SolidMesh Development

Only true periodic faces will be considered for this procedure, i.e. faces that can be determined to be a matrix transformation of one another. Thus the periodic faces must not only be topologically similar, but geometrically similar as well. To consider this, some minimum face requirements are needed.

The periodic faces do not need to be of the same surface type, i.e. NURBS, trimmed, discrete, and composite can be mixed. However, the same number of loops needs to exist on each face. For NURBS there is only one loop, the surface boundaries; for trimmed, discrete, and composite faces, the number can vary. Lastly, the same number of edges needs to exist per loop. Once the similar periodic topology has been established, the pairing and orientation can proceed.

Periodic Pairs and Orientation

Additional pieces of information are required by the user to determine the periodic surface pairs: namely a reference point pair ($P_{1,2}$) and a reference axis ($dot = \vec{v} \bullet \vec{V} \square \vec{A}$). The reference point is an identical and corresponding point on each periodic face and must be a distinct point related to each face. The reference point is not required to be a vertex on a loop or boundary. The reference axis is used as a guide to help determine orientation, but should indicate the rotational axis if the faces are rotated. If a vector is not selected, the reference axis is automatically determined by comparing the known geometric normal of the faces $\left(\vec{N}\right)$ by computing the dot product, $d = \vec{N}_1 \bullet \vec{N}_2$. In the case if $|d| < 1$, the reference axis is determined from the cross product, $\vec{A} = \vec{N}_1 \times \vec{N}_2$. To determine the two cases when $|d| = 1$, the vector between the reference points is used, $\vec{V} = P_2 - P_1$. In the case the faces are translated, $\left|\vec{V} \bullet \vec{N}_1\right| = 1$, the vector between the reference point of the first face and the centroid of the first face is used as the reference axis. In the case the faces are 180-degrees apart, $\left|\vec{V} \bullet \vec{N}_1\right| < 1$, the cross product between the normal of the first face and vector between the reference points is used as the reference axis, $\vec{A} = \vec{N}_1 \times \vec{V}$.

The loop pairings are determined by the loop centroids and arc length. The loop centroids (C_l) are determined by averaging the loop vertices of the component edges. They are only counted once. The loop arc lengths are determined by summing the arc lengths of the component edges. The initial pairing is determined by minimizing the distance from the reference point to the loop centroids, $\min(\|C_l - P\|)$, but only paired if the arc length of the loops is equal within the loop tolerance. The loop tolerance (σ) is determined by multiplying the number of edges (N_l) within the loop by the glue tolerance (ε), i.e. $\sigma = N_l * \varepsilon$. This pairing method fails if the selected reference point is symmetric with respect to the loops; two or more loops that have the same arc length and distance from the reference point. In this case, different reference points need to be selected. An example of a surface containing an axis of symmetry is depicted in Fig. 1. Selecting vertex P_4 or P_1 will create a vector $\vec{V}_1$ that is the axis of symmetry. A better choice would be P_2, P_3, or P_5 - P_{20}, any of these points selected as a reference point will not result in a symmetric domain. Once all the loops have been paired, then the orientation and pairing of the component edges and vertices can be determined.

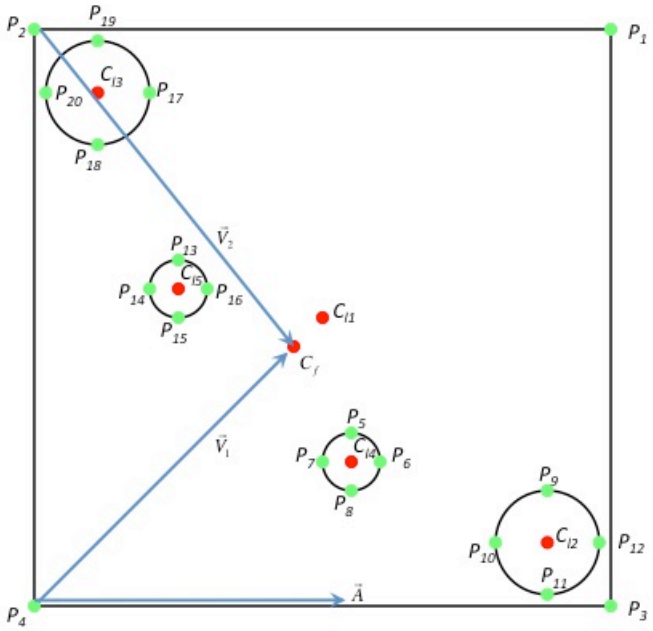


Figure 1. Symmetry example.

A normalized reference vector ($\vec{V}_r$) for the surface is determined from the reference point to the face centroid (C_f), where the face centroid is determined by averaging all the loop centroids. To determine component parings and orientation on each face, for each periodic loop pair, the closest point on the loop to the reference point is found by using the vector ($\vec{v}$) from the reference point to the vertex and calculating the dot product with respect to the reference vector ($\vec{v} \bullet \vec{V}_r$), the normal of the local vector with respect to the reference vector ($\vec{n} = \vec{v} \times \vec{V}_r$), and the actual distance between the reference point and vertex ($\|\vec{v}\|$). The vertex with the minimum dot product is selected. However, if the dot products are equal, then the vertex with the maximum dot product between the local normal and the face normal ($\vec{n} \bullet \vec{N}$) is selected and that distance is used.

The loop direction and orientation must also be determined. The two attached edges at the paired vertices are compared. The edge vectors ($\vec{e}_1, \vec{e}_2$) are calculated, i.e. the vector formed from the edge end vertices, with the paired vertices as the root point. The edge vector that is aligned most with the reference vector is selected $\left(\max\left[\vec{e}_1 \bullet \vec{V}, \vec{e}_2 \bullet \vec{V}\right]\right)$. If they are equal, then the edge vector that has the normal $\left(\vec{n}_i = \vec{e}_i \times \vec{V}_r\right)$ most aligned with the face normal is selected $\left(\max\left[\vec{n}_1 \bullet \vec{N}, \vec{n}_2 \bullet \vec{N}\right]\right)$.

Once these two edges are paired, their relative orientations are used to determine the periodic loop orientation. The edge vectors are determined based on the orientation of the previously connected edge; in the case that the first edge in the loop list has been selected, the comparison is made to the second edge in the loop. In terms of topology, the edge vertices are compared to the vertices of the previous edge; the match determines the direction of the edge vector. The normalized dot products $\left(a/|a|\right)$ of the two orientation vectors are compared and if they are of opposite sign, then the loops run opposite of each other and loop orientation of the pair is reversed.

Starting at the paired edges, each curve and vertex of the loop is paired by stepping through each edge in the loop list based on the periodic loop orientation. A periodic list is created on the edges and vertices, thus allowing for multiple periodic

connections that could occur through crystalline structures, spherical structures, and multiple stage structures like turbo-machinery. The orientation of each paired edge is determined with respect to each other in their respective loops. The relative orientation of the edges is based on the same determination used for the loop orientation. The normalized dot products $\left(a/|a|\right)$ of the two orientation vectors are compared; if they are of opposite sign, then the edges run opposite of each other and the periodic orientation of the pair is reversed.

This procedure could be modified to allow dissimilar topologies and dissimilar geometries by relaxing the restriction on the loop arc lengths. If they are sorted, then similar topologies and/or geometries could be compared and a mapped periodic pairing could be determined. This was not considered at this point since the algorithm in AFLR3 would not be able to rebuild on a mapped periodic boundary.

The pairings are checked afterwards. If a vertex on a loop or an edge on a loop has more than one paring, then the procedure is aborted. This can come about by using the wrong reference axis. In this case occurs, a vector needs to be selected that satisfies the criterion.

Periodic Transformation

The exact periodic transformation is determined from the face centroid and the paired periodic vertices. The distance between the face centroids is calculated as a reference distance $\left(d_r\right)$. If the difference between the reference distance and the distance $\left(d\right)$ among any of the paired periodic vertices is within the glue tolerance $\left(\left|d_r - d\right| \le \varepsilon\right)$, the faces are only a translation of each other. Otherwise, the two faces are considered to be a translation and rotation of each other. The translation vector $\left(\vec{T}\right)$ is the determined from the two provided reference points.

To determine the angle of rotation $\left(\theta\right)$, the orthogonal vectors $\left(\vec{v}_{1,2}\right)$ through the periodic vertex pairs from the axis of rotation (reference axis) are calculated. The vector for each periodic vertex pair from the reference point (as long as they are not parallel to each other and they have a non-zero magnitude) is projected along the axis of rotation

and a new vector is formed from the projected point to the periodic vertex. The angle is determined by $\theta = \left(\frac{\vec{n}}{\|\vec{n}\|} \bullet \vec{A} \right) * a\cos\left(\frac{\vec{v}_1 \bullet \vec{v}_2}{\|\vec{v}_1\|\|\vec{v}_2\|} \right)$, where $\vec{n} = \vec{v}_1 \times \vec{v}_2$. The variation in angle cannot exceed the glue tolerance. The variation in angle is determined by calculating the difference in angle length $\left(\Delta d = \sqrt{dx^2 + dy^2} \right)$, where $dx = .5 * \left(\|\vec{v}_1\| + \|\vec{v}_2\| \right) * \left(1 - \cos\left(|\Theta - \theta| \right) \right)$, $dy = .5 * \left(\|\vec{v}_1\| + \|\vec{v}_2\| \right) * \sin\left(|\Theta - \theta| \right)$, and Θ is the initial angle calculated on the first pass. If the variation in angle is exceeded, then the two faces are not paired as periodic and all periodic pairings are removed. The variation in angle is reported if there is a failure. The glue tolerance needs to be increased to this value to force the faces to be periodic.

To complete the transformation information, the transformation is validated for each periodic vertex pair. If all of the transformed points are within the glue tolerance, the first face is set to the translation vector; the second face is the negative of the translation vector. If there is a determined angle, the first face is set to the angle and the second face has the negative of the angle. These settings will insure that the correct transformation is done no matter which face is processed first during the surface mesh generation.

Periodic Inquiries

Selecting the face, edge, or vertex and using the Query option can determine the periodic partner(s) of a face, edge, or vertex. The periodic partner(s) will be selected and information will be reflected in the message window. There can be multiple periodic partners due to shared edges and vertices amongst adjacent periodic faces.

Periodic Point Spacing

When the point spacing is applied to a vertex, if a periodic vertex list exists, then each periodic vertex partner is applied the same point spacing and the edge distribution for all of the adjacent edges is automatically regenerated. Also, if the number of points is specified on an edge that has a periodic edge list, then each periodic partner is applied the

same number of points. The action is the same as if those edges were selected and applied the same number of points.

Periodic Surface Generation

The user does not have to do any additional work and there is no additional user interaction to generate a surface mesh containing periodic surfaces. For a given periodic surface pair, it is arbitrary as to which of the two surfaces is meshed first as they are supposed to be identical to one another. One of the periodic surfaces will be generated initially and the periodic partner will be generated immediately afterwards using the periodic topology and transformation information, i.e. the physical points and connectivity is copied and the physical points are transformed to the correct orientation. However, this does not complete the process. Since SolidMesh is based on a solid modeling data structure, the actual connectivity at the edges and vertices needs to be recovered or reconciled with the transformed mesh.

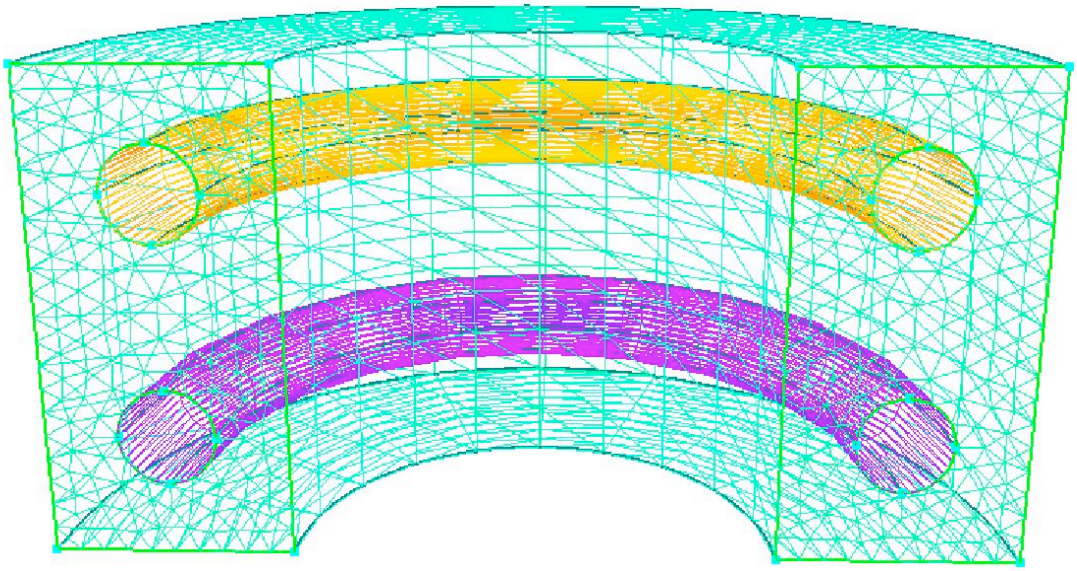
To recover the connectivity, the loops of the periodic surface are processed in the same order of the copied mesh. As each edge in the loop is recovered, the periodic edge partner is used to recover the edge point information.

Periodic Surface Attributes

Certain attributes need to be set in order for the periodic surfaces to be processed correctly by AFLR3 during the volume mesh generation. Individual periodic surfaces must reside in unique Groups and is done automatically within SolidMesh. Also, the Precedence surface mesh boundary condition should also be applied to periodic surfaces. SolidMesh automatically applies the Precedence boundary condition to periodic surfaces. In addition, the Reconnect volume mesh boundary condition must not be applied to Groups containing periodic surfaces. This is automatically addressed in SolidMesh and prevents the user from modifying this attribute. The Reconnect boundary condition allows the edges of adjacent triangles within the surface mesh to be swapped in order to improve the overall quality of the volume mesh. However, this is not allowed in order to preserve the connectivity of the periodic surface meshes.

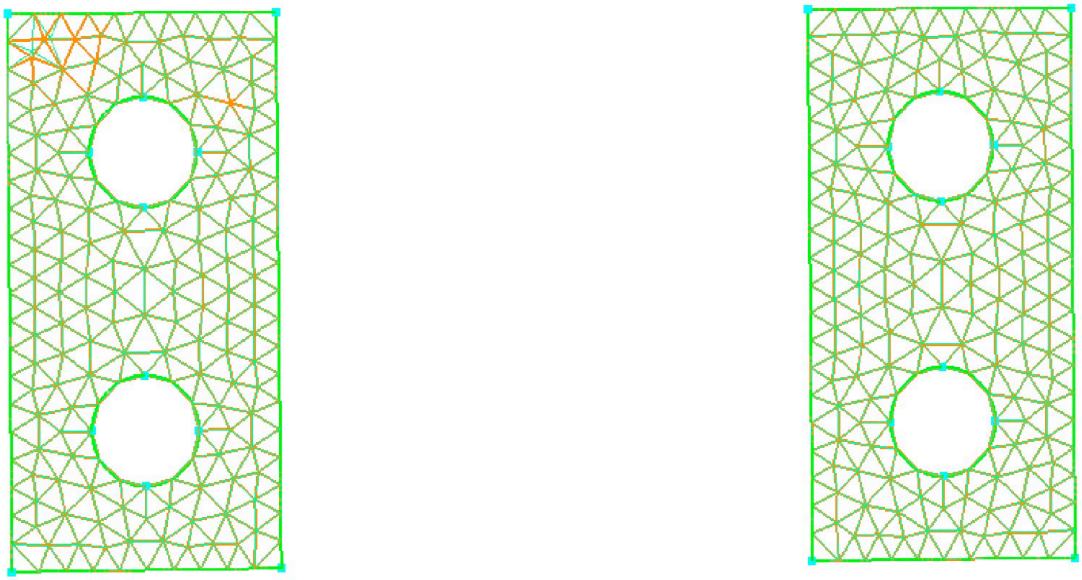
Periodic Example

Fig. 2 shows a 180-degree periodic geometry with two holes. The two periodic surfaces A and B will be placed in unique groups when the Periodic boundary condition is applied. Fig. 3 shows the differences in the surface mesh with and with out the Periodic boundary condition. The two meshes on surface A overlay each other except for a small discrepancy in the upper left corner, which matches the upper right corner of both meshes on surface B. Visually it is difficult to see the periodic matching. The details are apparent in the physical points and connectivity.



(A) (B)

Figure 2. 180-degree periodic example with holes.



(A) (B)

Figure 3. Differences between periodic and non-periodic surface meshes.

AFLR3 Development

The AFLR3 volume mesh generator was modified to provide the capabilities for meshing with arbitrary periodic and constraint surfaces. AFLR3 has always had the capability to fix any surface other than a constraint surface that intersects the BL region. This provided a rudimentary periodic capability useful for isotropic periodic surfaces only and limited applicability for cases requiring BL regions. There are several applications, turbomachinery for example, that require a more advanced periodic surface capability. Further the periodic surfaces in these applications are often curved and intersect the BL region. The development of the modified AFLR3 volume mesh generator therefore involved two primary tasks to develop a capability for both curved constraint surfaces that intersect the BL region and periodic surfaces that intersect the BL region. Both are described in more detail in the following sections along with examples that illustrate the capabilities.

Curved Constraint Surfaces

The AFLR3 volume mesh generator includes a capability to rebuild constraint surfaces that intersect the BL region. The prior capability required the constraint surface to be planar. Such surfaces are common in applications with symmetry planes or inlet/exit surfaces for ducts and engines. Since the BL volume mesh is not known a priori the initial surface mesh on any constraint surface that intersects a BL region must be regenerated to match the BL region and the desired point spacing in the isotropic region. For a curved constraint surface the internal BL volume mesh generator was modified so that the BL point generation at the constraint surface follows the curved geometry. The geometry on the constraint surface is defined by the input surface mesh discretization. The BL point generation and BL normal vectors are projected on the given discrete representation as each layer is generated. Techniques for interpolation and projection on discrete surface were previously developed for the AFLR4 surface mesh generator and were adapted for use by the volume generator.

In addition to the BL region, the subsequent isotropic tetrahedral element region also required modifications to handle curved constraint surfaces. The volume mesh generator uses the AFLR2 planar surface mesh generator to regenerate the isotropic triangular faces on the isotropic portion of the constraint surfaces. In the modified procedure this same technique is used only with planar generation taking place in a u,v transformed space. Surface mesh generators and CAD geometry systems use a transformed space concept to efficiently deal with geometric operations on a surface. In the volume mesh generator only the discrete representation is available and a u,v transformation must be created. The transformation from x,y,z physical space to u,v transformed space on the constraint surfaces is obtained using an average plane projection. In this approach the average plane is defined the plane that includes the centroid of the discrete surface definition and has a normal vector equal to the least-squares average surface normal vector for the discrete surface definition. Once the plane is defined all surface points are then projected upon the plane, and that projection defines the transformation. While transformation based upon projection imposes restrictions on the constraint surface curvature, it doesn't limit applicability in applications of interest as they include surfaces that are reasonable as constraint surfaces. Surfaces that could have

issues include ones with multiple folds in varying directions, and such surfaces are not reasonable for individual or periodic constraint surfaces. Far more arbitrary curvature could readily be supported with the current approach by using the generalized topological mapping employed within the full AFLR4 surface mesh generator. For the present application it was determined that this additional complexity was not justified.

To demonstrate the capability to rebuild curved constraint surfaces a horn configuration test case was created. The configuration and geometric definition for this case are illustrated in Figs. 4 and 5, respectively.

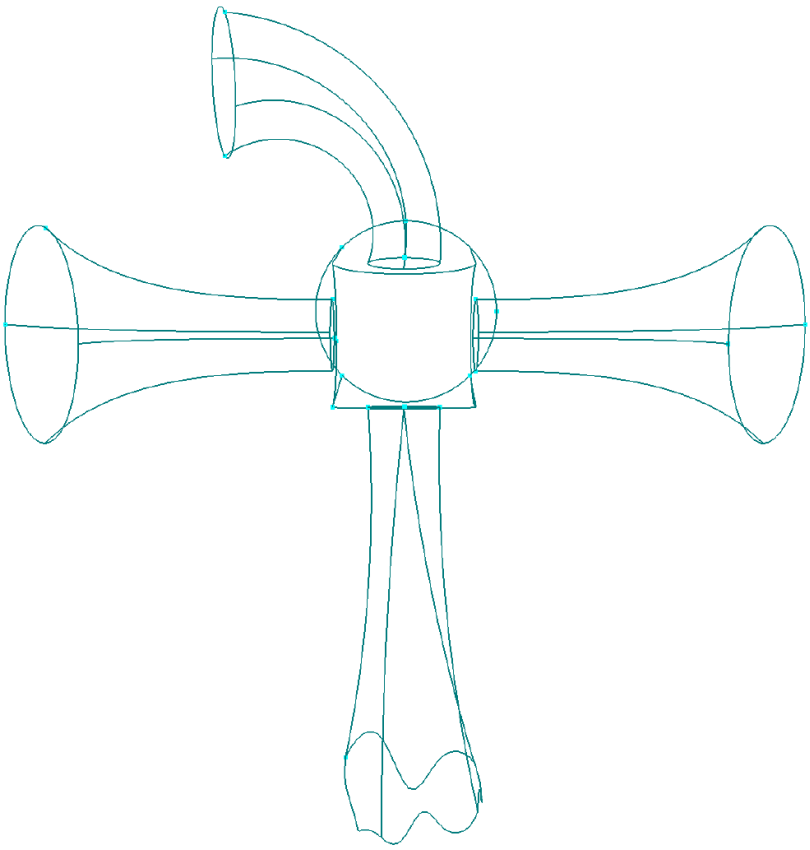


Figure 4. Horn configuration test case that illustrates capability to rebuild curved boundaries, which intersect the BL region.

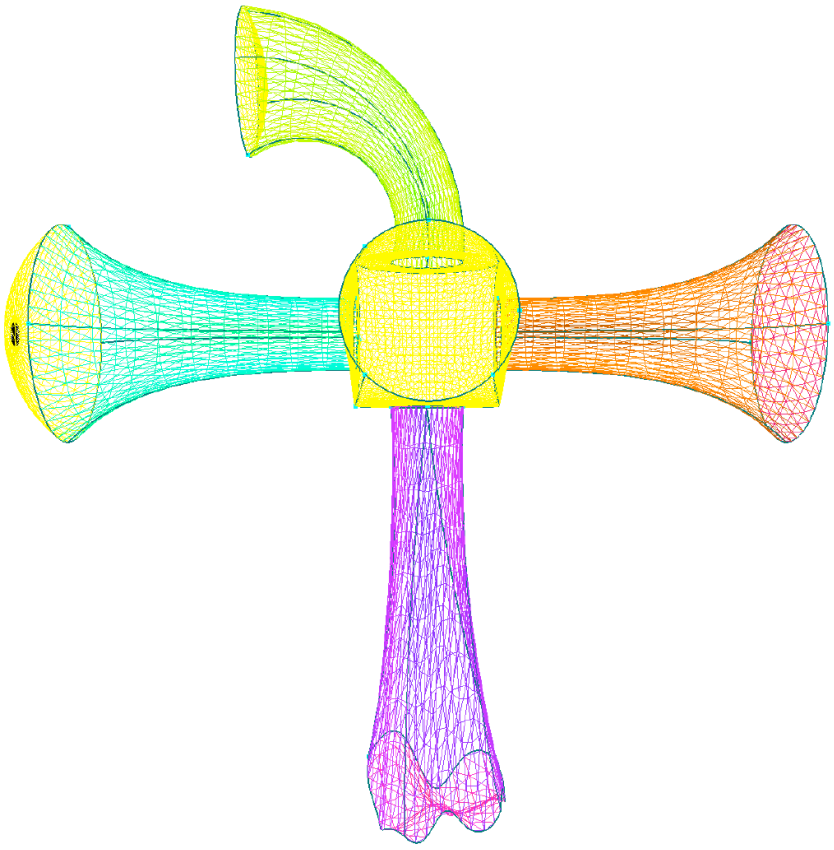


Figure 5. Geometric surfaces for horn case.

In this case a planar inlet plenum leads to four horns with differing exit surface types. The horn exit surface types are planar (left), convex (right), concave (top), and highly curved (bottom). The inlet and four exit surfaces are treated as constraint surfaces that intersect the BL region. All other surfaces have a BL region attached. After generation of the volume mesh the final surface mesh with rebuilt constraint surfaces is illustrated in Fig. 6.

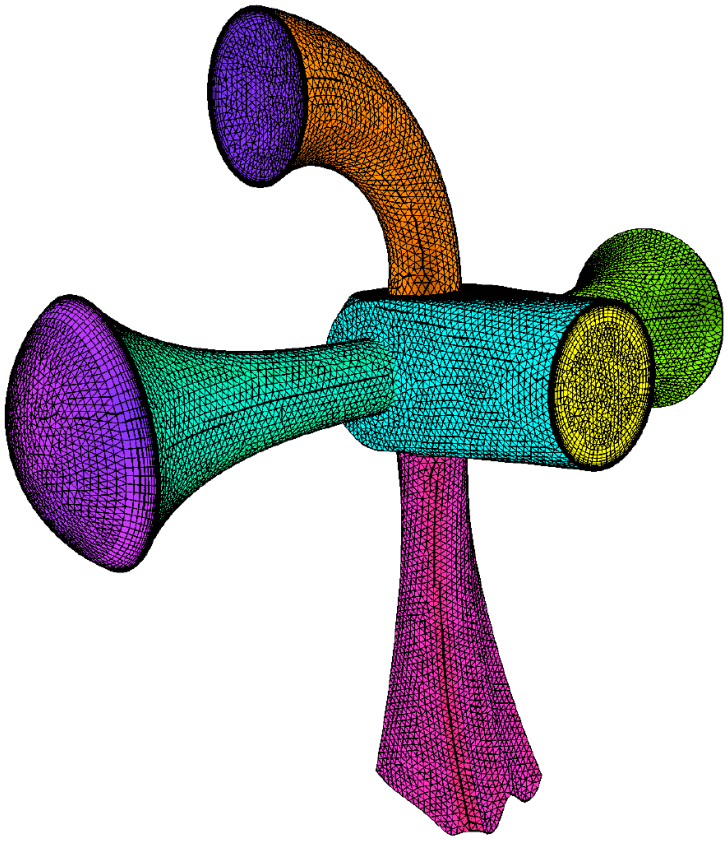


Figure 6. Final surface mesh for horn case with rebuilt curved surfaces that intersect the BL region.

Rebuilt constraint surfaces for the planar inlet, planar left horn exit, convex right horn exit, concave top horn exit, and highly curved bottom horn exit are illustrated in Figs. 7, 8, 9, 10, and 11, respectively.

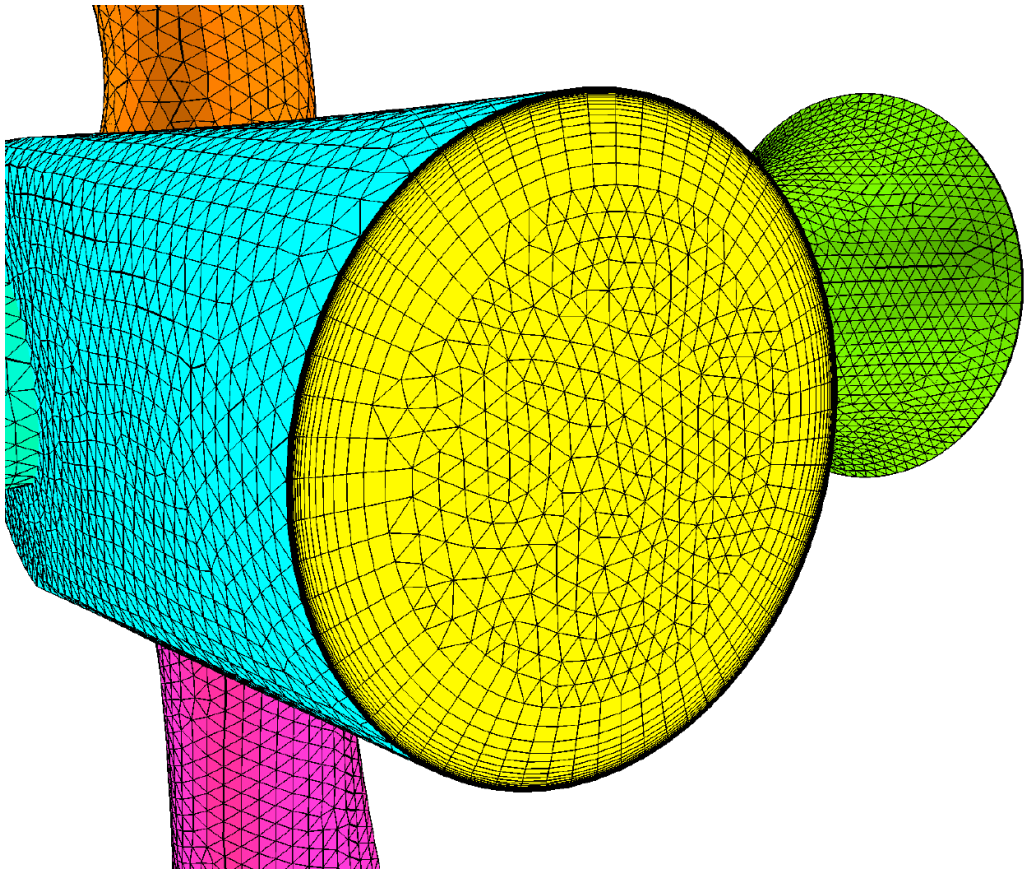


Figure 7. Detail view of rebuilt planar inlet surface mesh.

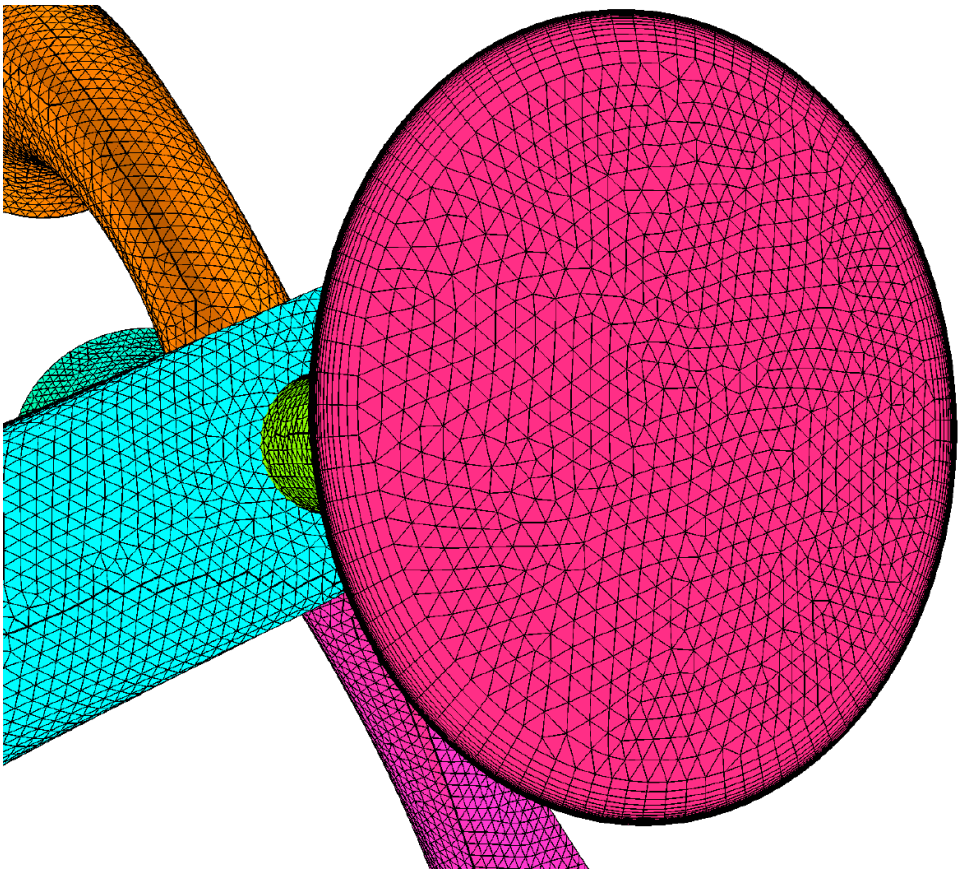


Figure 8. Detail view of rebuilt right horn planar exit surface mesh.

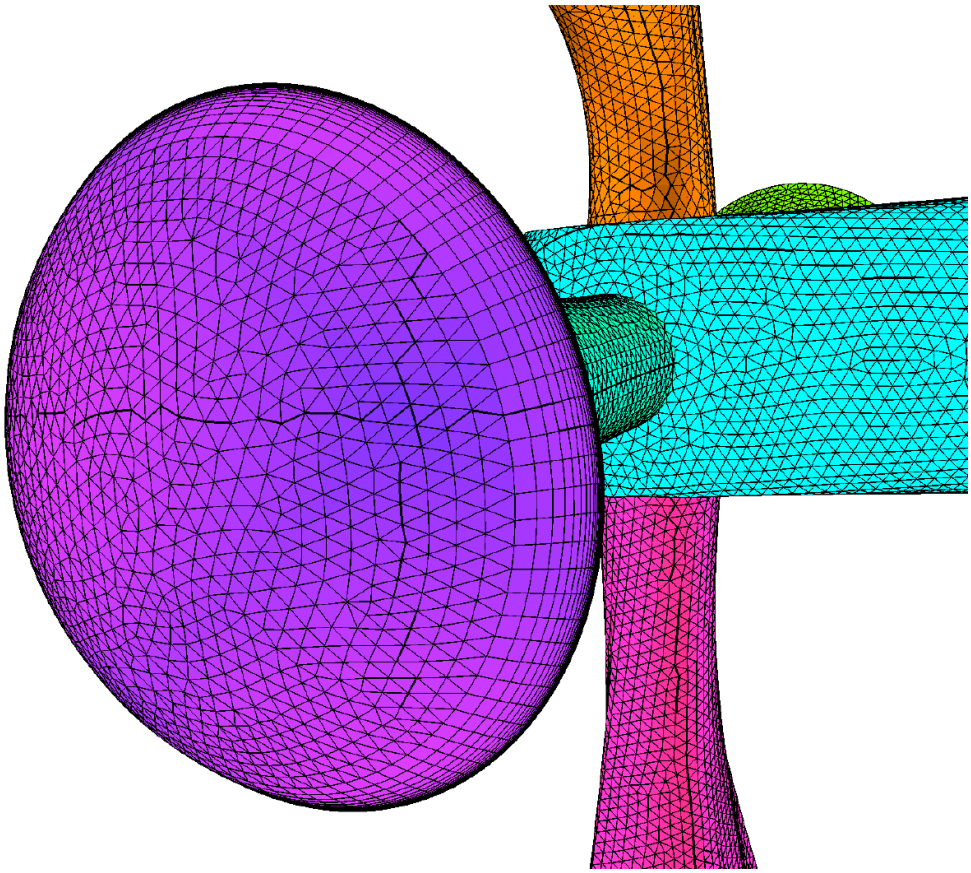


Figure 9. Detail view of rebuilt left horn convex exit surface mesh.

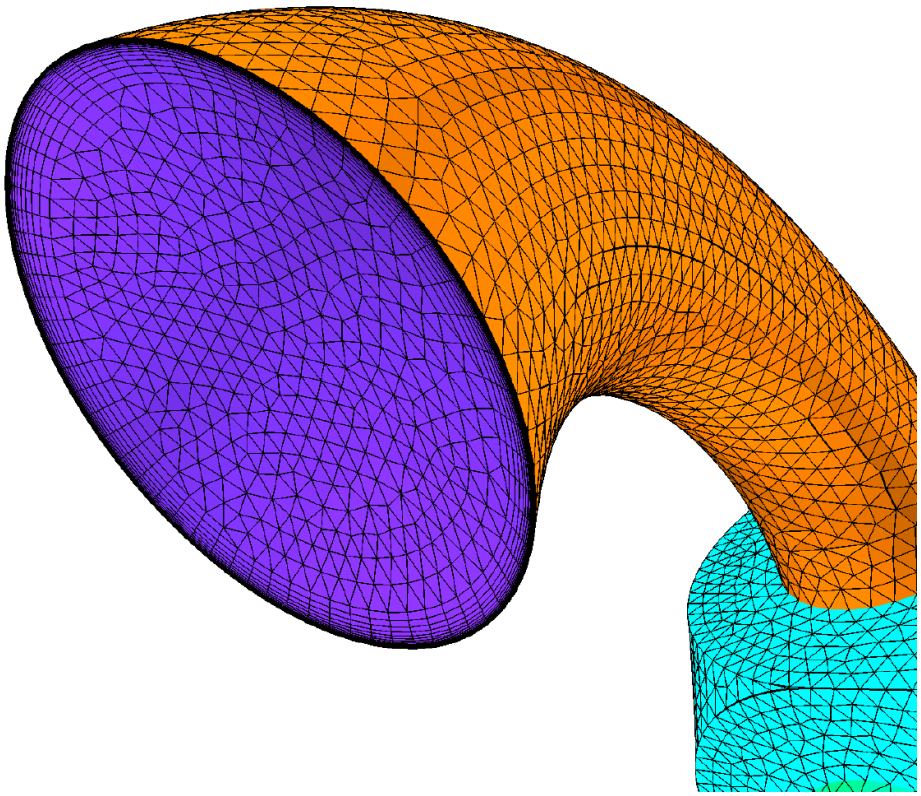


Figure 10. Detail view of rebuilt top horn concave exit surface mesh.

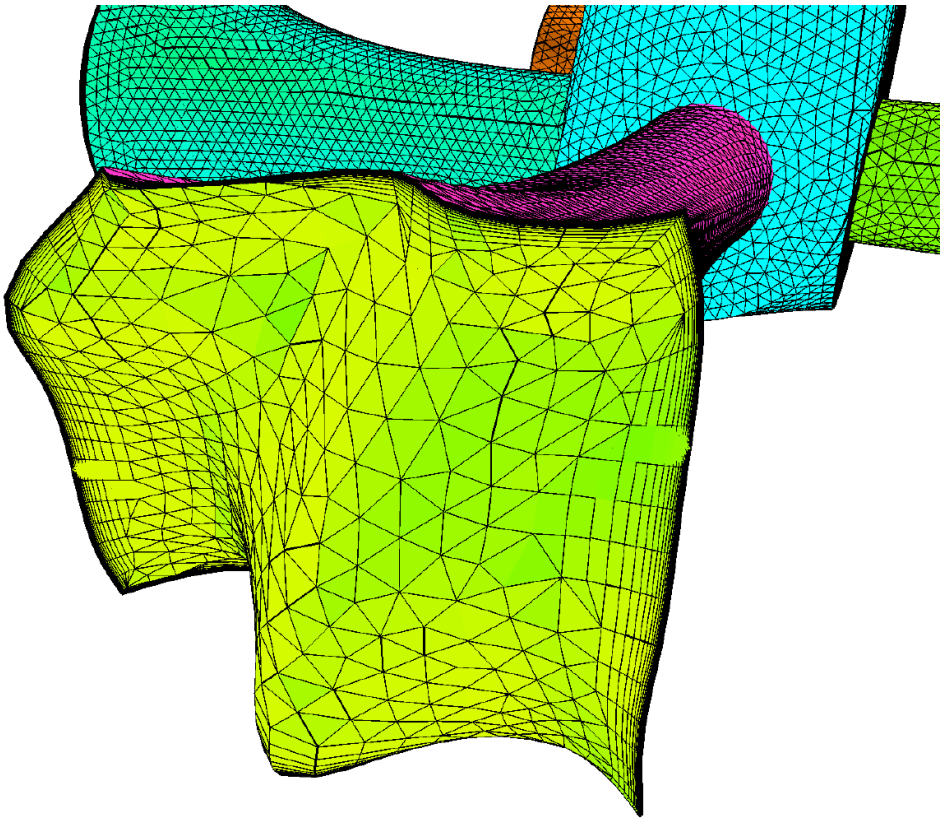


Figure 11. Detail view of rebuilt bottom horn highly curved exit surface mesh.

For the most extreme example of curvature in this case, the bottom horn exit shown in Fig. 11, there are no issues related to the projection based mapping. In fact, the more limiting factor is that high curvature on the constraint surface imposes skew on the BL region elements. The resulting element quality suffers and limits the extent of the BL mesh generation, if reasonable limits on element quality are enforced. For completeness a cross-section view of the volume mesh elements is illustrated in Fig. 12. As expected the BL region smoothly intersects the specified constraint surfaces.

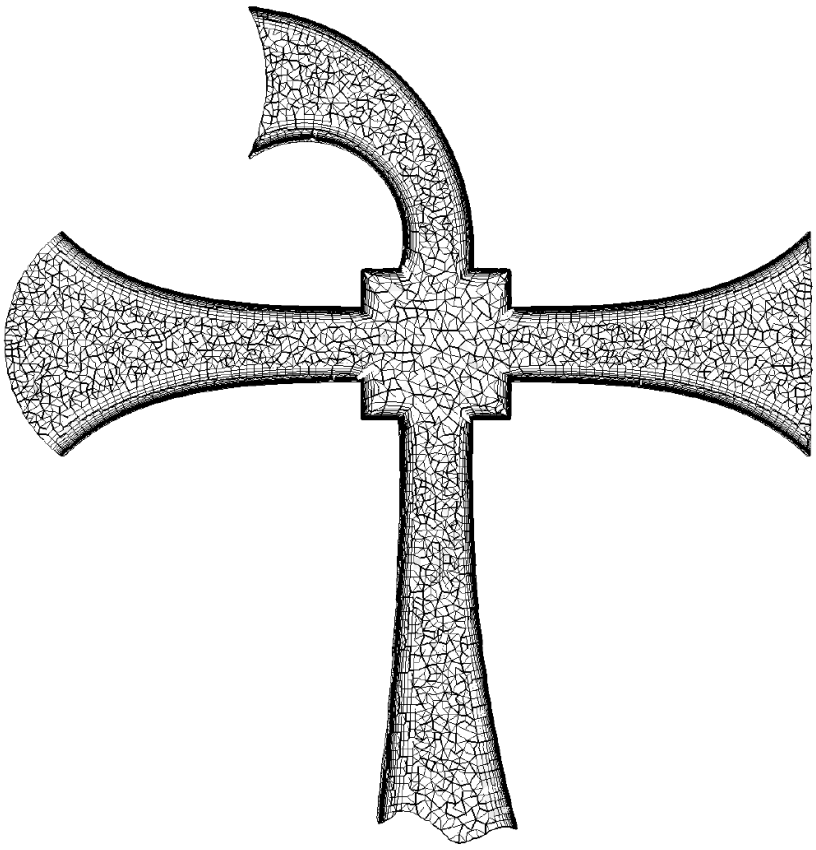


Figure 12. Volume element cross-section view.

Periodic Constraint Surfaces

The AFLR3 volume mesh generator never has included an explicit periodic constraint surface capability. As previously mentioned, isotropic constraint surfaces could be used as periodic by fixing the surface mesh. However, that doesn't allow for constraint surfaces that intersect the BL region. For the present work that capability was added by building upon the curved constraint surface capability previously described. For periodic constraint surfaces, planar or curved, both surfaces must be modified exactly the same. That implies that both the BL and isotropic mesh generation must behave periodically at the constraint surface intersection. The AFLR3 BL volume mesh generator was modified so that for any constraint surface labeled as periodic its corresponding periodic child or partner constraint surface will have the exact same mesh in the BL

region. During BL mesh generation as each layer is created the parent and child for each periodic surface pair are compared. If an additional layer can't be generated on either then both are terminated and the exact same topology is assured. Re-evaluating the child coordinates based upon the periodic coordinate transformation specification of the parent constraint surface enforces geometric periodicity. Upon completion of the BL mesh region the isotropic region is generated. Within the isotropic mesh generation process the input surface mesh is kept fixed at periodic constraint surfaces to insure exact periodicity. However, that assumes that the child surface of the staring surface mesh is exactly periodic with the parent surface. To insure that the rebuilt isotropic surface mesh for the constraint surfaces must re-generated with exact periodicity. That is accomplished by first duplicating the topology of the interface between the BL and isotropic mesh regions at constraint surfaces. Then the isotropic portion of each constraint surface is re-generated using the process described in the previous discussion on curved constraint surfaces. In the periodic case only the parent surface is re-generated and the child is evaluated using the periodic coordinate transformation specification of the parent constraint surface.

For cases with periodic constraint surfaces additional input is required to define the periodic coordinate transformation specifications of the parent constraint surfaces. SolidMesh will automatically generate the required information. For import of surface meshes from other systems the data must be generated manually or by some other procedure. From a user perspective, no additional input is required if the surface mesh is generated with SolidMesh. The periodic transformation specification data is stored in a file named *case_name.psdata*. SolidMesh generates that file automatically. If that file is present and there are matching constraint surfaces then periodicity will be enforced. The data required for each periodic pair includes the surface group ID for both parent and child constraint surfaces in the periodic pair, the coordinate system translation vector between the parent and child, and the coordinate system rotation matrix or transformation matrix between the parent and child. The transformation matrix (***M***) provided by SolidMesh to AFLR3 is composed of the direction cosine matrix $(M_{ij}, i = 1,3, j = 1,3)$ and the translation vector $\left(\vec{T}\right)$. The transformation matrix can be calculated from the provided axis of rotation ($dot = \vec{v} \bullet \vec{V} \square \vec{A}$), rotation angle $\left(\theta\right)$ and the reference point

(P). If there is not a rotation angle, the direction cosine matrix is the identity matrix. The transformation between points about an arbitrary point of reference can be expressed as $X' = M(X - P) + \vec{T} + P$. This can be simplified into one matrix multiplication

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = M \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \text{ where}$$

$$M = \begin{bmatrix} A_x^2 + \cos\theta\left(1 - A_x^2\right) & A_x A_y\left(1 - \cos\theta\right) - A_z \sin\theta & A_x A_z\left(1 - \cos\theta\right) + A_y \sin\theta & T_x - M_{11}P_x - M_{12}P_y - M_{13}P_z + P_x \\ A_x A_y\left(1 - \cos\theta\right) + A_z \sin\theta & A_y^2 + \cos\theta\left(1 - A_y^2\right) & A_z A_y\left(1 - \cos\theta\right) - A_x \sin\theta & T_y - M_{21}P_x - M_{22}P_y - M_{23}P_z + P_y \\ A_x A_z\left(1 - \cos\theta\right) - A_y \sin\theta & A_y A_z\left(1 - \cos\theta\right) + A_x \sin\theta & A_z^2 + \cos\theta\left(1 - A_z^2\right) & T_z - M_{31}P_x - M_{32}P_y - M_{33}P_z + P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This information is then written in the *casename.psdata* file with the corresponding data: *periodic id*, *parent periodic group id*, *child periodic group id*, M_{14}, M_{24}, M_{34}, M_{11}, M_{12}, M_{13}, M_{21}, M_{22}, M_{23}, M_{31}, M_{32}, M_{33}.

To demonstrate the basic periodic mesh capability a wedge configuration was created. The configuration and geometric definition for this case are illustrated in Figs. 13 and 14, respectively. In this case front and back surfaces are periodic planar constraint surfaces, the left and right surfaces are periodic planar constraint surfaces, and the top is a curved surface with attached BL region. This case also demonstrates a basic level of topological complexity that allows the periodic surfaces to be adjacent and share a common edge (the centerline in this case).

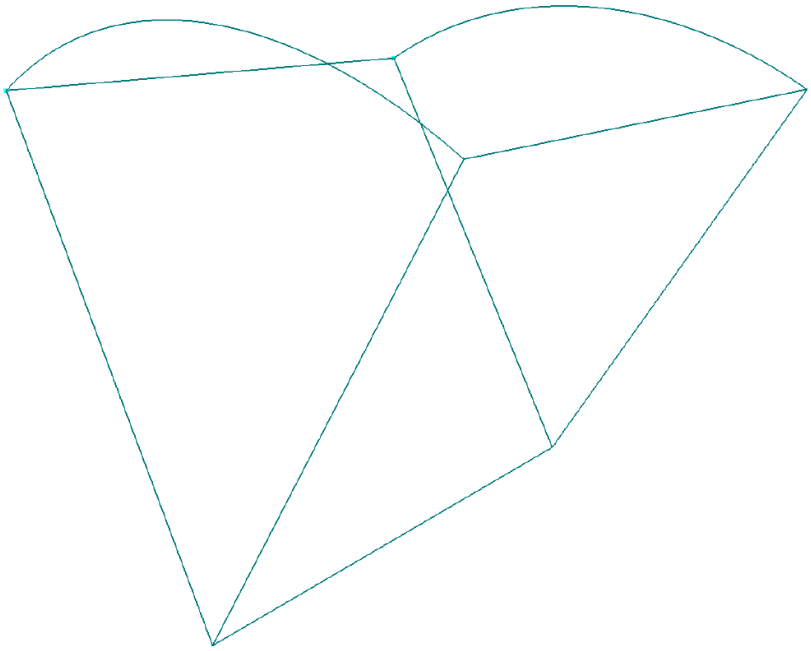


Figure 13. Simple wedge configuration test case that illustrates basic periodic mesh capability.

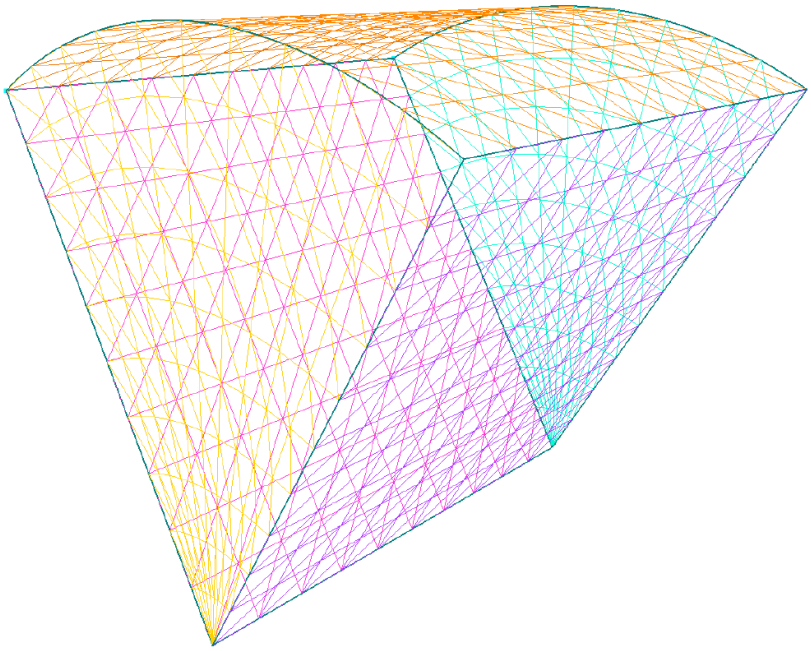


Figure 14. Geometric surfaces for wedge case.

After generation of the volume mesh the final surface mesh with rebuilt constraint surfaces is illustrated in Fig. 15. As expected the front and back periodic surfaces are identical as are the left (not shown) and right periodic surfaces.

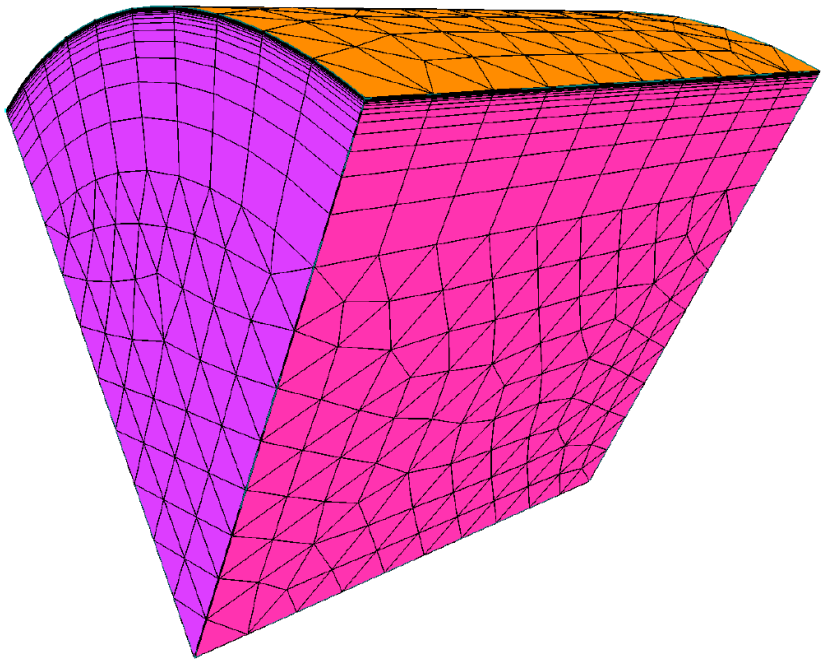


Figure 15. Final surface mesh for wedge case with periodic rebuilt surfaces that intersect the BL region.

An overlaid view of the final front and back surface meshes is shown in Fig. 16. This figure illustrates exact periodicity for these surfaces.

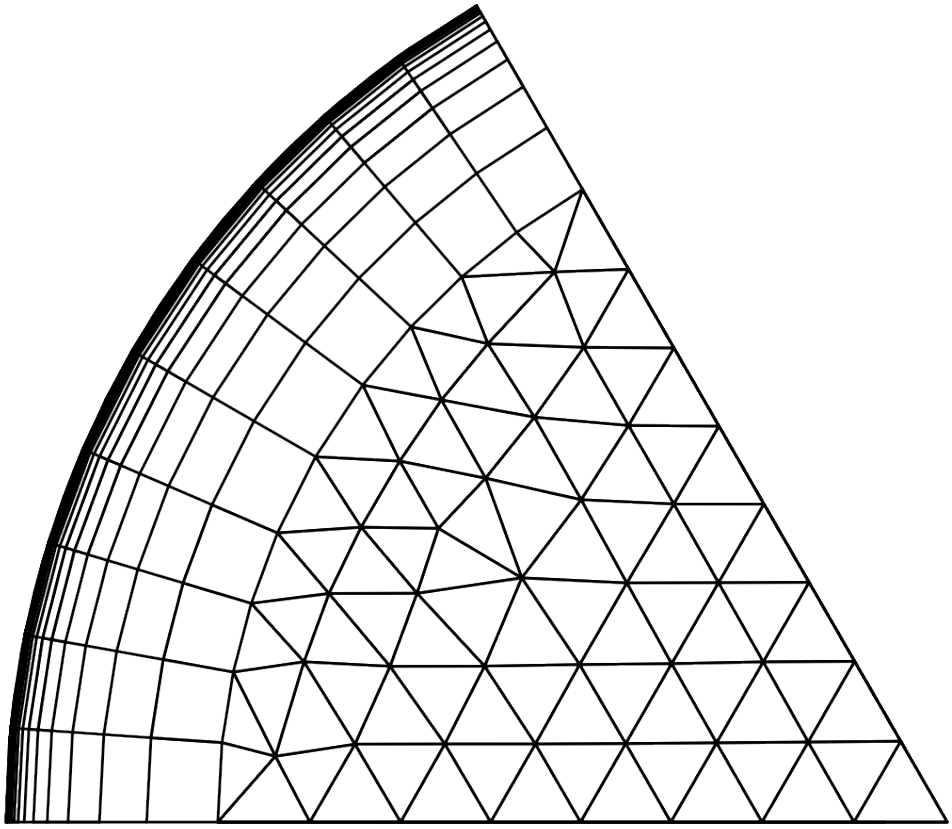


Figure 16. Overlaid view of rebuilt right and left end periodic surfaces showing exact matching.

To demonstrate the capability to rebuild curved constraint surfaces that are periodic a duct configuration test case was created. The configuration and geometric definition for this case are illustrated in Figs. 17 and 18, respectively. In this case there is an inlet with a planar constraint surface to the right and two exits with curved periodic constraint surfaces to the front and back. All other surfaces have an attached BL region.

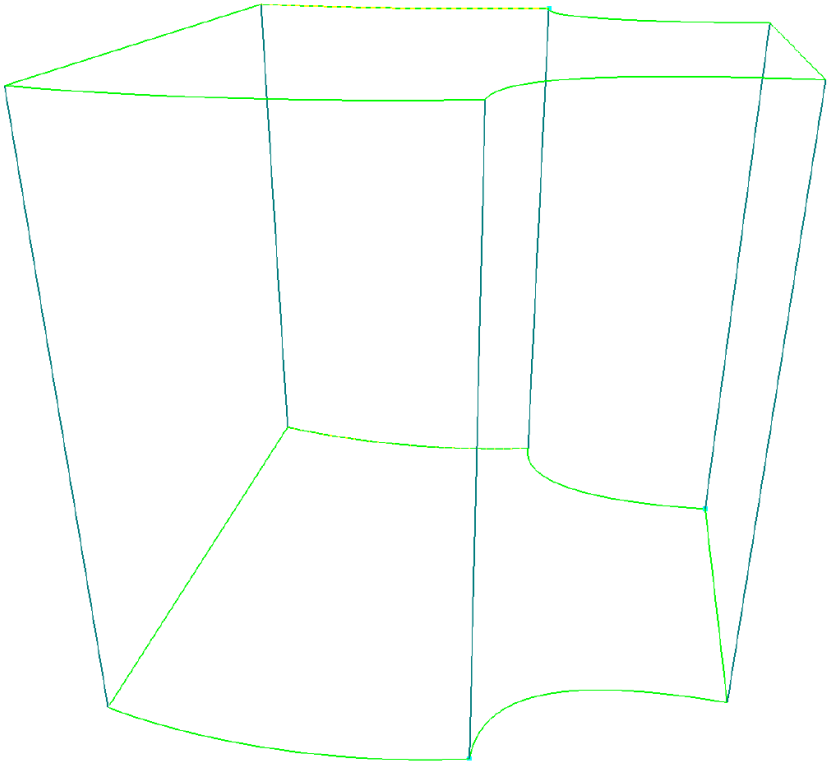


Figure 17. Duct configuration test case that illustrates periodic mesh capability with curved periodic surfaces.

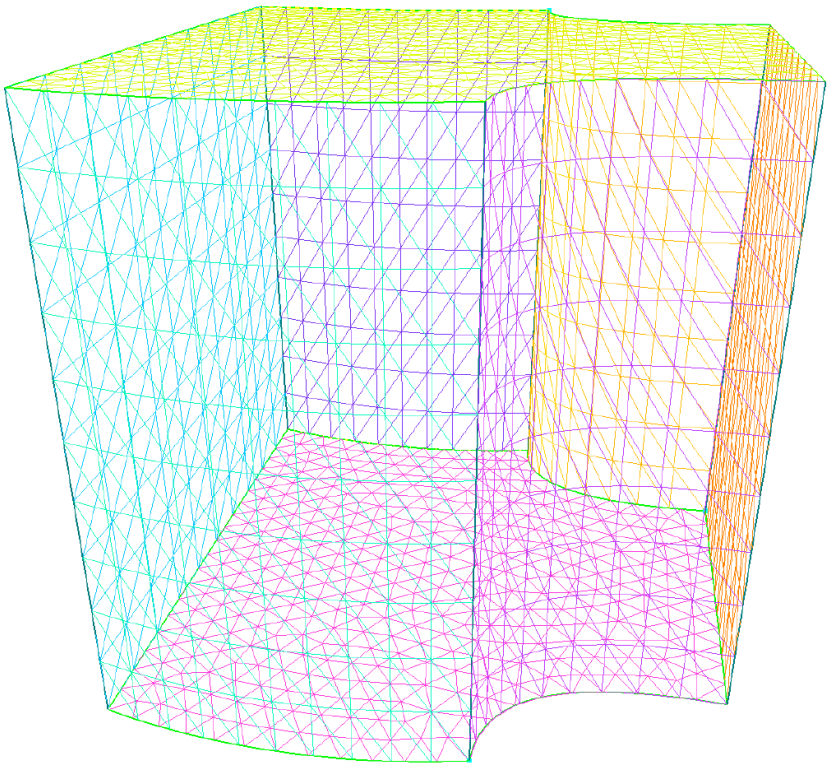


Figure 18. Geometric surfaces for duct case.

After generation of the volume mesh the final surface mesh with rebuilt constraint surfaces is illustrated in Figs. 19 and 20. As expected the front and back curved periodic surfaces are identical.

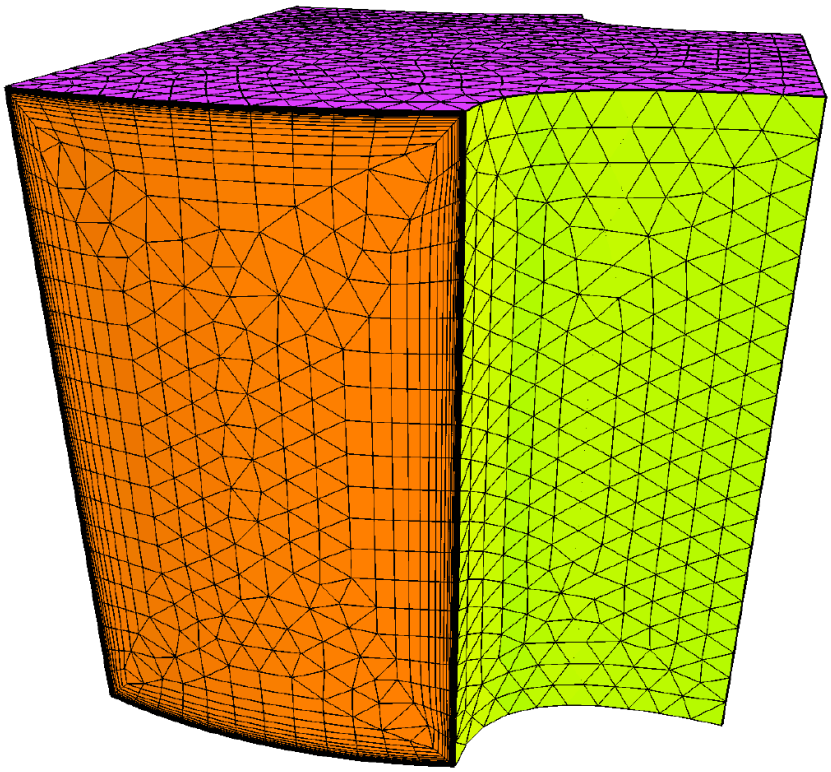


Figure 19. Final surface mesh for duct case with periodic rebuilt curved surfaces that intersect the BL region.

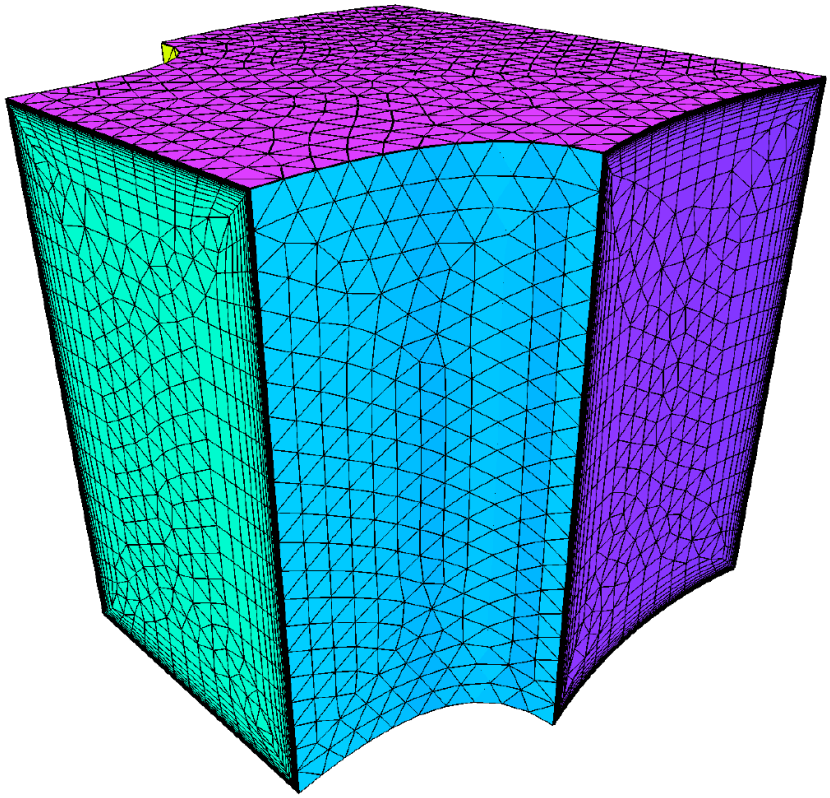


Figure 20. Back view of final surface mesh for duct case showing planar rebuilt inlet surface and back periodic rebuilt curved surface.

For completeness a cross-section view of the volume mesh elements is illustrated in Fig. 21. As expected the BL region smoothly intersects the specified constraint surfaces.

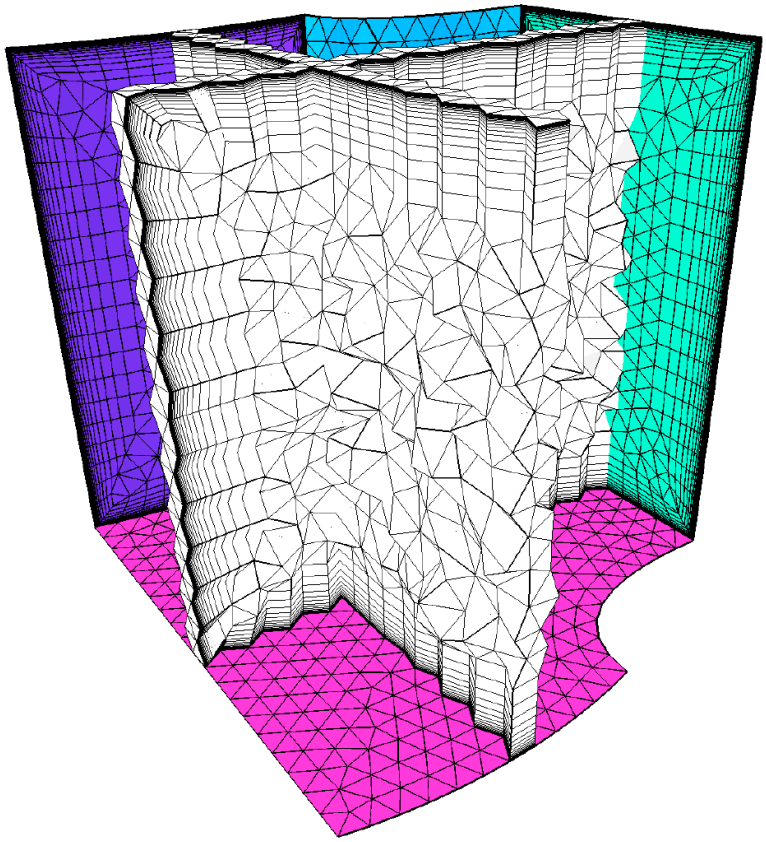


Figure 21. Volume element cross-section view.

An overlaid view of the final front and back periodic surface meshes is shown in Fig. 22. This figure illustrates exact periodicity for these surfaces.

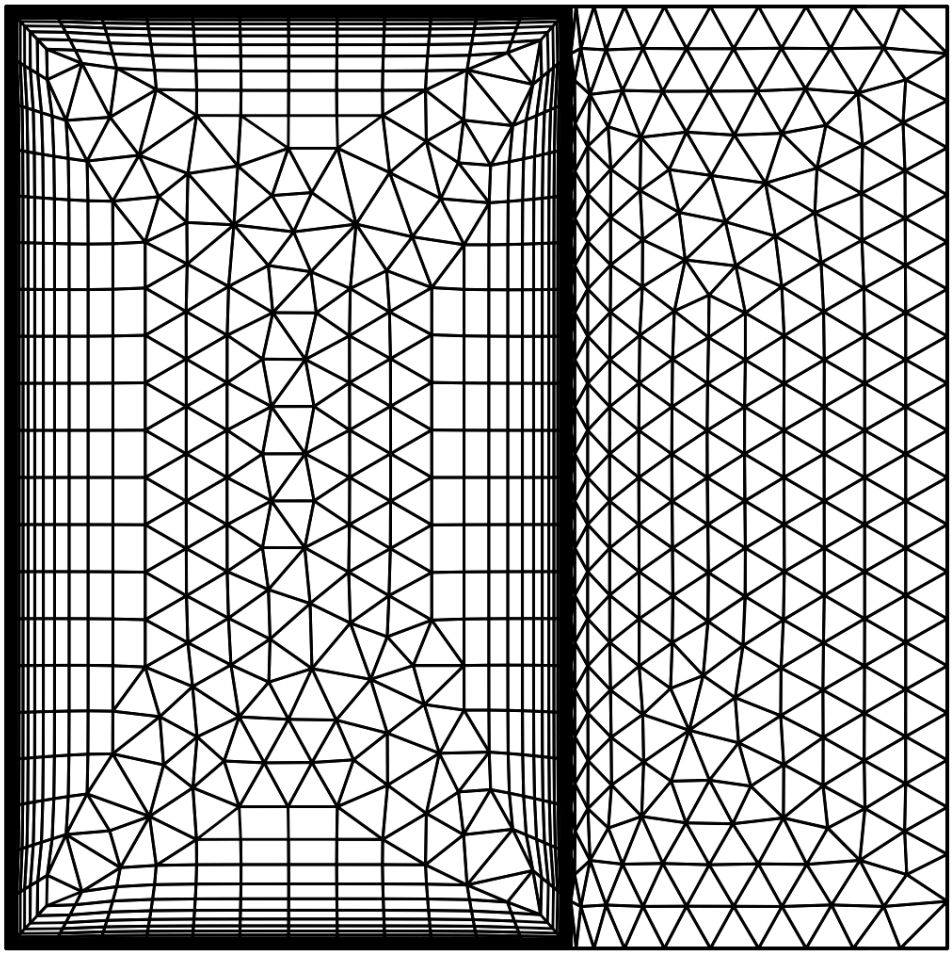


Figure 22. Overlaid view of rebuilt front and back periodic surfaces showing exact matching.

To demonstrate the capability to rebuild constraint surfaces that are periodic with asymmetric internal features a cube-column configuration test case was created. The configuration and geometric definition for this case are illustrated in Figs. 23 and 24, respectively. In this case the left and right surfaces are periodic planar constraint surfaces. There is an internal trapezoidal column adjacent to the left periodic constraint surface. The internal mesh on the left and right are expected to differ significantly left to right due to the column.

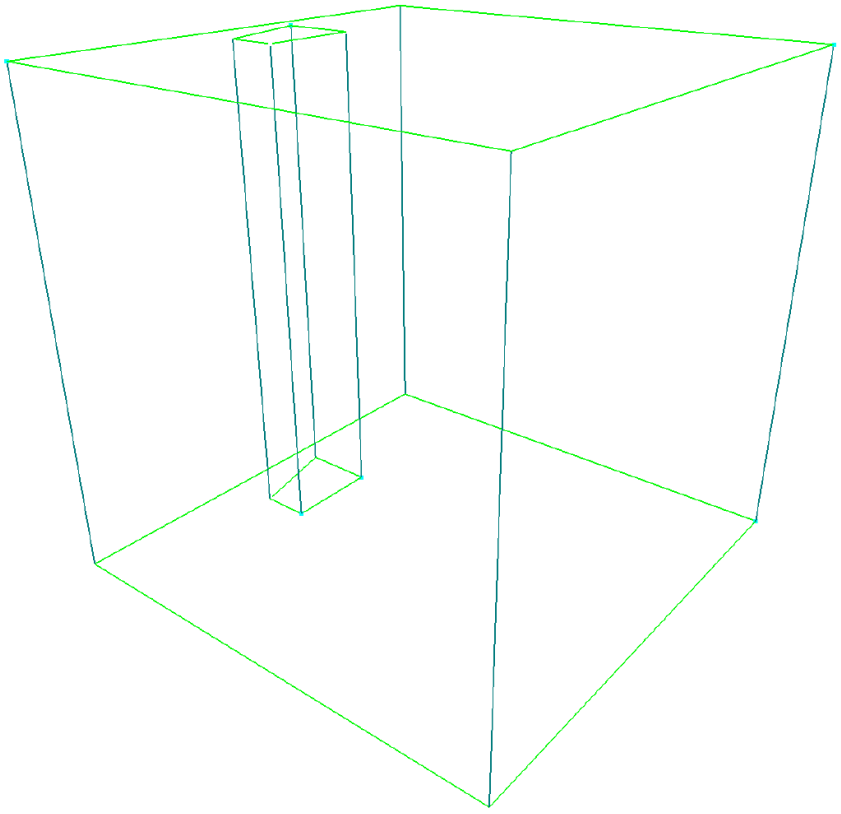


Figure 23. Cube and pillar configuration test case that illustrates periodic surface mesh capability with asymmetric internal features.

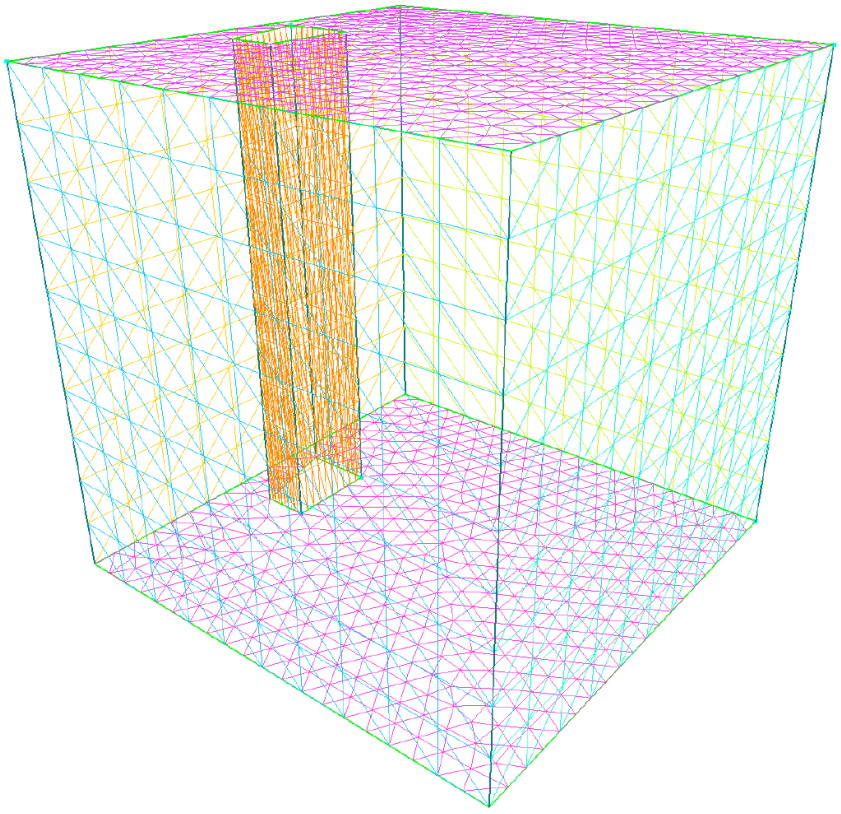


Figure 24. Geometric surfaces for cube and pillar case.

After generation of the volume mesh the final surface mesh with rebuilt constraint surfaces is illustrated in Fig. 25. As expected the left and right periodic surfaces are identical and exhibit exactly the same influence of the internal column. Comparing the surface mesh on the periodic constraint surface top and bottom edges versus left and right edges reveals that presence of the column reduced the BL region.

A cross-section view of the volume mesh elements is illustrated in Fig. 26 near the column. As expected the BL region smoothly intersects the specified constraint surfaces and the impact of the column is clearly illustrated to the left.

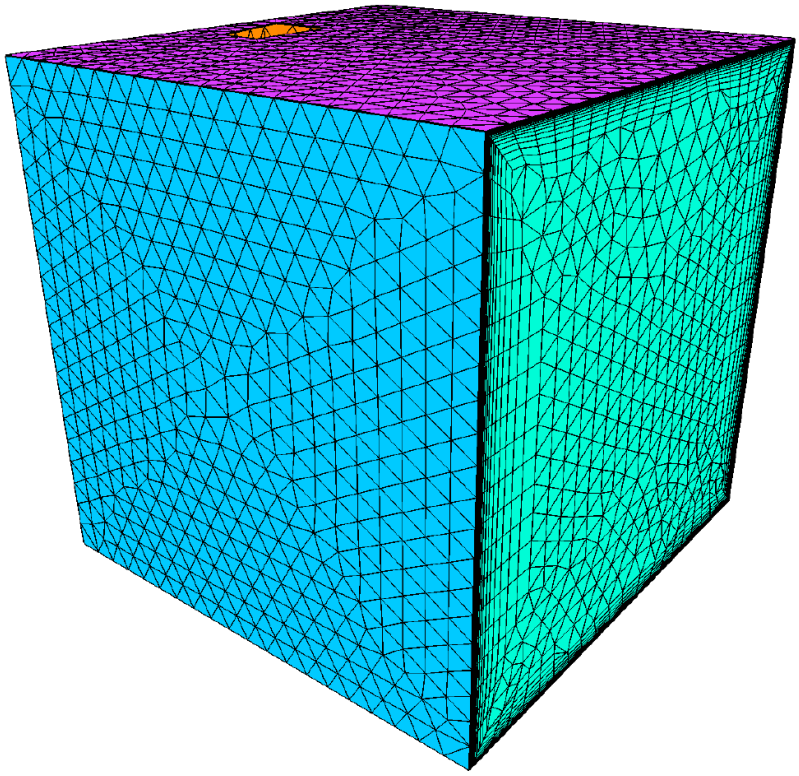


Figure 25. Final surface mesh for cube and pillar case with rebuilt surfaces that intersect the BL region showing that the internal feature impacts both periodic surfaces.

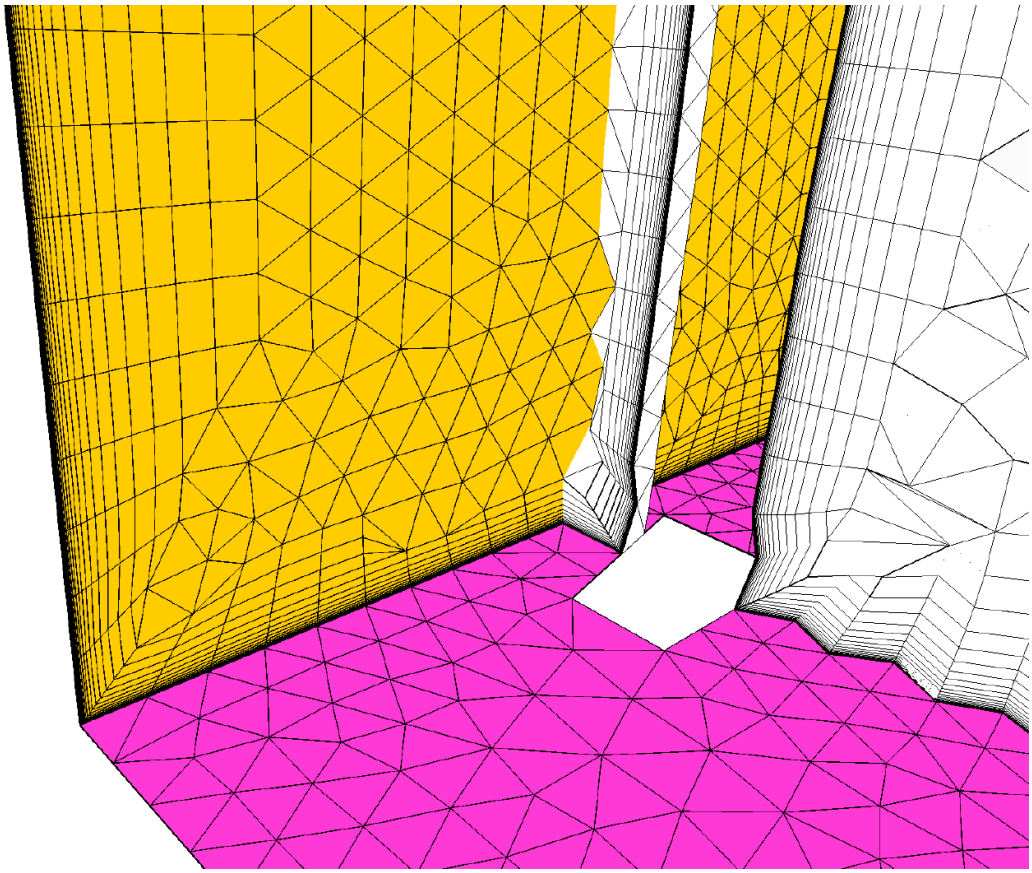


Figure 26. Volume element cross-section detail view showing mesh near asymmetric pillar

Finally, an overlaid view of the final left and right surface meshes is shown in Fig. 27. This figure illustrates exact periodicity for these surfaces with the expected matching influence of the internal feature. For comparison the results for the same test case without the periodicity condition is shown in Fig. 28. As expected the final left and right surface meshes differ significantly from left to right due to the column near the left constraint surface.

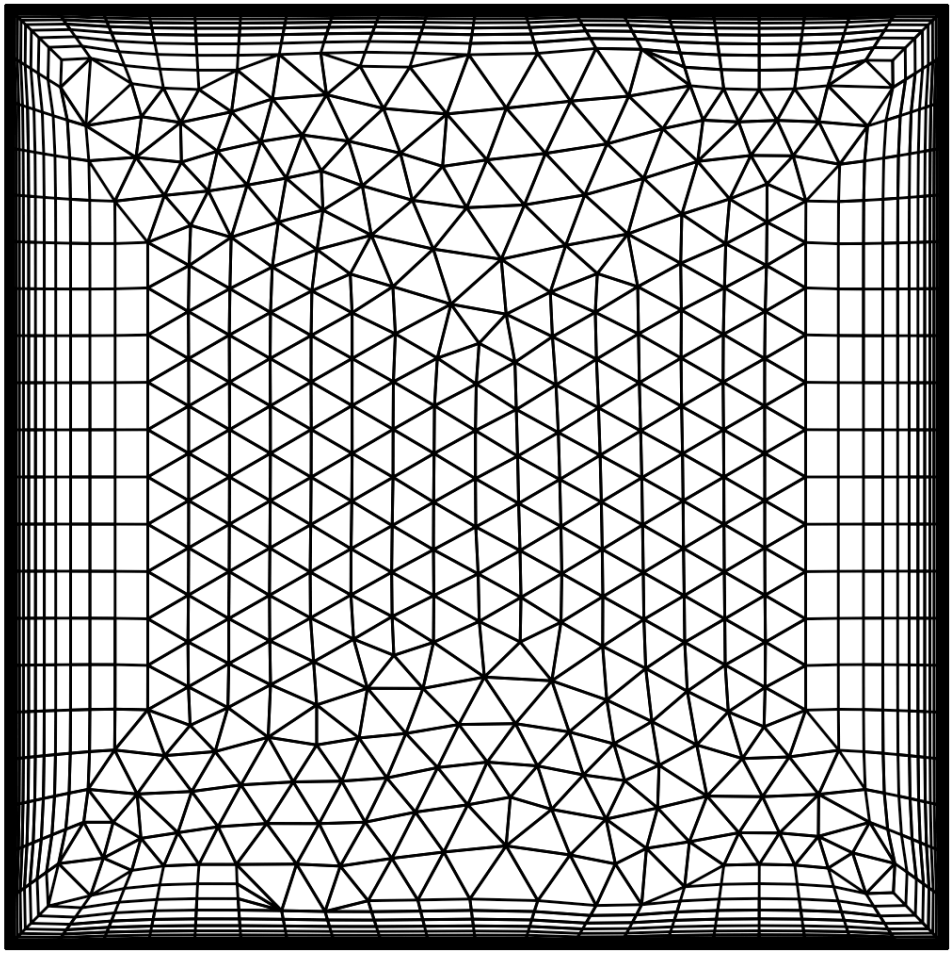


Figure 27. Overlaid view of rebuilt right and left end periodic surfaces showing exact matching and impact of asymmetric feature on BL region (top/bottom edges vs. left/right edges).

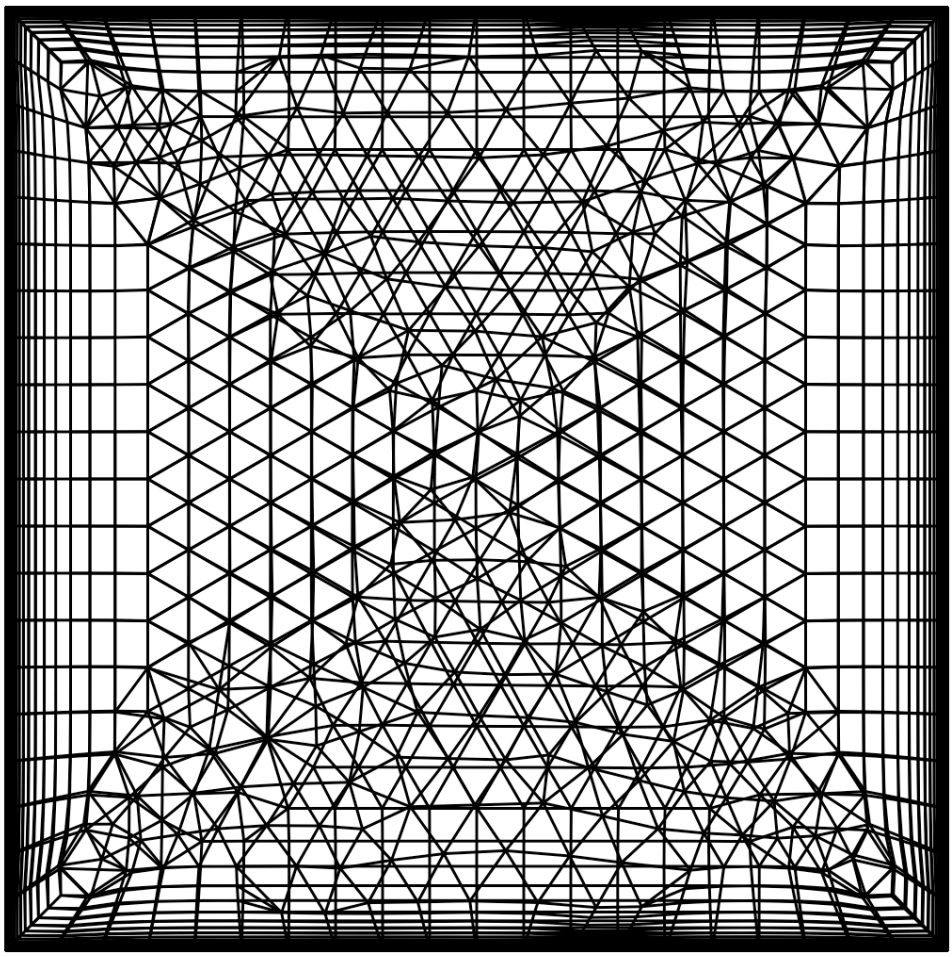


Figure 28. Overlaid view of rebuilt right and left end constraint surfaces showing significant differences without periodicity due to asymmetric internal features.

Deliverables

All the deliverables listed in the proposal have been met and the enhanced versions of SolidMesh and AFLR3 have been posted on the SimCenter Software Forum (http://www.simcenter.msstate.edu/software/forum) and initially made available to Dr. William Zilliox of ASC and the user partners. The initial capability was released in SolidMesh v5.60. In addition to the SolidMesh and AFLR3 executables, HTML-based documentation and tutorials have also been posted to the SimCenter Software Forum and

are available to all the subscribers. A perpetual license has been provided to install and use the delivered versions of SolidMesh and AFLR3 executable on DoD HPCMP sites.

Acknowledgements

This publication made possible through support provided by DoD High Performance Computing Modernization Program (HPCMP) Programming Environment and Training (PET) activities through Mississippi State University under the terms of Contract No. GS04TO1BFC0060.

References

1. Gaither, J. A., Marcum, D. L., and Mitchell, B., "SolidMesh: A Solid Modeling Approach to Unstructured Grid Generation," Numerical Grid Generation in Computational Field Simulations, Proceedings of the 7th International Grid Generation Conference, Whistler, British Columbia, September 2000.
2. Marcum, D.L., "Unstructured Grid Generation Using Automatic Point Insertion and Local Reconnection," *The Handbook of Grid Generation,* edited by J.F. Thompson, B. Soni, and N.P. Weatherill, CRC Press, p. 18-1, 1998.