Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 00 (2014) 000–000

SAND2015-1705J

Procedia Engineering

www.elsevier.com/locate/procedia

LayTracks3D: a new approach to meshing general solids using medial axis transform

William Roshan Quadros*

[1]Sandia National Laboratories,P O Box 5800, MS 0897,Albuquerque, NM – 87185 USA

Abstract

This paper presents an extension of the all-quad meshing algorithm called LayTracks to generate high quality hex-dominant meshes of general solids. 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 hex-dominant meshes are shown in the result section. Work is underway to extend LayTracks3D to generate all-hex meshes.

Peer-review under responsibility of organizing committee of the 23rd International Meshing Roundtable (IMR23)

Keywords: Medial Axis Transforms; Hex Mesh Generation; Hex-dominant Meshing, Geometry Decomposition; Assembly Meshing

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

* Corresponding author. Tel.: +1-505-220-9458; fax:+1-505-284-2418.
E-mail address: wrquadr@sandia.gov

1Sandia 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

Peer-review under responsibility of organizing committee of the 23rd International Meshing Roundtable (IMR23).

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.

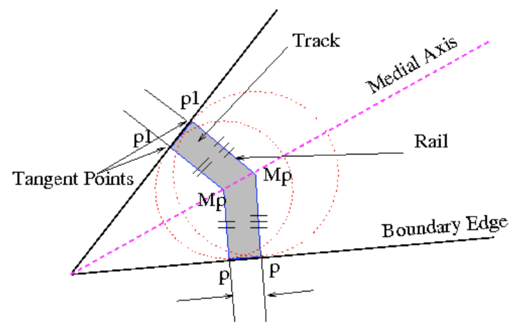


Figure 1 Track generated using MA map

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. The current focus has been to generate hex-dominant meshes.

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

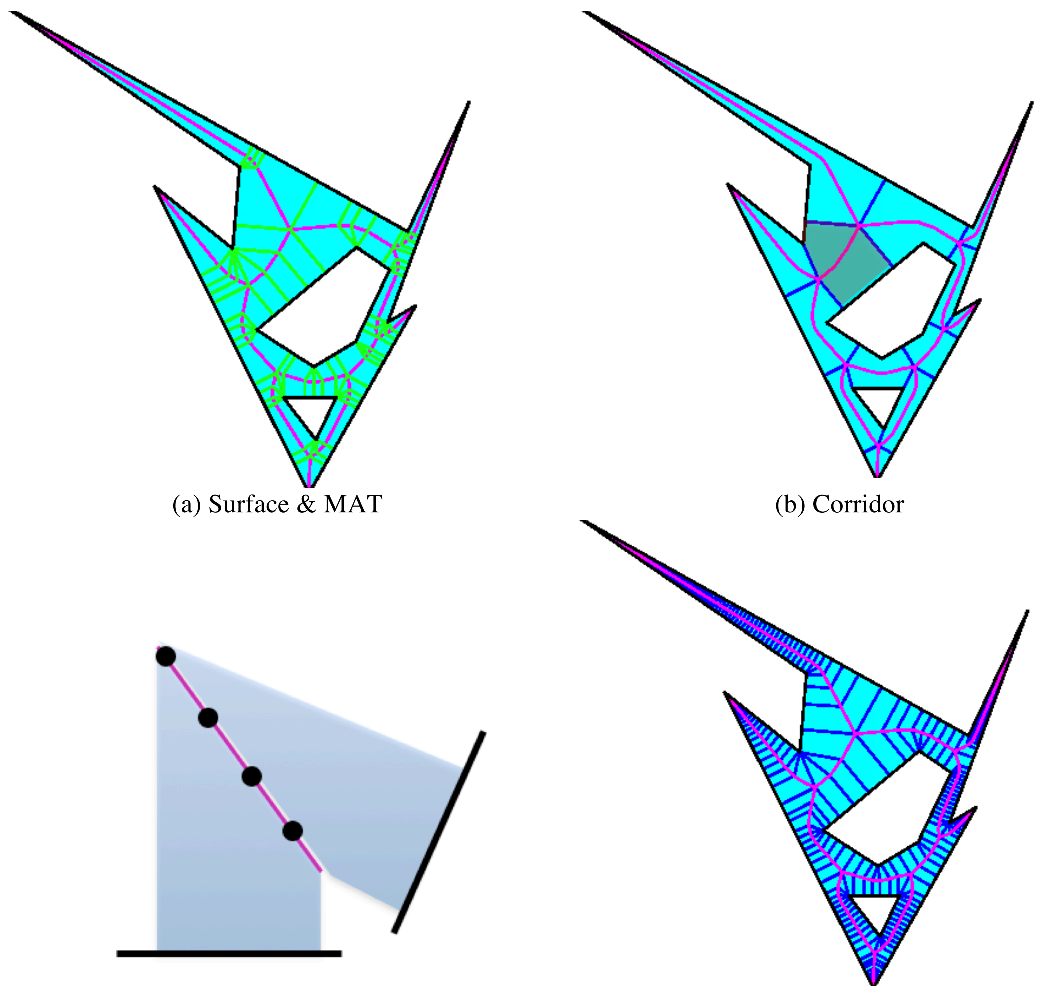
2. Overview of LayTracks

As it is easier to understand the algorithm in 2D, here we quickly recap the original 2D LayTracks [1]. For simplicity, first LayTracks on a single surface is discussed. Section 5 covers LayTracks on multiple surfaces. 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; 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 (see Figure 1), irregular nodes restricted at farthest distance from the boundary, an automatic conformal mesh at the interface of surfaces, and an all-quad mesh.

Figure 2 (a) shows a single surface in 2D and its MA. It also shows 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 2(b) shows the decomposition of the 2D domain into a set of connected simpler domains. Using the map, the branch point is connected to its corresponding tangent points on the boundary via line segments (see Figure 2(b)). The set of these connected line segments is called a Rail (see blue lines in Figure 2(b) and Figure 2(d)). A rail is 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 (one of the corridors is highlighted in Figure 2(b)).

Note that the MA skeleton representation reduces the surface meshing problem into curve meshing. The next step is propagating the rails inside each corridor as the MA curve segments are meshed by placing nodes based on the input mesh size. Uniform node spacing on the MA generates a uniform mesh and a varying node spacing based on the radius of the medial ball generates a geometry-adaptive mesh [6]. Figure 2(c) shows the uniform node spacing on a MA segment of a corridor. Figure 2(d) shows all the rails generated using non-uniform node spacing on the MA segments of each corridor. Here the spacing is directly proportional to the MA radius. The region between two adjacent rails' paths inside each corridor is called a Track [5] (see Figure 2(e)). Smaller width tracks are generated where the MA radius is smaller. Similarly, wider tracks are generated where MA radius is large.

A set of connected tracks is a much simpler domain to mesh compared to the original 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 1. 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 is the basis for a theoretical guarantee for an all-quad mesh [5].

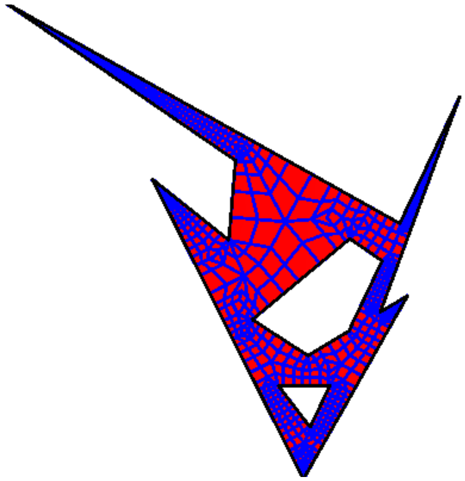


(a) Surface & MAT

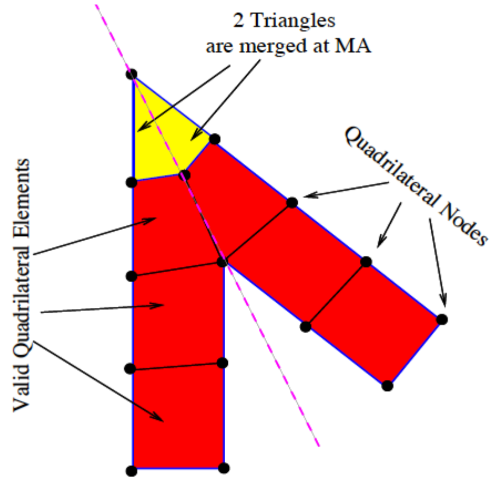
(b) Corridor

(c) Mesh on corridor's MA

(d) Tracks inside the corridors



(e) Mesh tracks



(f) All-quad mesh inside a track

Figure 2 Overview of quad meshing via LayTracks

3. **Overview of LayTracks3D**

LayTracks3D works analogous to 2D LayTracks [1] in decomposing a general solid/assembly into tunnel like tracks in 3D. Figure 3 shows the overview of LayTracks3D. Step 1 is to generate a 3D medial surface (see Figure 4(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 4(c)).

In Step 3, the 3D meshing is reduced to 2D meshing on the MA (see Figure 4(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 15.

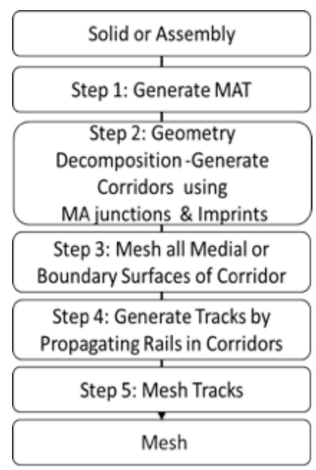


Figure 3 LayTracks3D overview

Step 4 involves subdividing the corridor into tracks, which look like tunnels with quad/tri cross sections (see Figure 4(e)). . 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. 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 (see Figure 5). 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 (see Figure 5). Section 7 gives more details on how to avoid non-hex elements at the MA to achieve all-hex meshes.

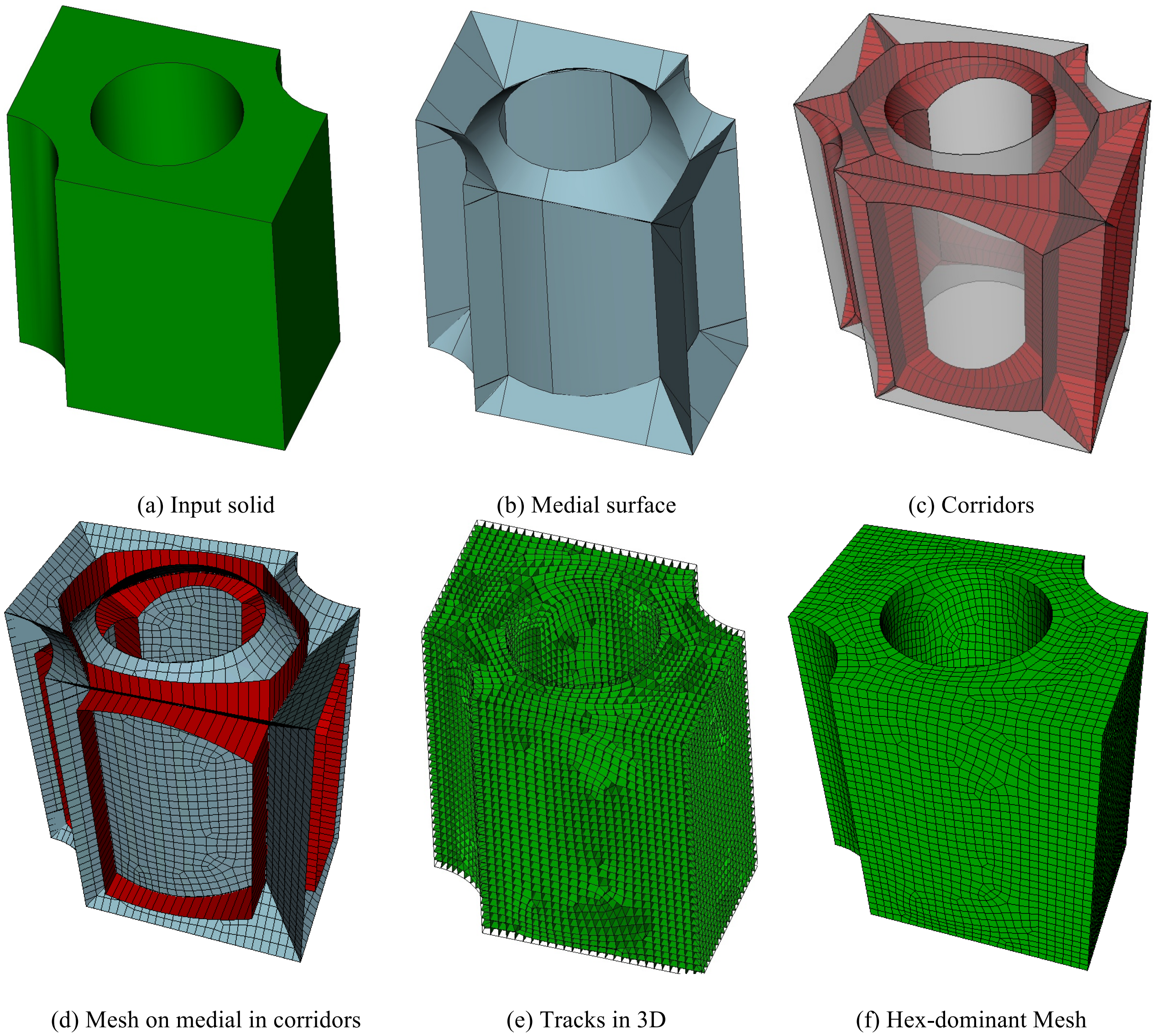


Figure 4 Overview of LayTracks3D

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 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.

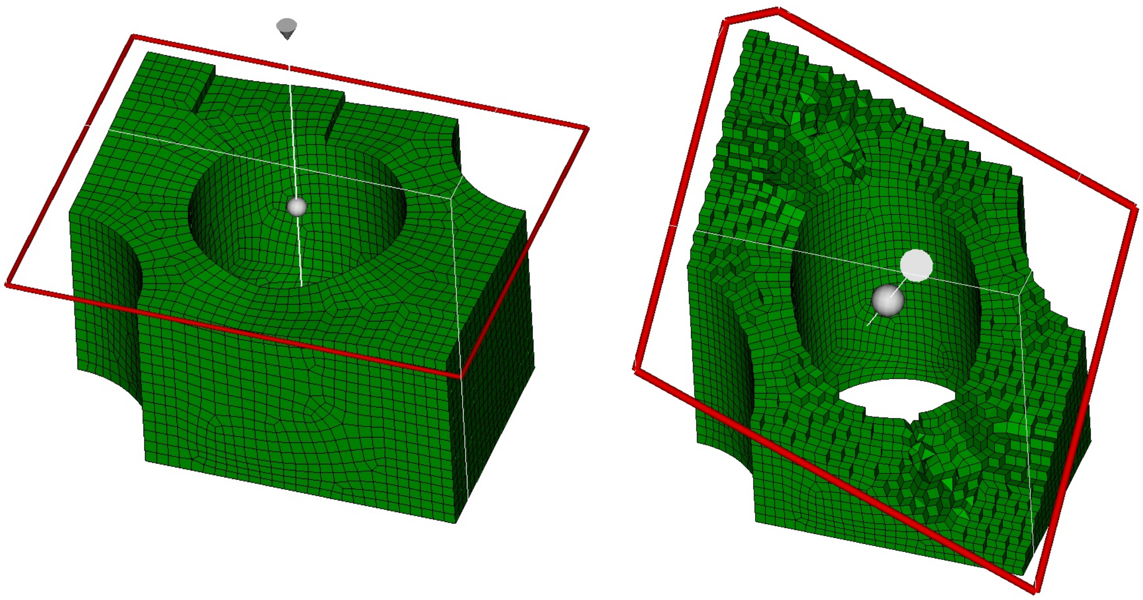


Figure 5 Sectional views of an hex-dominant mesh

- **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 multiple solids 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[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.

4. 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, the method of decomposition and advancing front methods are discussed as they are relevant to LayTracks3D.

The 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 Sampl [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., swept meshes are generated on the thin-sheet and long slender regions with unstructured tet meshes 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 direct extension of most 2D methods to 3D. Vyas and Shimada [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]. The 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 3D frame field on the surface 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 [15,16] 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 that there is no guarantee that a skeleton generated from the frame field can split the domain into a block-structured mesh in 3D.

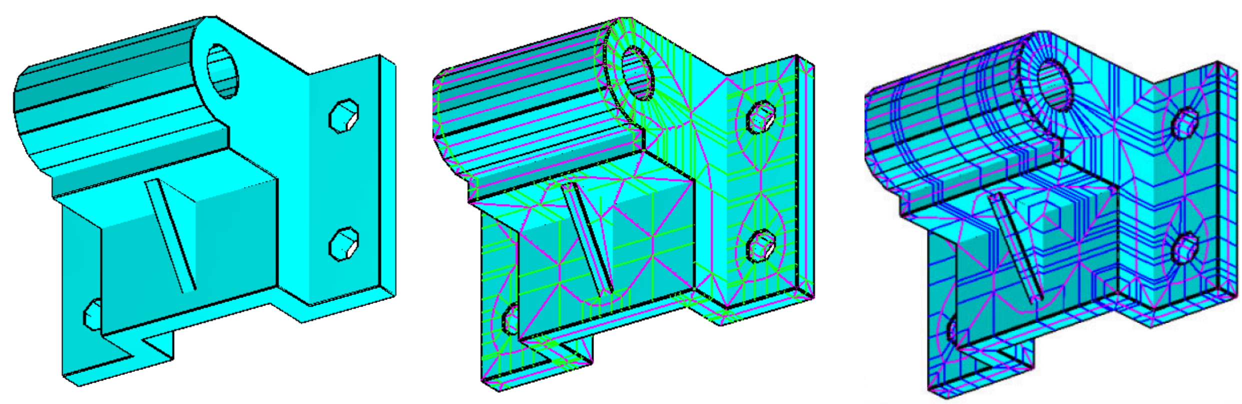
Another popular approach is advancing front methods because of their success in 2D. The extension of 2D advancing front all-quad meshing algorithm paving [17] to 3D plastering [18] has very limited success. Plastering starts with a pre-meshed boundary and places hex elements in an advancing-front manner, 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 [18,19,20] 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 these two popular mesh generation techniques, the method of decomposition and the 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 hex fronts propagated from the boundary are 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.

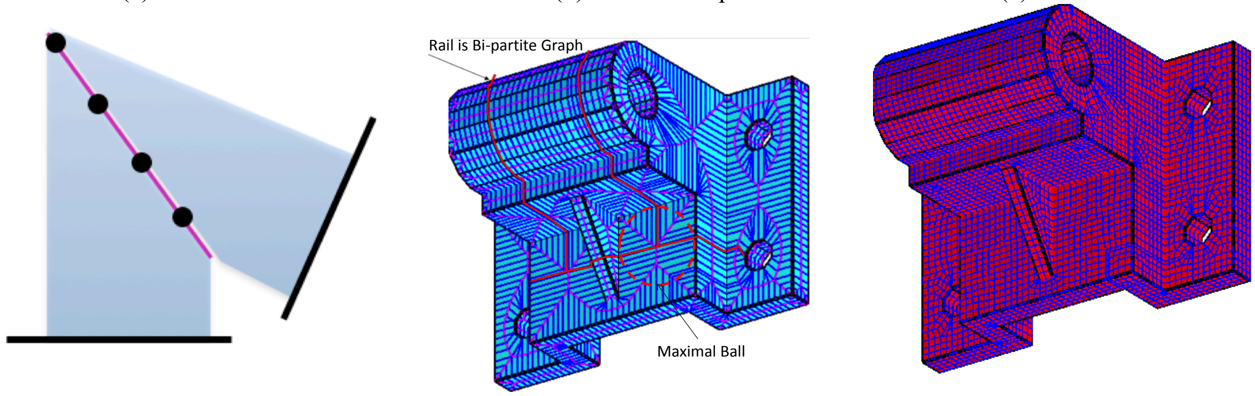
5. Extension of LayTracks to Multiple Surfaces

In this section LayTracks on multiple surfaces is discussed as it is difficult to visualize tracks propagating across multiple solids in 3D. First, the MA of each surface is generated independently. Figure 6 (b) shows the map between MA and boundary segments in each surface.

As described in Section 3, the critical part of subdividing the domain into simpler corridors is in placing the points at the branch points (see Figure 6 (c)). The rails are then propagated across multiple surfaces in breadth-first-traversal using the map. The propagation of the rail will terminate in three cases: (1) at an edge incident on only one surface, (2) at a concave vertex, or (3) at the starting point forming a loop. Figure 7 shows that the tangent points at the interfaces are equidistant from the common vertex incident on all the surfaces. Therefore, the rails form a perfect loop instead of forming a spiral. This is true even in 3D in the Euclidian space. It is possible that, the branch points in the MA of a surface are very close to each other, or the rails laid through the branch points of different surfaces come close. In such cases, the rails result in narrow corridors (see Figure 8). These narrow corridors need to be removed, otherwise they will result in quad elements with poor aspect ratio. Such a narrow corridor can be easily collapsed as a corridor is bounded by paths of two bipartite graphs [5]. Here we have removed the narrow corridors only by deleting/modifying the rails. Note that the MA is not altered, thus retaining the correct two-way mapping between the boundary and the MA.



(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 6 Overview of quad meshing via LayTracks

Figure 6 (d) shows meshing the MA inside each corridor. Thus corridors are subdivided into simpler tracks, which propagate across multiple surfaces as shown in Figure 6 (e). Note that rails cut the interface of the multiple surfaces orthogonally. Thus conformal mesh can be generated with near square elements along the boundary/interface of each surface as shown in Figure 6 (f).

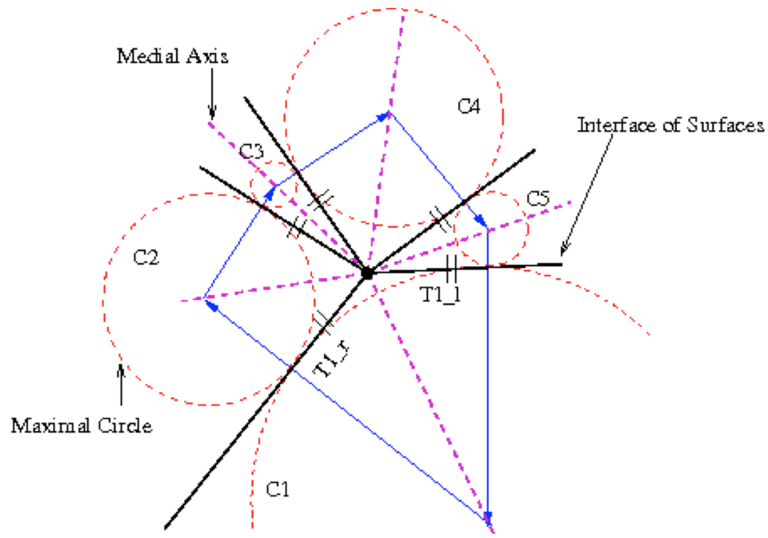


Figure 7 Rails can form a loop

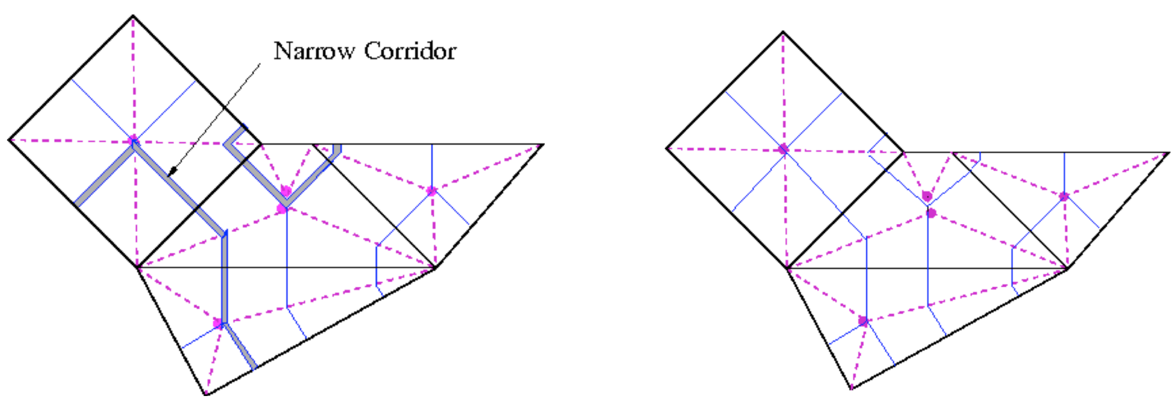


Figure 8 Removing narrow corridors on multiple surfaces

6. **Extension of LayTracks3D to Multiple Solids**

Meshing multiple solids 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 multiple solids, 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 9 shows the overview of meshing one solid of an assembly model via LayTracks3D. Figure 9(a) shows an assembly model with eight parts and Figure 9(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.

In is not trivial to get a hex mesh of the simple solid shown in Figure 9(b) 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 solid into sweepable subregions. For example, sweeping the circular imprinted on the left surface will intersect with sweeping the rectangle from the top. Even with plastering, imprinted surfaces 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. These disconnected imprints - which are located on different surfaces spatially - are resolved by transforming them from the solid's boundary to the MA using the map. As the MA is a lower dimensional skeleton representation of the 3D solid, all the boundary imprints are mapped onto the MA (see Figure 9 (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.

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. 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 9 (e) shows the tracks respecting all boundary imprints. Thus, a conformal mesh has been generated by respecting all the imprints and by ensuring the orthogonality condition at the boundary. The mesh quality at convex vertices can also be further improved by having a layer of tri elements along 3-manifold MA curves, which will be discussed later in Section 7

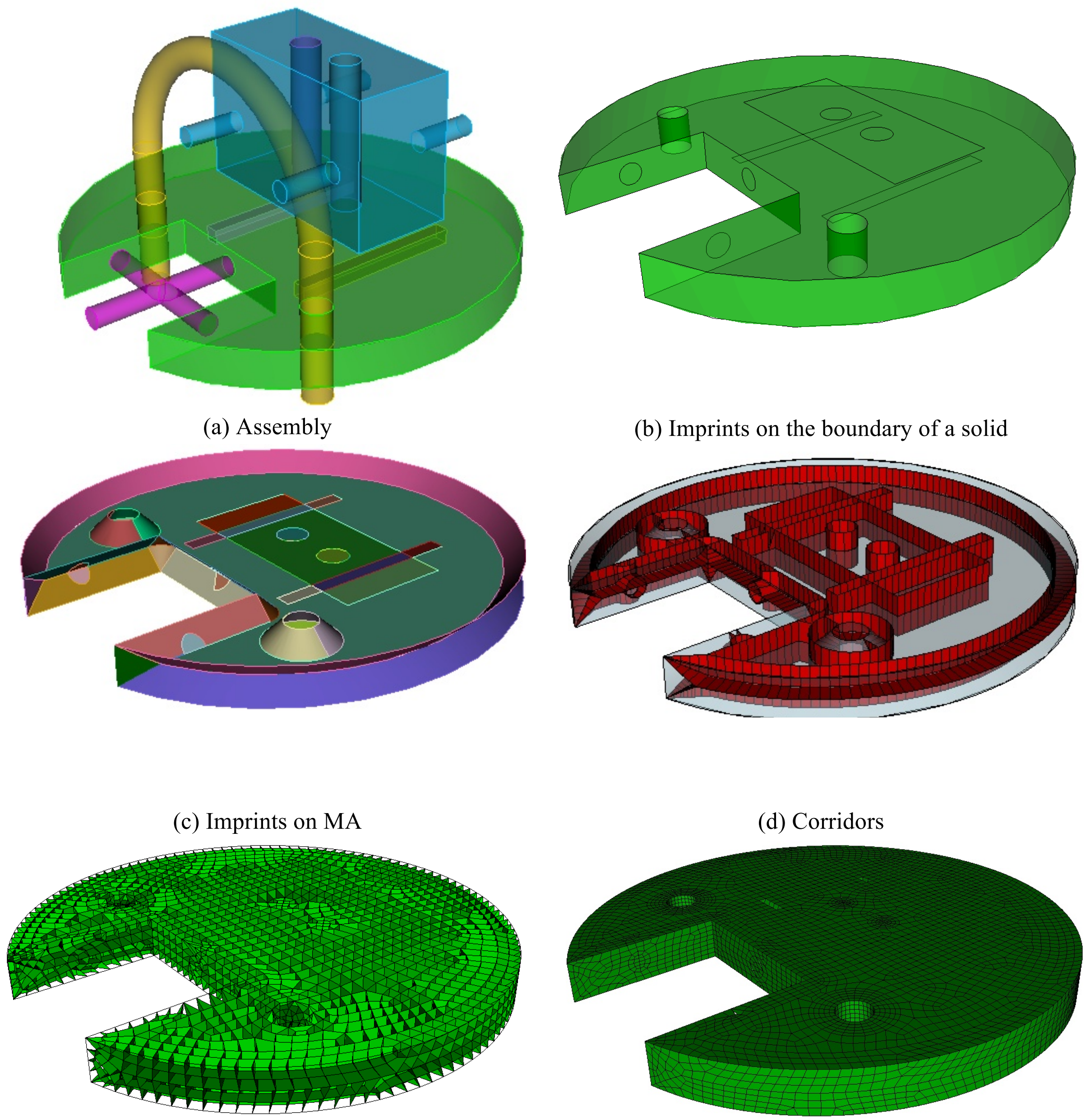
7. **Extension to All-Hex Meshing**

LayTracks3D is built on the strong mathematical foundation that is inherent in the MAT and hence has a very high potential in generating all-hex meshes of general solids. The LayTracks3D algorithm is based heavily 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 A be an n-dimensional compact sub-manifold of R^n and let MA(A) be its medial axis. Let P be an open subset of δA which is G^1 and piecewise C^2 continuous. Then for every point $p \in P$ there is one and only one maximal ball touching p. Furthermore, the function $M: P \rightarrow MA(A)$, which maps each point $p \in P$ to the center of its maximal ball, is continuous.

The mapping function $M: P \rightarrow MA(A)$ in the above lemma connects a point p on the boundary to maximal ball center *Mp* as shown in Figure 1. This lemma is central to LayTracks3D in (1) creating corridors by connecting branch points with its tangent points (see Figure 4(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 1), (3) projecting imprints from all directions onto the MA (see Figure 9 (c)), and (4) cutting the assembly interface orthogonally respecting all imprints to guarantee conformal mesh (see Figure 9 (d)). Therefore, all the major steps of LayTracks3D are very robust as they are derived from $M: P \rightarrow MA(A)$.



(a) Assembly (b) Imprints on the boundary of a solid

(c) Imprints on MA (d) Corridors

(e) Tracks (f) Final mesh

Figure 9 Overview of LayTracks3D on assembly model

Let us now prove the following two essential conditions for all-hex meshing:

Condition 1: A general solid can be decomposed into a set of connected quad cross-section 3D tracks

Condition 2: A quad cross-section 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 sections below show how these two conditions are met theoretically, although a full implementation is not completed at this time.

Condition 1: Decompose General Solid into Set of Connected Quad Cross-Section 3D Tracks

The quad mesh on the MA represents the 3D track decomposition via the mapping function $M: P \rightarrow MA(A)$. Thus, completing conformal quad meshing on the MA will guarantee connected 3D track decomposition. Note that a quad mesh on the MA may not always give tracks with quadrilateral cross section everywhere. This is because the map may not be 1-to-1 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 $p \in P$ there is one and only one maximal ball touching p"* does not hold. Therefore, a 1-to-1 map between *p* and *Mp* cannot be guaranteed at concave edges/vertices. Figure 10 shows all possible types of tracks that can arise from the 1-to-N and N-to-1 maps.

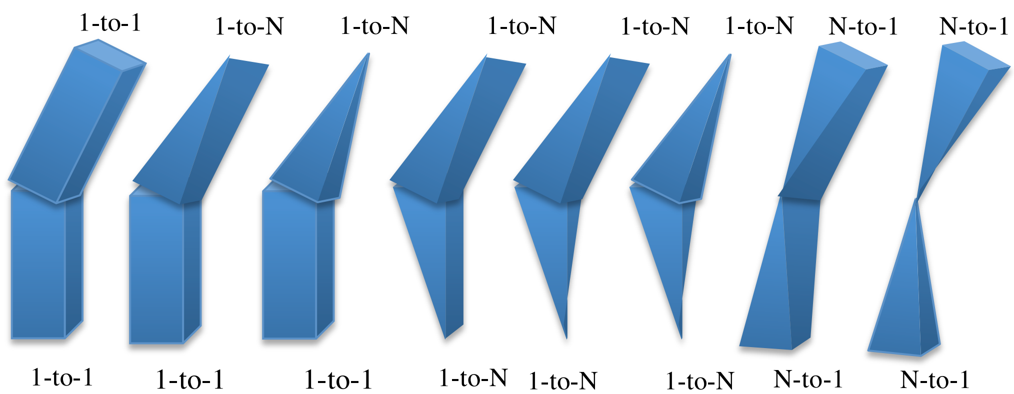
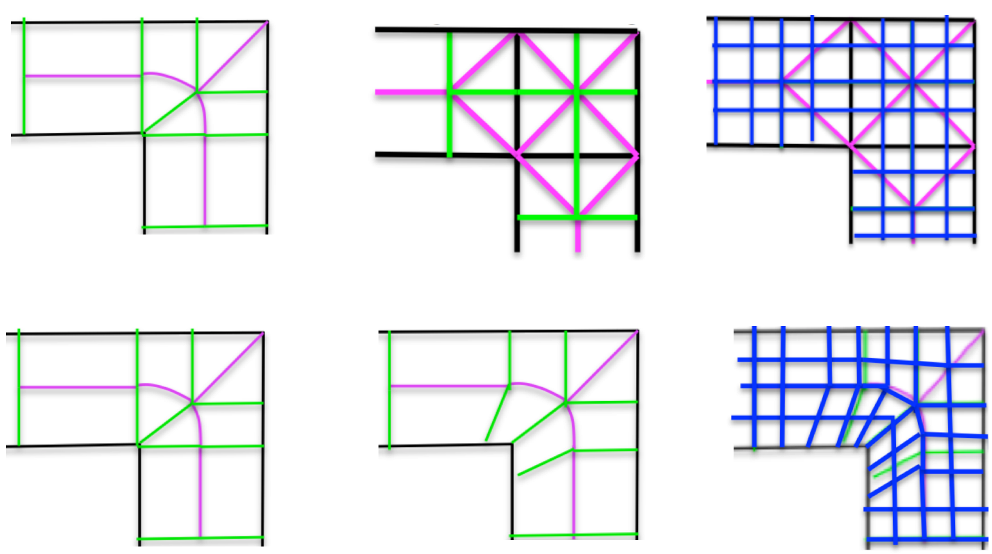


Figure 10 Types of tracks in 3D

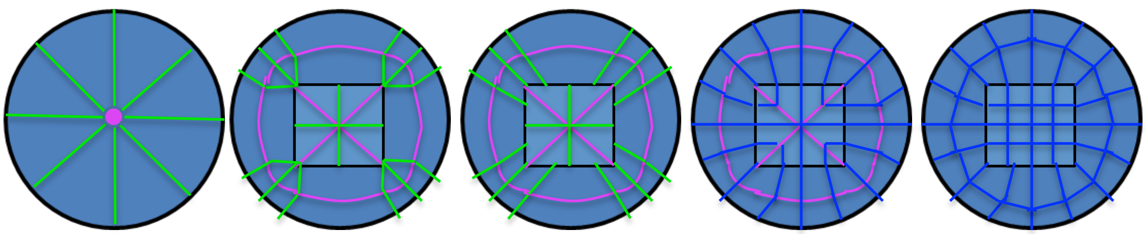


(a) 1-to-N map (b) 1-to-1 map (c) mesh on 1-to-1 map

Figure 11 1-to-N map transformed to 1-to-1 map

Two potential solutions for 1-to-N map at a concave edge/vertex are shown in Figure 11. One of the solutions is to remove the concave vertex by converting single surface into multiple convex surfaces. The second row shows another solution, which perturbs the 1-to-N map to obtain a 1-to-1 map 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 concave regions.

One of the solutions for the finite contact where the map is N-to-1 is shown in Figure 12. Figure 12(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 a multiple solids problem by introducing a core at the N-to-1 map of the MA regions. Figure 12(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 region (see Figure 12(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 12(c)). Thus the N-to-1 map is transformed completely to a 1-to-1 map.



(a)N-to-1 (b)1-to-N (c)1-to-1 (d)Tracks (e)Mesh

Figure 12 N-to-1 map transformed to 1-to-1 map

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 10) can be generated on general solids.

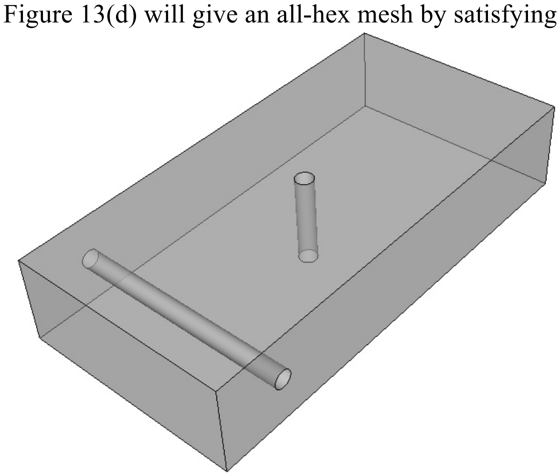
Condition 2: Mesh a Quad Cross-section 3D Track with All-Hex Elements

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

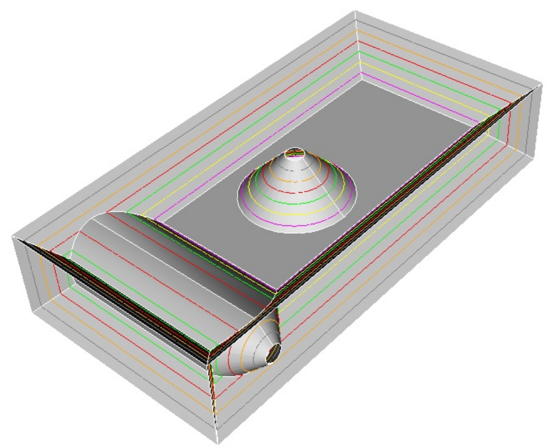
Here the author would like to point out why the current implementation discussed in Section 0 does not guarantee an all-hex mesh and how to overcome this limitation to achieve an all-hex mesh. In the current implementation, non-hex elements can arise at the MA when the hex elements are built inside the track in an advancing front manner. This is because LayTrack3D described in Section 0 has no control on the number of intervals present on the rails of the tracks. The quad mesh on the medial shown in Figure 4(d) is randomly generated using the Paving algorithm for a given mesh size without any special constraints. 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 1). 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, Case 1: all four rails must have the same intervals or Case 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.

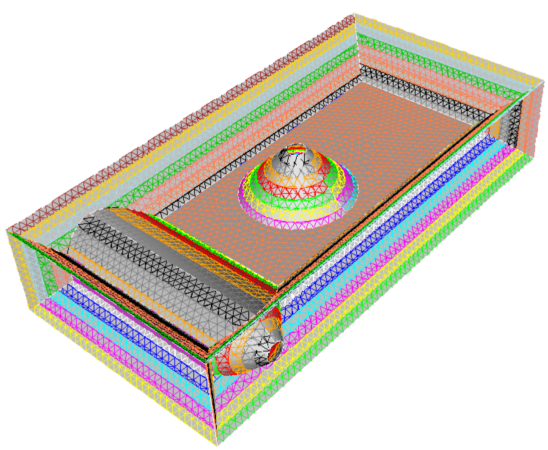
The big question is how to generate a mesh on the medial surface such that all tracks satisfy either Case 1 or Case 2? In order to satisfy Case 1 or Case 2 in every track, the isocontours of the radius function of the MA have been utilized. Figure 13(b) shows isocontours of the radius function of the MA obtained for mesh size increments. Isocontours and non-manifold MA curves split the MA into different regions/segments as shown in Figure 13(c). Figure 13(c) also shows the underlying facets of these MA segments. Figure 13(d) shows the quads on the MA that represent the quad cross section tracks in 3D. The intervals on a rail generated at a node can be easily determined by knowing the isocontour number (see Figure 14). Figure 13(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 equal 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 14. 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. Thus, the quad mesh shown in Figure 13(d) will give an all-hex mesh by satisfying either Case 1 or Case 2 in every track.



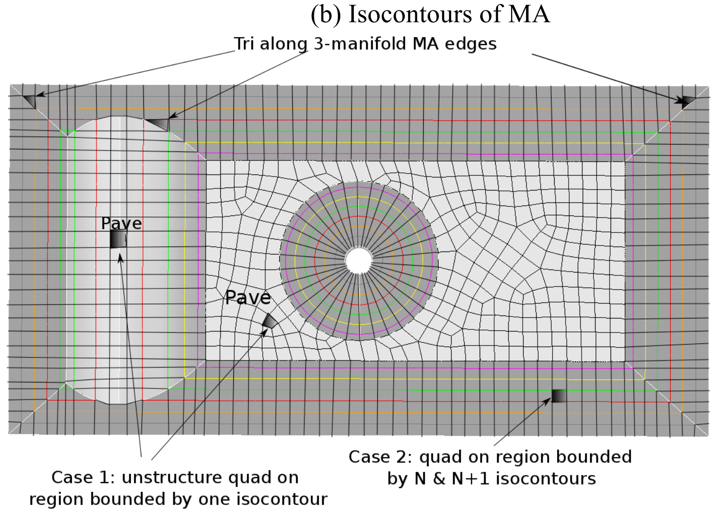
(a) Non-sweepable solid



(b) Isocontours of MA



(c) MA segments



(d) Meshing the MA using isocontours to obtain desired number of intervals on rails.

Figure 13 Meshing the MA using isocontours of radius function to achieve all-hex mesh.

In order to improve the hex quality at 3-manifold MA curves (MA curve incident on three MA surfaces), a layer of tri elements are generated all along the 3-manifold MA curve as shown in Figure 13(d). Figure 15(a) shows a 3-manifold MA curve at a convex vertex. Figure 15(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 15(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 4(d) and Figure 4(f) show that having a tri along the 3-manifold MA curve will result in a high quality hex elements at the convex corner. Figure 9 shows that having quads along the 3-manifold MA curve will often result in poor quality elements at the convex vertex. Therefore, using tri elements will give high quality hex elements that are conformal with the hex elements of the adjacent quad cross section tracks.

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 13(d) shows the unstructured quad mesh in the region bounded by one isocontour. In Figure 13(d), the regions bounded by two isocontours have a structured quad meshes. Therefore, the hex mesh originating from this quad mesh will have a structured mesh.

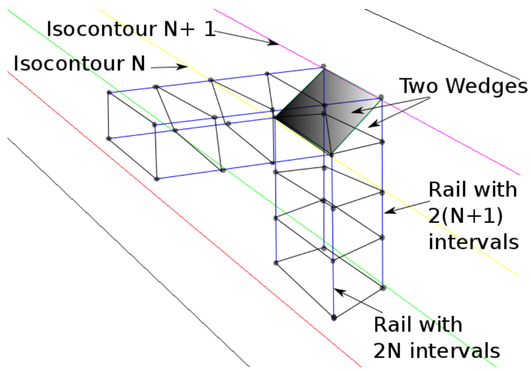
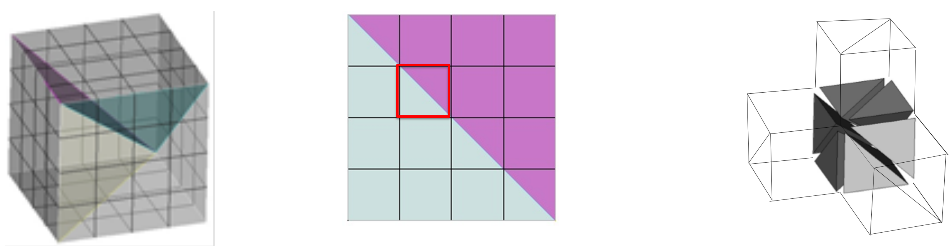


Figure 14 Case 2 forms a hex at the MA by merging two wedges



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

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

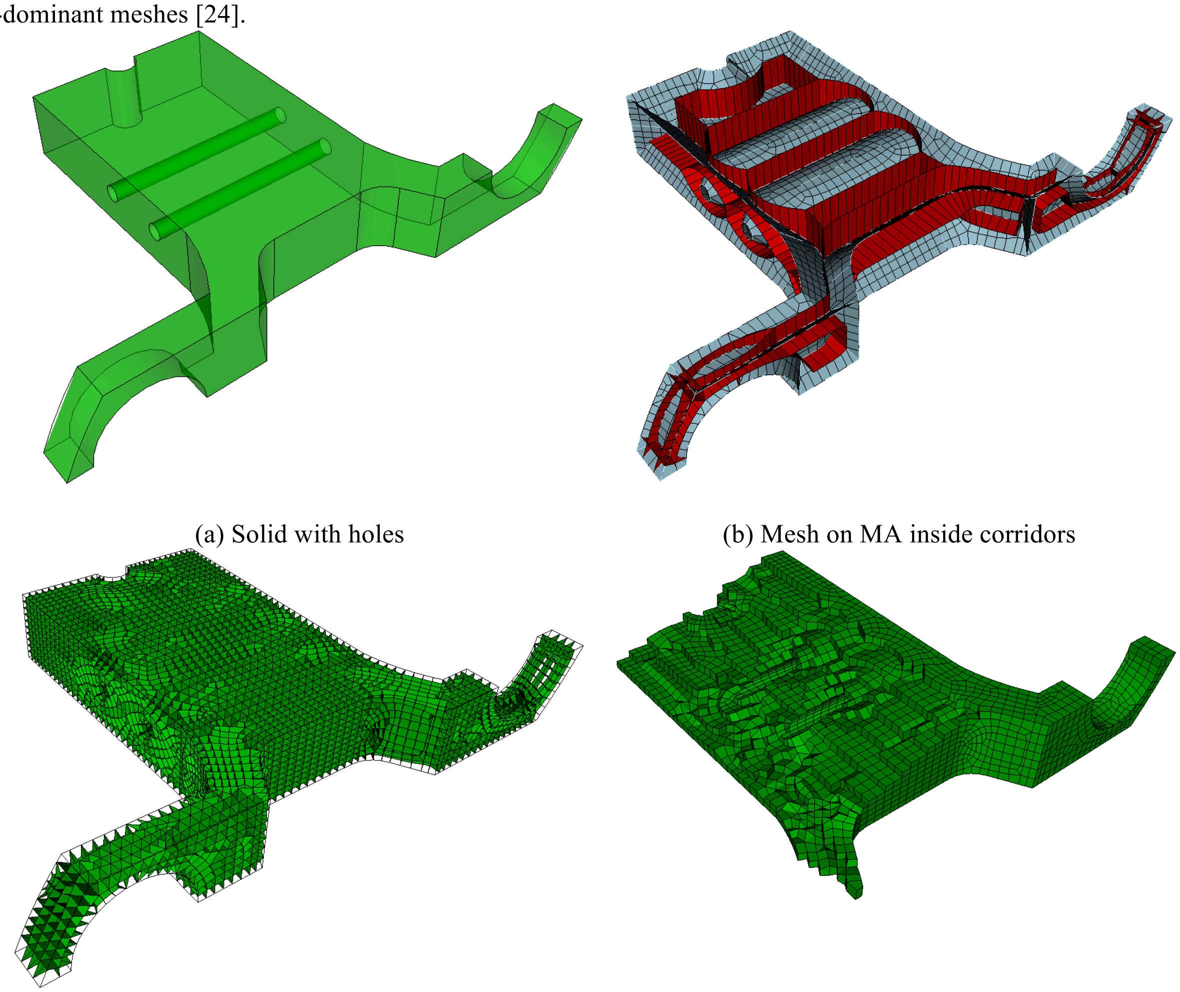
(c) Six tri-cross section tracks form an hex at MA edge

Figure 15 Case 3 forms a hex by merging 6 tets at 3-manifold medial edge

8. **Results**

LayTrack3D has been implemented in CUBIT [23] using the CADFix [24] medial object library. Note that the generation of the 3D MA itself is a challenging problem. The proposed method relies on the availability of a robust MA and does not do any cleanup of MA. 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 the user time required in methods requiring manual decomposition for hex meshing.

Currently, hex-dominant meshes can be generated by meshing the MA using the Paving algorithm [17] as shown in Figure 4. 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 fully 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 [24].

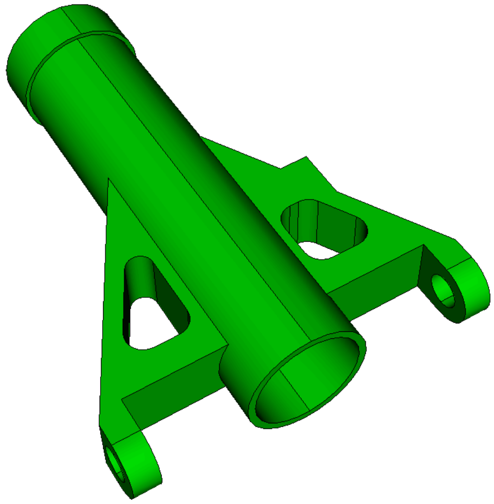


(a) Solid with holes (b) Mesh on MA inside corridors

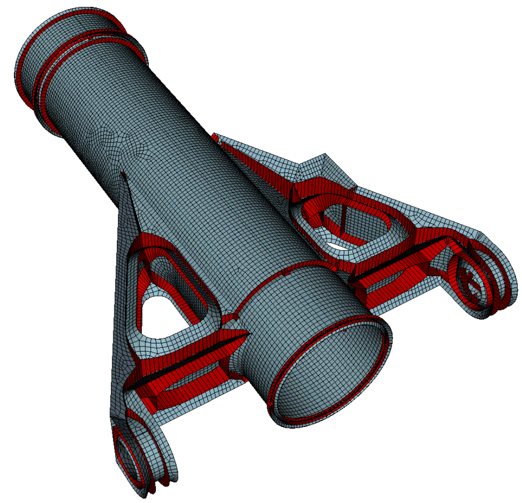
(c) Tracks in 3D (d) Mesh cross section

Figure 16 Meshing a Bracket model containing two through holes

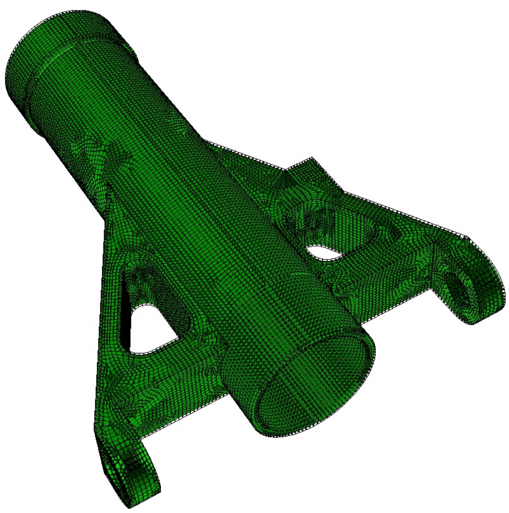
The proposed algorithm has been tested on a number of models in generating hex-dominant meshes. Figure 16 to Figure 19 show four such industrial parts containing through holes, sharp features, fillets, concave regions, and thin-thick wall cross sections. Image (b) in Figure 16 to Figure 19 show the corridors obtained by propagating rails (shown in red) from the non-manifold MA junction curves to boundary, which represent critical singularity planes. The MA is meshed using the paving algorithm with a layer of tri elements along the 3-manifold MA curves touching convex vertices. Image (c) in Figure 16 to Figure 19 shows the tracks, which look like tunnels in 3D and are much simpler to mesh than the original solid. These tracks are orthogonal to the boundary and take a turn at the MA. 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 (see image (d) in Figure 16 to Figure 19).



(a) Thin and thick wall solid



(b) Mesh on MA inside corridors

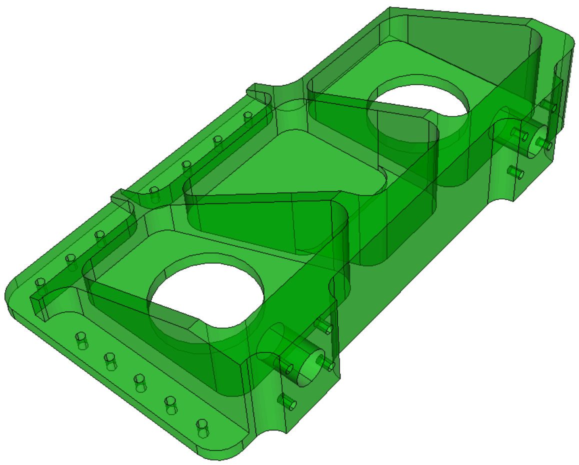


(c) Tracks in 3D

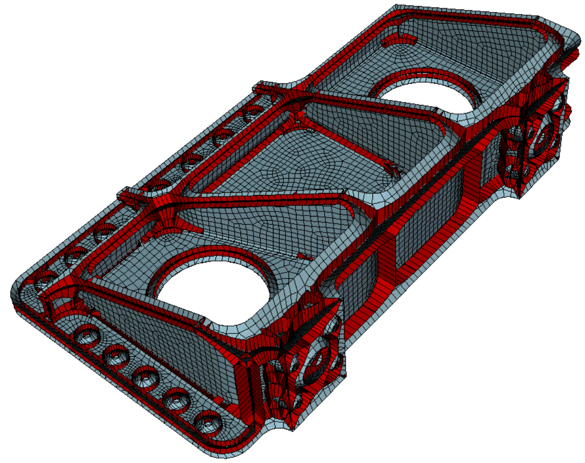
(d) Hex-dominant mesh

Figure 17 Meshing a Barrel model containing thin-wall

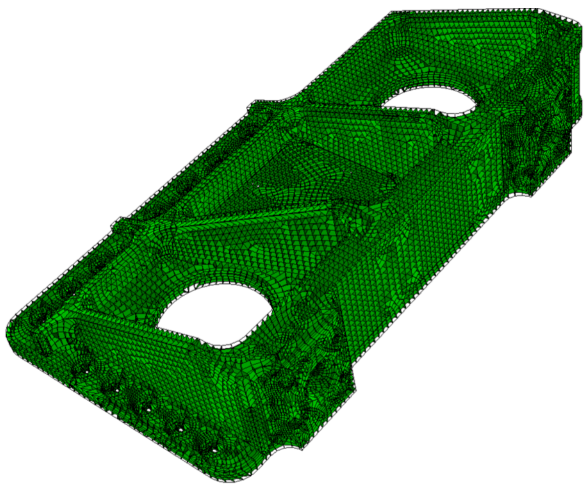
Table 1 shows the percentage of non-hex elements in the hex-dominant meshes. Most of the models tested result in having about 2% to 3% non-hex elements. These non-hex elements currently exist at the MA or at the concave edges/vertices. Currently, isocontours of the radius function are not utilized in meshing the MA. As discussed in Section 7, by using the isocontours correct intervals can be obtained on the rails to avoid non-hex elements at the MA. Also, the non-hex elements at the concave edges/vertices can be eliminated by either perturbing the map or by partitioning the concave region into convex regions.



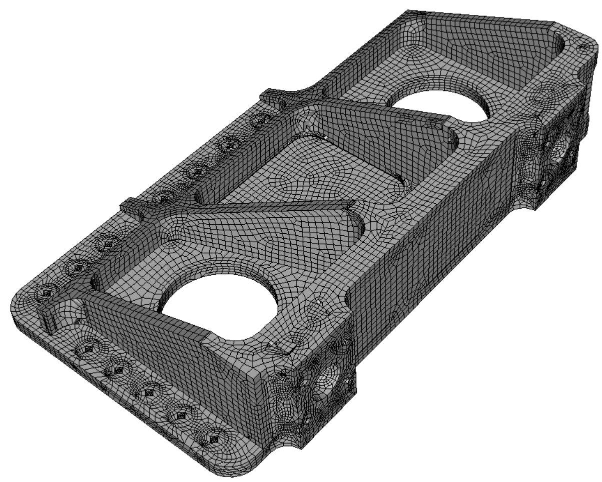
(a) Input solid in transparent mode



(b) Mesh on MA inside corridors



(c) Tracks in 3D

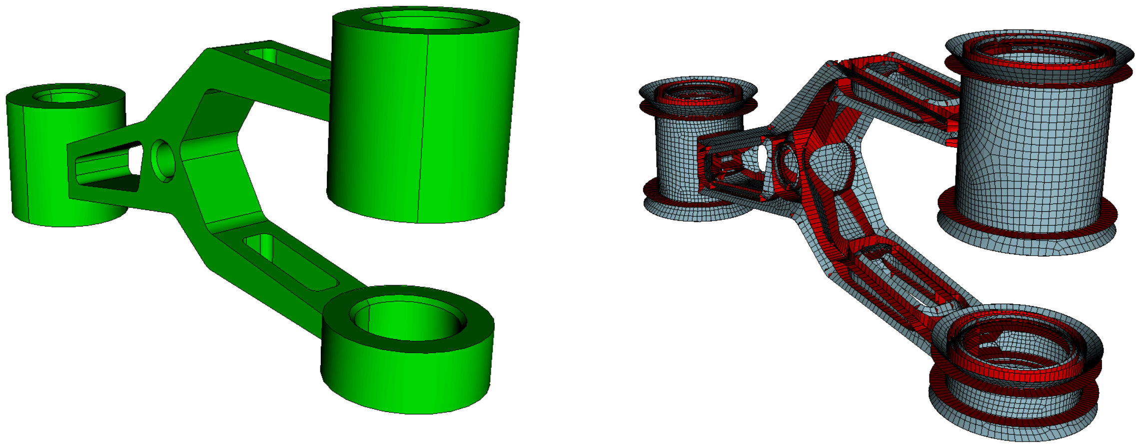


(d) Hex-dominant mesh

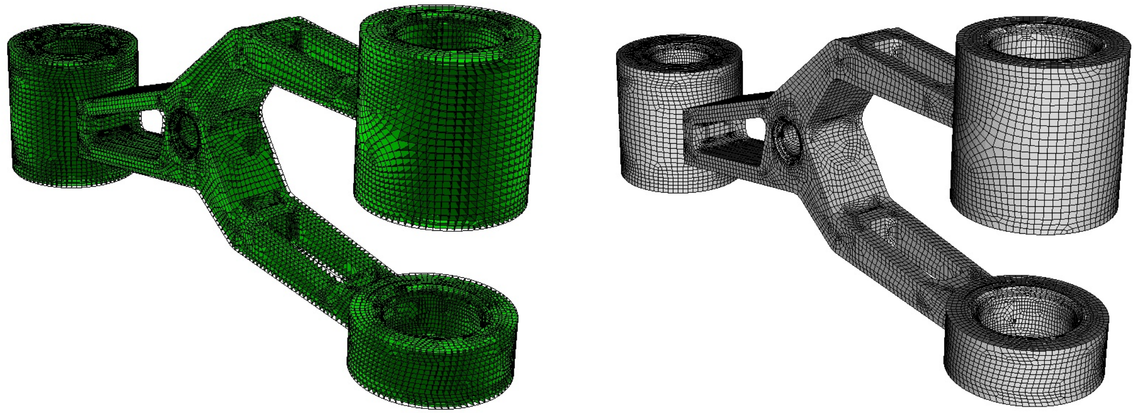
Figure 18 Meshing a Housing model with multiple holes and fillets

Table 1 Percentage of non-hex elements in the hex-dominant meshes

Model	Number of Hex Elements	Number of Non-Hex Elements	Percentage of Non-Hex Elements
Bracket	10,260	536	5.2%
Barrel	61,932	1,546	2.4%
Housing	98,507	2,873	2.9%
Pipe-joint	151,338	4,142	2.7%



(a) Input solid (b) Mesh on MA inside corridors



(c) Tracks in 3D (d) Hex-dominant mesh

Figure 19 Meshing a Pipe-Joint model

9. Conclusion

This paper presents an extension of LayTracks to generate hex-dominant meshes of general solids using the MAT. LayTracks3D combines the merits of two popular mesh generation techniques, the method of decomposition and the advancing front methods. The algorithm first decomposes any general solid into simpler sweepable like regions called corridors using the internal critical singularities, i.e., non-manifold MA junction curves and the boundary imprints. Next, the corridors are further subdivided into simpler tunnel like regions called tracks. The tracks are then meshed in an advancing front manner by laying hex elements from the boundary towards the MA without any interference checks. The algorithm generates hex-dominant 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.

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, pp 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", Engineering with Computers, Vol 30, Issue 3, pp 345-361, Dec 2012

[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, "Domain partitioning using frame fields, and its applications to quadrilateral and hexahedral meshing", Ph.D. Thesis, Universite, Pierre et Marie Curie, Dec 2013

[16] N. Kowalski, F. Ledoux, and P. Frey, "Block-Structured Hexahedral Meshes for CAD Models using 3D Frame Fields", 23rd International Meshing Roundtable, Oct, 2014

[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