Identifying and Meshing Thin Sections of 3-d Curved Domains

Luzhong Yin, Xiaojuan Luo, Mark S. Shephard

Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY 12180.
lyin@scorec.rpi.edu; xluo@scorec.rpi.edu; shephard@scorec.rpi.edu

Summary: Realization of the full benefits of variable p-version finite elements requires the careful construction of prismatic elements in thin sections. This paper presents a procedure to automatically isolate the thin sections using the points on an approximate medial surface computed by an octree-based algorithm. Using the pairs of triangles associated with medial surface points, in conjunction with adjacency, classification and distance information sets of surface triangles that are opposite face patches in thin sections are identified. Mesh modifications are then executed to match the surface triangulations on the opposite face patches such that prismatic elements can be generated without diagonal edges through the thickness directions.
Keywords: thin sections, medial surface, prismatic elements

1. Introduction

Historically, the methods used to analyze thin sections involved applying deformation assumptions to the 3-D elasticity equations allowing the problem dimensionality to be reduced [1]. The application of such methods requires a reduced dimensional domain model. Application of these methods requires the identification of the thin sections and then the application of model dimension reduction on those portions [2]. Handing the interconnection between two-dimensional reduced elements to fully three-dimensional solid elements is another source of difficulty [3].

Since the assumptions corresponding to those deformation models are equivalent to allowing only low order deformation modes in the thickness direction, an alternative is to apply full three-dimensional model discretized with p-version finite elements with low polynomial order through the thickness [4, 5]. Tetrahedral meshes cannot effectively be used to implement the appropriate low order deformation modes through the thickness due to the presence of the through the thickness diagonals. Therefore a mesh that contains a single element through the thickness without through the thickness diagonals is needed.

The automatic generation of meshes for general 3-D domains with such elements (prism of hexahedra) in the thin sections is not a straightforward process, particularly in the case where adaptive p-version finite element methods are applied that will require large curved elements of high polynomial order in the other

directions. This paper reports on the status of efforts to development mesh generation procedures aimed at producing the desired p-version curved finite elements for models containing thin sections.

Starting from a general curved solid model with a classified surface triangulation, the thin sections for the model are identified using points on an approximate medial surface computed by an octree-based algorithm. The thin sections are then meshed by prismatic elements and a general volume mesh generator can be applied to fill the remaining domains for p-version adaptive analysis. The geometric approximation required by the p-version finite elements is achieved by applying a curving procedure in [7, 8].

Section 2 presents the criteria to identify points on the medial surface for a classified surface triangulation of the model and gives an octree-based algorithm for their determination. Each of these medial surface points is associated with a pair of opposite triangles on the thin sections. Section 3 discusses the procedure that given those pairs of triangles determines any missing thin section triangles and isolates the thin sections. Section 4 considers the procedures for then meshing the thin sections.

2. Determination of Medial Surface Points and Associated Triangle Pairs

2.1 Criteria to Define Thin Section Triangle Pairs

The definition of a thin section is closely related to size of and order of the elements in the mesh. The geometric characteristic for a thin section is the dimension through the thickness is far less than the "in-plane" dimensions. We identify the thin sections using a surface triangulation of the model. The basic idea is to find pairs of triangles on "opposite model faces" that are close to each other relative to their size, thus indicating they are within a thin section.

A point on the medial surface can provide the local thickness [10, 11] by the distance to its closest boundary points and 'opposite' boundary points by relating the closest boundary points. Therefore, the concerned pair of the opposite triangles can be defined based on a medial surface point as follows.

A pair of triangles M_i^2 and $M_{i'}^2$ is candidate *thin section triangle pair* if there exist a pair of closest boundary points P_1 and P_2 from a medial surface point E_i^0, such that $P_1 \subset \overline{M}_i^2$ and $P_2 \subset \overline{M}_{i'}^2$ (over bar means closure of a triangle), where the P_1 and P_2 have following properties:

(1) The ratio of thickness (defined as the diameter of the maximum inscribed sphere associated with E_i^0) to the average size of M_1^2 and M_2^2 is smaller than a default value, for example 1/2 of the average edge length of the element.

(2) The angle formed by the outward normal to M_1^2 and M_2^2 is close to π.

The situation of the medial surface point defined by conditions of (1) and (2) is shown in Fig. 1.

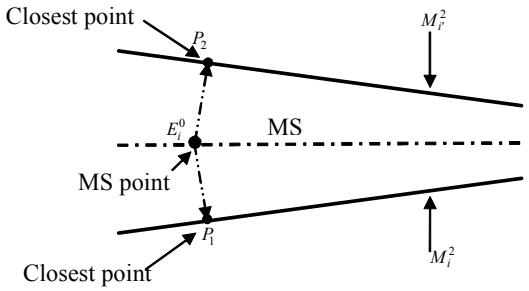


Fig. 1. A thin section triangle pair identified by a medial surface point E_i^0

A candidate thin section triangle pair is further processed to ensure that all points on their closures meet those conditions. The key step to identify the thin section triangle pairs is to calculate the points on the medial surface of the classified surface triangulation. We use octree to calculate the medial surface points with the goal of identifying most, but not all, triangles in the thin sections.

2.2 Octree-Based Algorithm to Compute the Medial Surface Points

The medial surface points are calculated for a classified surface triangulation. The classification information of the surface triangulation is used to ignore the medial surface branches of the triangulated model that do not exist in the smooth curved model. That is, the two closest points of a medial surface point on two adjacent triangles that are classified on one C^1 continuous model face will be ignored in the calculations. From the property that a closed geometrical model has a closed set of medial surfaces [12], a medial octant tracing algorithm was constructed. In this algorithm, medial octants are defined as octants that intersect medial surface as Figure 2 (a). The steps of the tracing algorithm are as follows.

- Construct octree by inserting surface mesh entities into boundary octants.
- Determine an octant with an edge that intersects the medial surface.
- Resolve all intersections of that octant edge by a traversal algorithm.
- Continue the traversal on the other edges until all intersections are resolved.
- Move to neighboring octants of the intersection points to process their other octant edge/medial surface intersections. See Fig. 2 (b).

To control the medial octant size, before calculating intersections, recursively subdivide the neighbor medial octants to be no more than one level different. Further subdivide the medial octant to the same order of the size of surface triangles that are closest to the octant vertices. Denote an octant as O_i, an octant edge as O_i^1 and its two bounding vertices as O_j^0 and O_k^0.

The goal of the algorithm is to determine the intersection between an octant edge and the medial surface. An octant edge can have multiple intersections. To determine those intersections, we employ the relationship among Voronoi regions and medial surface [10] for polyhedron. An octant edge with bounding vertices closer to two different surfaces (in different Voronoi region) that are not adjacent to one re-entrant edge or corners, intersects the medial surface. To judge the Vo-

ronoi region a vertex O_1^0 is in the closest points of O_1^0 on the boundary are used. The current octree is employed to determine the closest point information.

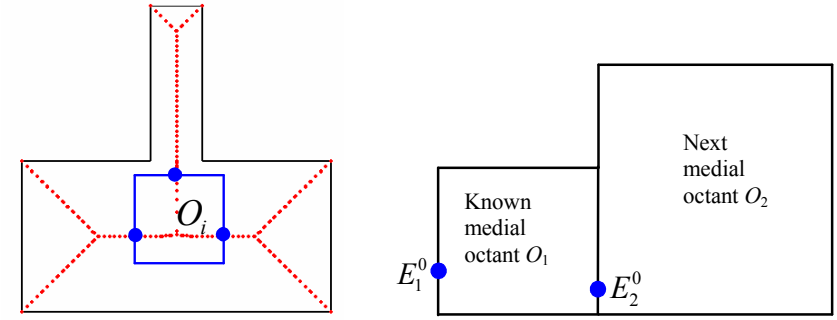


Fig. 2. (a) A situation of intersections between octant edges and medial surface (b) Move from one resolved medial octant to next neighbor medial octant to calculate the other new intersection locations.

The procedure to resolve the intersections is as follows:

Assume there is just one intersection on the edge bounded by O_1^0 and O_2^0 which can be found through solving a scalar t in the interpolation formulation $\mathbf{E}_i^0 = (1-t)\mathbf{O}_1^0 + t\mathbf{O}_2^0$ due to the equidistant condition to the boundary entities whose associated Voronoi regions O_1^0 and O_2^0 are in, where the bold letters denotes the location vectors at the corresponding vertices. After getting the assumed intersection, we request its closest points on the boundary. If multiple closest points are returned there is a single intersection. If a single closest point is returned there are multiple intersections, in which case subdivide the edge at that location and repeat until the intersections are resolved.

The efficiency of the above algorithm is demonstrating using Fig. 3, where $p_i(\bullet)$ denotes a closest point for the entity $\bullet$. Given the octant edge whose bounding vertices are O_1^0 and O_2^0, we obtain an assumed medial surface intersection point E_j^0. The closest point to E_j^0 is a single point $p(E_j^0)$, thus indicating the point is not on the medial surface. After subdivide the edge at E_j^0 obtain the correct intersection points, one on each sub-edge bounded by (O_1^0, E_j^0) and (E_j^0, O_1^0).

It is noted the tracing algorithm can start from a convex model edge whose interior dihedral angle is $w_i < \pi$, or by a medial surface point calculated on a ray in the direction of the normal to the triangle to the interior of model by the above edge intersection algorithm. Operations to calculate medial surface points are not applied to octant edges external to the model.

Fig. 4 shows a result of the above algorithm for an example. The medial octants are square boxes (Fig.4(b)) and medial surface points are shown in Fig. 4(c) by circle dots.

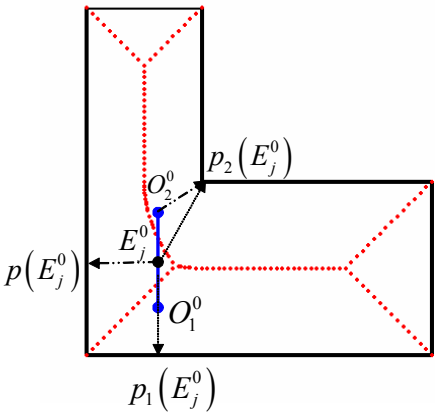
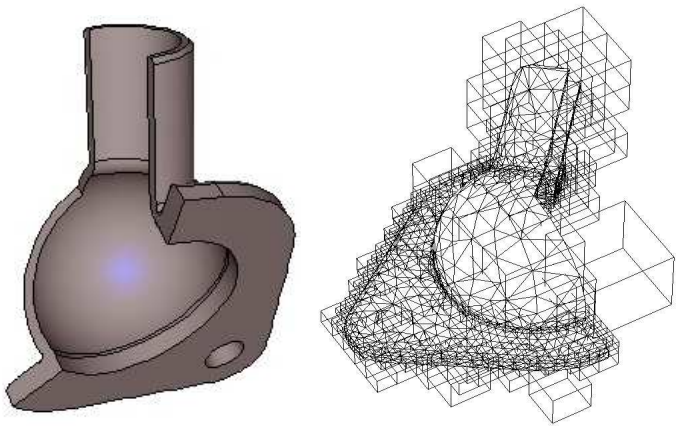
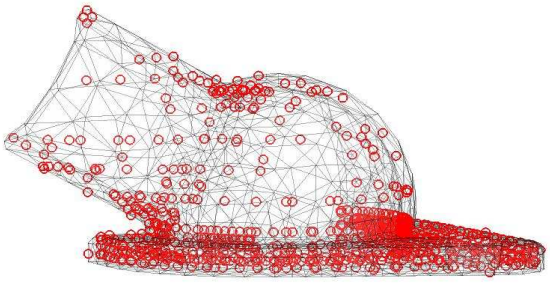


Fig. 3. Multiple intersection on the edge bounded by O_1^0 and O_2^0



(a) (b)



(c)

Fig. 4. (a) Model; (b) traversed octants; (c) medial surface points

3 Defining Thin Sections

The medial surface point calculations provide a set of an unorganized and incomplete thin section triangle pairs. We organize the thin section triangles in the sets by their classification information. After that, the missing thin section triangles are determined. The procedure has three steps:

- Collect starting triangle sets using classification information.
- Complete the triangle sets to define thin section surface patches
- Construct the loops for each thin section surface patch and match the loops on the boundary of opposite thin section patches.

3.1 Collect Starting Thin Section Triangle Sets

Given medial surface point E_i^0 , introduce

$$\left|E_i^0\right|_* = \begin{cases} 1 & \text{t hin} \\ 0 & \text{not thin} \end{cases} \tag{1}$$

to indicate whether E_i^0 define a thin section triangle pairs, where 1 indicates the medial surface point is associated with thin triangle pair and 0 means they are not part of a thin section. Denote the triangle pair as

$$\left[E_i^0\right]_* = \left\{M_k^2, M_{k'}^2\right\}. \tag{2}$$

With the above notations, a starting thin section triangle set is defined as

$$\hat{G}_j^2 = \left\{M_i^2 \middle| M_i^2 \left[\; G_j^2 \text{ and } M_i^2 \in \left[E_l^0\right]_* \text{ and } \left|E_l^0\right|_* = 1\right\}\right. \tag{3}$$

Note that each $\hat{G}_j^2$ is uniquely associated with G_j^2 . For this unique association, the identity tags of "opposite" sets for $\hat{G}_j^2$ can be recorded during the construction of $\hat{G}_j^2$. Generally, $\hat{G}_j^2$ may have one or more opposite sets denoted as $opp(\hat{G}_j^2)$. A simple example in Fig. 5 shows $\hat{G}_1^2 = \left\{M_a^2, M_b^2\right\}$, $opp(\hat{G}_1^2) = \left\{\hat{G}_3^2\right\}$; $\hat{G}_2^2 = \left\{M_c^2, M_d^2, M_e^2\right\}, opp(\hat{G}_2^2) = \left\{\hat{G}_3^2\right\}$ and $\hat{G}_3^2 = \left\{M_{a'}^2, M_{b'}^2, M_{c'}^2, M_{d'}^2, M_{e'}^2\right\}$, $opp(\hat{G}_3^2) = \left\{\hat{G}_1^2, \hat{G}_2^2\right\}$.

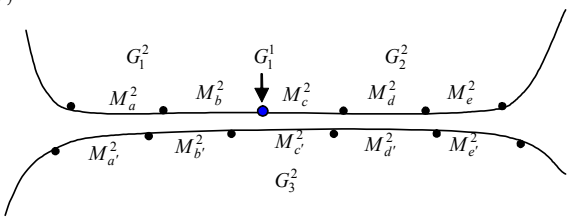


Fig. 5. An example to demonstrate the starting triangle sets

Note sets at this point may have to be later split or merge to represent a thin section surface patch.

3.2 Determining the Missing Thin Section Triangles

The majority of thin section triangles are identified by the medial surface points in the tracing algorithm but some are missed, see Figure 6 (a), where the dark shaded triangle faces are the identified thin section triangles. To determine whether a missing triangle M_e^2 on G_j^2 belongs to $\hat{G}_j^2$, local thickness h_e at M_e^2 is compared with the local thickness h_i at an edge adjacent triangle M_i^2 that is inside $\hat{G}_j^2$. If $|h_e - h_i|/h_i$ is smaller than a default value, place M_e^2 in $\hat{G}_j^2$. The local thickness h_e at M_e^2 is obtained by searching for the closest point on the triangles classified on the model faces that are known to be opposite to G_j^2. The triangle $M_{e'}^2$ that the closest point is on is defined as opposite triangle to M_e^2. Also place $M_{e'}^2$ in the set that is opposite to $\hat{G}_j^2$ if it is not there already. Note, $M_{e'}^2$ must be in the neighborhood of $M_{i'}^2$ which is opposite to M_i^2. The search is a local operation. Fig 6 shows an example before (Fig 6 (a)) and after completing the thin section triangle sets (Fig. 6(b)). Note the boundary edges of $\hat{G}_j^2$ can also be identified as those used by only one triangle in the set.

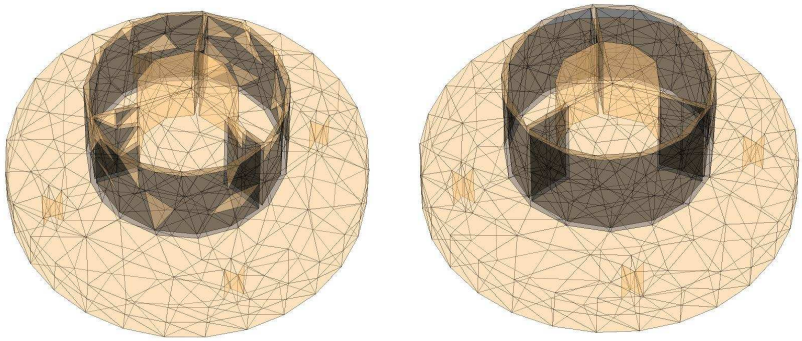


Fig. 6. (a) Thin section triangles obtained by medial surface points; (b) The completed thin section surface patches.

3.3 Construct the Boundary Loops on Thin Section Surface Patches

To complete the definition of the thin section surface patches opposite each other, the loops on the boundary of surface patches have to be matched. The process can lead to the need to split surface patches. Figure 7 (a) shows an example that has three thin section surface patches, where $\hat{G}_3^2$ is opposite to $\hat{G}_1^2$ and $\hat{G}_2^2$ with each loop on each of the sets. In this case, splitting $\hat{G}_3^2$ to form two loops on $\hat{G}_3^2$ to match the loops on $\hat{G}_1^2$ and $\hat{G}_2^2$ is needed as shown in Fig. 7 (b). Note the loops in $\hat{G}_1^2$ and $\hat{G}_2^2$ cannot be merged to form one loop since the model edge must be used in the volume mesh generation.

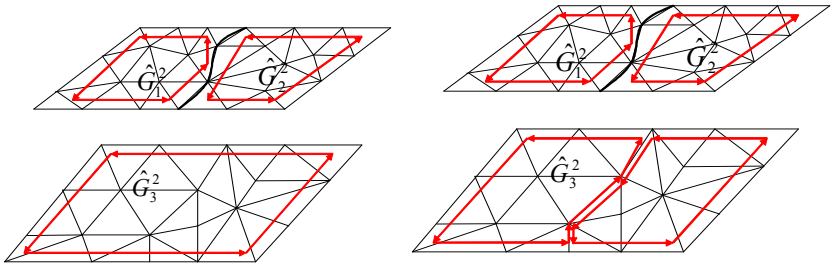


Fig. 7. (a) Loops on thin section surface patches (b) Opposite loops

The procedure to create opposite loops looks for its opposite edge $M_{j'}^1 \in \partial \hat{G}_{j'}^2$ using the opposite triangles and connectivity information.

4. Meshing Thin Sections

The thin section information obtained from Section 3 is characterized by a pair of opposite thin section surface patches with paired closed opposite loops as boundaries. To generate structured prismatic elements without long diagonal edges through the thickness the opposite triangulation sets for thin section surface patch must be topologically matched and be geometrically similar.

4.1 Overall Algorithm

The overall procedure to mesh each thin section consists of the following steps:

- Apply local mesh modifications to match the thin section boundaries
- Delete the surface triangulation of one triangle set
- Copying the remaining triangulation to its opposite model face
- Connect the matched triangulations to form prismatic elements

4.2 Boundary Matching

The procedure to match the boundaries of the triangle sets for a thin section can be divided into two continuous operations. First, apply split or collapse operations to modify the mesh topology to ensure the mesh edges in each paired opposite loops are one-to-one matched. Second, the desired target locations for the vertices in the loops are computed and local mesh modifications as in [10] are applied to incrementally move the vertices towards to the target locations.

4.2.1 Topological Matching

For each pair of opposite loops, the process begins to traverse one loop through vertex adjacency information from one selected starting mesh edge and vertex to match the topological configuration to its opposite loop. Split and/or collapse operations are used to keep a loop L_i one-to-one matching with its opposite loop $L_{i'}$. The procedure starts from a vertex with lowest dimensional classification.

For each mesh edge M_i^1 in the closed loop L_i, retrieve the attached edges $\{M_j^1\}$ in the opposite closed loop $L_{i'}$. Let

$$|M_{j'}^1|^* = \begin{cases} 1, assigned \\ 0, unassign \end{cases} \tag{4}$$

be the operation to determine whether the mesh edge $M_{j'}^1$ already associates with one mesh edge in loop L_i. If only one mesh edge $M_{j'}^1$ is attached and $|M_{j'}^1|^* = 0$, let $M_{j'}^1$ associate with M_i^1 and update $|M_{j'}^1|^* = 1$ and continue to next mesh edge M_{i+1}^1. Otherwise, either split or collapse is applied to produce or eliminate mesh edges to maintain the one-to-one matching. As examples, Figure 8 shows that a split operation is applied on mesh edge $M_{1'}^1$ to produce one more edge $M_{2'}^1$ to obtain the matched pair edges $(M_1^1, M_{1'}^1)$ and $(M_2^1, M_{2'}^1)$ then update $|M_{2'}^1|^* = 1$. Considering the mesh vertices will be moved in the next step, the split operation applied currently does not snap the new introduced vertices to the model boundary.

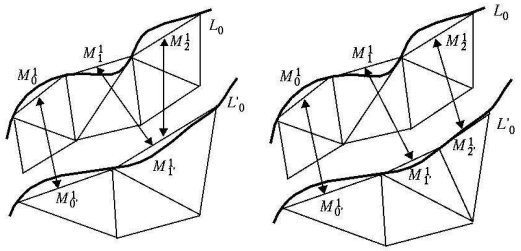


Fig. 8. Split operation to assign one-to-one match for mesh edge M_2^1

4.2.2 Target Location for Vertices in the Opposite Model Face

To achieve the geometrical similarity between the two triangle sets for a thin section, each vertex M_i^0 in one triangle set need to compute an appropriate location for its matched vertex $M_{i'}^0$ on the opposite model face. The target location for $M_{i'}^0$ is obtained by first computing the closest point P_i^0 on the mesh face $M_{i'}^2$ classified on the opposite model face to M_i^0, and then, projecting P_i^0 to the model face by the model parameters of P_i^0 determined by the interpolation of P_i^0 on the triangle face $M_{i'}^2$.

4.2.3 Incrementally Move Vertices on the Thin Section Boundary

The movement of the mesh vertices on the thin section boundary loop $L_{i'}$ can cause the surface mesh to become invalid. Fig 9 shows an example where two tri-

angle faces marked as shaded in Fig 9(b) become invalid because of moving vertices M_2^0 and M_3^0 to their target locations. This problem is avoided by applying the following procedure [7]:

- Put all of the vertices with attached target locations into a list
- Traverse the list and deal with one vertex at each step
- If the vertex moves to the target location without causing any problem, move it and remove it from the list. Otherwise, apply local mesh modifications to correct the invalid elements. Remove the vertex from the list (See reference [7]).

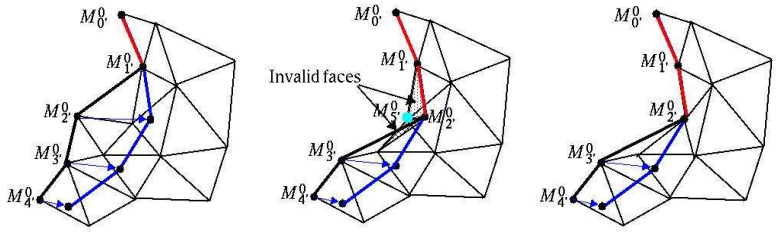


Fig. 9. (a) Move M_0^0 and M_1^0. (b) Move M_2^0, invalid faces marked as shaded. (c) Collapse M_5^0 to M_2^0.

4.3 Surface Triangulation Matching

The surface triangulation matching between the two triangle sets for a thin section is achieved through the triangulation deletion of one triangle set and the copying of the remaining triangle set to the opposite model face. The key technique is the location computation of the copied vertices on the opposite model face. This is accomplished by the interpolation strategy discussed in Section 4.2

4.4 Volume Mesh Generation

With the topologically and geometrically matched surface triangulation for thin sections, the volume mesh generation procedure constructs prismatic elements by directly connecting each paired triangles classified on the two opposite model faces. Because the generalized volume mesh generator requires the exposure mesh faces to be triangles, one layer of pyramid elements are added neighboring to the interior quadrilateral faces of the structured prismatic elements.

4.5 Examples

Two example models with thin sections are given in Figure 10. The first row shows the input surface triangulation for the two models. The second row shows just the isolated opposite thin section surface patches after loop construction and loop matching. As can be seen from the figures the two surface meshes are not yet matched. The bottom row shows the final meshes where the thin section meshes have been matched, the prismatic thin section elements created, volume mesh completed and the mesh properly curved to the domain boundary.

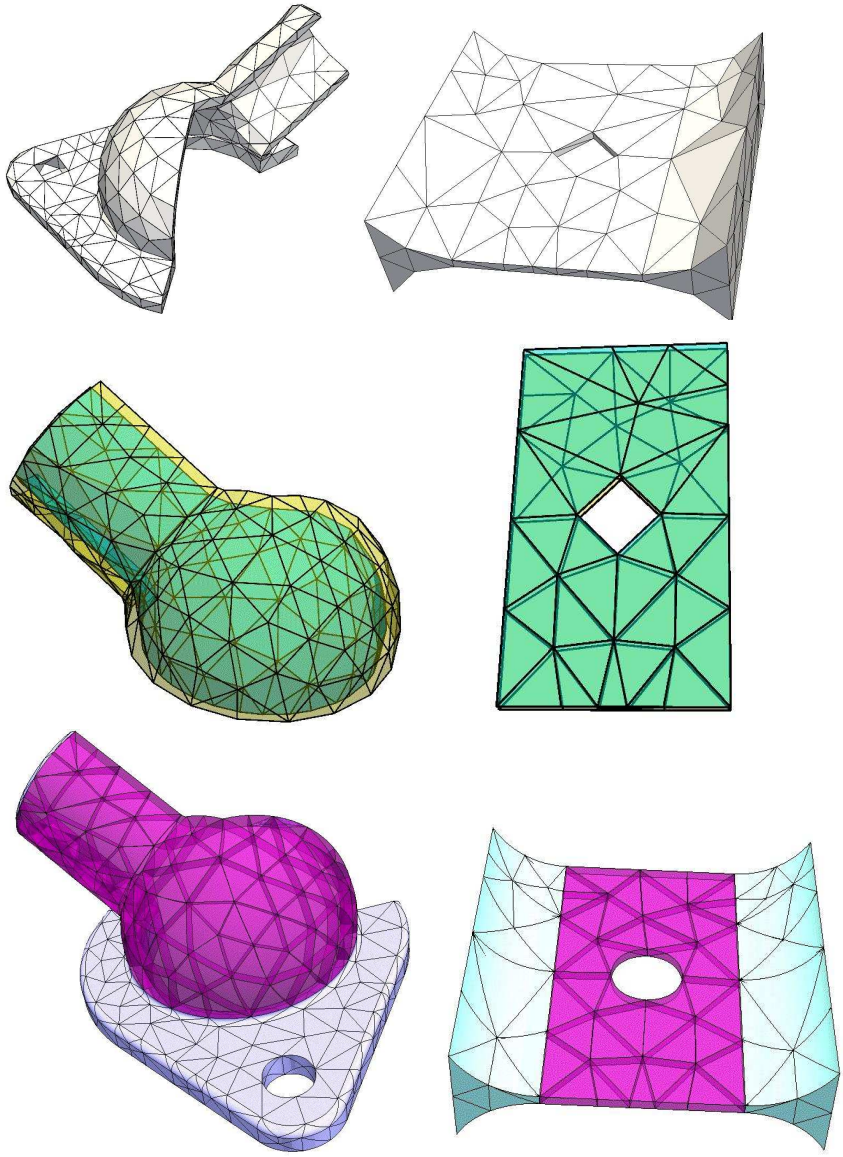


Figure 10: Examples meshes for models with thin sections: Surface triangulations (top). Thin sections (middle). Curved meshes with prismatic elements (bottom).

5. Closing Remarks

The paper presented a procedure to automatically isolate and mesh thin sections of 3-D solid models with prismatic elements for directional p-version finite element analysis. Key ingredients of the procedure are:

- Construction of an octant tracing algorithm to calculate a limited number of medial surface points to define the thin section triangle pairs,
- A strategy to organize the thin section triangle pairs to define thin section face patches that are opposite to each other,
- A procedure to generate prismatic volume mesh by copying one side of surface mesh to the other side and connecting the corresponding opposite vertices.

References

1. J.N. Reddy, Theory and analysis of elastic plates. 1999, Taylor & Francies.
2. R.J. Donaghy, C.G Armstrong, M.A. Price (2000) Dimensional Reduction of surface models for analysis. Eng. with Computers 16: 24-35.
3. RL Actis, SP Engelstad, BA Szabo (2002) Computational requirements in design and design certification, Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, pp. 1379-1389.
4. Duster, H. Broker, E. Rank (2001) The p-version of the finite element method for three dimensional curved thin walled structures, Int. J. Numer. Methods Engrg, 52: 673-703.
5. C.A. Duarte, I. Babuska (2002) Mesh-independent p-orthotropic enrichment using the generalized finite element method, Int. J. Numer. Methods Engrg, 55: 1477-1492.
6. S. Dey, M.S. Shephard, J. E. Flaherty (1997) Geometry representation issues associated with p-version finite element computation, Comput. Methods. Appl. Mechanics. Engrg., 150: 39-55.
7. X. Li, M.S. Shephard, M.W. Beall (2003) Accounting for curved domains in mesh adaptation, Int. J. Numer. Methods Engrg, 58: 247-276.
8. X.J. Luo, M.S. Shephard, R.M. Obara, R. Nastasia, M.W. Beall (2004) "Automatic p-version mesh generation for curved domains", Engrg. Comput. 20: 265-285.
9. C.K. Lee, Q.X. Xu (2005) A new automatic adaptive 3D solid mesh generation scheme for thin-walled structures. Inter. J. Numer. Meth. Engng 62: 1519-1558.
10. N. M. Patrikalakis, H. N. Gursoy (1990) Shape interrogation by medial axis transform, ASME Advances in Design Automation, 16th design automation conf., Chicago, DE-vol 23, 77-88.
11. P. Sampl (2001) Medial axis construction in three dimensions and its application to mesh generation. Engineering with Computers 17: 234-248.
12. A. Lieutier (2004) Any open bounded subset of r^n has the same homotopy type as its medial axis, Computer-Aided Design 36: 1029-1046.