}_0$ and so on. The simplest mapping that does this exactly is bilinear in ξ and η, and linear in ζ:

$$\vec{x} = \vec{A} + \vec{B}\xi + \vec{C}\eta + \vec{D}\xi\eta + \vec{E}\zeta \tag{1}$$

where

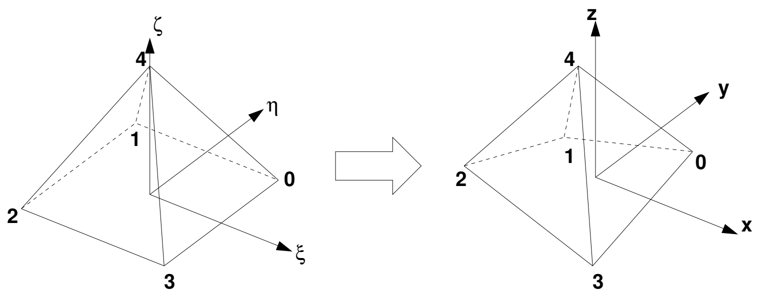


Figure 1. Mapping from reference to physical pyramid.

Point	Reference	Physical
0	(1,1,0)	$\vec{x}_0 = \vec{A} + \vec{B} + \vec{C} + \vec{D}$
1	(-1,1,0)	$\vec{x}_1 = \vec{A} - \vec{B} + \vec{C} - \vec{D}$
2	(-1,-1,0)	$\vec{x}_2 = \vec{A} - \vec{B} - \vec{C} + \vec{D}$
3	(1,-1,0)	$\vec{x}_3 = \vec{A} + \vec{B} - \vec{C} - \vec{D}$
4	(0,0,1)	$\vec{x}_4 = \vec{A} + \vec{E}$

Solving, we get:

$$\begin{aligned}
\vec{A} &= \frac{\vec{x}_0 + \vec{x}_1 + \vec{x}_2 + \vec{x}_3}{4} \\
\vec{B} &= \frac{\vec{x}_0 - \vec{x}_1 - \vec{x}_2 + \vec{x}_3}{4} \\
\vec{C} &= \frac{\vec{x}_0 + \vec{x}_1 - \vec{x}_2 - \vec{x}_3}{4} \\
\vec{D} &= \frac{\vec{x}_0 - \vec{x}_1 + \vec{x}_2 - \vec{x}_3}{4} \\
\vec{E} &= \vec{x}_4 - \vec{A}
\end{aligned}$$

This mapping satisfies our consistency requirements, because the shape of the quadrilateral base in the physical space does not change if we cyclically permute the vertices in the physical space, nor if viewed from the opposite side. The volume of this pyramid in the physical space can be found by computing

$$\begin{aligned}
|Pyr_{ABCDE}| &= \iiint_{phys} 1\, dV \\
&= \iiint_{ref} J\, dV
\end{aligned} \tag{2}$$

where J is the Jacobian of the change of variables, det $\left(\partial\vec{x}/\partial\vec{\xi}\right)$. For this bilinear transformation, we get:

$$J = \begin{vmatrix} \vec{B} + \vec{D}\eta \\ \vec{C} + \vec{D}\xi \\ \vec{E} \end{vmatrix} = \left(\vec{B} \times \vec{C} \cdot \vec{E}\right) + \left(\vec{B} \times \vec{D} \cdot \vec{E}\right)\xi + \left(\vec{D} \times \vec{C} \cdot \vec{E}\right)\eta$$

where the $\xi\eta$ term includes $\vec{D} \times \vec{D}$ and so is zero. Substituting this back into Equation 2, the terms involving ξ and η integrate to zero by symmetry, and we are left with

$$|Pyr_{ABCDE}| = \frac{4}{3}\left(\vec{B} \times \vec{C} \cdot \vec{E}\right)$$

where the constant is the volume of the reference pyramid. This formula gives us the exact volume of the bilinear pyramid without quadrature.

Having evaluated cell volumes, we can now check that all cells have positive volume, which is the case for all the meshes tested here. With the help of cell-to-cell neighbor connectivity, we can also compute the ratio of sizes of adjacent cells.[a] This is of particular interest for stretched anisotropic meshes, because we expect to see peaks in the size ratio distribution at specific values. For example, for a prism-dominant boundary layer, we expect that prisms sharing a quadrilateral face (that is, prisms in the same layer of the advancing layers mesh) will have a volume ratio that is almost exactly one, while prisms sharing a triangular face (that is, prisms in consecutive layers) will have a size ratio that is quite close to the prescribed stretching ratio.

II.B.2. Determining Cell-to-Cell Connectivity

Before calculating size ratios, however, we need to construct cell-to-cell connectivity, which both the UGRID and VTK file formats lack. The naive approach for this is to create a list of oriented tuples for the vertices of each face of each cell (tagged with the cell index), then sort these using standard library functions so that the two oppositely oriented copies of the same face are adjacent in sequence. Then the connectivity from cell to cell can be read directly from the cell indices associated with the two copies of each face. When memory allows, this approach is both simple and fast. However, memory usage can be quite large. Assuming that vertex and cell indices can be stored using 32-bit integers (that is, assuming fewer than $2^{32} \approx 4$ billion cells), each copy of a triangular face requires 20 bytes (24 for a quadrilateral face). For the committee-provided medium prism-tet mesh, this adds up to a tractable total of around 6 GB of data. For the extra-fine mesh of the same family, however, the total climbs to around 50 GB.

To create cell-to-cell connectivity with a smaller memory footprint, we take advantage of the functionality of the set container in the C++ Standard Template Library. Each boundary face and interior cell is read in turn, and its constituent triangles and quadrilaterals are tested against a set of previously unmatched faces of that topology. If a face is in the set (that is, if it has been seen before), then the current face and its mate in the set are used to create a new cell-to-cell connectivity entry, and the copy in the set is removed. If a face is not already in the set, it is added to the set. Despite storage overhead in the set, this approach requires substantially less memory than the naive implementation, because there is no need to store connectivity data for all faces at once. This comes at the cost of substantially higher runtime for a given mesh size.

As a side effect of this process, we can easily confirm that the mesh topology is valid: if it is, then every face will have a counterpart, leaving no unmatched faces at the end of the process.

II.B.3. Distortion of Quadrilateral Faces

Ideally, the quadrilateral faces in these mixed-element meshes will be almost exactly flat. If we treat these quads as nearly-degenerate tetrahedra, there are as many ways to assess quadrilateral flatness as to assess degeneracy of tetrahedra, the only difference being that in this case we actually want the quad to be flat (the tet to be degenerate). The method we choose here is a non-dimensional volume to surface area measure. We interpret a non-planar quadrilateral $\square 0123$ as a tetrahedron with faces $\triangle 012$, $\triangle 013$, $\triangle 123$, and $\triangle 203$. Computing the volume of the tetrahedron and the areas of these four triangles (and treating all these quantities as positive), we can then compute the distortion measure D for a quadrilateral as:

$$D = \sqrt{\frac{27\sqrt{3}}{8}} \frac{V}{\left(\frac{1}{4}\sum A_i\right)^{3/2}}$$

where the denominator includes the average area of the four faces. D is zero for any planar quadrilateral and is unity for an equilateral tetrahedron — which would be a very badly non-planar quadrilateral indeed! Anticipating that most quadrilateral faces will be close to planar, we will produce histograms of this data on a logarithmic scale.

II.B.4. Edge and Dihedral Angle Checks

Computing angles between edges that meet at a vertex requires only the standard calculation of angle between two vectors. Dihedral angles, however, are more challenging because non-planar quadrilaterals will produce varying dihedral angles with the faces they share an edge with. To provide a measure of the typical

[a] Note that this ratio is computed as the size of the smaller cell divided by that of the larger cell, and so always falls between 0 and 1.

angle between normals for two faces sharing an edge across the entirety of the faces, we use the dihedral angle at the center of non-planar quadrilateral faces. More precisely, we find the normal at the point in physical space to which the parametric point $(0, 0, 0)$ maps. This normal is defined by

$$\begin{aligned} \frac{d\vec{x}}{d\xi} \times \frac{d\vec{x}}{d\eta} &= \left(\vec{B} + \vec{D}\eta\right) \times \left(\vec{C} + \vec{D}\xi\right) \\ &= \vec{B} \times \vec{C} \end{aligned}$$

when evaluated at $\xi = \eta = 0$. Both the edge and dihedral angles are reported using histograms.

II.C. Mesh Spacing Checks

The meshing guidelines provided by the High Lift Workshop committee specify normal spacing at the wall, spacing on the airplane surface, and stretching ratio. The last of these is not so easy to assess directly in an unstructured mesh, but the size ratio of adjacent cells discussed above provides an indirect measure.

To provide the wall normal and tangential spacings, we scan the data file and identify each boundary vertex. Next, we scan the mesh connectivity information. For each boundary vertex found in a given cell, we compute the distance from that vertex to the nearest boundary and interior vertex in the same cell.[b] The minimum distance over all cells can be identified on the fly from the per-cell data.

II.D. Geometry Conformance Checks

Boundary vertices in the mesh should lie on the surface geometry. Verifying this requires using a geometry kernel to project mesh boundary vertices onto the geometry, producing statistics about errors along the way. For this work, we used the Pointwise Geode geometry kernel,[4] because it was readily available, as was a surface geometry representation that Geode can read.

Regardless of what geometry kernel had been chosen, the same problem inevitably occurs: a mesh that has been properly generated using that kernel has all of its points precisely on the surface,[c] while another mesh generated using the same geometry input file but a different geometry kernel may have points that lie some distance away from the surface. Differences may also arise because of the difference between the actual geometry specified for the same nominal geometry when saved in different file formats. Differences in geometry cleanup (manual or automatic) can also play a role, as can differences in projection tolerances. Finally, for points on parts of the surface with two nearly coincident overlapping surfaces, which surface the mesh point is projected to can cause an error much larger than typical projection errors.

III. Results

Seventeen sets of meshes were analyzed for the current paper, both for quality and for conformance to the HiLiftPW-3 meshing guidelines[5] (referred to below as "the guidelines"). These include nearly all of the unstructured meshes submitted by participants at the workshops. Seven of these (including the three related committee meshes) were generated using Pointwise; two each using ANSA, Capstone, and AGPS/AFLR3; and one each using Rhinoceros, Solar, HeldenMesh, and Symmetrix. It is worth mentioning that only HeldenMesh appears to have any direct linkage to VGRID, once a workhorse for advancing layer mesh generation in 3D. Table 1 summarizes the contributed meshes and their sizes. It is worth noting that several mesh families — i, j, and s — do not follow the recommended mesh growth rate of about a factor of three in entity count between mesh levels.

In three cases (the two Bombardier mesh sets and the KHI contribution), the mesh was actually a multiblock unstructured mesh; the inner block was extracted and analyzed, with no quality assessment performed on the far-field block.

There were three mesh sets that were not analyzed. A set of high-order meshes was submitted, but not analyzed because straight-mesh quality assessment is not appropriate for curved meshes. A set of overset unstructured meshes was generated using the near-field part of one of the Pointwise mesh sets that *were* analyzed; since the analysis focuses on near-field mesh features, these meshes would have given duplicate

[b] It is possible that, within a given cell, a boundary vertex may have only boundary or only interior neighbors.

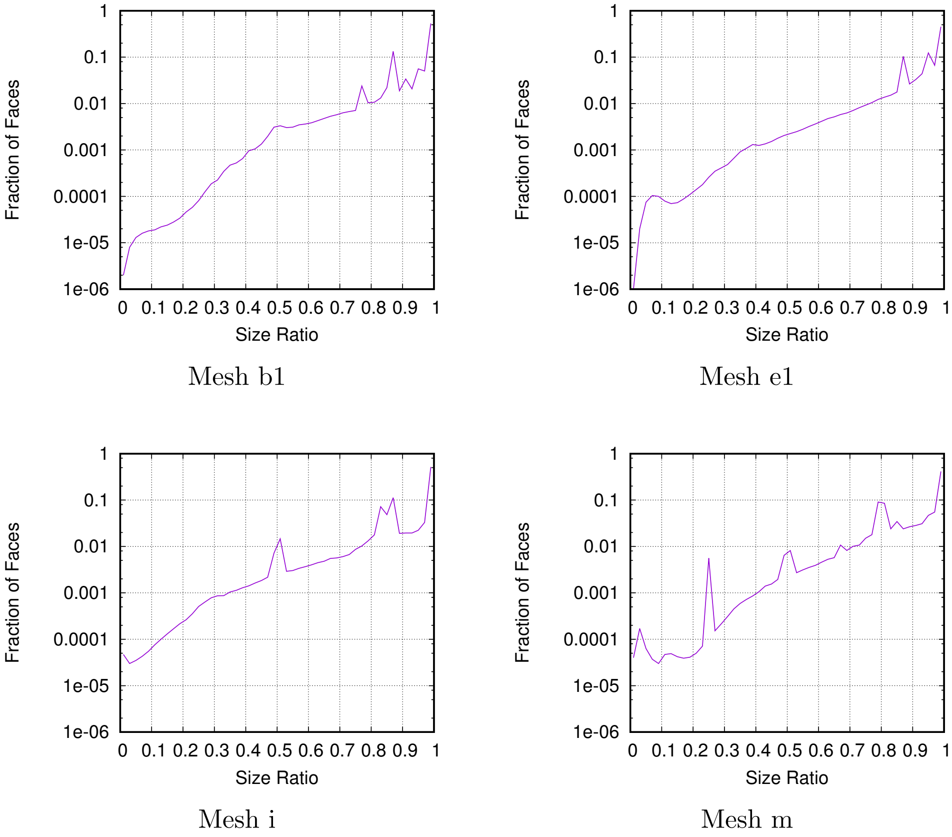
[c] To within the projection error tolerance and/or file I/O precision.

results. Finally, one set of meshes was submitted in Fluent mesh file format instead of the mandated CGNS or UGRID formats; without access to a translator or reader for these files, they were not analyzed.

Finally, although analysis was run on each of the meshes shown in the table, results are presented here only for the medium mesh in each family.

III.A. Mesh Quality

III.A.1. Cell Volume Ratios



Mesh b1

Mesh e1

Mesh i

Mesh m

Figure 2. Size ratio between adjacent tetrahedra

There are actually twelve different potential cell volume ratios that one can examine, ringing the changes on all possible cell types on either side of a triangular or quadrilateral face. Most of these, however, yield unsurprising and uninteresting conclusions like "adjacent pyramids and tetrahedra are often a factor of two different in volume", and so are not discussed here.

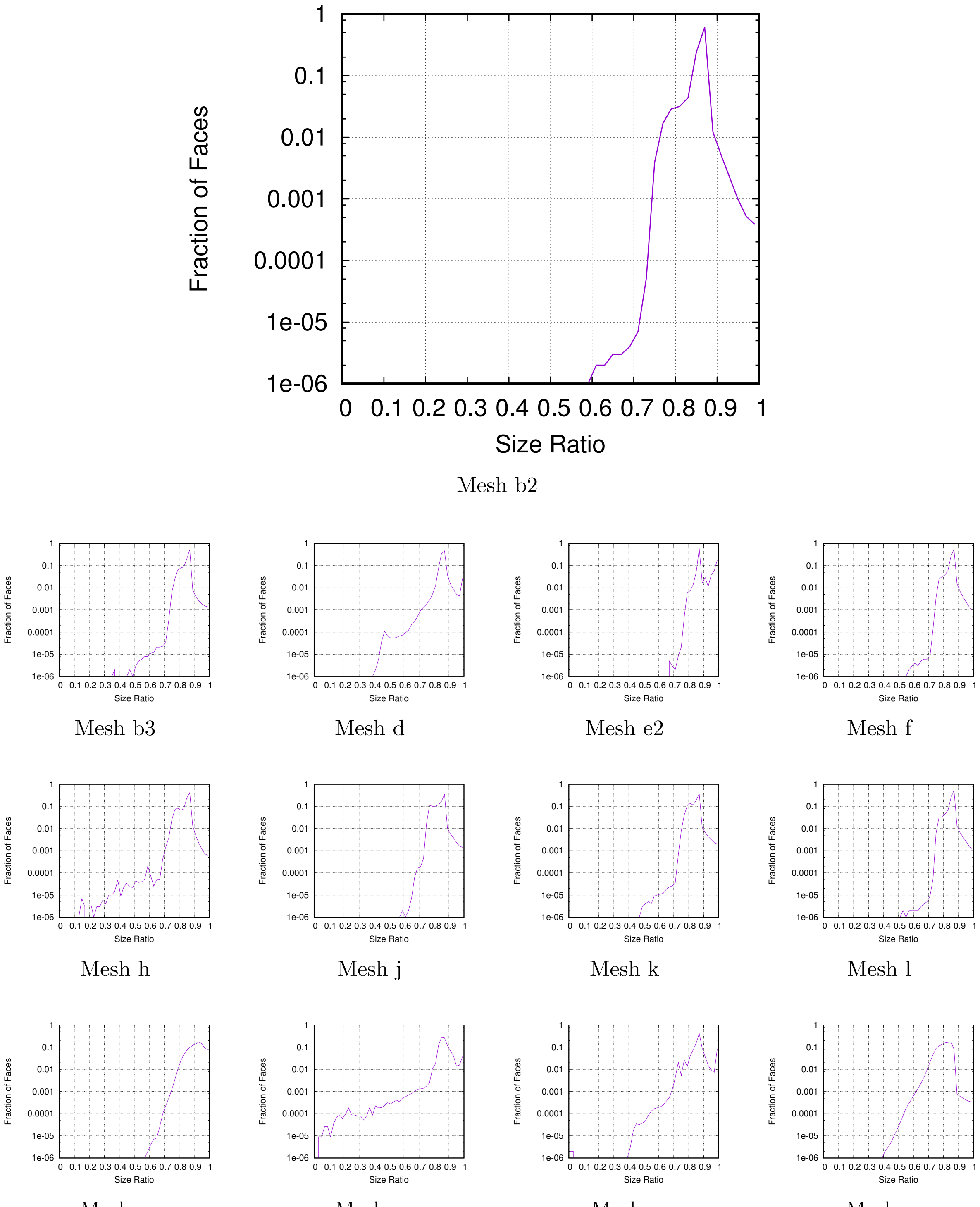
Of more interest are the results that show stretching ratio for anisotropic meshes. Figure 2 shows the size ratio between adjacent tetrahedra for all-tet meshes. We expect to see many faces with nearly unit size ratio, which should occur in isotropic regions of the mesh and between tets in the same layer of an advancing layer mesh; and a smaller concentration of faces with size ratio equal to the stretching ratio in the boundary layer. For medium meshes, the guidelines mandate a stretching ratio of 1.16, which corresponds to a size ratio of $1/1.16 = 0.86$. As the figure shows, most meshes match this closely; Mesh m stretches somewhat more quickly than the guidelines suggest.

Figure 3 shows analogous results for prismatic layers in a boundary layer, for all thirteen meshed containing prisms; Mesh b2 is typical of these. In all but one case, the peak in size ratio again corresponds reasonably well to the expected stretching. Mesh n has a less sharp peak in size ratio than the other meshes, and at a different size ratio; however, this prism-hex mesh has only about 4% as many triangular faces separating prisms as quadrilateral faces separating hexahedra, so this result may not be particularly indicative.

Tool Used	Organization of User	Label	Cell Types	Coarse Mesh					Medium Mesh					Fine Mesh					Extra-Fine Mesh				
				#V	#T	#Py	#Pr	#H	#V	#T	#Py	#Pr	#H	#V	#T	#Py	#Pr	#H	#V	#T	#Py	#Pr	#H
Pointwise	Pointwise	b1	T	8.09	47.8				26.5	157				69.9	416				206	1228			
	Pointwise	b2	TPP	8.09	9.58	0.47	12.4		26.5	17.7	1.39	45.6		69.9	45.1	2.26	122		206	196	4.52	341	
	Pointwise	b3	TPPH	8.30	10.0	1.54	0.95	5.52	27.0	18.7	3.90	5.41	19.6	70.7	46.8	6.55	10.3	55.1	208	203	12.9	24.4	157
	Bombardier 1	f	TPP	9.37	6.34	0.16	16.3		21.7	12.5	0.31	38.4		53.3	27.8	0.61	96						
	ONERA	h	TPPH	5.01	2.68	0.83	0.30	4.05	11.3	7.29	1.37	0.82	9.08										
	Auburn	k	TPPH	13.8	13.4	2.66	1.71	9.58	42.4	30.7	5.70	5.30	32.3	118	72.4	10.4	9.2	96.7					
	Bombardier 2	l	TPPH	7.97	4.23	0.20	13.4	0.35	20.9	12.4	0.29	37.0		65.9	30.0	1.03	111	4.59					
ANSA	BCAE-Oxford	d	TPPH	46.4	30.9	0.75	34.1	23.3	81.5	40.7	0.97	6.29	42.1	137	52.5	1.15	112	71.0					
	Embraer	r	TPPH	27.7	10.6	0.70	4.67	23.0	93.0	33.2	1.66	11.3	80.4	298	103	3.69	27.2	264					
AGPS / AFLR3	Boeing	e1	T	15.7	93.2				41.9	249				108	643								
	Boeing	e2	TPP	15.7	16.3	0.10	25.4		41.8	35.7	0.20	70.9		108	66.5	0.56	192						
Capstone	NRL[a]	i	T	5.48	32.4				18.6	110													
	NRL	j	TPP	6.52	3.57	0.10	11.6		64.9	62.3	0.69	107											
Symmetrix	UC Boulder	m	T	7.15	42.0				21.7	126													
Rhinoceros	KHI[b]	n	PH	58.8			6.38	53.5	176			15.3	164										
Solar	DLR[c]	p	TPPH	11.5	8.93	0.32	0.05	9.63	36.4	24.0	0.74	0.12	31.6	123	65.1	1.70	0.30	111					
Helden Mesh	Gulfstream	s	TPP	33.3	60.6	0.25	44.9		41.4	76.0	0.33	55.5		54.5	101	0.41	72.8		77.9	146	0.55	103	

Table 1. Meshes analyzed and their size. Entity counts are given in millions.

[a] Naval Research Laboratory
[b] Kawasaki Heavy Industries
[c] Deutsches Zentrum für Luft- und Raumfahrt



Mesh b2

Mesh b3

Mesh d

Mesh e2

Mesh f

Mesh h

Mesh j

Mesh k

Mesh l

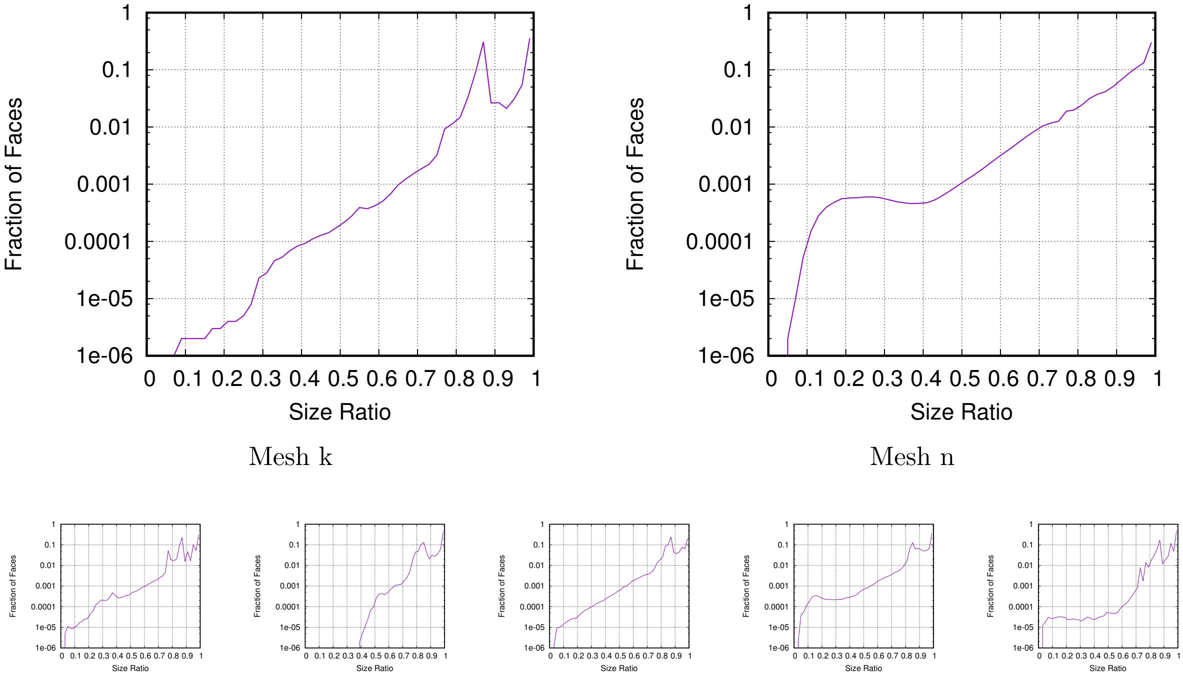
Mesh n

Mesh p

Mesh r

Mesh s

Figure 3. Size ratio between adjacent prisms through triangular faces



Mesh k

Mesh n

Mesh b3

Mesh d

Mesh h

Mesh p

Mesh r

Figure 4. Size ratio between adjacent hexahedra

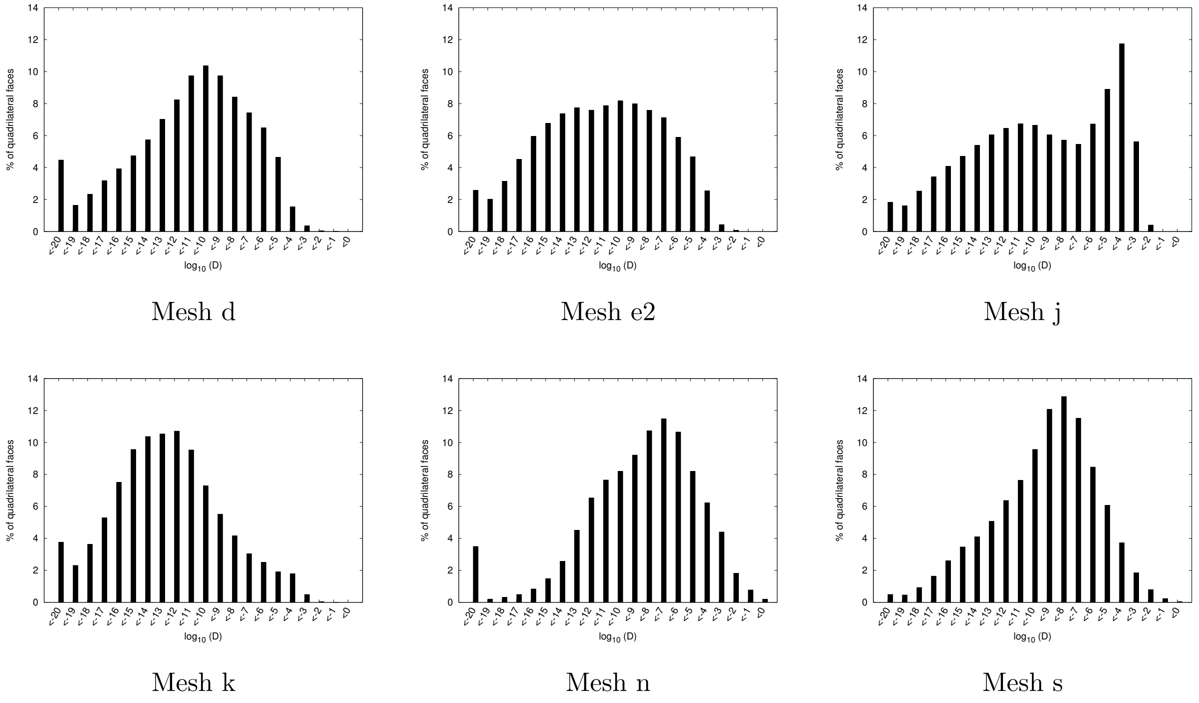
For the size ratio of adjacent hexahedra, shown in Figure 4, we expect to see distributions that are qualitatively similar to those for tetrahedra, for similar reasons. Six of the seven meshes with hexes follow this pattern, with an inferred stretching ratio in line with the guidelines. Mesh n has very distinctly different behavior, suggesting that this mesh is not stretched in the conventional way moving away from the wall; the large size of these meshes (the medium mesh UGRID file is 9.5GB) made it prohibitive to manually explore this mesh to learn more.

The only surprising result from the other size ratio data is that 30% of all triangular faces separating a prism and a tetrahedron in Mesh j have a size ratio below 0.04! This is a surprisingly large number of such faces. It appears, based on the mesh imprint on the symmetry plane, that the anisotropic layers in this mesh do not march out to isotropy, resulting in isotropic tetrahedra adjacent to medium-aspect ratio prisms; the tetrahedra appear to be the larger cell in these cases. The companion all-tet Mesh i has the same spacing characteristics, but this shows up only as a small blip in the graph in Figure 2, simply because there are so many other triangles washing out this problem.

III.A.2. Distortion of Quadrilateral Faces

Examination of the distortion of the quadrilateral faces in these meshes revealed that there are very few faces with significant distortion, as shown in Figure 5. The thirteen meshes with quadrilateral faces included some clusters of similar distributions that seem to be based more on mesh type than on which mesh generator was used. For instance, Meshes d, h, p, and r are all hex-dominant in the boundary layer and show similar profiles for face distortion despite coming from three different mesh generators; other hex-dominant meshes generated using Pointwise (b3, k) are more similar to these than to the prism-dominant meshes from Pointwise, which instead show results more similar to the AGPS/AFLR3-generated Mesh e2.

Finally, recall that the distortion measure used is a volume-surface area ratio. Starting with a unit square and lifting one corner straight up to distort the quadrilateral, a distortion of 0.1 is reached at a height of about 0.015; since the distortion measure is scale invariant, movement scales down with face size. Only two meshes — n and s — have more than 0.001% of faces with even this small level of distortion.



Mesh d

Mesh e2

Mesh j

Mesh k

Mesh n

Mesh s

Figure 5. Distortion of Quadrilateral Faces. Results were similar for meshes (d,h,p,r), for (b3, k) and for (b2, e2, f, l).

III.A.3. Edge and Dihedral Angles

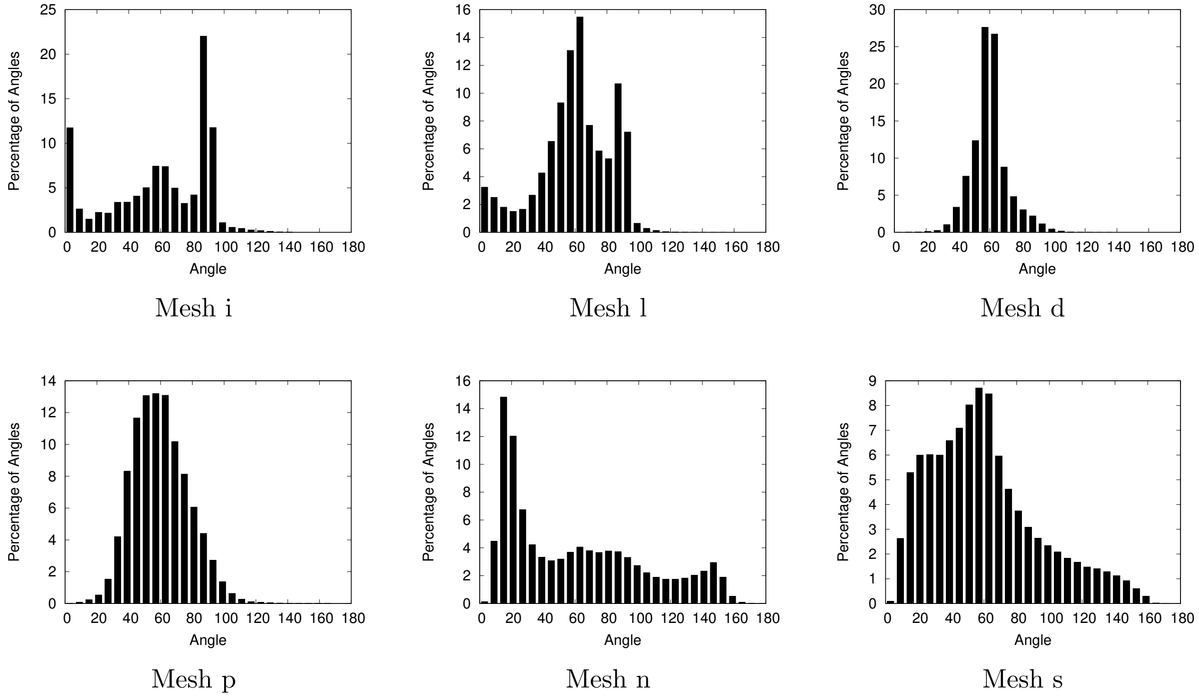
The behavior of face angles for triangular faces (see Figure 6) shows similarities between meshes that are somewhat familiar from examining size ratios. We would expect the all-tet meshes (b1, e1, i, m) to have many triangles with two right angles and one near zero, and a smaller number of isotropic triangles, and this is what we see. The meshes with prism-dominant boundary layers (b2, e3, f, j, l) are similar, but with fewer ϵ, 90° 90° triangles because the prisms lack the internal triangles that the tetrahedral meshes add. The meshes with hex-dominant boundary layers show two distinct behaviors, with the ANSA meshes (d, r) producing much more nearly equilateral triangles than the others (Pointwise (b3, h, k), and Solar (p)). Again, Meshes n and s are outliers. For Mesh s, this appears to be because the wing trailing edge features a surface mesh that looks like a scaling of equilateral triangles, producing many triangles with angles of ϵ, ϵ, and $180° - 2\epsilon$.

The face angles for quadrilateral faces are completely predictable: nearly all of those faces are nearly perfect rectangles, as expected from the way these meshes are constructed.

The dihedral angles between triangular faces all arise in tetrahedra or pyramids; as the cell counts in Table 1 indicate, nearly all of these come from tetrahedra. These distributions are shown in Figure 7. Again, the all-tet meshes produce similar distributions of these dihedral angles, dominated by the behavior of high-aspect ratio tetrahedra in the boundary layer. There is a second cluster of similar distributions from the Pointwise mixed-element meshes, which seem to contain a respectable number of tetrahedra or pyramids with nearly right dihedral angles. The other meshes show a tri-tri dihedral angle consistent with isotropic meshes.

The behavior of dihedral angles for edges between quadrilaterals and triangles (almost exclusively in prisms) reveal another similarity in the behavior of meshes with hex-dominant boundary layers generated by Pointwise and Solar (see Figure 8). These four meshes all show a tail of large quad-tri dihedral angles. The other mixed-element meshes (not shown) have a very sharp concentration of these dihedral angles at almost exactly 90°.

When examining the distribution of dihedral angles between two faces (shown in Figure 9), it is easy to distinguish between prism-dominant boundary layer meshes (similar to b2 in the figure) and hex-dominant boundary layers (similar to p). Mesh s is again an outlier, for the same reason it was an outlier for edge angles in triangular faces.



Mesh i

Mesh l

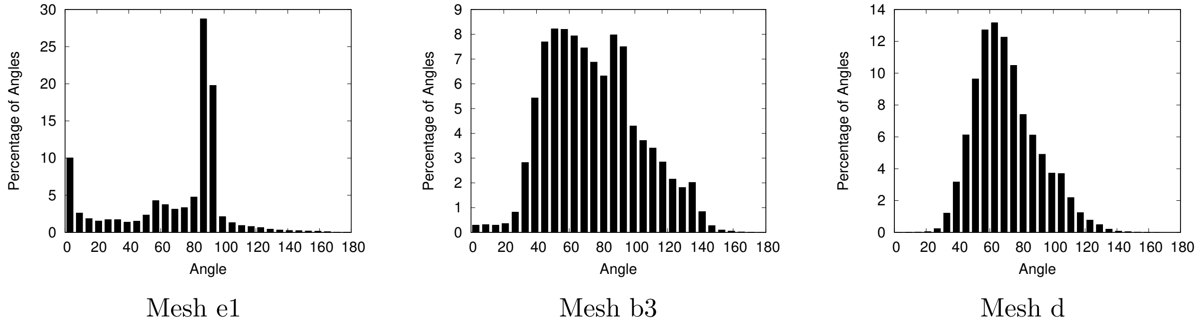
Mesh d

Mesh p

Mesh n

Mesh s

Figure 6. Angles between edges of triangular faces. Results were similar for meshes (i,b1,e1,m), for (d,r), for (b2, e2, f, j, l) and for (b3, h, k, p).

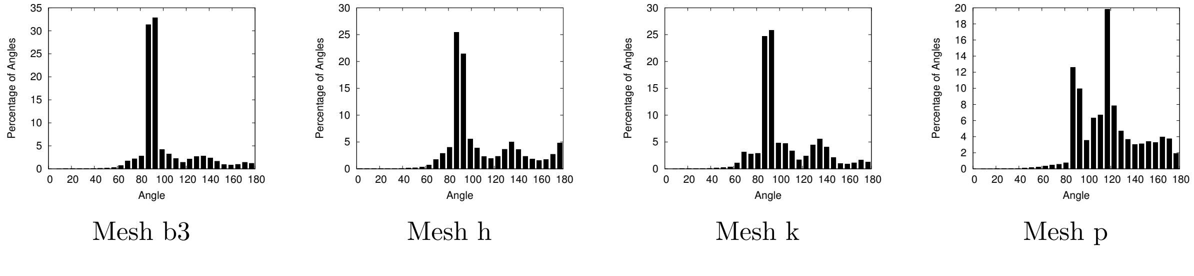


Mesh e1

Mesh b3

Mesh d

Figure 7. Distribution of dihedral angles at edges common to two triangular faces. Mesh e1 is typical of all-tet meshes. Mesh b3 is typical of mixed meshes with a significant number of anisotropic tets (b2 and to a lesser extent h and k). Mesh d (and the others that aren't named) is closer to the sort of angle distribution typical of isotropic tet meshes.



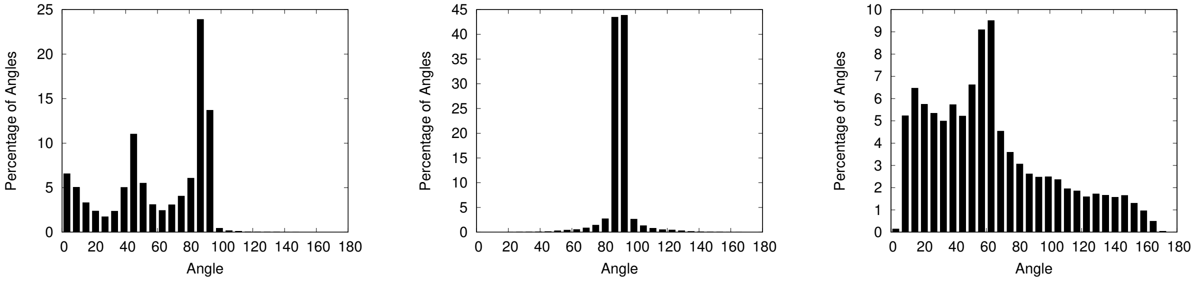
Mesh b3

Mesh h

Mesh k

Mesh p

Figure 8. Quad-tri dihedral angles for meshes with hex-dominant boundary layers generated by Pointwise and Solar. Other meshes have angle distributions dominated even more completely by right angles.



Mesh b2

Mesh p

Mesh s

Figure 9. Distribution of dihedral angles at edges common to two quadrilateral faces. Meshes b2 and p are representative of prism- and hex-dominant boundary layer meshes, respectively.

III.B. Mesh Spacing

The HiLiftPW-3 meshing guidelines recommend grid spacing in the normal direction at the surface, and also spacing along the surface at the leading and trailing edges of the wing, the root and tip of the wing, and the nose and tail of the fuselage.

For normal spacing at the wall, the analysis code identifies the minimum, 5%ile, median, 95%ile, and maximum spacing. Figure 10 shows these results as candlestick plots, including the recommended spacing of 0.00117 inches. Most meshes follow the guideline closely; the upper line on the candlesticks typically reflects a handful of points in corners on the geometry which have no directly normal point, so the distance to the nearest interior point (our proxy for normal distance) is actually a cell diagonal. The NRL and Gulfstream meshes appear to match the off-wall spacing for the coarse mesh instead.

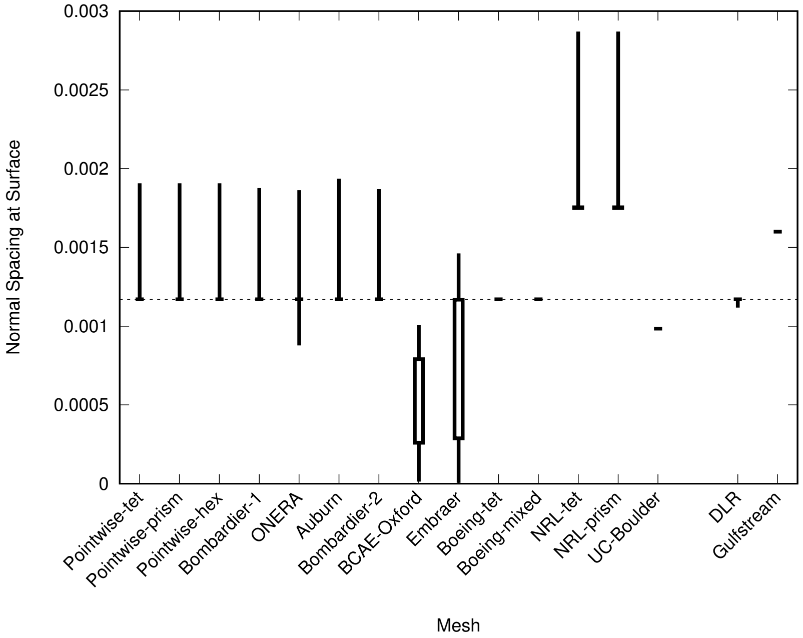
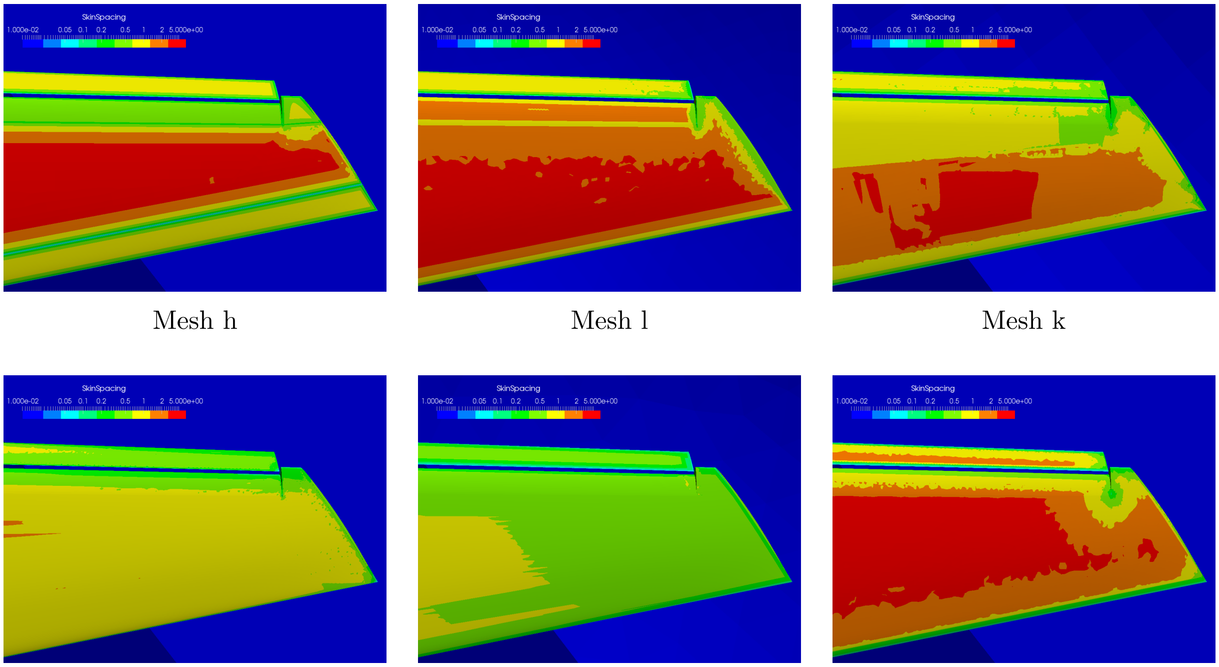


Figure 10. Normal spacing at the wall

For spacing along the surface, statistical extraction of data is impractical, so we must resort to visual inspection. Note throughout these comparisons that Meshes b1, b2, and b3 share a common set of surface points, as do Meshes e1 and e2. Mesh n is a multiblock mesh with the near-body block extending less than a chord from the surface, making visualization of the actual surface impossible to do consistently with Paraview. Thus, there are only 13 meshes to compare here.

Figure 11 shows surface spacing — the closest distance between points on the surface for several meshes



Mesh h

Mesh l

Mesh k

Mesh p

Mesh r

Mesh e

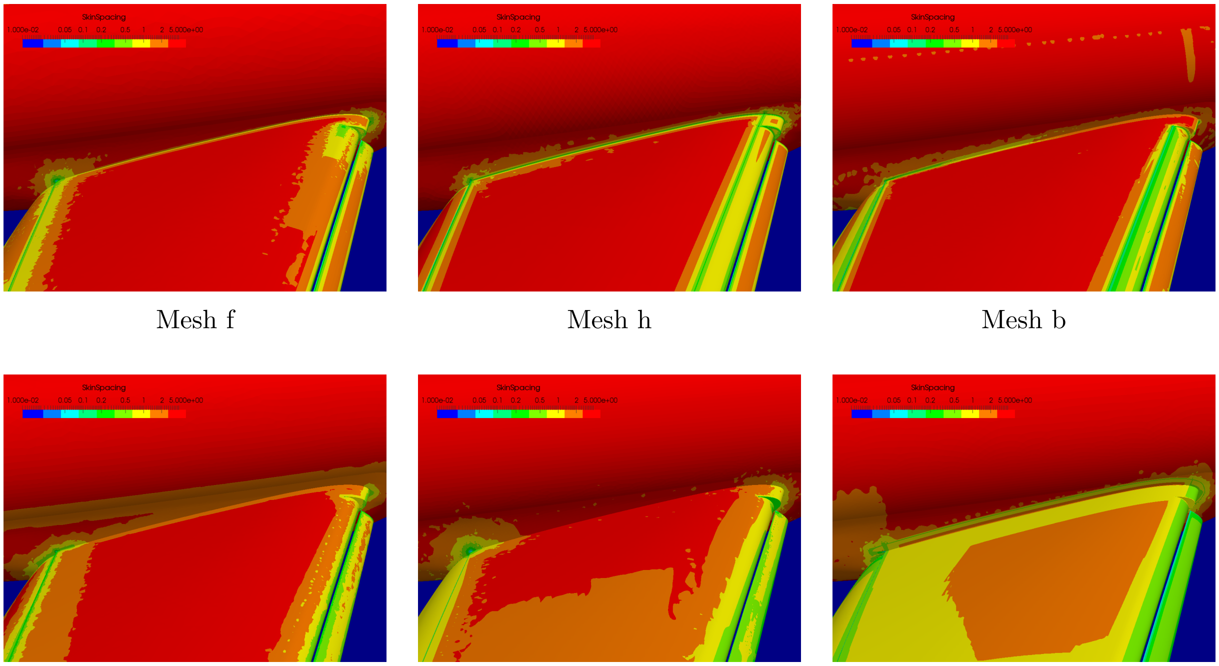
Figure 11. Surface Spacing near the Wingtip. The meshes not shown are qualitatively similar to Mesh m.

— near the wingtip, including part of the leading edge slat. The target spacing at the leading and trailing edges is 0.1% of chord, which amounts to about 0.1 inch at the wingtip; on the color scale used, this is at the interface between treen and cyan. At the wingtap, the target spanwise spacing is 0.1% of semispan, or about 1 inch (yellow). Mesh h shows distinctive bands of fine spacing at both the leading and trailing edges, in regions corresponding to the under-slat surface and the extension of the trailing-edge flap. Mesh l shows a less-pronounced version of the same leading-edge variation, while Mesh k extends this finer spacing to about 40% of chord. Meshes p and r both show more resolution at mid-chord than the other meshes, but neither — especially Mesh p — reach the required spacing at the trailing edge; these are the only two meshes which deviate significantly from the recommended leading and trailing edge spacing. Other meshes show a pattern similar to Mesh e.

At the wing-body junction (WBJ), the target surface spacing is again 0.1% of semi-span to provide resolution for the boundary layer interaction and possible flow separation in this region. Figure 12 shows several typical results; the color scale is the same as at the wingtip, so again the target spacing is in the yellow range. Mesh f (and l) has this spacing along the entire WBJ. Mesh h (and e, m) has somewhat finer spacing along the entire WBJ. The b meshes are have slightly smaller spacing than the target at the trailing edge, but slightly larger at the leading edge. Mesh s is somewhat underresolved along the WBJ, while Mesh p is significantly underresolved compared with the target spacing. Mesh r has an interesting artifact along the side of the fuselage: the surface spacing in the corner is higher than the target. This actually *decreases* moving away from the wing surface on the fuselage before reaching a minimum; at this minimum, the anisotropic boundary layer mesh imprinted on the side of the fuselage transitions abruptly to equilateral triangles. It would be interesting to know whether this mesh artifact is also visible in solution (and/or solution gradient) data at this location.

Figure 13 shows surface spacing near the flap gap at the trailing edge. The target surface spacing at the leading and trailing edges for the flaps is 0.1% of the *element* chord, or about 0.01% of total chord: about 0.03 inches. Mesh r (and b, m) comes close to this value; from visual inspection, it is difficult to determine whether the spacing is actually quite that small. Mesh l (and all other not mentioned) meets the spacing target on the main wing, and repeats this spacing on the flaps. Mesh p, as on the main wing, does not approach the target spacing closely.

The mesh spacing target near the nose is 1% of reference chord, about 2.75 inches. On the scale using



Mesh f

Mesh h

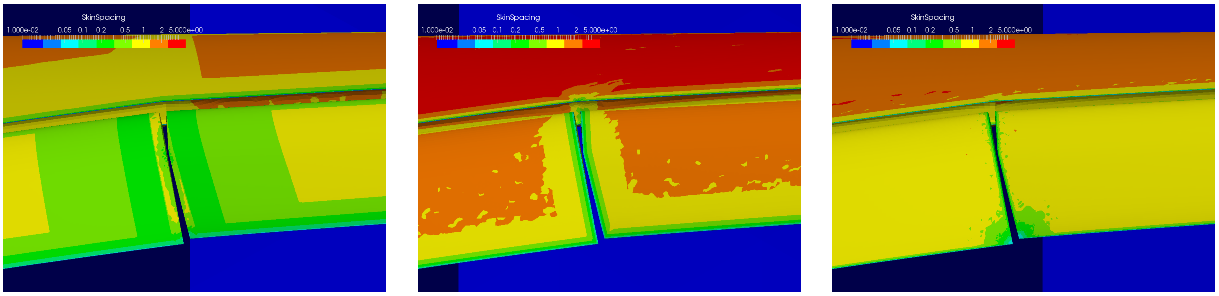
Mesh b

Mesh s

Mesh p

Mesh r

Figure 12. Surface Spacing near the Wing-Body Junction. Mesh l is similar to f; meshes e and m are similar to h; meshes d and k are similar to s; and meshes i and j are similar to p.

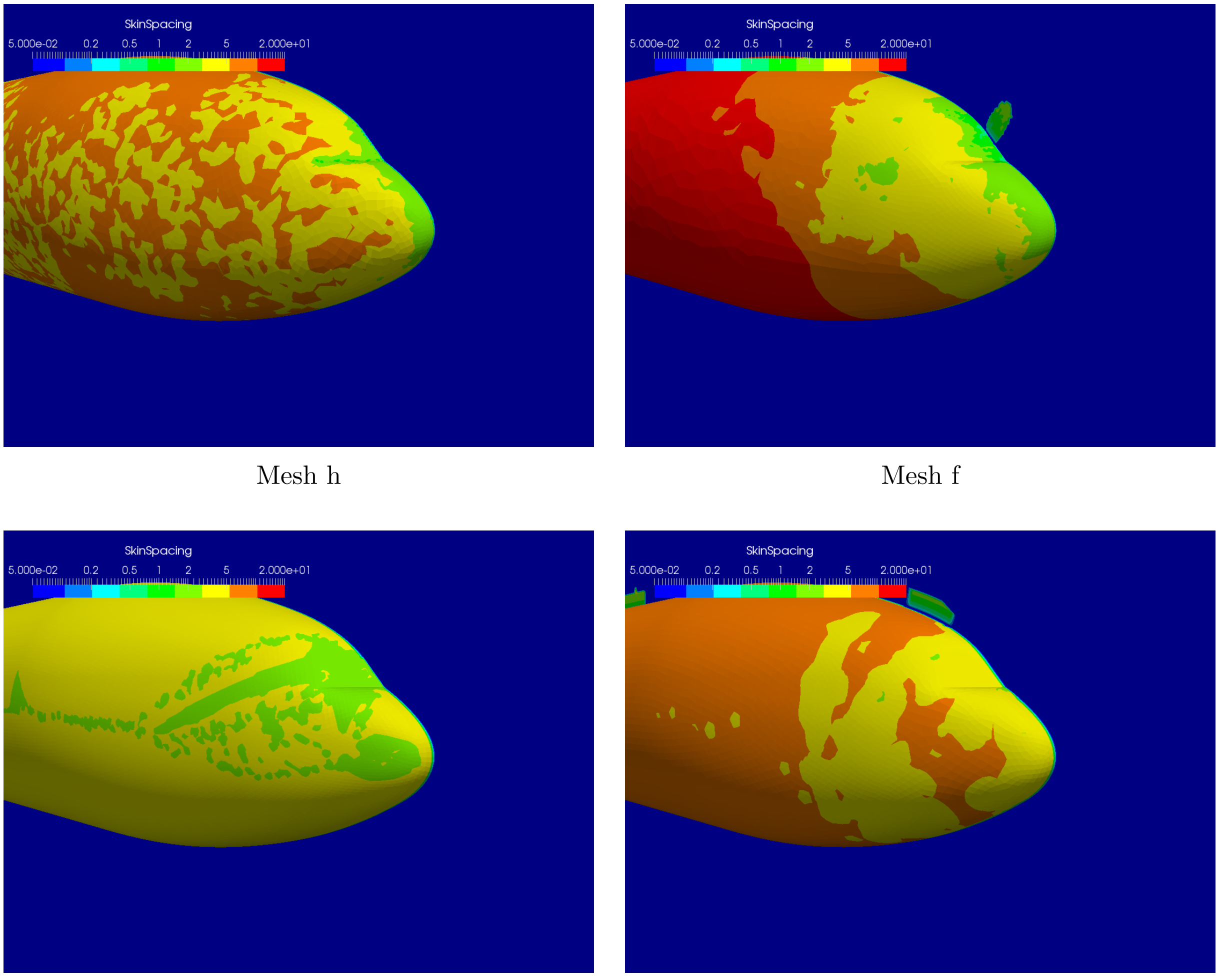


Mesh r

Mesh l

Mesh p

Figure 13. Surface Spacing near the Flap Gap. Meshes b and m are similar to r; other meshes not shown are similar to l.



Mesh h

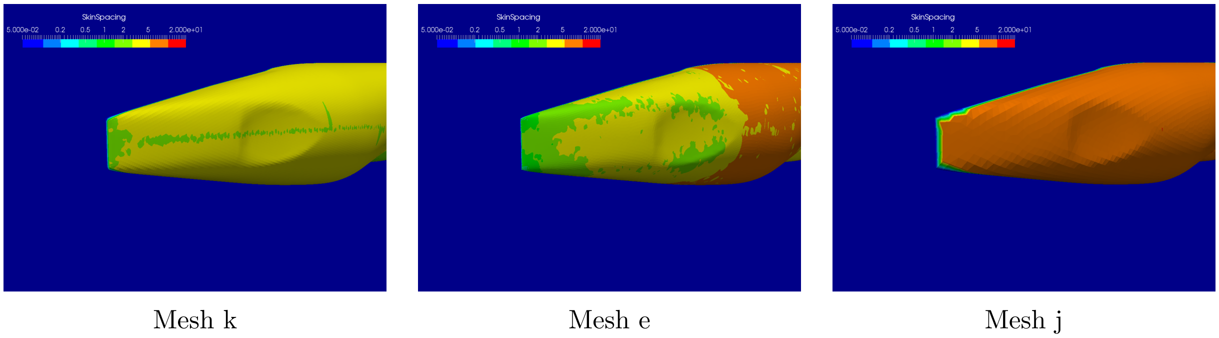
Mesh f

Mesh b

Mesh d

Figure 14. Surface Spacing near the Fuselage Nose. Meshes m and p are similar to h; meshes e, l, and s are similar to f; mesh k is similar to b; and meshes i, j, and r are similar to d.

in Figure 14, this corresponds to about the border between the yellow and green contours. All meshes are really quite close to meeting this target. One could stretch a point to argue that Mesh h is just right, while Mesh b and Mesh d are slightly too coarse. The biggest difference, though, is that Mesh f (and e, l, s) has a bit of extra resolution around the windshield and grading in surface size to much larger cells on the straight fuselage aft of the cockpit. Note that the patches of color on the symmetry plane (and the abrupt change in color in the cell on the fuselage adjacent to the symmetry plane) are artifacts from blanking out the symmetry plane (assigning a spacing of zero), but imperfectly, because some meshes have symmetry plane points that do not lie exactly on $y = 0$.



Mesh k

Mesh e

Mesh j

Figure 15. Surface Spacing near the Fuselage Tail. Meshes b, h and m are similar to k; meshes d, f, p, and s are similar to e; and meshes i, l and r are similar to j.

At the fuselage tail, the target surface spacing is the same as at the nose (see Figure 15). Mesh k (and b, h, m) match this target with relatively uniform spacing. Mesh e (and d, f, p, s) show noticeable grading in the tail region. Mesh j (and i, l, r) is much coarser (at least a factor of two) than the target; the colored fringe in the last triangle on the tail is an artifact of having zero length assigned at the body-symmetry plane junction.

III.C. Geometry Conformance

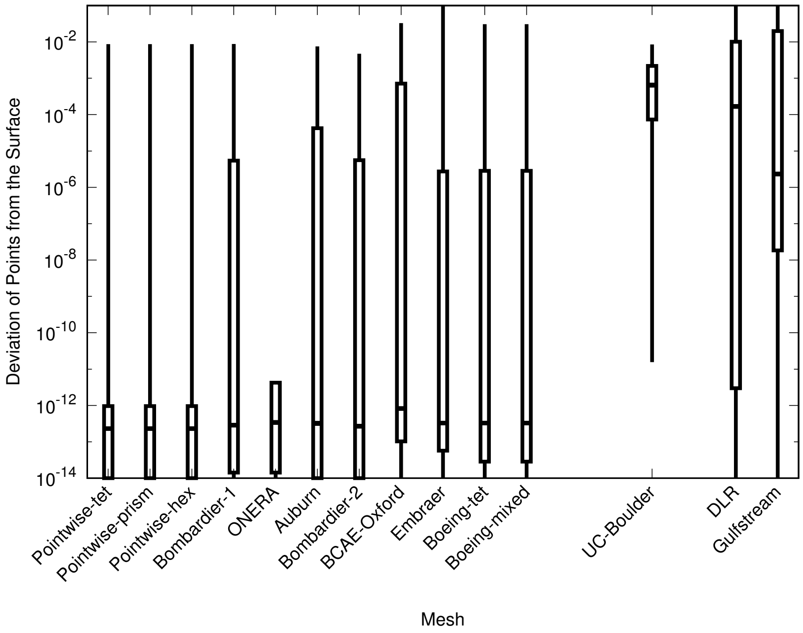


Figure 16. Conformance of meshes to the surface definition

Finally, we turn our attention to the question of conformance to the geometry. Just as with surface normal spacing, the candlestick plots in Figure 16 show the extremal, median, and 5%/95%ile values for distance of surface points from one instance (file + geometry kernel) of the actual surface. All meshes experience occasional excursions from the surface. These typically lie at patch boundaries, especially where the wing fairing meets the fuselage; small differences in point location can cause projection to two different surfaces in such situations. Because the analysis code uses the Pointwise geometry kernel, the deviation of points from the Pointwise-generated meshes (which used exactly the same file, geometry cleanup, and kernel) tells us that the projection tolerance of the kernel is 10^{-12}. Most of the meshes have a median projection distance smaller than this. The BCAE-Oxford mesh (Mesh d) has a 95%ile projection distance approaching the minimum mesh spacing, but the other meshes to the left have errors that are small compared to mesh dimensions. The UC-Boulder (m), DLR (p), and Gulfstream (s) meshes have large numbers of points which are relatively far from the test geometry. The data here is insufficient to draw conclusions about the source of the difference, however. In particular, there has as yet been no analysis done to determine how much difference there is between different geometry kernels reading the same geometry file, nor between the same geometry kernel reading the geometry from two different file formats.

Note that the NLR meshes (i, j) were not included in this comparison because those meshes used a translated geometry, which placed the fuselage nose at the origin. While that is perhaps a more logical location for the model, undoing that translation precisely enough for comparison of surface geometries is an optimization problem with an expensive objective function (norm of projection distance).

IV. Summary

The analysis results are summarized in Table 2, where a ✓indicates that a mesh satisfies the HiLiftPW-3 meshing guidelines, ✗indicates that the guideline was not met, — indicates that it was not possible to assess a mesh's compliance with a guideline, and ∼indicates a guideline that was approximately met. For flap LE/TE spacing, Windicates meshes that met the main wing spacing requirement instead of the flap requirement. The column labeled “Cells on TE” indicates whether a mesh had the recommended eight cells across the blunt trailing edge of the wing; note that some meshes had fewer cells on the TE of the flap, slat, and/or flap cove. Finally, all meshes had the far field sufficiently far away.

Overall, most meshes were mre or less compliant with most of the guidelines. In fact, allowing for wing LE/TE spacing on the flap, eight of the seventeen meshes met all guidelines satisfactorily (b1, b2, b3, f, h, k, e1, e2). Four more deviated from the guidelines in only one respect (l, d, r, m). One mesh (n) proved quite difficult to extract information from because of its large size and the multiblock nature of the mesh.

There are several mesh characteristics that have not been analyzed by the current work, and should be in future extensions of this analysis tool.

- Are there abrupt transitions in cell or face aspect ratio within the mesh? These were observed, sometimes by accident, when examining several meshes. It should not be too difficult to identify these during analysis and highlight them for examination.
- How closely do edges and triangles conform to the shape of the boundary? Assuming vertices are all very near the boundary, this would give a measure of how well the shape of the airplane is resolved by the linear mesh.
- Do the meshes generated as a sequence by a single user actually form a mesh refinement sequence? I have examined this question in the past,[6] but lacked resources to upgrade that tool to apply it to the current much larger meshes.
- How do meshes from different families differ in the way they distribute cells? The mesh comparison tool used to examine mesh refinement sequences can also answer this question.

Finally, of course, the long-term question is what we can learn from this sort of mesh analysis, to improve both mesh generation software and its usage by humans to create better meshes.

References

[1] Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D., “CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences,” Tech. rep., NASA, 2014, NASA CR-2014-218718.

[2] Dannenhoffer, III, J. F. and Ollivier-Gooch, C., “Methods for the Evaluation of Candidate Meshes for the Geometry and Mesh Generation Workshop,” *Proceedings of the Twenty-Third AIAA Computational Fluid Dynamics Conference*, 2017.

Tool Used	Organization of User	Label	Cell Types	Spacing							Stretch Ratio	Cells on TE	Volume Ratio	Surface Conform	Far Field
				Normal	LE/TE	Flap	Root	Tip	Nose	Tail					
Pointwise	Pointwise	b1	T	✓	✓	✓	∼	✓	✓	✓	✓	✓	✓	✓	✓
	Pointwise	b2	TPP	✓	✓	✓	∼	✓	✓	✓	✓	✓	✓	✓	✓
	Pointwise	b3	TPPH	✓	✓	✓	∼	✓	✓	✓	✓	✓	✓	✓	✓
	Bombardier 1	f	TPP	✓	✓	W	✓	✓	✓	✓	✓	✓	✓	✓	✓
	ONERA	h	TPPH	✓	✓	W	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Auburn	k	TPPH	✓	✓	W	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Bombardier 2	l	TPPH	✓	✓	W	✓	✓	✓	✗	✓	✓	✓	✓	✓
ANSA	BCAE-Oxford	d	TPPH	✓	✓	W	✓	✓	✓	✓	✓	6	✓	✓	✓
	Embraer	r	TPPH	✓	∼	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
AGPS / AFLR3	Boeing	e1	T	✓	✓	W	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Boeing	e2	TPP	✓	✓	W	✓	✓	✓	✓	✓	✓	✓	✓	✓
Capstone	NRL	i	T	✗	✓	W	✓	✓	✓	✗	✓	1	∼	—	✓
	NRL	j	TPP	✗	✓	W	✓	✓	✓	✗	✓	4 (iso)	✗	—	✓
Symmetrix	UC Boulder	m	T	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	?	✓
Rhinoceros	KHI	n	PH	—	—	—	—	—	—	—	✗	—	—	—	—
Solar	DLR	p	TPPH	✓	✗	✗	✗	✓	✓	✓	✓	4	✓	?	✓
Helden Mesh	Gulfstream	s	TPP	✗	✓	W	∼	✓	✓	✓	∼	4 (iso)	✓	?	✓

Table 2. Summary of mesh analysis results.

[3] Dannenhoffer, III, J. F., "Analysis of GMGW1 Structured Grids," *Proceedings of the Fifty-Fifth AIAA Aerospace Sciences Meeting*, 2018.

[4] Pointwise, Inc., "Geode Core V6.4," [Computer software], Fort Worth, TX, 2016.

[5] HiLiftPW3 Organizing Committee, "Third AIAA CFD High Lift Prediction Workshop Gridding Guidelines," 2016.

[6] Ollivier-Gooch, C., "Assessing the Validity of a Mesh Refinement Sequence, with Application to DPW-3 Meshes," *Proceedings of the Forty-Seventh AIAA Aerospace Sciences Meeting*, 2009.