The Cooper Tool

Ted Blacker
Fluid Dynamics International
500 Davis St. #600
Evanston, IL 60201
ted@fdi.com

Abstract. The cooper tool is a new approach to automation of hexahedral meshing (or a wedge/hexahedral element mix if desired) for a general subset of geometries. The tool recognizes applicable geometries and decomposes such into logically single axis swept subvolumes (barrels) with barrel caps bounding the decomposed pieces. It then enforces strict compatibility constraints by imprinting, intersecting, and matching all bounding barrel caps to rapidly generate a well formed continuous mesh throughout the geometry.

1.0 Introduction

Numerous approaches have been proposed and investigated ([2], [4], [5], [6], [7], [8], [9], and [10]) for automation of the hex meshing process for complex geometries. As with many difficult problems, each of the proposed solutions contains both positive and negative attributes associated with the technique employed. However powerful the tool's claims are individually, counter examples of geometries where another tool is definitely superior can be easily generated. However, the conglomeration of the techniques into a toolbox assortment may be able to realistically defend a claim of global optimality, especially if the tools themselves (or a tool manager) can determine a-priori the tool most suited to the geometry at hand. The cooper tool, described below, offers a unique automation tool to consider in such a toolbox.

A "cooper" is an artisan who practices the historic trade of barrel making. Although today's society has a fairly low demand for wooden barrels, the practise of forming tight fitting ribs of wood with bounding barrel caps is a natural analogy of the process described in this paper - hence the name "cooper tool". The cooper tool recognizes and subdivides applicable geometries into multiply connected subvolumes or "barrels". Each barrel is logically a single axis sweep, bounded on the sides by barrel "ribs" (edges of regular quadrilateral rows of elements) and on the ends by "caps" and "subcaps" (topologically equivalent paired faces or sets of paired faces). Although barrels are always logically single axis sweeps, geometrically they are arbitrary. This technique enforces mesh compatibility by ensuring one-to-one matches between all subcaps on each barrel cap, using projection, imprinting, intersection, and seaming techniques to accomplish an optimal matching. It uses the barrel ribs to insure a valid transformation for all subsequently generated interior nodes along the projection direction. Element connectivity thus becomes completely determined by the cap meshes and thus actual element generation is extremely rapid.

As will be described in this paper, the cooper tool has the following positive attributes:

- Complete automation for applicable geometries,
- High element quality,
- Boundary sensitivity (best elements along the boundary),
- Orientation insensitivity (same mesh for transformed geometries),
- Speed.

The cooper tool will mesh geometries which are logically decomposable into barrels. This includes a wide range of geometries as shown by the example in Figure 1. This technique is equally applicable to the generation of wedge elements (5 faced triangular prism) and/or mixed wedge/hexahedral meshes by simply allowing triangle and/or mixed triangle/quadrilateral cap meshes.

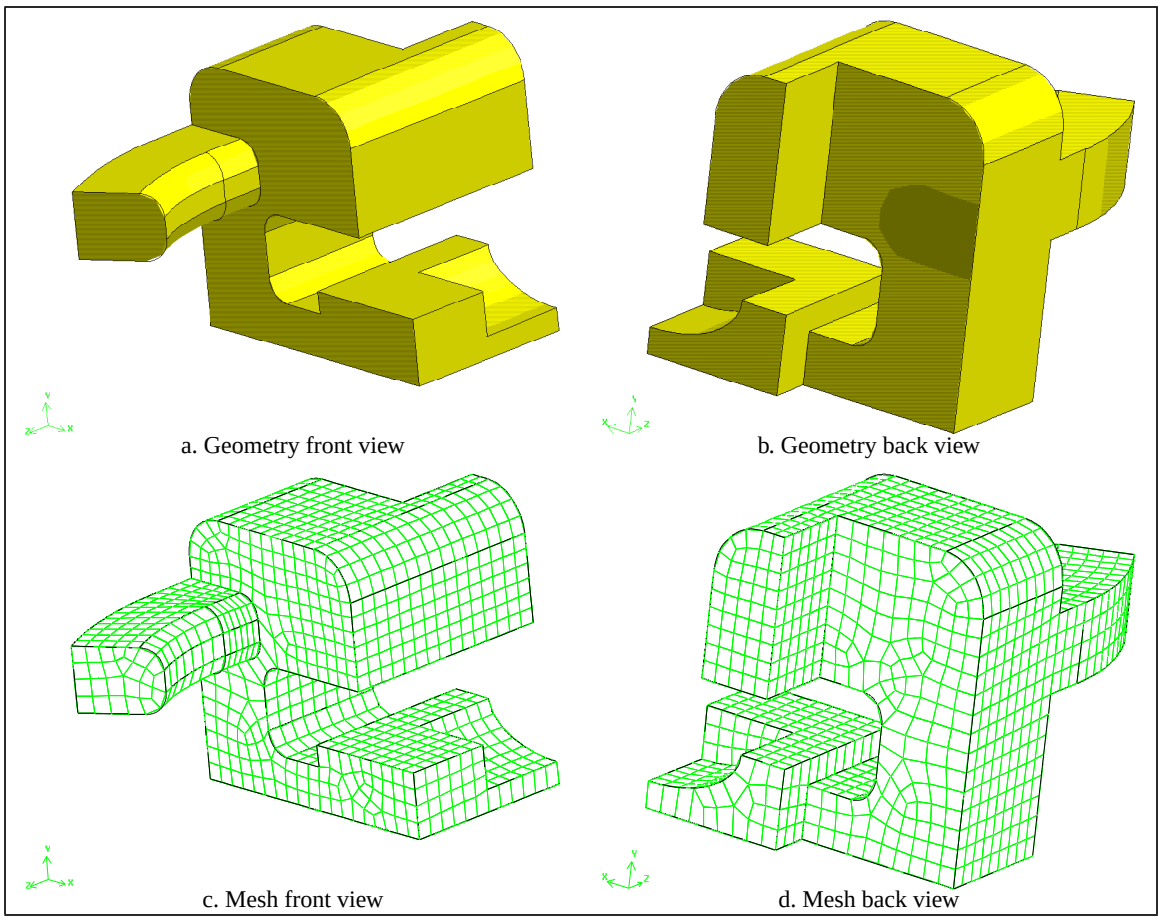


Figure 1. Example geometry meshed using the cooper tool.

2.0 Cooper Process

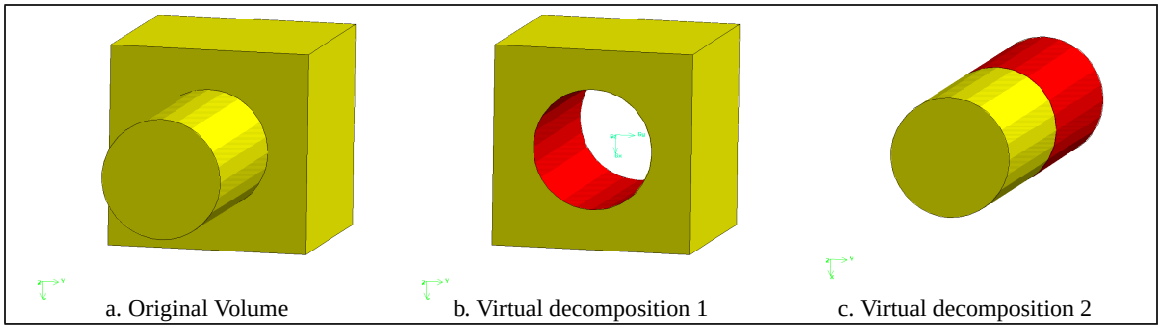
The process of meshing using the cooper tool couples geometric evaluation, virtual decomposition, and projection. It requires the combination of a general purpose, robust face meshing tool such as the paving algorithm [1], and the use of a predictable regular mesh generation technique called submapping [9] and [10]. This process involves the following steps:

1. Classification of applicable volumes and sorting of faces into sides and caps.
2. Meshing of all side faces using submapping.
3. Formation and separation of barrel volumes.
4. Iteratively matching and imprinting all subcaps to produce stacks of caps.
5. Meshing of cap stacks.
6. Projection of the barrel volume interior nodes.
7. Formation of volume elements in each cap stack.

3.0 Volume Classification

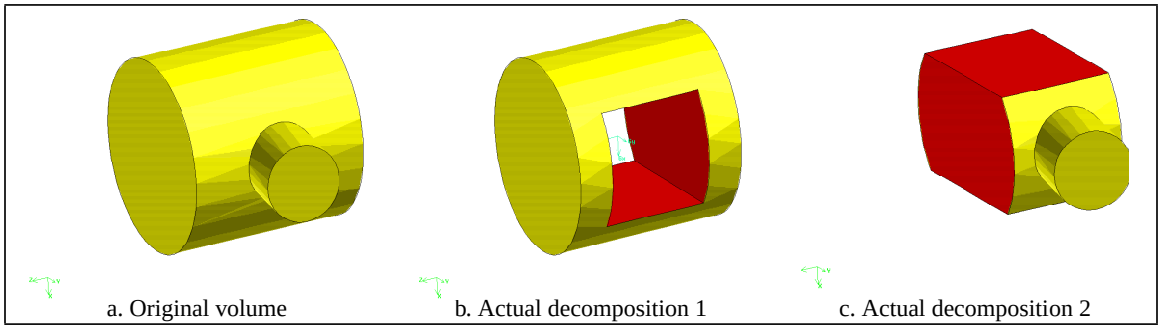
In general terms, a volume can be automatically meshed using the cooper tool if a logical decomposition can be found which would admit coupled single axis projections. For example, the volume in Figure 2 could be logically divided into 2 projections, each of which is a single axis projection. The cooper tool currently

requires all shared “virtual” faces (the faces that would be present if an actual decomposition occurred) between such projections to be in the same direction. The volume shown in Figure 2a meets this criteria and is correctly classified by the cooper tool while the volume shown in Figure 3a does not - again correctly determined by the cooper tool. To mesh the volume shown if Figure 3a, the user would be required to manually decompose the original volume into pieces, each of which can be meshed using the cooper tool. Such a decomposition is shown in Figure 3b and Figure 3c. The lifting of this common projection restriction is a natural next step in this work.



a. Original Volume b. Virtual decomposition 1 c. Virtual decomposition 2

Figure 2. Volume meshable with the cooper tool. The projection direction is consistent along the shared virtual face (dark shaded).



a. Original volume b. Actual decomposition 1 c. Actual decomposition 2

Figure 3. Decomposition of a volume unmeshable using the cooper tool into 2 volumes which are meshable with this tool. The projection direction on the shared face (dark shaded) need not be in the same direction.

In practice, to determine if a volume can be meshed using the cooper tool, all faces of the volume are categorized as either a side face or a cap face. A side face must admit a submapping mesh (i.e. be submappable) and the combination of all adjacent side faces must also admit a submapped mesh. To illustrate this, Figure 4 shows three submappable faces which can all be side faces, as their combination is itself submappable. Conversely, only two of the three submappable faces shown in Figure 5 can be side faces, as their combination is

not submappable. Determining a submappable face is accomplished by a simple calculation based on topol-

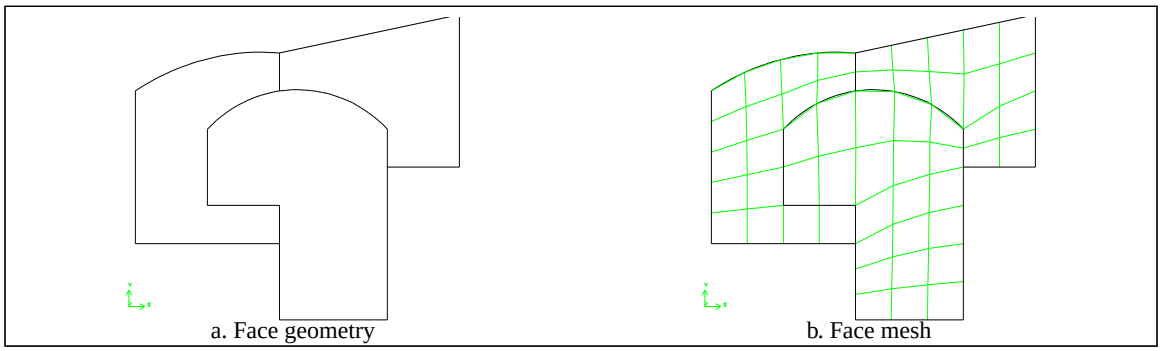


Figure 4. Allowable combination of side faces, each of which is submappable, and whose combination is also submappable.

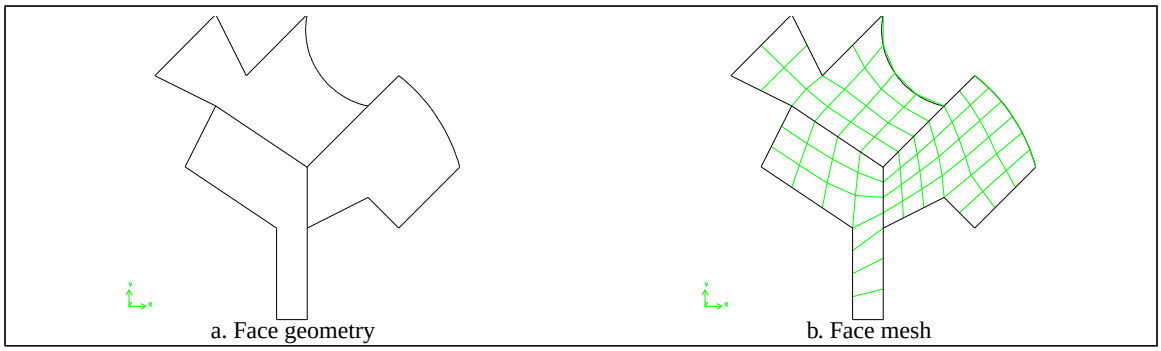


Figure 5. Disallowed combination of side faces. Although each face is submappable, the combination, in its current configuration, is not.

ogy and angle designations and does not require a meshing attempt.

A cap face is simply defined as any face which is not a side face.

If a valid sorting of the volume faces into caps and sides can be found, the only remaining constraint for the volume to be cooperable is that the volume contain at least two caps whose dihedral angles with the attached side faces is less than π. This insures that the projection along the side faces not produce inverted elements. Figure 6 shows the classifications of the faces of the volume shown meshed in Figure 1 which enable it to be cooperable.

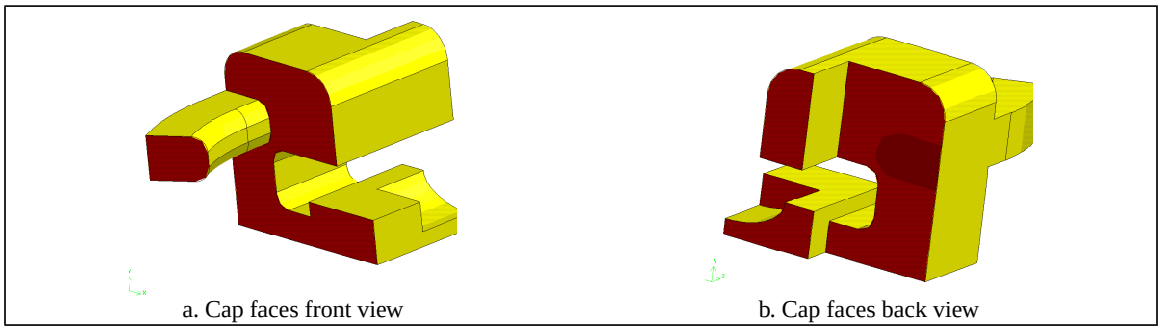


Figure 6. Classification of the faces of an example volume into side faces (lightly shaded) and cap faces (dark shaded).

If a volume is cooperable, all side faces are meshed at the requested density using submapping. If cap faces are unmeshed, they initially remain so to allow greater flexibility during the barrel formation and meshing process.

4.0 Barrel Formation

Since any cooperable volume will contain at least two caps whose dihedral angles with the attached side faces is less than π, these caps are collected in a list and termed “end caps”. A barrel is defined by two barrel caps, an upper and a lower, and a set of exterior ribs.

Barrel formation begins by choosing one of the unused end caps as the upper cap for the new barrel. All exterior nodes on this end cap are gathered to form the genesis of the ribs which will progress along the outside of the barrel. Due to the constrained regularity of the submapped side faces, these ribs can then progress logically down a barrel’s side. Since each interior side face node is connected logically to only four other nodes, this process involves simply choosing the only node connected to the end of the rib which is not yet part of a rib, as shown in Figure 7. All of the exterior ribs of the barrel progress a complete layer at a time.

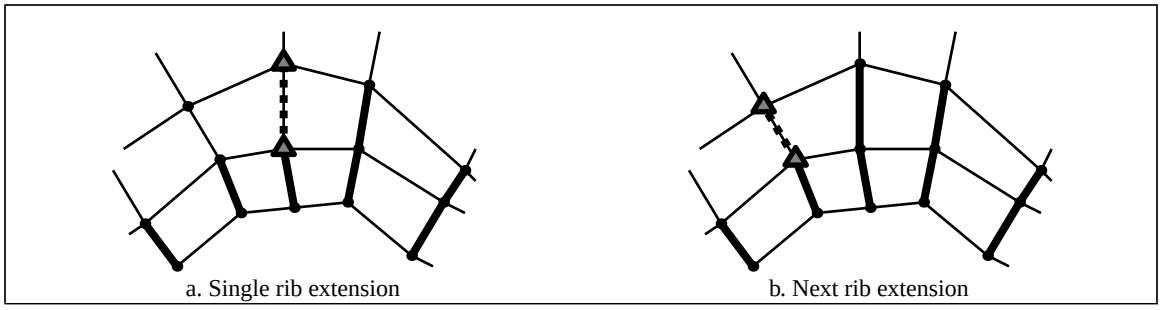


Figure 7. Extension of one rib to the only attached node on the side face not yet part of a rib. Existing ribs are shown in heavy lines.

As soon as any rib on the front reaches another existing cap, rib formation ceases. A barrel is guaranteed to have a minimum of one layer.

Once propagation has ended, a lower barrel cap is formed using the terminated nodes of the barrel’s ribs. If this new cap matches an available end cap it replaces the lower cap of the barrel. Otherwise, the barrel has terminated into one or more barrel caps which are not ends. To complete the barrel, the lower cap must then be reformed to contain subcaps which account for the attached caps. For example, for the volume in Figure 8a the first barrel formed using the side mesh in Figure 8b is shown in Figure 8c. The lower cap must then be combined with the adjacent cap to form an end cap as shown in Figure 8d. The barrels for a given volume are always a reproducible unique set and thus independent of the formation sequence.

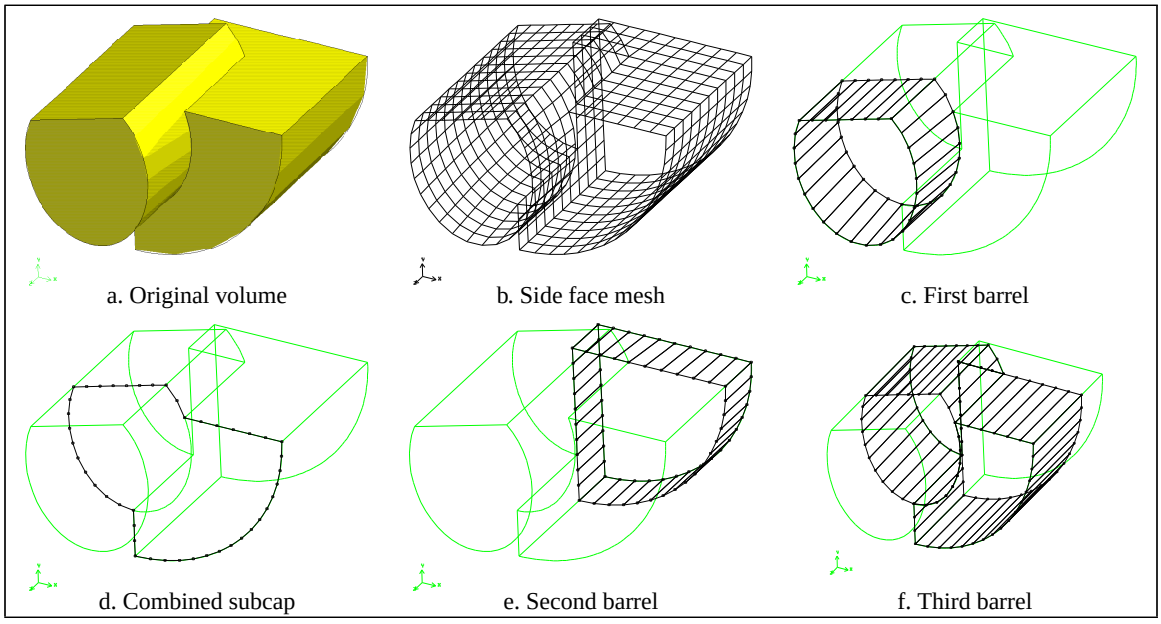


Figure 8. Formation of the barrels in a cooperable volume.

5.0 Imprinting Subcaps

As described above, each barrel will have an upper and a lower cap (designation of which cap is upper/lower is arbitrary) which logically connect exterior boundary nodes. This connection is maintained with the exterior barrel ribs. However, for all nontrivial volumes, one or both of the barrel caps are composed of subcaps as shown in Figure 9. These subcaps rarely match, and even when subcaps do match no direct link between interior nodes (i.e interior ribs) of the subcaps has been established. In order for a valid projection to occur, each of the subcaps must be paired logically (node for node) similar to the matching of the upper and lower cap of the barrel. Let a be a subcap of the upper barrel cap and b be a subcap of the lower barrel cap. A matching operator ${}^{B}\Lambda$ is defined for each barrel B such that $a = {}^{B}\Lambda_{b}$ (read subcap a matches b on barrel B) and inversely $b = {}^{B}\Lambda_{a}$.

This matching has the side effect of generating interior ribs within the volume, which contain two nodes, one on the upper and one on the lower cap.

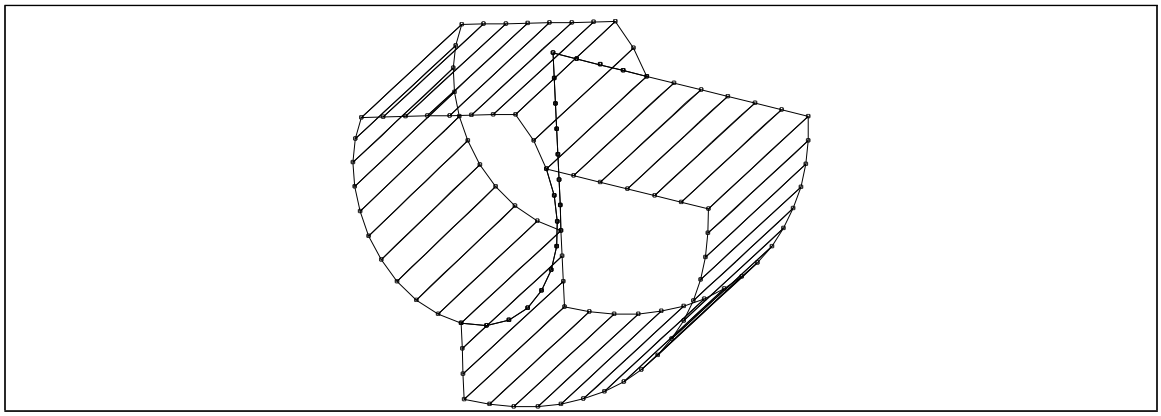


Figure 9. The barrel subcaps on a typical barrel. The upper and lower caps contain subcaps which do not match (see Figure 8e which shows the barrel formation).

Subcap matching is accomplished by choosing one of the barrel end caps and systematically imprinting each of the cap's subcaps onto the opposite cap of the barrel. This imprinting process involves classification, projection, intersection, seaming, and resolving overlaps as will be explained below.

The imprinting process relies on the information inherent in the existing rib connections of the subcaps with each other. Initially these will only occur around the exterior of the boundary. However, as matching progresses these rib connections will include interior boundaries as well.

The subcap to be imprinted is first classified against all subcaps on the opposing end of the barrel. Let $\Gamma_{a}^{\ i}$ be loop i of boundary cap a, and likewise $\Gamma_{b}^{\ i}$ be loop i of boundary cap b, as shown in Figure 10a and Figure 10b. In this and subsequent schematics, two circles representing the two barrel caps are shown. Nodes at corresponding positions (e.g. farthest right to farthest right, top to top, etc.) may be assumed to be

connected through barrel ribs as shown in Figure 10c. A subcap boundary loop can be divided into two sets:

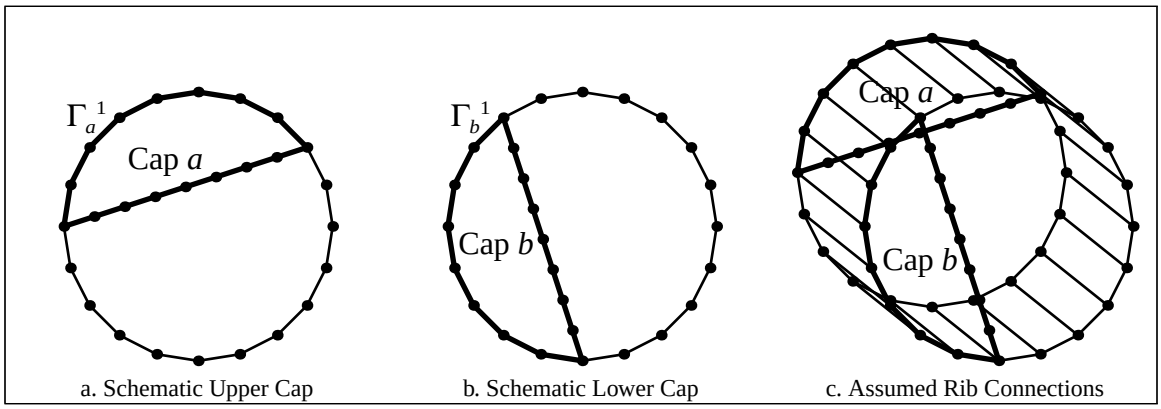


Figure 10. Barrel cap schematic showing an upper and lower cap which are assumed to be connected through ribs along the complete exterior boundaries by connecting corresponding locations.

${}^{R}\Gamma$ which contains the portions of the loop which are attached to barrel ribs and ${}^{S}\Gamma$ which contains the portions of the subcap loops which are not attached to barrel ribs as shown in Figure 11. Γ_{ab} can then be

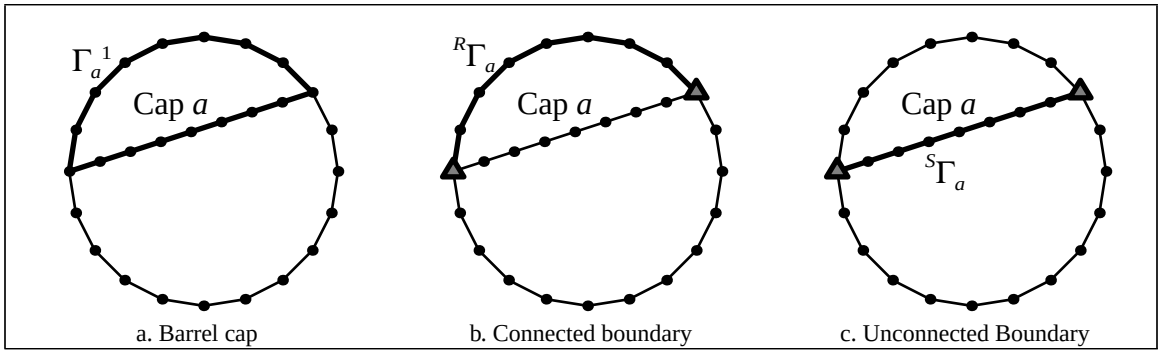


Figure 11. Division of a barrel cap loop into two sets based on the connection of nodes to existing ribs of the barrel. The cap boundary shown in (a) is divided into nodes with ribs (b) and nodes without (c).

defined as the subset of ${}^{R}\Gamma_a$ that is connected to Γ_b through the existing ribs of the barrel as shown in Figure 12a. Inversely, Γ_{ba} is the portion of ${}^{R}\Gamma_b$ which is connected to Γ_a through the same ribs as shown in Figure 12b. Γ_{ab} may be continuous or discontinuous, and may span multiple loops of one/both of the caps. The characteristics of Γ_{ab} determine the type of matching and/or imprinting that will occur.

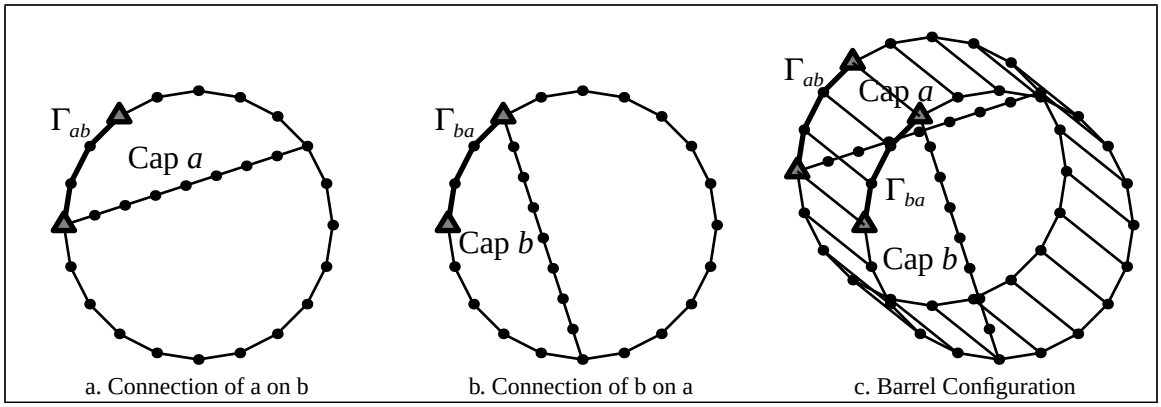


Figure 12. Determination of connected segments of barrel cap loops using connected ribs of the barrel.

Upon successful completion of subcap imprinting, each subcap of the barrel will be paired with an equivalent subcap on the opposite end of the barrel, i.e.

$$(a = {}^{B}\Lambda_b)\forall(a)$$

$$(b = {}^{B}\Lambda_a)\forall(b)$$

This pairing is stored as a double pointer on each cap/subcap. A cap/subcap may have at most two matching barrel caps because it may be attached at most to two barrels.

5.1 Subcap Matching

If Γ_{ab} is continuous on each loop i of cap a, $\{\Gamma_{ab} = \Gamma_a^{\,i}\}\forall i$ and $\{\Gamma_{ba} = \Gamma_b^{\,i}\}\forall i$, then the two caps are one-to-one matches, and the barrel match pointers are set between the two caps a and b. If there are no subcaps on either end of the barrel, then the caps by definition meet this criteria.

5.2 Subcap Subtraction

A subcap subtraction must occur when the projection of a subcap is completely within an opposing cap or subcap. More precisely, if Γ_{ab} is continuous on one loop i of subcap a, $\Gamma_{ab} \neq \Gamma_a^{\,i}$ and $\{\Gamma_{ac} = 0\}\forall(c \neq b)$, then a subcap subtraction must be performed. This occurs when a subcap's boundary is connected along one portion of its boundary to only one other opposing cap/subcap. A subcap is subtracted by projecting the "non-shared" boundary portion onto the opposing cap/subcap and then splitting the opposing cap/subcap into two appropriate caps. Let $\hat{\Gamma}_{ab}$ be the nonshared boundary portion of $\Gamma_a^{\,i}$, or

$$\hat{\Gamma}_{ab} = \Gamma_a^{\,i} - \Gamma_{ab}$$

$\hat{\Gamma}_{ab}$ is then projected onto cap b to form a new boundary segment $\hat{\Gamma}_{ba}$. Two new caps, $\hat{b}$ and b' are now formed, where

$$\hat{b} = (\hat{\Gamma}_{ba} + \Gamma_{ba})$$

$$b' = \{\hat{\Gamma}_{ba} + (\Gamma_b^{\,i} - \Gamma_{ba})\}$$

The projected boundary segment, $\hat{\Gamma}_{ba}$ is then connected to the original non-shared boundary portion $\hat{\Gamma}_{ab}$ by interior ribs. These ribs contain only the two end nodes, and not any intermediate nodes along their length

through the barrel. Figure 13 shows one such subtraction. The newly formed cap $\hat{b}$ is now a match to the projected cap a and the matching barrel pointers are set accordingly.

$$a = {}^{B}\Lambda_{\hat{b}}$$

$$\hat{b} = {}^{B}\Lambda_{a}$$

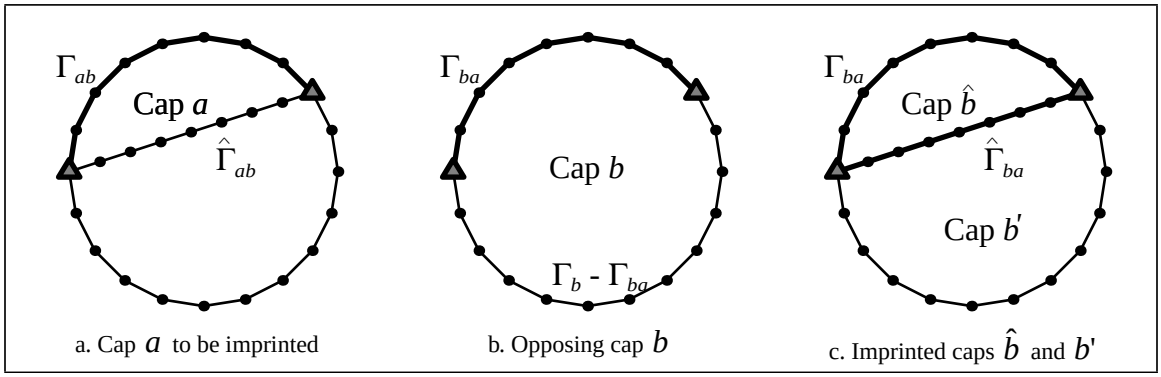


Figure 13. Barrel cap subtraction to form matching subcaps on either end of a barrel. Interior ribs are formed connecting the new nodes.

This example is for a rather simplistic case, but the same principles hold for more complex cases such as when Γ_{ab} spans multiple boundaries i of cap b or when $i > 1$ and the overlapping portion of the boundary is a complete loop of cap b or

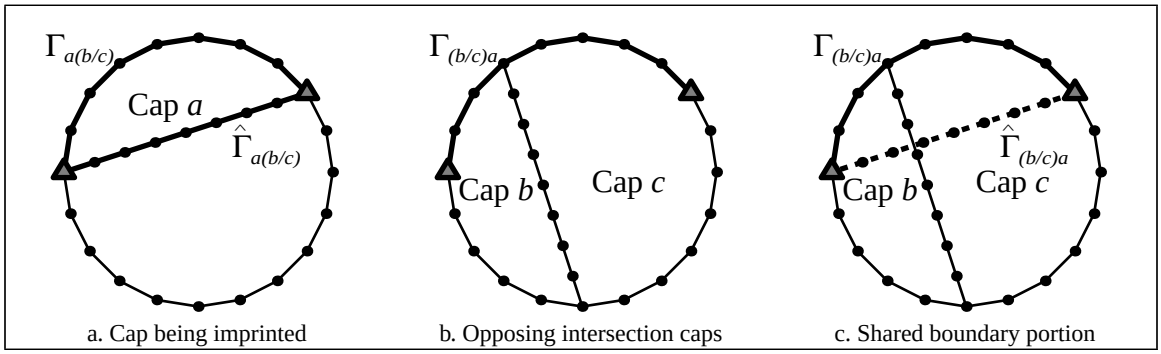
$$\Gamma_{ba} = \Gamma_b{}^{j}; (j \in (2, i))$$

5.3 Subcap Intersections

A subcap is said to intersect an opposing subcap if the projection is only partially in the opposing cap. More precisely, if one loop i of subcap a, contains a continuous boundary segment $\Gamma_{a(b/c)}$ which is connected through ribs to two subcaps b and c such that the inverse of $\Gamma_{a(b/c)}$, $\Gamma_{(b/c)a}$ is defined as

$$\Gamma_{(b/c)a} = (\Gamma_{ba} + \Gamma_{ca})$$

then a subcap intersection exists and must be resolved. This case is shown in Figure 14. Let $\hat{\Gamma}_{a(b/c)}$ be the



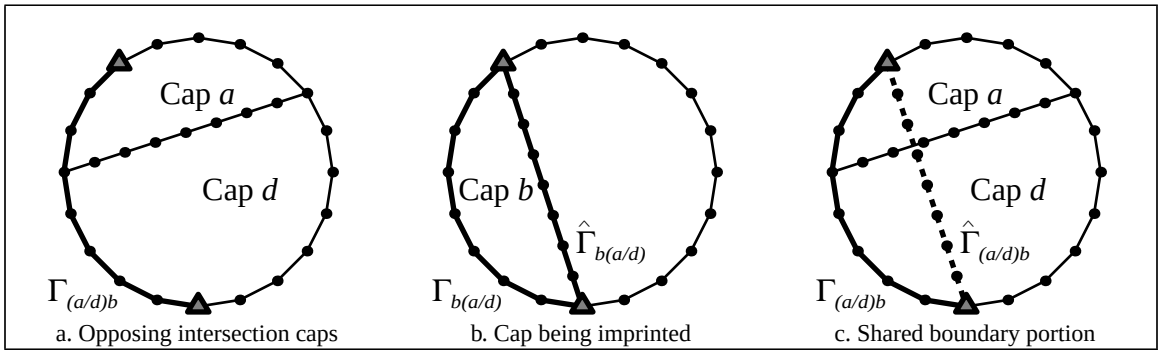
a. Cap being imprinted b. Opposing intersection caps c. Shared boundary portion

Figure 14. Intersection subcap schematic showing the intersection of barrel subcap a into opposing subcaps b and c.

nonshared boundary portion of $\Gamma_a^{\,i}$, or

$$\hat{\Gamma}_{a(b/c)} = (\Gamma_a^{\,i} - \Gamma_{a(b/c)})$$

From the definition of barrels, it can be shown that there will always exists an equivalent intersection case viewing the barrel from the opposing end using either cap b or cap c. In other words, assuming we use subcap b, a boundary segment $\Gamma_{b(a/d)}$ will exist where subcap d is on the same barrel end cap as subcap a as shown in Figure 15.



a. Opposing intersection caps b. Cap being imprinted c. Shared boundary portion

Figure 15. Intersection subcap schematic showing the intersection of barrel subcap b into opposing subcaps a and d.

In either case, there is one common cap, cap a in this case, which forms the basis of an intersection. $\hat{\Gamma}_{a(b/c)}$ is then projected onto caps b/d and the closest intersection node p is found. This intersection node must meet the even parity constraint if generating an all-quadrilateral/hexahedral mesh. Let Γ_{bp} be the boundary segment connecting Γ_{ba} to the intersection point p, and let $\hat{\Gamma}_{ap}$ be the projected portion of $\hat{\Gamma}_{a(b/c)}$ also connecting Γ_{ba} to the intersection point p as shown in Figure 16a. A new cap $\hat{a}$ is then defined as

$$\hat{a} = \Gamma_{ba} + \Gamma_{bp} + \hat{\Gamma}_{ap}$$

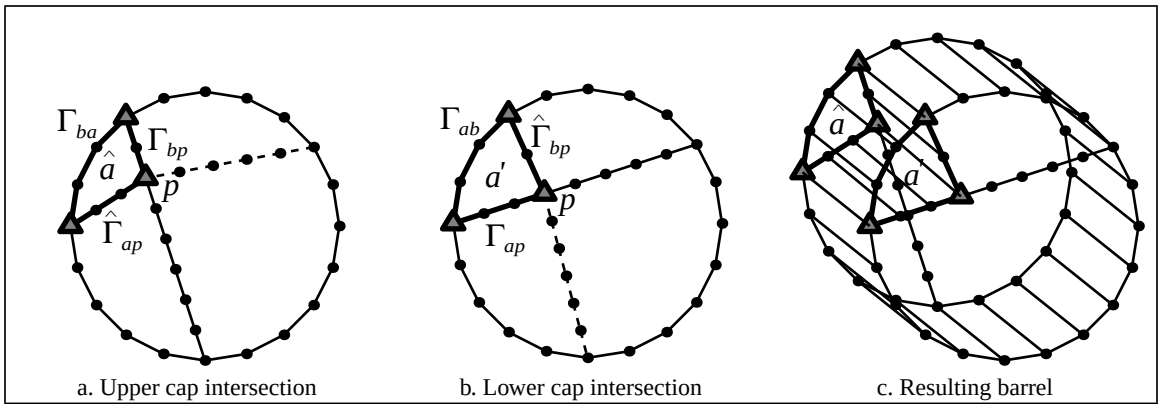
Similar projections are then performed in the reverse direction as shown in Figure 16 and a second new cap can then be defined as

$$a' = \Gamma_{ab} + \hat{\Gamma}_{bp} + \Gamma_{ap}$$

The boundary portions Γ_{ap} and Γ_{bp} are then respectively connected by interior ribs to $\hat{\Gamma}_{ap}$ and $\hat{\Gamma}_{bp}$ as shown in Figure 16c, and the matches between these two caps are set. The new caps $\hat{a}$ and a' are appropri-

$$\hat{a} = {}^{B}\Lambda_{a'}$$

$$a' = {}^{B}\Lambda_{\hat{a}}$$



a. Upper cap intersection b. Lower cap intersection c. Resulting barrel

Figure 16. Barrel cap intersection to form one set of matching subcaps on either end of a barrel. Interior ribs are formed connecting the new nodes.

ately subtracted from caps a and b respectively. These new subcaps are not projected as well, as a recategorization and processing will accomplish this with minimal additional work.

5.4 Subcap Seaming

Upon projection of subcaps, occasionally small cracks can occur between the projected cap and the existing cap as depicted in Figure 17. These cracks can be eliminated by combining adjacent nodes similar to the seaming operation which occurs during paving [Blacker and Stephenson, 90]. Care must be taken to insure

that nodes on edges of the external boundary are not combined by checking rib connections throughout the barrels.

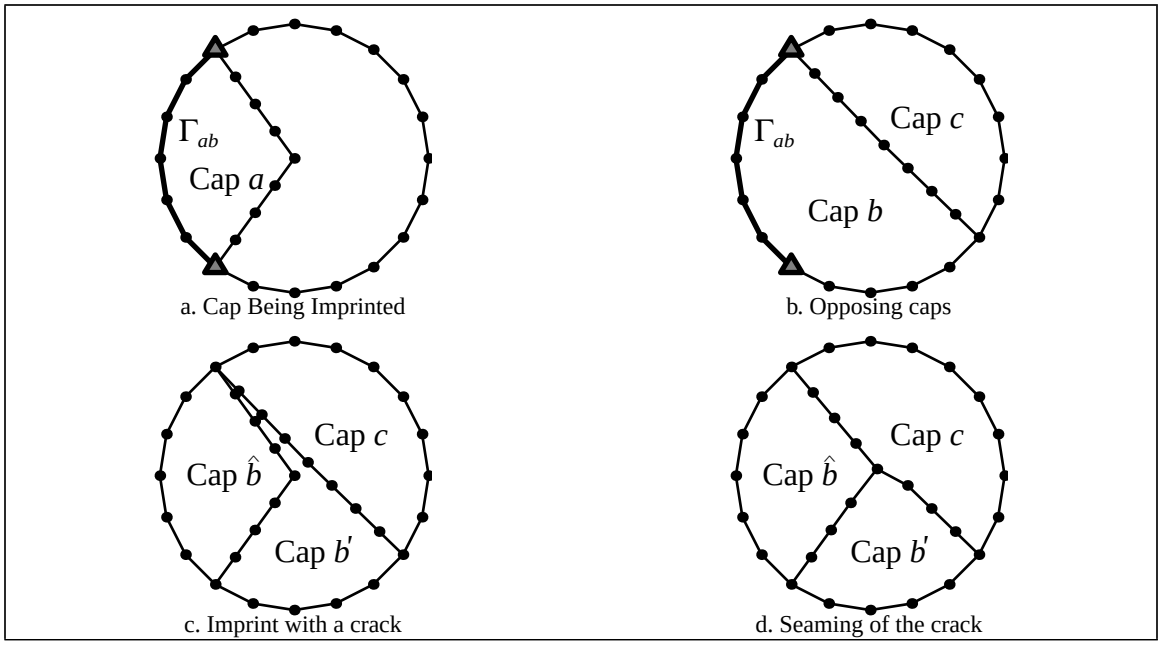
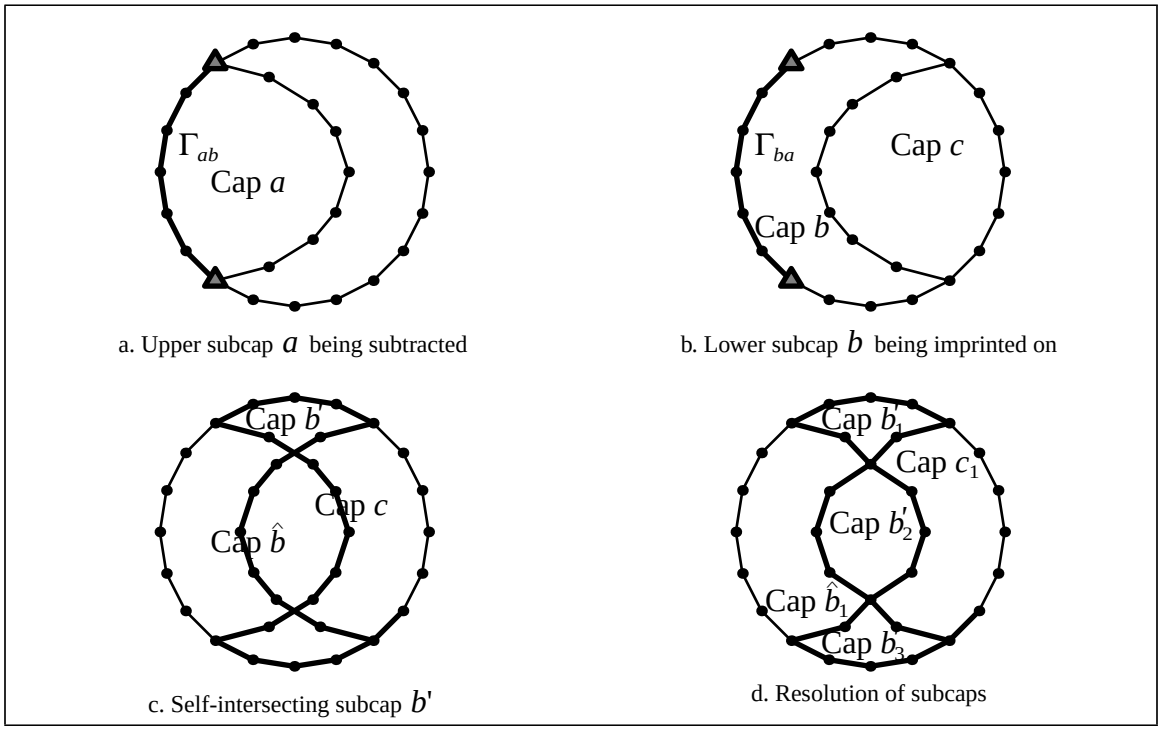


Figure 17. Seaming of projected subcaps

5.5 Self-Intersecting Subcaps

A subcap subtraction (Section 5.2 on page 8) may produce a new cap b' which is self-intersecting. That is, the two boundary portions that were combined to form cap b', $\hat{\Gamma_{ba}}$ and $(\Gamma_b^{\ i} - \Gamma_{ba})$ may overlap or intersect each other as shown in Figure 18. Because these two boundary portions form the bounding loop of a single cap, the cap is said to be self-intersecting. Since the boundary of the cap is continuous, the number of intersections, n, must be even. The resolution is simply the formation of $n + 1$ new caps by connecting the overlapping portions of the boundary. As with subcap intersection, care must be taken if an all-hexahedral mesh is desired to insure that the crossings produce loops with an even number of nodes. After the formation of the $n + 1$ new caps (caps b'_1 b'_2 and b'_3 in Figure 18d) any completely interior caps (b'_2)

must be subtracted from adjacent caps (caps $\hat{b}$ and cap c in Figure 18c). These adjusted caps $\hat{b}_1$ and c_1 are also shown in Figure 18d.



a. Upper subcap a being subtracted

b. Lower subcap b being imprinted on

c. Self-intersecting subcap b'

d. Resolution of subcaps

Figure 18. Self-intersecting subcap example and its resolution by connecting the overlapping boundary portions into an odd number of new caps.

6.0 Meshing Barrels

Barrel meshing is performed by first forming barrel cap stacks as will be described below. These are then meshed to produce the complete surface meshes of all the barrels. Interior nodes are then generated using the original barrels to complete all the interior ribs. These ribs and the barrel cap stacks are then combined to produce the volume mesh.

6.1 Forming Barrel Cap Stacks

Recall that upon completion of the barrel subcap imprinting, each subcap a will have a matching subcap $b = {}^{B}\Lambda_a$ for each of the barrels B which a is attached to. Also, all the boundary nodes of the subcaps will be attached to corresponding nodes of its matching subcaps through ribs (either external or internal ribs of the barrel). Barrel stacks can now be formed which simply collect a stack of all matching caps through all attached barrels.

To generate a cap stack, a cap a (not yet in a stack) is chosen. The attached barrel B is used to find the matching cap $a' = {}^{B}\Lambda_a$. This cap is then added to the stack. The next barrel is found and the process continues in both directions until all matched caps are in the stack.

Each stack can then be meshed by simply meshing either the first or the last cap (or using an existing mesh from either if it has already been meshed). The paving algorithm is currently used for this meshing process. This mesh is then projected to each of the caps in the stack using a least square weighted residual method (beyond the scope of this paper) which maintains the overall shape and quality of the mesh even for drastic

distortions in geometry. Each node on the mesh is connected to the next cap with an interior rib, identical to the process used during subcap imprinting.

As is obvious from Section 5.0, subcap imprinting will have sectioned existing faces into subcaps which are meshed independently. A smooth of the independent subcaps forming an exterior face of the volume is performed to optimize element quality. Since this smooth does not affect the topology of the mesh, the structure of the ribs and barrel caps remains unaffected.

6.2 Generating Interior Barrel Nodes

Since each barrel now has a complete exterior mesh, this can be used to generate all the interior nodes of the mesh. This is done using a weighting of the same least square residual method mentioned in Section 6.0. Let $\boldsymbol{x}_{ij}^{U}$ be the least square residual projection of node i from the upper cap to layer j. Similarly, let $\boldsymbol{x}_{ij}^{L}$ be the projection of node i from the lower cap to layer j. Then the location of node i on layer j is defined as

$$\boldsymbol{x}_{ij} = \boldsymbol{x}_{ij}^{U}\left(1 - \frac{j}{j_{max}}\right) + \boldsymbol{x}_{ij}^{L}\left(\frac{j}{j_{max}}\right)$$

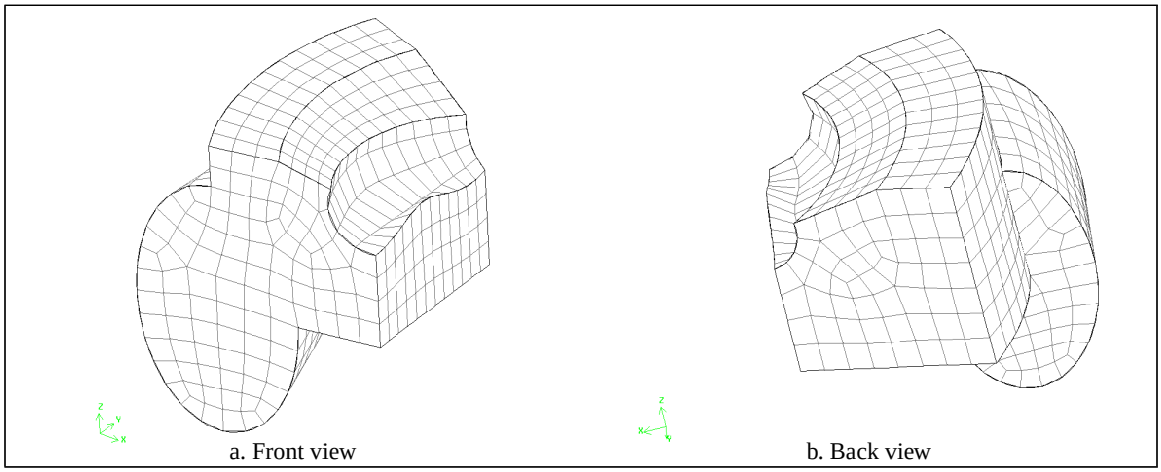
The nodes are maintained in order on the interior ribs as they are generated from the upper to the lower cap for each barrel.

6.3 Volume Element Generation

With all interior nodes generated, the formation of volume elements is a trivial task best performed using the barrel cap stacks. The surface mesh connectivity of the end cap of the stack forms the basis for the element generation. The nodes for each of the mesh faces (3 for a triangular face, 4 for a quadrilateral face) are used to collect the attached ribs. These ribs are then progressed, layer by layer, to form the volume element connectivity. No smoothing is required, as the projections to form the ribs have already optimized the node locations.

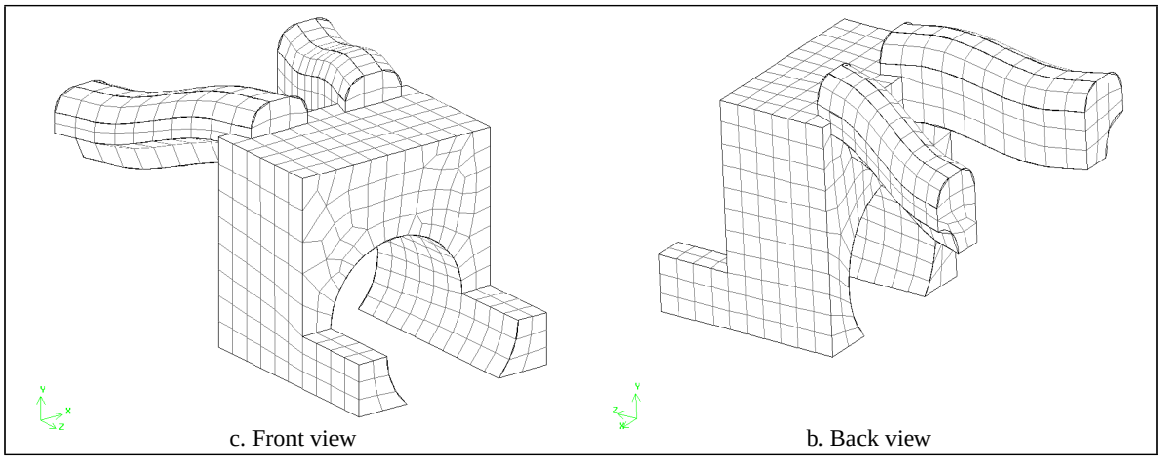
7.0 Examples and Evaluation

Figure 1 and Figure 19 through Figure 21 show examples of cooper tool generated meshes. These examples



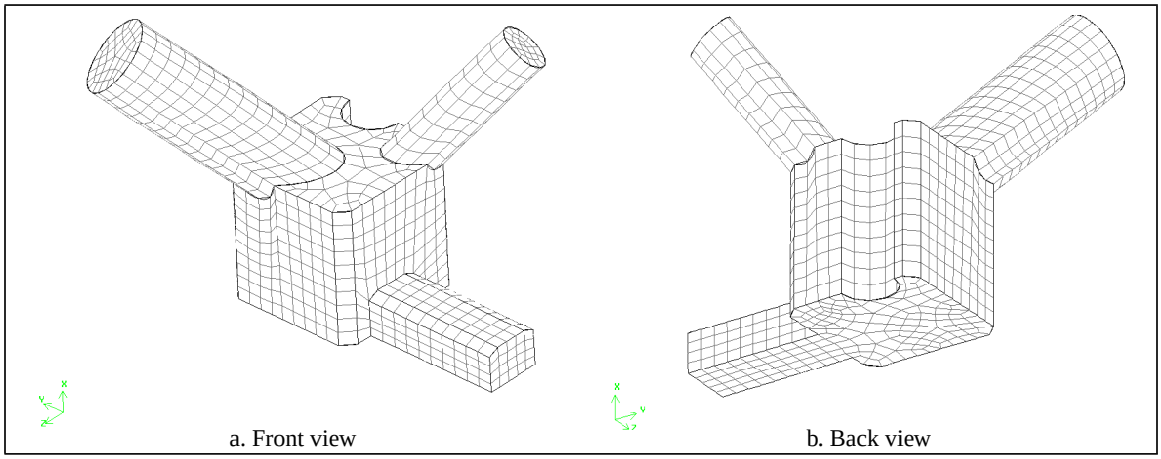
a. Front view b. Back view

Figure 19. Example 1 showing varying cross sections and simple topology.



c. Front view b. Back view

Figure 20. Example 2 showing a geometry with several appendages and nurb surfaces.



a. Front view b. Back view

Figure 21. Example 3 showing a geometry with large deviations in projections.

include only geometries that can be meshed with the cooper tool without decomposition. For all the examples except Figure 21 the only user input is an element size. Figure 21 required limited input to set side submapping constraints.

Several meshing densities for each of the examples are reported in Table 1, but only the coarsest mesh is shown in the figures. The run times of the problems shown in this paper at the varying densities have been calculated on an HP9000 machine and are shown in Table 1. The method is linear with the number of elements, and roughly constant at over 100 elements per second. The slower times were mainly attributable to longer surface meshing times for nurb side faces.

TABLE 1. Cooper Tool Performance

Figure	Density	Elements	Elements/Second
1/6	0.35	2,623	133.7
1/6	0.20	14,123	126.5
8	1.00	1,322	142.9
8	0.70	6,328	138.3
19	0.70	948	81.0
19	0.30	12,244	101.84
20	1.00	789	48.3
20	0.70	2,383	48.9
21	1.00	2,137	121.0
21	0.60	8,652	137.4

8.0 Conclusions

A new cooper tool has been presented which automates hexahedral meshing of a large class of geometries. The method decomposes a volume into a unique set of logical barrels which are then meshed using a single axis projection. The method is fast and robust. Element quality is assured through the generation of consistent surface meshes for projection. The cooper tool meshes are boundary sensitive (best mesh at the boundary) and orientation insensitive (same mesh for transformed geometries).

9.0 References

[1] Blacker, T. D., Stephenson, M. B. (1991) “Paving: A new approach to automated quadrilateral mesh generation”, Int. J. Numerical Methods, 32:811-847.

[2] Blacker, T. D., Meyers, R. J. (1993) “Seams and Wedges in Plastering: A 3-D Hexahedral Mesh Generation Algorithm”, Eng. w/Computers, 9:83-93.

[3] Blacker, T. D., Mitchell, S. A., Tautges, T. J., Murdoch P., Benzley, S. (1996) “Forming and Resolving Wedges in the Spatial Twist Continuum”, to appear in Eng. w/Computers.

[4] Holmes, D. I., (1995)“Generalized Method of Decomposing Solid Geometry into Hexahedron Finite Element”, proceedings 4th Int. Meshing Roundtable, Sandia National Laboratories.

[5] Price, M. A., Armstrong, C. G., and Sabin, M. A. (1995) “Hexahedral Mesh Generation by Medial Axis Subdivision: I. Solids with convex edges”, Int. J. Numerical Methods, 38.

[6] Schneiders, R., (1995) “Automatic Generation of Hexahedral Finite Element Meshes”, proceedings 4th Int. Meshing Roundtable, Sandia National Laboratories.

[7] Taghavi, R. (1994) Automatic, parallel and fault tolerant mesh generation from CAD on Cray research supercomputers”, proceedings CUG Conference, Tours, France.

[8] Tautges, T. J., Blacker, T. D., Mitchell, S. A. (1996) “The Whisker Weaving Algorithm. A Connectivity-Based Method for Constructing All-Hexahedral Finite Element Meshes”, to appear in Int. J. Numerical Methods.

[9] White, D. R., Mingwu, L., Benzley, S. E., Sjaardema, G. (1996) “Automated Hexahedral Mesh Generation by Virtual Decomposition”, proceedings 4th Int. Meshing Roundtable, Sandia National Laboratories.

[10] Whitely, M., White, D. R., Benzley, S., Blacker, T. D., (1996) “Two and Three-Quarter Dimensional Meshing Facilitators,” to appear in Eng. with Computers.