HEXAHEDRAL MESH GENERATION USING MULTI-AXIS COOPER ALGORITHM

Katsuhiro Miyoshi* and Ted BlackerÜ

* Corporate Information Technology Department
Toyota Motor Corporation
1 Toyota-chou, Toyota-city, Aichi Prefecture, Japan
Email: kats@mail.toyota.co.jp

Ü Gambit Project Manager
Fluent Inc.
500 Davis Street, Suite 600, Evanston, IL 60201
Email: ted@fluent.com, Web page: http://www.fluent.com

Key words: Hexahedral mesh generation, Cooper tool, Sweep

Abstract. The multi-axis cooper tool is a new tool to create a hex mesh by using multi-axis imprinted sweeps. The tool automatically recognizes applicable geometries and divides them into a hierarchy subvolume, which are then meshed by existing single-axis sweep tools. The resulting meshed volumes, called inlay volumes, contain mesh which is non-conformal or discontinuous at their interfaces. The nonconformal sections of the mesh are then removed and replaced with a conformal mesh using the cooper tool. In this paper, we systematically describe the new algorithm, its applicability and strengths, and discuss its future potential.
1 INTRODUCTION

Numerical analyses continue to play a very important role in technology intensive engineering efforts, such as the process of designing new automobiles. Fortunately, quadrilateral and triangular surface and tetrahedral volume mesh generation techniques have improved very dramatically in the last decade. They have proven effective for very complicated geometries, such as those typical of automotive parts. On the other hand, a lot of hexahedral meshing algorithms have been introduced, but these algorithms have not been nearly as successful. In this paper, we will introduce the multi-axis cooper tool, which automatically discretizes geometries into a hex mesh by using multiple-axis sweeps. The generated mesh has the following positive attributes:

1) Full automatic meshing for applicable geometry
2) High element quality
3) Boundary sensitivity (best elements along the boundary)
4) Orientation insensitivity (same mesh for transformed geometries)

Numerous approaches have been proposed and investigated ([1] – [16]) 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 can determine a-priori the tool most suited to the geometry at hand. The multi-axis cooper tool, described below, will be provided as one of the toolbox assortment available in the Gambit code, a commercially available pre-processing package.

Figure.1 is the example of the multi-axis cooper tool.
2 AUTOMATICALLY DETERMINING APPLICABLE GEOMETRY

When any number of tools are available to the user, the task of choosing the most appropriate one becomes significant. The tool should be able to assist with this task by identifying those geometries that are within its domain. The multi-axis cooper tool accomplishes this by classifying face types and looking at potential combinations.

There are three types of faces that the multi-axis cooper tool classifies: a side face, an inlay face, and a cap face.

A side face is characterized by the type of mesh that it will permit. In particular, a side face admits a submap mesh. This means that the face can be meshed with a structured mesh or that the face is easily decomposable into pieces that can be meshed with a structured mesh.

An inlay face is similar to a side face except for the presence of a hole. This interior loop (hole) is connected to a side face or another inlay face. With the interior loop included, the
inlay face cannot be submapped. However, if the interior loop is removed the inlay face would admit a submap mesh.

A cap face is any face that is not a side face or a submap face.

When classifying a volume, the multi-axis cooper tool first classifies the faces. It then combines any cap faces that share edges into a single cap. After this process is complete, the volume will qualify for the multi-axis tool if it has at least one side face, one inlay face, and three cap faces. It also must be divisible into an inlay volume hierarchy as described in section 3.

3 ESTABLISH INLAY VOLUME HIERARCHY

In order to mesh a volume using the multi-axis cooper tool, we must identify and then decompose it into a hierarchy of inlay volumes. An inlay volume is a volume that is meshable with the standard single-axis cooper tool. For example, in Figure 2, IV0 - IV6 represent cylindrical inlay volumes. For ease of explanation this example is limited to cylindrical inlay volumes. In reality, they can be much more complex.

Figure 2b shows the hierarchy structure of the sample geometry in Figure 2a. In this case, IV0 is the parent inlay volume of IV1 and IV3. Conversely IV1 and IV3 are the children of inlay volume IV0. Similar parent-child relationships extend up the tree. A child is always one level higher than the parent. Currently we support a single parent for each inlay volume. Extending this, to allow multiple parents will be discussed in section 9.

This hierarchical level is used to determine meshing order with inlay volumes at the lower level meshed first.

a. Sample geometry  b. The inlay volume hierarchy

Figure 2 The sample geometry and its inlay volume hierarchy. It has a four level hierarchy
4 CONTROL STRUCTURE

In this section, we will explain the overall flowchart of the multi-axis cooper tool and present a detailed explanation for each step in subsequent sections. Steps (1) and (2) are explained in previous sections.

(1) Classification of applicable geometries and sorting of faces into inlays, sides and caps.
(2) Create the hierarchy of inlay volumes.
(3) Set the caps for each inlay volume
For each inlay volume (from lower hierarchy to higher)
{
   (4a) Surface mesh on the side faces
       If inlay volume has child inlay volume
       {
           (5a) Establish child footprints
           (5b) Identify inlay node set on the cap
           (5c) Process inlay node sets on the cap
           (5d) Divide caps into subcaps and mesh
       }
   (4b) Mesh inlay volume using standard cooper tool
       If inlay volume has parent inlay volume
       {
           (6a) Core the parent’s hex elements below the footprint of the inlay volume
           (6b) Mesh the footprint cap
           (6c) Mesh the footprint volume using standard cooper tool
       }
}

By following this sequence of operations, the complete conformal hex mesh is generated.

5 INLAY VOLUME MESHING PROCESS

In this section, the meshing of an individual inlay volume is described. It includes steps (4a) - (4b) as introduced in control structure outline. If the inlay volume does not have any children or it’s at the top of hierarchy, steps (5a) - (5d) are skipped. Steps (5a) - (5d) are basically preparations for accommodating the children mesh yet to be generated.

5.1 Surface meshing of side faces

The first step in meshing an inlay volume is to create a submap mesh on all the side faces.
For those side faces connected to a child inlay volume (i.e. inlay faces), a submap mesh is created by ignoring the loops connected to the children. This inlay mesh (see Figure 3a) is temporary. It will later be adjusted to accommodate the child. The adjustment will occur during the child's coring process.

5.2. Preparation for the child footprints

We are meshing these volumes starting with the lower level parent and working up the hierarchy. In order for this to work, the parent mesh must be controlled such that the child mesh may be later connected. This control is accomplished by establishing the child footprints. From the footprints we identify inlay node sets on the cap. These sets are then processed and used to split the cap as needed.

5.2.1 Establish child footprints

To prepare for connection to the child inlay volumes, we mark the footprint of the child on the inlay side mesh. We currently restrict the footprint to be a rectangular hole in the submap mesh. In Figure 3a, a parent, IV0, and a child, IV1, is shown. The footprint must include the edge loop which connects the parent IV0 and the child IV1. To minimize interference with other features, the footprint should be as small as possible. This footprint will be used to modify the nonconformal meshes during the coring process of the parent by the child (section 6).

To determine the footprint, we overlay the common loop onto the inlay mesh (see Figure 3b). We then determine nodes which lie on the interior of the loop (nodes marked in Figure 3b) and then remove all elements attached to these nodes. These elements are shown shaded in Figure 3b. The footprint can then be expanded to form a logical rectangle.

Each child inlay volume must have its own footprint. The parent in Figure 3a has only one child inlay volume, so IV0 must have one child footprint. If there is more than one footprint, there is a possibility that the footprints will interfere, or overlap with each other. Such would occur if child inlay volumes were relatively close. Currently we do not support interfering footprints, but hope to handle these as explained in the future work section 8.1.

Once the footprint has been established for the child inlay volume, we use the total number of nodes around the perimeter of the footprint to set the number of nodes on the connecting loop. This allows us to use the inlay meshing technique described in Section 6.2.
5.2.2 Identifying inlay node sets on the cap

At this stage of the process, we have a side mesh and footprint as explained in the previous section. Since the side face terminates at the cap face, the edges of the cap face will be meshed. We define the inlay nodes as the "pseudo" projection of the footprint boundary on the cap edge.

This projection is accomplished by traversing the side face mesh. Since the side face mesh is submapped, (i.e. it is a structured mesh), it allows the definition of a rib. A rib [1] is the regularly connected line of nodes running logically perpendicular to the cap edge.

Let "projection ribs" be the ribs which connect to a footprint boundary as shown in Figure.4a. A controlling footprint side is found (one of four possible sides), which is logically closest to the cap and logically perpendicular to the ribs. The projection ribs uniquely define a set of inlay nodes on the cap edges which are connected to the controlling footprint boundary side.

For each inlay child volume's footprint, $F_a$ (see Figure.4b), a corresponding set of inlay nodes must be determined, $I_a$. Note that this inlay set may overlap other inlay sets, even though the footprints, themselves, do not overlap. Each inlay set is assigned a minimum and maximum row height associated with the footprint position. The lower height, $L_a$, is the number of rows between the cap and the closest node on the footprint. Likewise, the upper height, $U_a$, is the number of rows between the cap and the furthest node on the footprint. In Figure.4b, $L_a = 2$ and $U_a = 5$, $L_b = 1$ and $U_b = 3$. 

Figure.3  Mesh the inlay face and determine the footprint.
5.2.3 Processing inlay node sets on the cap

In order to mesh a cap appropriately, each set of inlay nodes must have a valid opposing match. The processing of inlay sets establishes these valid sets and matches. New inlay sets may be created as needed to establish the matches, as explained below.

Since a cap face is essentially a surface, we can view it as a logically 2D circular diagram. The cap nodes, are the set of nodes which form the perimeter of the cap. We constrain all caps to have an even number of nodes to permit an all-quad surface mesh. For simplicity, we will begin with the case of two sets of inlay nodes, \( I_1 \) and \( I_2 \). For any two inlay sets, there are 3 possible relationships. The inlay node sets, \( I_1 \) and \( I_2 \) may: 1) overlap or share nodes, 2) be adjacent with no nodes between the inlay sets, or 3) be independent with one or more nodes between the inlay sets.

Figure 5a shows a case where \( I_1 \) and \( I_2 \) overlap. When inlay nodes overlap, the sets must be merged into a single set of inlay nodes, \( I_{12} \). Let \( n_{i12} \) be the number of the inlay nodes in \( I_{12} \). If \( I_1 \) and \( I_2 \) are not overlapping but are adjacent, these sets must also be merged. If the total number of cap nodes is \( n_c \), then in order to mesh the cap, the following relationship must hold:

\[
2 \times n_{i12} + 2 \leq n_c \tag{1}
\]

If this relationship (1) is met, then matching can occur as explained below. If the relationship (1) does not hold, strategies to overcome this problem are given in the future work section 8.2.
I1 and I2 overlap     Merged into I12     Matching I12'

Figure 5 Processing the inlay nodes, I1 and I2 (overlap)

Even when I1 and I2 are independent, it is sometimes advantageous to merge them. Let \( n_b \) be the number of nodes between I1 and I2. Then, if

\[
2 \times (n_{I_1} + n_{I_2} + n_b) + 2 \leq n_c
\]  

these two sets may be merged. This will allow sufficient intervals for the path between the matches (see Section 5.2.4) to be created. Currently merging must continue until there is either a single inlay set, or two inlay sets whose footprint heights do not overlap. This then reduces the problem to a manageable set of alternatives. In the future work section we describe how cases which do not meet this criteria could be handled.

As mentioned before, all inlay sets, I12, must have an appropriate match, I12'. When merging has been completed, these matches can be established. To generate the optimal mesh, it is advantageous to separate the matching sets as far away from each other on the perimeter as possible.

If following the merging, there is only a single set left, I12, then this set's match, I12', must be created. The set I12' is positioned to balance the number of nodes remaining in the cap, as shown in Figure 5c. This then dictates that there will be \( n_s \) nodes between the two caps, where \( n_s \) is defined by:

\[
n_s = (n_c - (2 \times I_a)) / 2
\]  

If there are two sets left, I1 and I2, then these two must be matched against each other (see
If the number of nodes are equal, i.e. \( n_1 = n_2 \), then the sets can be matched without adjusting them. If the \( n_1 > n_2 \), then a super set of \( I_2 \) is created as \( I_1' \), by adding additional nodes to \( I_2 \) until \( n_1 = n_1' \). Again, to optimize mesh shape, these additional nodes are added to the side of \( I_2 \) which is furthest from \( I_1 \) as shown in Figure.6b. Again matching is restricted to those sets whose related footprint heights, \( L_a \) and \( U_a \), do not overlap.

![Figure 6a: Matching Pattern](image)

- The case of \( I_1 > I_2 \)
- \( I_1' \) is created by adding nodes to \( I_2 \)

### 5.2.4 Divide caps into subcaps and meshing

Once the matching sets are established, then connections between the sets are made to establish a path between the matches, \( I_a \) and \( I_a' \). This path is created by connecting both end nodes of the set \( I_a \) to the closest corresponding end node of the matching set \( I_a' \) as shown in Figure.7a. The number of intervals along the path is calculated by using the maximum length of the connecting path edges, and the relevant desired element size. This connection divides the cap into three subcaps.

Each subcap must be meshed correctly. The subsequent coring operation will require a regular projection mesh. In preparation for this, the path, or middle subcap, must be meshed with the mapping algorithm. This subcap is marked to force such meshing. Actual meshing occurs during the generation of the volume mesh using the cooper tool. The only constraints on the other two caps are that they be meshed with an all-quad mesh. We currently employ the paving algorithm [17] during the coopering process when meshing these subcaps.
5.2.5 Meshing the inlay volume

Once the cap has been divided to create a path between matching inlay sets, the inlay volume can be meshed using the standard cooper tool. This tool will imprint the path (any other subcap divisions) onto opposing ends of the volume. An all-hex mesh can then be generated in the inlay volume. The cooper tool [1] is a single axis projection tool, and thus requires that the footprint area be a structured mesh. The structured mesh of the path will insure that this projected mesh can later be cored to provide a conformal mesh.

6 CORING PROCESS

After a child inlay volume has been meshed, its mesh does not conform to the mesh of the parent. This is because the footprint area has been meshed with a structured mapped mesh, instead of accommodating the pattern of the child imprint on the parent. The process of coring rectifies this problem. It replaces elements in the parent with a conformal mesh that then connects parent and child.

6.1 Removing the hex elements beneath the footprint

The first step in the coring process is to remove the nonconformal elements in the parent inlay volume. These elements are the elements directly "below" the footprint of the child on the parent side face mesh. Since this footprint was used when the parent was meshed, a
structured mesh will exist below the footprint. This structured mesh is removed, leaving structured side meshes along the boundaries of the hole (see Figure.8a).

### 6.2 Mesh the footprint cap

This cored section of the parent inlay volume can be viewed as a volume itself, which we call the footprint volume. The sides of the footprint volume are defined by the structured mesh along the hole where the elements were removed as shown in Figure.8a. The top and bottom of the volume must now be established.

As a top for this footprint volume, the side face footprint and the common edge with the child are used. As explained above, a footprint is always a rectangular structured mesh. The boundary of this footprint now becomes an exterior loop of the top subcap being formed. The actual connecting boundary edge between the parent and child becomes an interior loop, or hole, on the top cap. To complete the meshing of this top cap, a mesh must be created between the exterior and interior loops of this face.

We mesh the remaining portion of the top cap using an inlay technique [18] as shown in Figure.8b. This technique connects the inner loop of nodes to the outer loop by using a single row on elements. Since we control the number of intervals on the common loop (section 5.2), we are assured that there will be an equal number of intervals on each loop. Once connected, we perform a smooth of the surface mesh to improve quality.

### 6.3 Mesh the footprint volume

With the top cap meshed, the footprint volume can now be meshed using the standard cooper tool techniques. This will produce an all-hex mesh which is conformal with the parent and connects directly to the child. The final hex mesh is shown in Figure.8c.
Figure 8  a. Removing the hex elements beneath the footprint
   b. Footprint meshing
   c. Final hex mesh

7 EXAMPLES

Figure 9  This volume has 22 faces. (35786 nodes, 32277 hex elements)
Figure 1 and Figure 9 through Figure 11 show examples of multi-axis cooper tool generated mesh. The volume in Figure 9 has five inlay volumes and three hierarchical level.

Figure 10 This volume has 24 faces. (13119 nodes and 11431 hex elements)
8 FUTURE WORK

This work is a type of automated decomposition technique. As such the set of volumes that can be meshed is limited to those which fall within its domain. The challenge with any decomposition approach is to identify a tool which is expandable; allowing it to be applicable to an increasingly wider range of geometries. Although it may never become completely general, if the domain of applicable geometries is large enough it can be quite effective for the user. Future work is thus centered on expanding this tool and by so doing expanding its domain.

8.1 Allowing overlapping footprints

One of the simplest enhancements will be to allow overlapping footprints. Figure.12a shows two child inlay volumes, whose footprints overlap. These two footprints could be combined into a larger rectangular footprint which contains both interior loops of the children. The mesh on the footprint cap could no longer be a simple inlay type, but the paver would be employed to generate the mesh (see Figure.12b). This also relaxes the need to preset the number of intervals on the common edges.
Figure 12 Overlapping footprints should be combined to a larger new footprint.

8.2 Adding rows for easier inlay set matching

When matching inlay node sets, it is possible that condition (1) will not be met. This simply indicates that there are not enough nodes to allow the match (see Figure 13a). The solution would be to add additional nodes to the cap boundary (or columns of elements in the side face mesh (see Figure 13b)). With this additional flexibility, the number of applicable geometries are greatly increased. It will also allow us to create a higher quality surface mesh on the top cap and consequently a better volume mesh.
8.3 Matching inlay sets with overlapping heights

Currently, the matching of inlay node sets requires that there is no overlap in the respective footprint heights. This height is based strictly on the connectivity of the structured side face mesh (see Figure.14a).

As shown in Figure.14b, one way of avoiding the overlapping heights is to add and/or reroute rows in the side face mesh. This then uncouples the inlay sets and allows for more general matching. Producing more matches and fewer merges usually increases the top cap mesh quality and consequently the generated volume mesh.

![Figure.14](image.png)

a. Overlapping the height between $F_a$ and $F_b$  
b. Add the rows to avoid the height overlapping

Figure.14 The methods to avoid the height overlapping of footprints.

8.4 Dividing top cap with crossing paths

Another method of improving the top cap mesh quality would be to allow crossing paths as shown in Figure.15. This would be particularly useful when the inlay node sets are not directly opposite each other on the cap. If there are few nodes between the sets on one side, and a large number on the other, this disparity causes merges and/or matches that suffer in quality and number of elements. By allowing crossing paths, these problems are eliminated. However, in order for crossing paths to be allowed, the inlay node sets cannot have height overlaps. When such occur, ideas in section 8.3 can be employed.
a. Ideal positions of I₁ and I₂ for crossing  b. Creating the crossing paths

Figure.15  Creating the crossing paths and divide the cap into nine subcaps.

9 CONCLUSIONS

A new multi-axis cooper tool has been presented which automates hex meshing of a large class of geometries. This technique automates meshing using a decomposition technique which in some sense mimicks the more manual approaches currently in use for complicated geometry. The advantage of this approach is that element quality can be maintained while automation significantly reduces the time required to produce a mesh. This approach is easily expanded, which insures that the applicable domain of meshable geometries can progressively increase.

REFERENCES


