SAND2014-15041C

LayTracks3D: A New Approach to Meshing General Solids using Medial Axis Transform

William Roshan Quadros

[1]Sandia National Laboratories, Albuquerque, USA, wrquadr@sandia.gov

Abstract. This paper presents an extension of the all-quad meshing algorithm called LayTracks to generate high quality hex and hex-dominant meshes of 3D assembly models. LayTracks3D uses the mapping between the *Medial Axis* (MA) and the boundary of the 3D domain to decompose complex 3D domains into simpler domains called *Tracks*. Tracks in 3D are similar to tunnels with no branches and are symmetric, non-intersecting, orthogonal to the boundary, and the shortest path from the MA to the boundary. These properties of tracks result in desired meshes with near cube shape elements at the boundary, structured mesh along the boundary normal with any irregular nodes restricted to the MA, and sharp boundary feature preservation. The algorithm has been tested on a few industrial CAD models and work is underway to achieve all-hex meshes on general solids.

1 Introduction

Many computational simulations such as non-linear solid mechanics require all-hex meshes. Currently, there is no ideal automatic hex meshing algorithm to mesh general solids or assemblies with commonly desired features in a hex mesh as this is a very challenging problem. In most cases, users have to resort to using meshes of suboptimal quality or spend a significant amount of time generating them. In complex cases, the creation of a desirable hexahedral mesh may take months even for an expert user. This hex mesh generation process tends to dominate the overall cost of numerical simulations. Therefore, improvements in the hex meshing technology are of significant importance to the computational simulation community.

The goal of LayTracks3D is to generate hex meshes of solids and assembly models with desirable features such as boundary sensitivity, orientation insensitivity, sharp feature preservation, high-quality mesh, and the handling of general solids. The mesh generator should have the ability to generate a variety of meshes by controlling sizing and anisotropy, generate geometry adaptive meshes, provide fast remeshing during FEM iterations, and should be scalable.

This paper is an extension of the all-quad meshing algorithm proposed by the author called LayTracks [1] for 3D solids. Section 2 gives an overview of the original 2D LayTracks algorithm. Section 3 reviews the literature on method of decomposition and advancing front methods as LayTracks3D combines the merits of these two methods. Section 4 gives an overview of LayTracks3D and the rest of the paper discusses extensions of LayTracks3D for assembly meshing and all-hex meshing. The results section shows hex-dominant meshes on few industrial models as the all-hex meshing is not fully implemented at this time.

[1]Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000

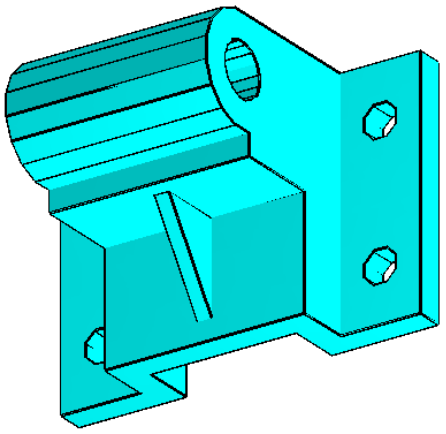
2 Overview of LayTracks

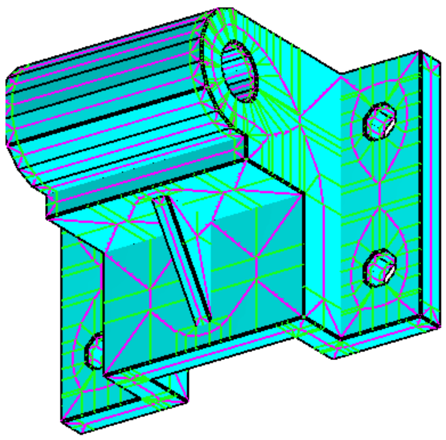
As it is easier to understand the algorithm in 2D, here we quickly recap the original 2D LayTracks [1]. LayTracks works analogous to the formation of railway tracks by laying rails on the ground to form a set of non-intersecting connected tracks on surfaces (see Figure 1(e)); hence, the name LayTracks. This algorithm uses a skeletal representation of the input domain called Medial Axis Transform (MAT) [2,3,4], which is a mathematically well-studied skeletal representation. LayTracks is built on the mathematically sound MA and guarantees many desirable properties such as orthogonality of mesh elements at the boundary, irregular nodes restricted at farthest distance from the boundary, automatic conformal mesh at interface of surfaces, and all-quad mesh.

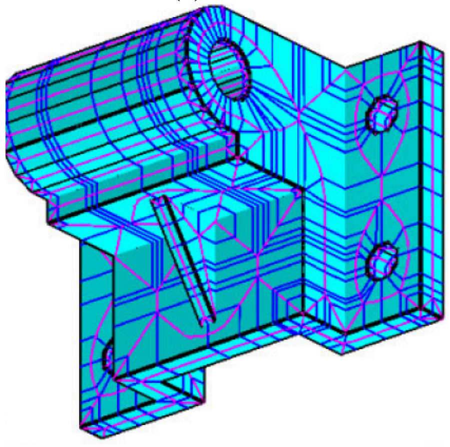
Figure 1 (a) shows an assembly of surfaces. Figure 1 (b) shows the MA and the map from each MA segment to the corresponding boundary segment. The points on the MA where more than two segments meet are called *Branch Points*. These points represent the critical singularity points in the interior of the domain. Figure 1(c) shows the decomposition of the original domain, i.e., assembly of surfaces into a set of connected simpler domains using the branch points. Using the map, the branch point is connected to its corresponding tangent points on the boundary via line segments (see Figure 1(e) & Figure 5). These line segments propagate from one surface to the other either from the MA to the boundary or from the boundary to the MA using the map. The set of these connected line segments is called a *Rail*. Figure 1(e) shows the rails in blue, which are mathematically defined as a bi-partite graph [5]. Each line segment of a rail (i.e. edge of the bi-partite graph) has two end points: one on the MA and the other on the boundary. Note that rails branch at the MA branch points and they enter and exit the boundary orthogonally. The region between two adjacent rails' paths [5] is called a *Corridor* (see Figure 1(c)).

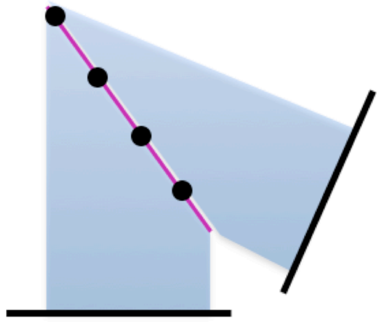
Note that the MA skeleton representation reduces the surface meshing problem into curve meshing. The next step is propagating the rails across the surfaces inside each corridor as the MA curve segments are meshed by inserting nodes based on the input mesh size. Uniform node spacing on the MA generates uniform mesh and a varying node spacing based on the radius of the medial ball generates a geometry-adaptive mesh [6]. Figure 1(d) shows the uniform node spacing on a MA segment of a corridor. Figure 1(e) shows all the rails generated using uniform node spacing on the MA segments of each corridor. The region between two adjacent rails' paths [5] inside each corridor is called a *Track (*see Figure 1(e)).

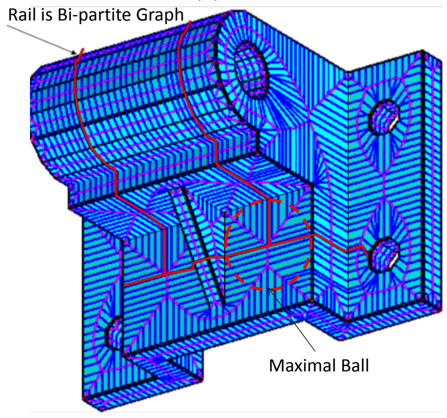
A set of connected tracks is a much simpler domain to mesh compared to the input surfaces. First the rails are meshed and then the quad elements are built inside each track. Note the rails are symmetric on either side of the MA as they are line segments connecting the center of the maximal ball to the tangent points as shown in Figure 5. Therefore, the total number of nodes on the two radii is always even. Thus, a track will be bounded by an even number of edges (i.e., even number of edges on the two rails and two boundary edges). This provides a theoretical guarantee for an all-quad mesh.


(a) Set of surfaces


(b) MA and map


(c) Corridors


(d) Mesh on corridor's MA



(e) Tracks inside the corridors

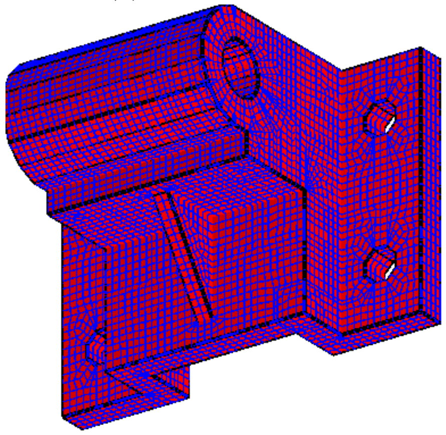

(f) Quad mesh on set of surfaces

Figure 1 Overview of quad meshing via LayTracks

3 Literature Review

Various hex meshing algorithms have been proposed in the literature; however there is no automatic all-hex meshing algorithm that gives all the desirable properties such as boundary sensitivity, orientation insensitivity, high quality mesh, sharp feature preservation and handling of general solids. Every algorithm has its own pros and cons. In the paragraphs below, method of decomposition and advancing front methods are discussed as they are relevant to LayTracks3D.

Method of decomposition works by decomposing a complex 3D domain into simpler meshable subdomains. One of the most practical approaches for generating all-hex meshes involves decomposing a complex solid into sweepable, mappable, or submappable subdomains, and then meshing these subdomains. This technique generally gives a high-quality mesh. The main disadvantage of this method is that it requires manual geometry decomposition, which is a very tedious task, and not trivial on complex models even for experts. This is a major bottleneck of this type of decomposition method.

Another decomposition-based method uses the MA. Here a quick review of the MA-based 3D meshing algorithms is presented. Price and Armstrong [7,8] described a subdivision yielding one subregion for each medial vertex, medial edge and medial face. The subregions are subsequently meshed by mid-point subdivision. This method can generate poor quality elements, which are not useful for simulation. Pete Smpl [9] presented a semi-structured meshing algorithm that generates mixed meshes with hex percentage ranging from 10.6% to 47%. It does not consider assembly models and therefore does not address respecting boundary imprints and obtaining conformal meshes. Makem et al. [10] used the MA for detecting thin and thick regions while generating a hybrid mesh, i.e., hex meshes are generated on the long slender regions and tetmesh on the rest of the domain.

Another decomposition-based method uses frame fields to design high quality hexahedral meshes. However, the automatic generation of frame fields that are useful for generating meshes of good quality is a complex problem. The generation of 3D frame fields is more complex than its 2D counterpart called cross fields; thus, preventing extension of most 2D methods to 3D. Ved et al. [11] made the first attempt to generate such a frame field using tensor metrics. The tensor field is first initialized at the boundary of the solid and then interpolated in the interior in an advancing front manner. This method generates a hex mesh at targeted regions and a hex-dominant mesh in the rest of the domain. The eigenvalues of the metric order the different directions of the tensor field. As a result, smoothing and interpolation operations treat the frame field as a set of 3 different direction fields, preventing the intertwining naturally occurring in the fields. Moreover, the regions surrounding umbilics, where several eigenvalues are starting to be identical, are highly unstable, and make the resulting tensor field unusable in these regions.

To overcome these problems, a method based on an energy formulation has recently emerged [12,13,14]. Energy formulation ensures that the order in which the directions of a frame are considered have no impact. Using the gradient of this energy, it is then possible to globally smooth the frame field. The initial field is computed using a given crossfield on the surface that is transformed into a surface frame field by adding the surface normal. The smoothing operations proposed are a massive step toward in the generation of frame fields. However, poor initial singularity locations cannot be improved through smoothing, as these only look for the closest local minima of the energy. The results are also very dependent on the input cross field and it is not clear if the singularity graph can always guarantee all-hex meshes.

Kowalski et al. [1516] proposed a method of generating frame fields, which does not depend upon the input surface crossfield. By computing streamlines of interest, a skeleton is obtained that allows the partitioning of the domain into multiple blocks that can be easily meshed through structured mesh generation methods. The biggest drawback of this method is dealing with domains for which conflicting patterns of streamlines arise. Also, computing the 3D frame field is very expensive.

Advancing front methods are popular because of their success in 2D. The extension of 2D advancing front all-quad meshing algorithm paving [17] to 3D plastering [18] has very limited success. Plastering starts with a pre-meshed boundary and places hex elements in an advancing-front manner, progressing toward the center of the domain. A heuristic set of procedures for determining the order of element formation is defined. Unconstrained paving and plastering [19,20,21] extend respectively the paving and the plastering algorithms by starting from a domain whose boundary is not pre-meshed. They use a background simplicial mesh to guide the placement of entire layers of cells at a time, reducing the frequency at which unmeshable voids appear. Plastering is automatic, produces high quality elements at the boundary, preserves sharp features, and handles general solids. However, the major drawback of this method is that it almost always contains interior voids that cannot be meshed through heuristics and generates poor quality elements at the interior.

LayTracks3D combines the merits of two popular mesh generation techniques, method of decomposition and advancing front methods. While the MAT has been used for domain decomposition before, this is the first attempt at using the MAT for the robust subdivision of a complex 3D domain into a well-defined simpler sub-domain called "Tracks". As the MAT exists where the advancing fronts collide, the fronts can be terminated without complex interference checks. Thus, LayTracks3D has the promise of generating high quality, boundary sensitive, orientation insensitive, sharp feature preserving hex meshes on general solids without any manual decomposition.

4 Overview of LayTracks3D

LayTracks3D works analogous to 2D LayTracks [1] in decomposing a general solid/assembly into tunnel like tracks in 3D. Note that in 2D we discussed meshing an assembly of surfaces, but here we discuss meshing a single solid instead of an assembly for clarity. Assembly meshing is discussed later in Section 6.

Figure 2 shows the overview of LayTracks3D. Step 1 is to generate a 3D medial surface (see Figure 3 (b)) and build data structures to hold a 2-way map from the MA to the B-Rep and the B-Rep to the MA. Establishing the map is the most critical step and subsequent steps depend heavily on the 2-way map.

In Step 2, the non-manifold MA junction curves, which represent critical singularities of the 3D shape are used to decompose the solid into corridors. First, the MA junction curves are meshed. Then, rails are propagated from mesh nodes to define critical partition surfaces that define simpler meshable sub-regions called corridors (see Figure 3 (c)).

In Step 3, the 3D meshing is reduced to 2D meshing on the MA (see Figure 3 (d)). Meshing all the surfaces of the MA inside each corridor will cover the entire 3D solid. LayTracks3D is not an inside out method, i.e., as an alternative, one can mesh the boundary surfaces of corridors instead of the MA. It is quite typical to have a 3-manifold medial curve at convex vertices. It is recommended to have a layer of tri elements along the 3-manifold MA edge in order to obtain a single hex element by combining six tets. This is discussed in detail in Section 7 as shown in Figure 11.

Step 4 involves subdividing the corridor into tracks, which look like tunnels with quad/tri cross section. First, rails are propagated at every node of the mesh on the MA. Second, tracks are automatically formed using the quad mesh topology on the MA.

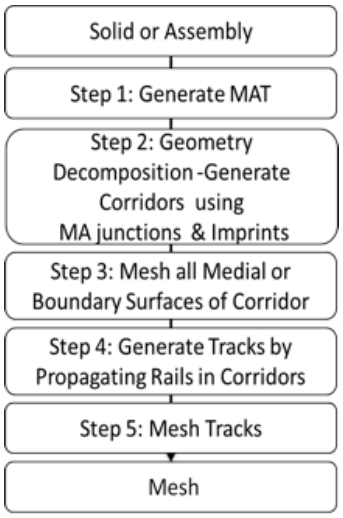

Solid or Assembly
Step 1: Generate MAT
Step 2: Geometry Decomposition -Generate Corridors using MA junctions & Imprints
Step 3: Mesh all Medial or Boundary Surfaces of Corridor
Step 4: Generate Tracks by Propagating Rails in Corridors
Step 5: Mesh Tracks
Mesh

Figure 2 LayTracks3D overview

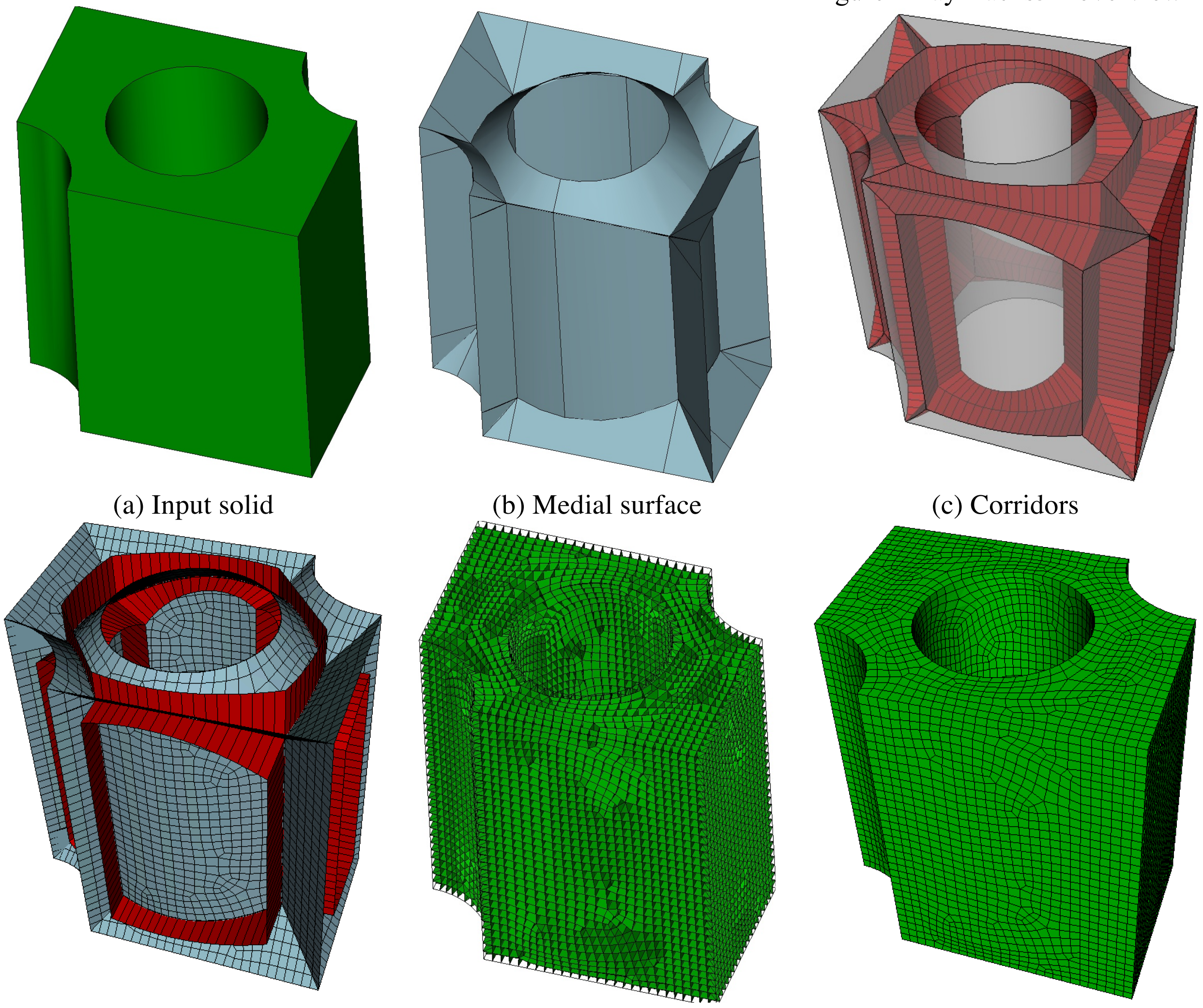


Figure 3 Overview of LayTracks3D

Unlike a rail, tracks do not branch. Tracks either form a closed tunnel or a tunnel with only one entry and only one exit.

In Step 5, tracks are meshed from the boundary towards the MA in an advancing front manner (without any interference checks) to achieve a boundary sensitive mesh. First rails are meshed using the defined mesh size. Next, hex elements are built using the nodes of the rails. At the MA, two wedges can be combined into a hex (e.g. at convex edge) or six tets can be combined into a hex (e.g. at convex vertex). Non-hex elements may arise at the MA if all-hex cannot be achieved inside each track. Section 7 gives more details on all of these cases.

5 Characteristics of LayTracks3D

As LayTracks3D is a new approach, some of the characteristics of the proposed method are highlighted below:

- **Handle General Solids**
  LayTracks3D can decompose any general solid into simpler tracks using a mathematically well-defined MA skeletal representation.
- **Boundary Sensitive**
  Rails/tracks cut through the boundary/interface orthogonally at tangent points giving a boundary sensitive structured mesh along the surface normal.
- **Orientation Insensitive**
  MAT is independent of the input model orientation and hence the mesh is orientation insensitive.
- **Dimension Reduction**
  MAT reduces hex meshing to quad meshing on the medial or the boundary surface of corridors.
- **Feature Preservation**
  All the sharp boundary features are preserved in the corridors, tracks, and the final mesh.
- **Sizing and Anisotropy Control**
  The size/anisotropy specified on the boundary surfaces can be mapped to medial surfaces, which controls the size/anisotropy of hex elements in two directions. Node spacing along rails controls the size/anisotropy in the third principal direction of a hex element. Note that tracks find the shortest path and hence limit the scope of specified size to a local region, which is quite hard to do with hex meshes.
- **Conformal Mesh**
  The 2-way map projects and resolves all the boundary imprints on the medial. Corridors then cut the interface of the assembly orthogonally and give an automatic conformal mesh respecting imprints. More explanation on assembly meshing is given in Section 6.
- **Geometry Adaptive**
  The radius function of the MAT and its gradients can be used to control element size, anisotropy, and orientation. Rails can be used as NURBS control points to generate non-linear tracks [2,5].
- **Fast Remeshing/Refinement**
  Recomputing the MAT or corridors is not required for remeshing/refinement as they depend only on geometry but not mesh size. Global or local remeshing can be performed by remeshing the medial surfaces of the corridors with a newer mesh size.
- **Mesh Morphing**

Old meshes can be morphed easily to new deformed geometry whenever the MA topology does not change.

- **Parallel Friendly**
  Decomposition-based methods are generally parallel friendly. Meshing rails and tracks can be easily parallelized.
- **Potential All-Hex**
  LayTracks3D is based on the strong mathematical foundation of the MAT. Section 7 discusses extension of LayTracks3D for all-hex meshing.

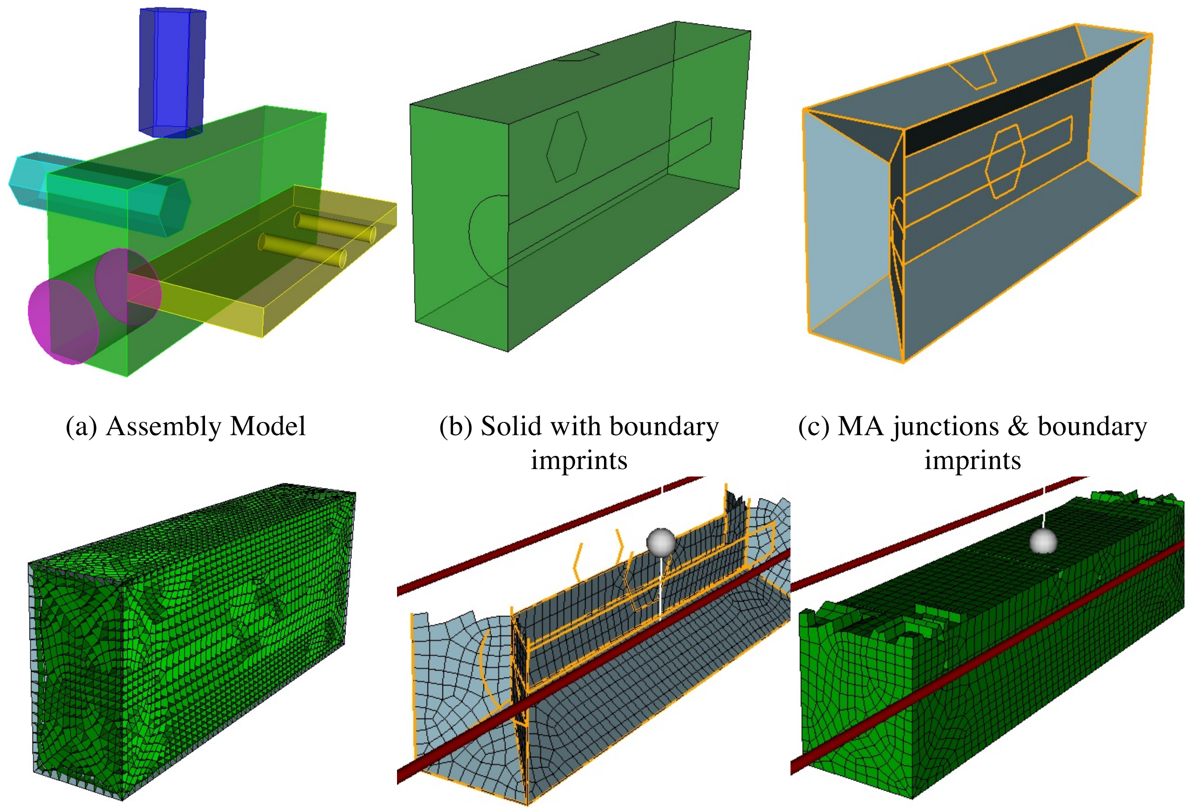
6 Extension to Assembly Meshing

Meshing an assembly model is more challenging than meshing a single solid. While meshing a single solid, the solid is decomposed into subdomains called corridors using only the non-manifold MA curves, which are critical singularities located in the interior of the solid. While meshing an assembly, even the imprinted boundary curves need to be taken into account in order to obtain a conformal mesh at the interface of the solids.

Figure 4 (b) shows a cuboidal solid of an assembly with imprinted curves on the top, left, right and front surfaces. In is not trivial to get a hex mesh of this simple cuboidal solid using existing hex meshing algorithms. Even though grid-based algorithms do a better job on solids aligned along Cartesian axes, they fail to respect the boundary imprints. Sweeping-based algorithms require manual decomposition and it is not trivial to decompose this simple cuboid into sweepable subregions. For example, sweeping the rectangular imprint on the right surface will intersect with sweeping the semicircle from the front, hexagon from the left and semi-hexagon from the top. Even with plastering, imprinted surfaces on the cuboid make it very difficult to close the inner voids.

One of the original contributions of LayTracks3D is that the MA is used to resolve all boundary imprints coming from all directions. Figure 4 shows the overview of meshing the cuboidal part of an assembly model via LayTracks3D. As the MA is a lower dimensional skeleton representation of the 3D solid, all the boundary imprints are mapped onto the MA (see Figure 4 (c)). The MA is then subdivided into different patches not only by non-manifold MA curves but also by all boundary imprints coming from all directions.

It is quite trivial to subdivide a 3D solid into corridors using the map after the MA has been subdivided into patches using imprints. Note that the corridors automatically cut the imprints orthogonally thus giving conformal mesh at the interface. Recall that the sweeping based manual decomposition propagates globally and thus intersects with other sweeps. Corridors are similar to sweepable regions but they do not propagate globally. Corridors automatically find the shortest path to enter and exit a solid orthogonally. Figure 4 (d) shows the tracks respecting all boundary imprints in front, top, right, and left surfaces. For example, the tracks at the semicircle enter the front surface orthogonally and then take a right turn at the MA (see Figure 4 (f)) and exit orthogonally at the right surface. Thus, a conformal mesh has been generated by respecting all the imprints and by ensuring the orthogonality condition at the boundary.


(a) Assembly Model (b) Solid with boundary imprints (c) MA junctions & boundary imprints

(d) Tracks respecting boundary imprints (e) Cross section of mesh on MA (f) Conformal mesh at interface

Figure 4 Overview of LayTracks3D on assembly model

7 Extension to All-Hex Meshing

LayTracks3D is built on the strong mathematical foundation that is inherent in the MAT. Every 3D solid has a unique MA and that MA is a continuous rich skeleton representation of the 3D solid. The MA reduces 3D hex meshing to quad meshing on the MA. Thus completing quad meshing on the MA will complete hex meshing the solid. The LayTracks3D algorithm is based on the uniqueness and continuity of the MAT as given by the following lemma [22].

Lemma 1.0 Uniqueness and Continuity of Mapping to MAT

Let $\mathcal{A}$ be an n-dimensional compact sub-manifold of R^n and let $MA(\mathcal{A})$ be its medial axis. Let $\mathcal{P}$ be an open subset of δA which is G^1 and piecewise C^2 continuous. Then for every point $p \in \mathcal{P}$ there is one and only one maximal ball touching p. Furthermore, the function $M\colon P \rightarrow MA(\mathcal{A})$, which maps each point $p \in \mathcal{P}$ to the center of its maximal ball, is continuous.

The mapping function $M\colon P \rightarrow MA(\mathcal{A})$ in the above lemma connects a point p on the boundary to maximal ball center Mp as shown in Figure 5. This lemma is central to LayTracks3D in (1) creating corridors by connecting branch points with its tangent points (see Figure 3 (c)) to get high quality boundary oriented elements, (2) creating tracks inside the corridor by placing rails using a 1-to-1 map from the boundary to the MA and from the MA to the boundary thus guarantying no branches inside the tracks (see Figure 5), (3) projecting imprints from all directions onto the MA (see Figure 4 (c)), and (4) cutting the assembly interface orthogonally respecting all imprints to guarantee conformal mesh (see Figure 4 (d)). Therefore, all the major steps of LayTracks3D are very robust as they are derived from $M: P \rightarrow MA(\mathcal{A})$.

Let us now use this map $M: P \rightarrow MA(\mathcal{A})$ to prove the following two conditions for all-hex meshing:

- Condition 1: A general solid can be decomposed into a set of connected 3D tracks
- Condition 2: A 3D track can be meshed with all-hex elements

Therefore, if these two conditions can be met, then any general solid can be meshed with all-hex elements using LayTracks3D. The below sections show how these two conditions are met theoretically, although full implementation is not completed at this time.

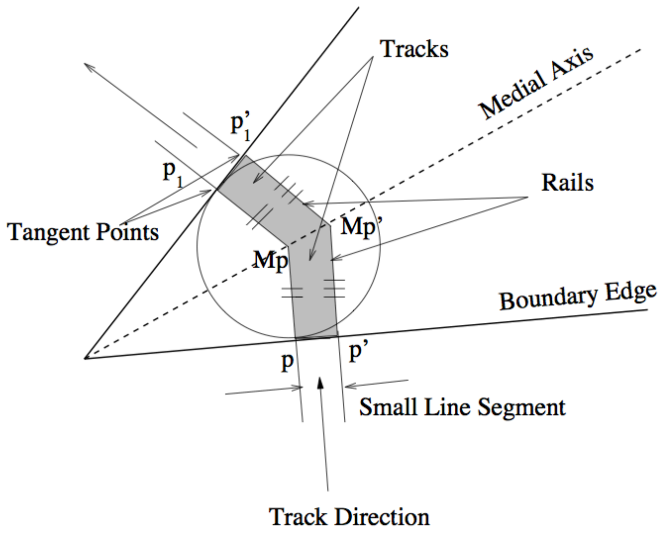


Figure 5 Track generated using MA map

Condition 1: Decompose General Solid into Set of Connected 3D Tracks

Let us now examine Condition 1. Even though a general solid can be decomposed into a set of connected tracks as stated in Condition 1, the tracks may not have quadrilateral cross section everywhere. Note that the above lemma assumes that the boundary is G^1 continuous (i.e. tangent direction is continuous) and piecewise C^2 continuous (i.e. second derivatives are continuous). In practice, at concave edges/vertices, where the map is 1-to-N, the statement *"every point $p \in \mathcal{P}$ there is one and only one maximal ball touching p"* does not hold. Therefore, a 1-to-1 map between *p* and *Mp* can not be guaranteed at concave edges/vertices.

Solution for concave edge/vertex, where the map is 1-to-N is shown in Figure 6. The solution is to perturb the 1-to-N map to 1-to-1 using smoothing operation. The multiple coincident points at the concave vertex will get spread out locally. Note that the orthogonality condition will be lost locally at the convex vertex.

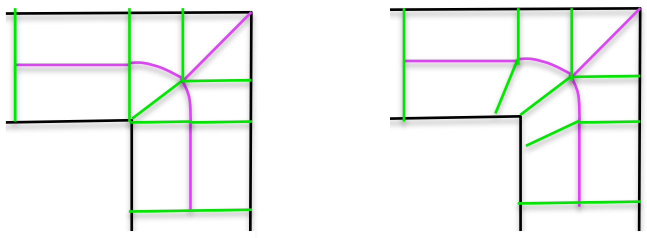


Figure 6 1-to-N transformed to 1-to-1 map

The solution for the finite contact where the map is N-to-1 is shown in Figure 7. Figure 7 (a) shows a finite contact case such as a cylinder or sphere where multiple points on the boundary map to a single point on the MA. This single solid problem is transformed to an assembly problem by introducing a core at the N-to-1 map of the MA regions. Figure 7 (b) shows a cuboidal core at the center of the sphere. Note the change in the MA shown in purple. The MA, which was a point at the center of the sphere, now becomes a sheet in between the outer sphere and inner cuboidal core. Thus we have removed the N-to-1 map throughout the domain. Note that there exists a 1-to-N map at the concave vertices of the outer sphere (see Figure 7 (b)). As discussed in the above paragraph, a 1-to-N map at the concave regions can be transformed to a 1-to-1 map by spreading the coincident nodes at the concave vertex (see Figure 7 (c)). Thus the N-to-1 map is transformed to a 1-to-1 map and an all-hex mesh can be generated on the sphere.

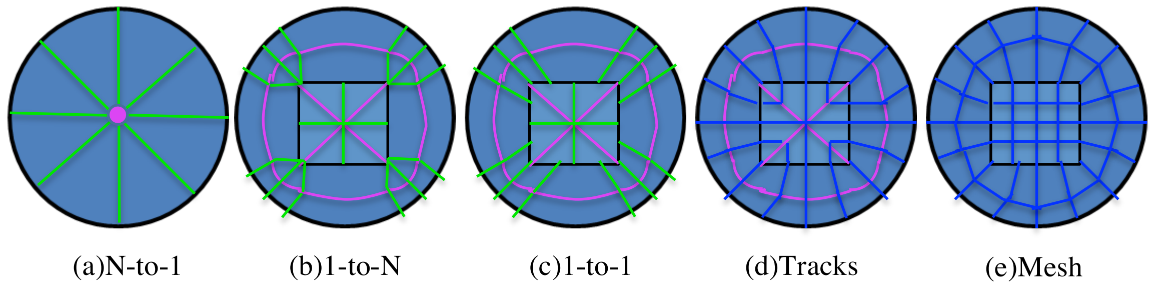


Figure 7 N-to-1 transformed to 1-to-1

Figure 8 shows all possible types of tracks that can arise from the 1-to-N and N-to-1 maps. As discussed in the above paragraphs, 1-to-N and N-to-1 maps can be transformed to a 1-to-1 map. The 1-to-1 map then satisfies Condition 1 and quad cross-section tracks (as shown in the first image of Figure 8) can be generated on general solids.

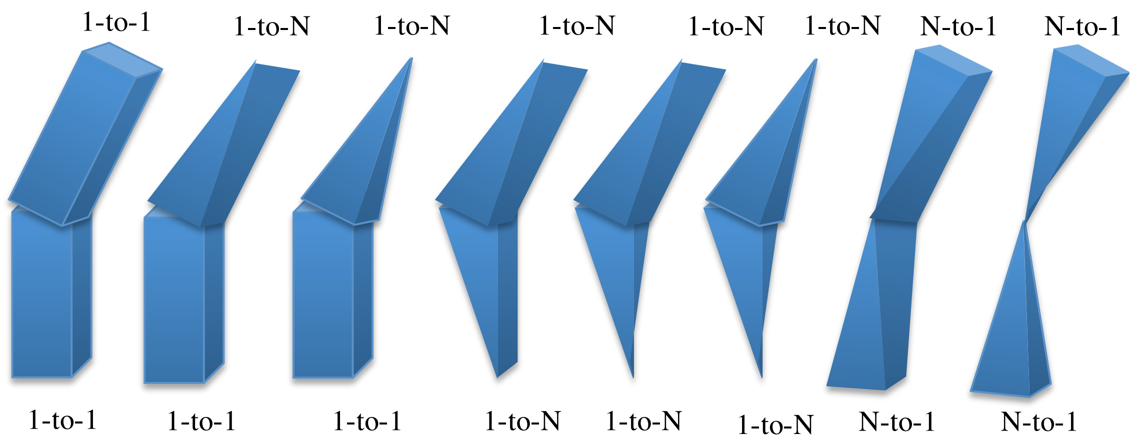


Figure 8 Types of Tracks in 3D

Condition 2: Mesh a 3D Track with All Hex Elements

Let us now examine Condition 2, which requires us to mesh every 3D track with all-hex elements. LayTrack3D first meshes all the rails and then meshes all the tracks. The hexes are built in advancing front manner inside each track, i.e., hex elements are built from the boundary towards the interior MA.

Here the author would like to point out why the current implementation discussed in Section 4 does not guarantee an all-hex mesh and how to overcome this limitation to achieve an all-hex mesh. In the current implementation non-hex elements can arise at the MA when the hex elements are built inside the track in advancing front manner. This is because LayTrack3D described in Section 4 has no control on the number of intervals present in the rails of the tracks. The quad mesh on the medial shown in Figure 3 (d) is randomly generated using the Paving algorithm for a given mesh size without any special attention. Paving is an unstructured meshing algorithm and the quad nodes can exist in an unstructured manner on the medial surface. Then the rails are generated from these quad nodes. Note that the rails on either side of a quad node will have the same length as they are radii of a maximal ball (see Figure 5). The rail length is then divided by the desired mesh size to set the intervals on the rails. Thus an even number of intervals on the rails of a track is set automatically without any special attention.

In order to obtain an all-hex mesh inside a quad cross-section track, (1) all four rails must have the same intervals or (2) two rails must have 2N intervals and the other two rails must have 2(N+1) intervals. With Case 1, all hex elements can be easily built on equal interval rails and with Case 2, two wedges at the MA can be merged to form a hex as shown in Figure 9. If the intervals on the four rails of a track do not satisfy Case 1 or Case 2, then non-hex elements will be generated at the MA.

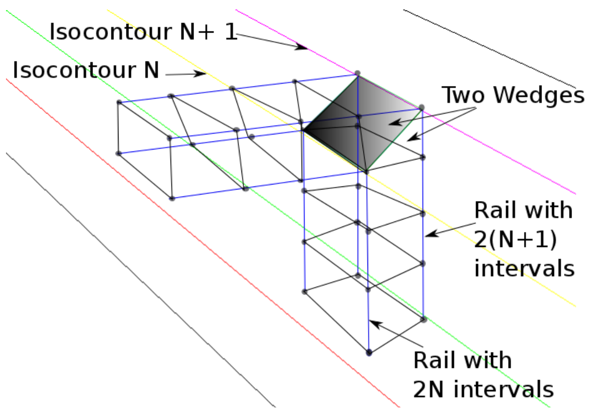
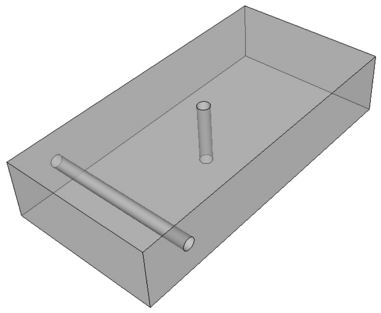


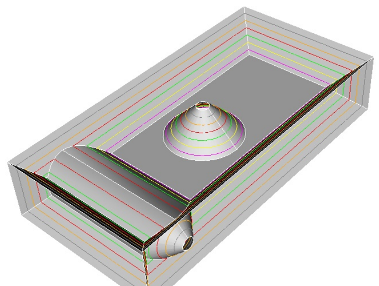
Figure 9 Merge two wedges to get all-hex

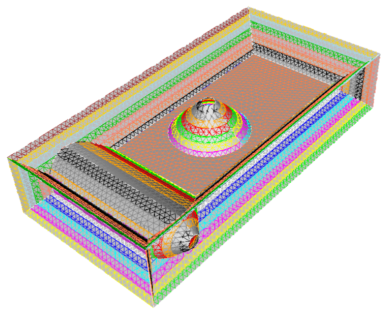
The big question is how to generate a mesh on the medial surface such that all tracks satisfy either Case 1 or Case 2? Figure 10 (d) shows the mesh on the MA that will give an all-hex mesh by satisfying either Case 1 or Case 2 in every track. In order to satisfy Case 1 or Case 2 in every track, the isocontours of the radius function of the MA have been utilized. Figure 10 (b) shows isocontours of the radius function of the MA obtained for mesh size increments. Isocontours and non-manifold MA curves split the MA into different regions/segments as shown in Figure 10 (c). Figure 10 (c) also shows the underlying facets of these MA segments. Figure 10 (d) shows the quads and tris on the MA that represent the quad and tri cross section tracks in 3D, respectively. The intervals on a rail generated at a node can be easily determined by knowing the isocontour number (see Figure 9). Figure 10 (d) shows two regions bounded by only one isocontour and non-manifold MA curves. These regions can be meshed using unstructured quad meshing algorithm such as paving. All the rails at the quad nodes of this an unstructured mesh will have the same intervals thus satisfying Case 1. The rest of the region is bounded by two isocontours N and N+1 and non-manifold MA curves. A quad in this region is shown in Figure 9. Two quad nodes lie on isocontour N and the two rails originated from these two quad nodes will have 2N intervals. Similarly, the other two quad nodes lie on isocontour N+1 and the two rails originated from these two quad nodes will have 2(N+1) intervals. Thus, the track satisfies Case 2.

In order to improve the hex quality at 3-manifold MA curves, a layer of tri elements are generated all along the 3-manifold MA curve as shown in Figure 10 (d). Figure 11(a) shows a 3-manifold MA curve at a convex vertex. Figure 11(b) shows a layer of tri elements all along the 3-manifold MA curve. The tracks corresponding to the tri will have triangular cross-section. Figure 11(c) shows how six tets form a hex at a 3-manifold MA edge and two wedges form a hex along the two tri cross section tracks. Figure 3 (d) and Figure 3 (f) show that having a tri along the 3-

manifold MA curve will result in high quality hex elements at the convex corner. Figure 4 (h) and Figure 4(i) show that having quads along the 3-manifold MA curve will result in poor quality elements at the convex vertex. Thus, using tri elements will give high quality hex elements that are conformal with the hex elements of the adjacent quad cross section tracks.


(a) Non-sweepable solid


(b) Isocontours of MA


(c) MA segments

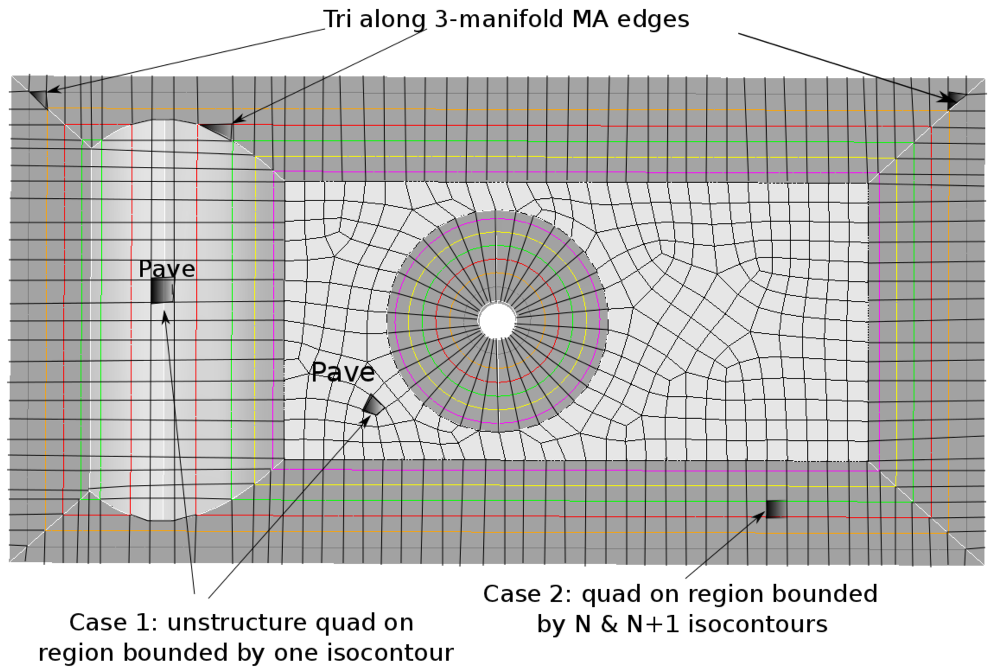
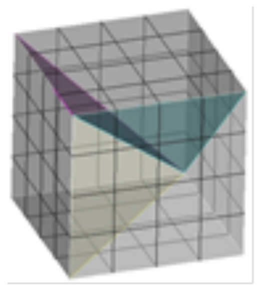
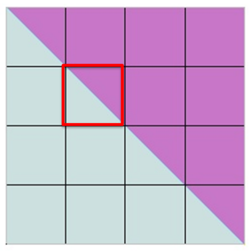

Tri along 3-manifold MA edges
Pave
Pave
Case 1: unstructure quad on region bounded by one isocontour
Case 2: quad on region bounded by N & N+1 isocontours
(d) Meshing the MA using isocontours to obtain desired number of intervals on rails.

Figure 10 Meshing the MA using isocontours of radius function to achieve all-hex mesh.


(a) All-hex at 3-manifold MA curve


(b) Top view of tri elements along 3-manifold MA curve

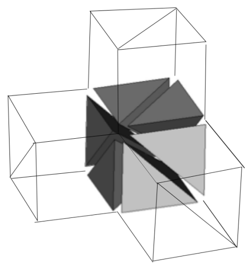
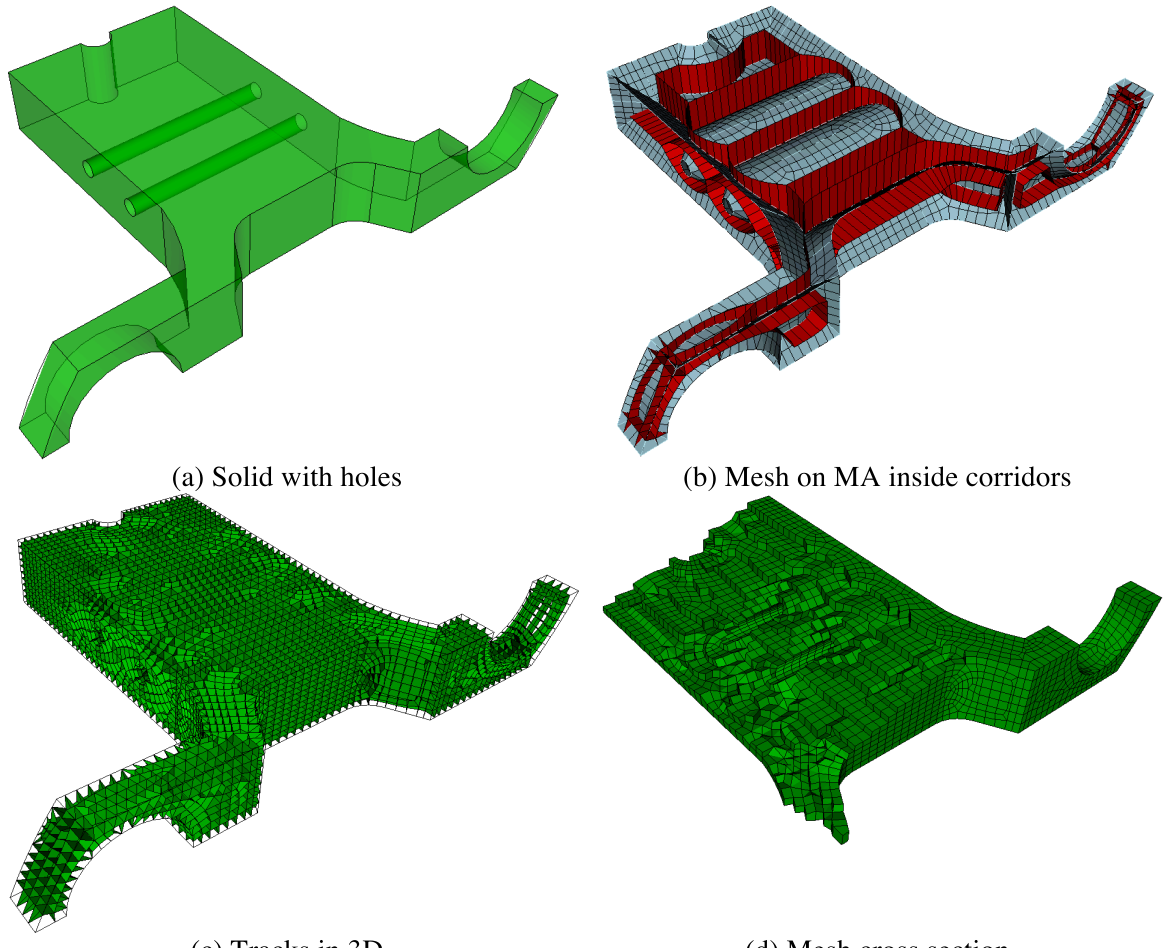

(c) Six tri-cross section tracks form an hex at MA edge

Figure 11 All-hex mesh at 3-manifold medial edge connecting a convex vertex

The final all-hex mesh topology can be visualized using the quad mesh topology on the MA. An unstructured hex mesh containing irregular nodes will exist in the tracks originating from the unstructured quad mesh. Figure 10 (d) shows the unstructured quad mesh in the region bounded by one isocontour. In Figure 10 (d), the regions bounded by two isocontours have structured quad meshes. Therefore, the hex mesh originating from this quad mesh will have a structured mesh.

8 Results

LayTrack3D has been implemented in CUBIT [23] using the CADFix [24] medial object library. The author is actively working on LayTracks3D and this section shows the current results obtained on some industrial models. The time taken to compute the MA of the below models is on the order of minutes. It takes about half a minute on a MacBook Pro for the rest of the process which includes importing the CAD model, importing the MA model, meshing the MA, building corridors, building tracks, meshing tracks, and exporting the mesh. This computation time is significantly lower than to the user time required in methods requiring manual decomposition for hex meshing.

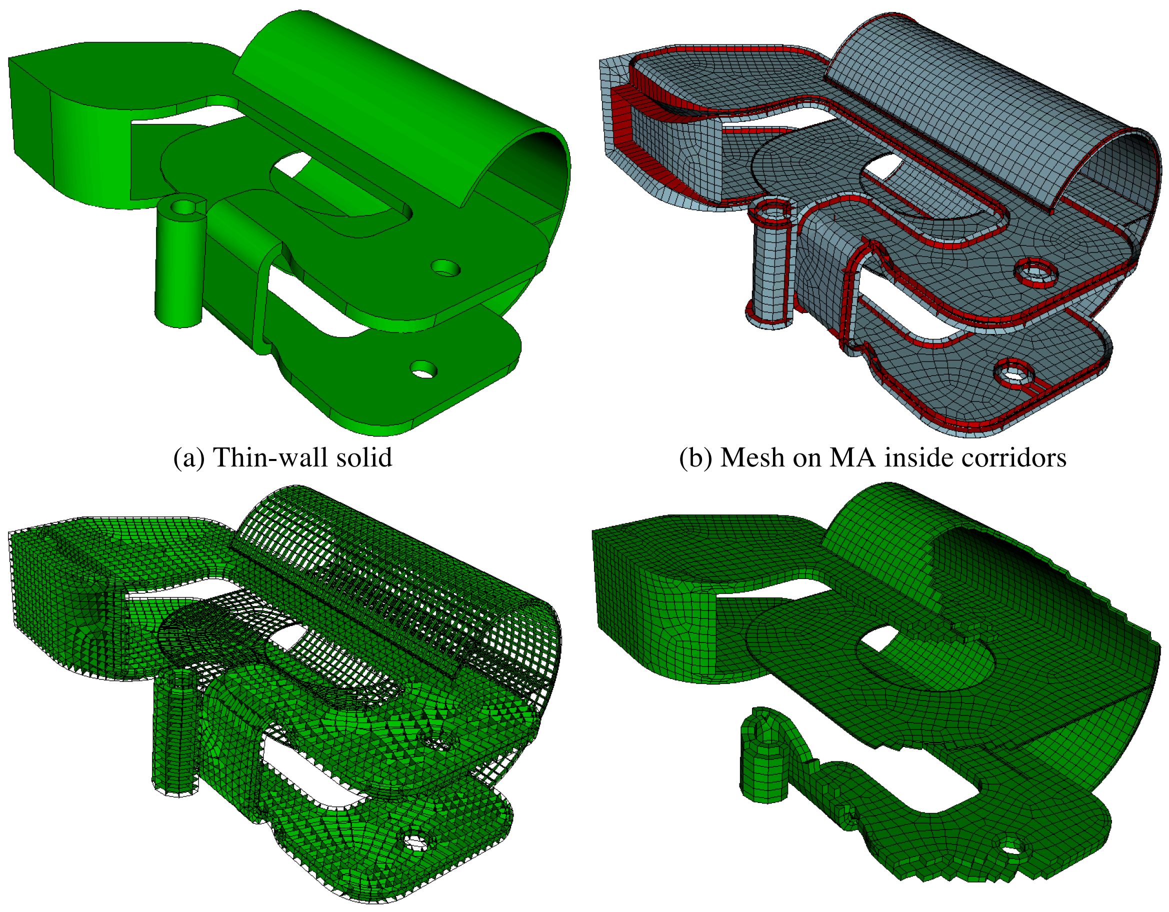

(a) Solid with holes (b) Mesh on MA inside corridors

(c) Tracks in 3D (d) Mesh cross section

Figure 12 Meshing a solid containing two through holes

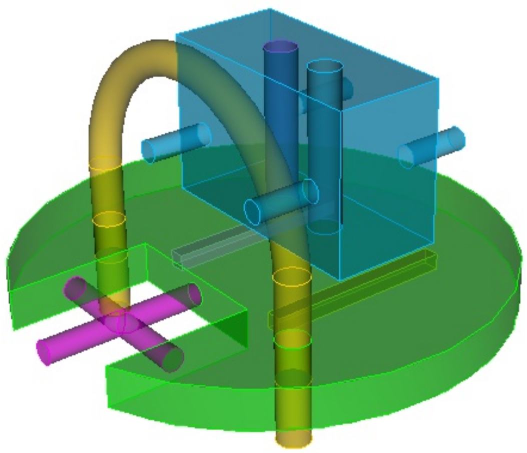
Currently, hex-dominant meshes can be generated by meshing the MA using the Paving algorithm (see Figure 3). Generation of the isocontours discussed in Section 7 has been implemented; however, the quad meshing on the MA using the isocontours to achieve all-hex meshes is not yet implemented. Therefore, in this section only the hex-dominant meshes are shown, which by itself is a significant step forward in the MA-based hex meshing research. While hex-dominant meshes are not suitable for all applications, numerous numerical simulations can use hex-dominant meshes [25].

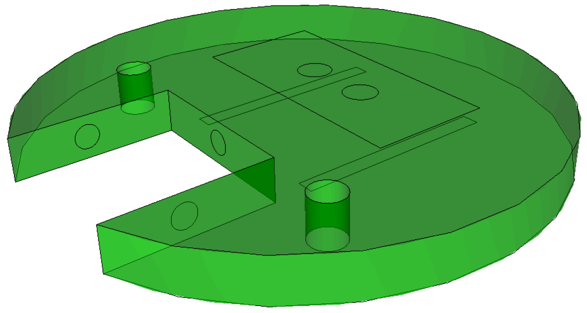
Figure 12(a) shows a typical industrial solid containing through holes, sharp features, fillets, concave regions, and thin-thick cross sections. Figure 12(b) shows the corridors obtained by propagating rails (shown in red) from the non-manifold MA junction curves, which represent critical singularities. The MA is meshed using the paving algorithm with a layer of tri elements along the 3-manifold MA curves touching convex vertices. Figure 12(c) shows the tracks, which look like tunnels in 3D and are much simpler to mesh than the original solid. An advancing front method has been used to mesh the tracks by building high-quality (near cubical shape) hex elements from the boundary towards the MA. Figure 12(d) shows the hex-dominant mesh containing 10,260 hex elements and 536 non-hex elements, i.e., 5.2% of elements are non-hex elements.

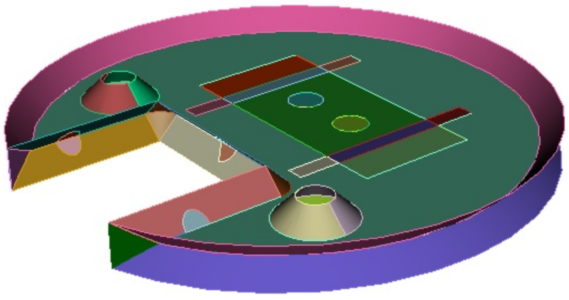

(a) Thin-wall solid (b) Mesh on MA inside corridors

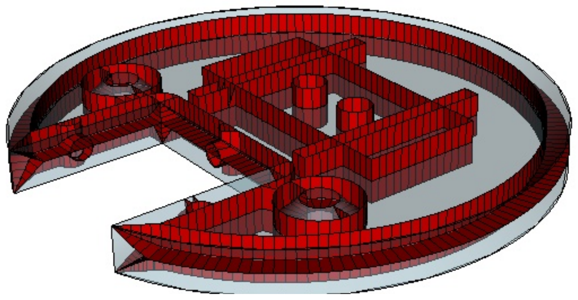
(c) Tracks in 3D (d) Mesh cross section

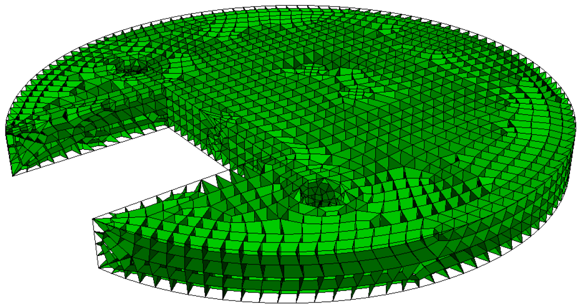
Figure 13 Meshing a thin walled solid


(a) Assembly


(b) Imprints on the boundary of a solid


(c) Imprints on MA


(d) Corridors


(e) Tracks

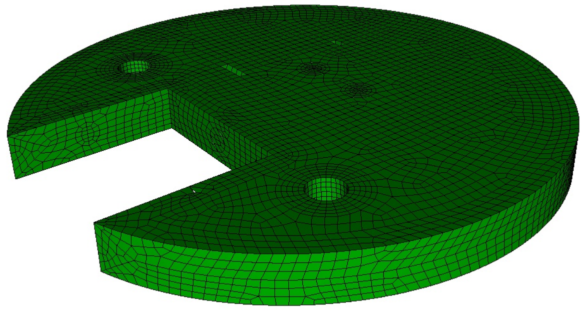

(f) Final mesh

Figure 14 Meshing an assembly model with imprinted interface

The proposed method can handle thin-walled solids irrespective of the relative scale between the thin and the thick regions. Figure 13(a) shows a typical industrial thin-walled solid. Figure 13 (d) shows that the final mesh is always aligned orthogonal to the boundary. Note that currently we get a minimum of one layer of hex on either side of the MA. Figure 13 (d) shows the mesh cross section containing two layers of hexes at thin walls. The hex-dominant mesh contains 14,104 hex elements and 185 non-hex elements, i.e., 1.3% of elements are non-hex elements.

Figure 14 shows an assembly model with eight parts and Figure 14 (b) shows the thick cylindrical plate in transparent mode with imprints from adjacent parts. The top surface contains a rectangular imprint with two inner circles. The bottom surface contains two rectangular imprints and the lateral surfaces of the plate contain three circular imprints. These disconnected imprints - which are located on different surfaces spatially - are resolved by transforming them from the solid to the MA using the map. Figure 14 (c) shows all the imprints on the MA. Corridors are then generated using the imprints and the non-manifold junction curves of the MA. Note that the corridors pass through the boundary imprints orthogonally and are simpler subdomains with a 1-to-1 map. Figure 14 (e) and Figure 14 (f) show tracks and the hex-dominant mesh respectively. As the mesh respects the boundary imprints, other schemes such as sweeping can be used on simpler parts of the assembly. The mesh quality at convex vertices can also be further improved by having a layer of tri elements along 3-manifold MA curves as shown in Figure 12 & Figure 13.

9 Conclusion

This paper presents an extension of LayTracks to generate hex-dominant meshes of general solids using the MAT. The algorithm first decomposes any general solid into corridors using boundary imprints and MA junction curves. Next, corridors are further subdivided into simpler tracks that look like tunnels in 3D. Tracks are then meshed in advancing front manner without any interference checks to generate high-quality hex-dominant meshes. The algorithm generates meshes on general solids and assemblies with desirable features such as boundary sensitivity, orientation insensitivity, sharp feature and imprint preservation, and high element quality without any manual decomposition/interaction. Work is under way to guarantee all-hex meshes using isocontours of the MA radius function. In the near future, the generated meshes will be validated using non-linear solid mechanics analysis codes.

Acknowledgements: The author would like to thank Henry Bucklow, Robin Fairey and Mark Gammon at ITI TranscenData Europe Ltd. for providing the CADfix medial object library.

References

1. W.R.Quadros, K.Ramaswami, F.B.Prinz and B.Gurumoorthy, "LayTracks: A New Approach to Automated Quadrilateral Mesh Generation using MAT", proceedings of the 9th International Meshing Roundtable, 239-250, Oct 2000
2. H Blum, A Transformation For Extracting New Descriptors Of Shape, Models for the Perception of Speech and Visual Form, Cambridge, MA, The MIT Press, pages 326-380,1967
3. H Blum, Biological shape and visual science (part I). Journal of Theoretical Biology 38:205–287, 1973
4. H Blum, Nagel RN, “Shape description using weighted symmetric axis features”, Pattern Recognition 10:167–180, 1978
5. W.R.Quadros, K.Ramaswami, F.B.Prinz, and B.Gurumoorthy, "LayTracks: A New Approach to Automated Geometry Adaptive Quadrilateral Mesh Generation using Medial Axis Transform," International Journal for Numerical Methods in Engineering, vol. 20, pp. 249-264, 2004
6. W.R.Quadros, K.Ramaswami, F.B.Prinz and B.Gurumoorthy, "Automated Geometry Adaptive Quadrilateral Mesh Generation using MAT", proceedings of ASME DETC, Sept 2001
7. M.A. Price and C.G. Armstrong, “Hexahedral Mesh Generation by Medial Surface Subdivision: Part I”, International Journal for Numerical Methods in Engineering, Vol 38(19), pp.3335-3359, 1995
8. M.A. Price and C.G. Armstrong, “Hexahedral Mesh Generation by Medial Surface Subdivision: Part II”, International Journal for Numerical Methods in Engineering, Vol 40, pp.111-136, 1997
9. P. Sampl, “Semi-structured mesh generation based on medial axis”, proceedings of the 9th International Meshing Roundtable, 21–32, Oct 2000

10. J. Makem et al., “Automatic Decomposition and Efficient Semi-Structured Meshing of Complex Solids”, proceedings of the 20th International Meshing Roundtable, Oct 2011
11. V. Vyas and K. Shimada, “Tensor-guided hex-dominant mesh generation with targeted all-hex regions”, proceedings of the 18th International Meshing Roundtable, pp. 377–396, 2009
12. J. Huang, T. Jiang, Y. Wang, Y. Tong, and H. Bao, “Automatic frame field guided hexahedral mesh generation”, Tech. report, State Key Lab of CAD & CG, College of Computer Science at Zhejiang University, Dec 2012
13. J. Huang, Y. Tong, H. Wei, and H. Bao, “Boundary aligned smooth 3d cross frame field”, ACM Transactions on Graphics, 30, no. 6, 143:1–143:8, 2011
14. Y. Li, Y. Liu, W. Xu, W. Wang, and B. Guo, “All-hex meshing using singularity-restricted field”, ACM Transactions on Graphics 31, no. 6, 177:1–177:11, 2012
15. N. Kowalski, F. Ledoux, and P. Frey, “A pde based approach to multi domain partitioning and quadrilateral meshing”, proceedings of the 21th International Meshing Roundtable, pp. 137–154, 2012
16. N. Kowalski, F. Ledoux, M.L. Staten, and S.J. Owen, “Fun sheet matching: towards automatic block decomposition for hexahedral meshes”, Engineering with Computers 28, 241–253, 2012
17. T.D. Blacker and M.B. Stephenson, “Paving: a new approach to automated quadrilateral mesh generation”, International Journal for Numerical Methods in Engineering 32, 811–847, 1991
18. T.D. Blacker and R.J. Meyers, “Seams and wedges in plastering:a 3d hexahedral mesh generation algorithm”, Engineering with Computers 2, no. 9, 83–93, 1993
19. M.L. Staten, R.A. Kerr, S.J. Owen, T.D. Blacker, M. Stupazzini, and K. Shimada, “Unconstrained plastering - hexahedral mesh generation via advancing front geometry decomposition”, International Journal for Numerical Methods in Engineering 81, 135–171, 2010
20. M.L. Staten, R.A. Kerr, S.J. Owen, and T.D. Blacker, “Unconstrained paving and plastering: Progress update”, proceedings of the 15th International Meshing Roundtable, pp. 469–486, 2006
21. M.L. Staten, S.J. Owen, and T.D. Blacker, “Unconstrained paving and plastering: A new idea for all hexahedral mesh generation”, proceedings of the 14th International Meshing Roundtable, pp. 399–416, 2005
22. E.C. Sherbrooke, N.M. Patrikalakis, F. Wolter, “Note on differential and topological properties of medial axis transforms”, Graphical Models Image Process 58:547–592, 1996
23. CUBIT Geometry and Meshing Toolkit, https://cubit.sandia.gov/public/14.1/Cubit-14.1-announcement.html, Version 14.1, Released Jan 13, 2014.
24. CADFix, ITI TranscenData Europe Ltd, www.cadfix.com
25. T. Shelton, N. Crane, J. Cox, “An exploration of accuracy and convergence of the degenerate uniform strain hexahedral element (a solution to the unmeshed void in an all-hexahedral mesh)”, ASME 2013 International Mechanical Engineering Congress and Exposition, Volume 12: Systems and Design, San Diego, California, USA, November 15–21, 2013