

Tetrahedral Mesh Generation With Multiple Elements Through The Thickness

Rao Garimella, Mark S. Shephard
Scientific Computation Research Center,
Rensselaer Polytechnic Institute, Troy NY 12180

ABSTRACT

The problem of automatically generating unstructured tetrahedral meshes having a user specified number of elements in the thickness direction without over-refinement in the other directions is addressed in this paper. The procedures generate meshes with multiple elements through the thickness by anisotropically enriching an initial mesh, using local mesh modification procedures, wherever the mesh does not have a sufficient number of elements through the thickness. An algorithm for detecting portions of the mesh with insufficient elements through the thickness is also given. The procedure to generate multiple elements through the thickness is completely automatic with the only user input being the minimum number of elements required through the thickness and the geometric model. The initial mesh is automatically generated by Finite Octree [9] and multiple elements are created through the thickness starting from this mesh.

INTRODUCTION

Finite element analysis of domains with thin sections and strong gradients through the thickness using lower polynomial order elements requires meshes with multiple elements through the thickness. Isotropic refinement of meshes to obtain multiple elements in the thickness direction results in a large increase in the number of elements and consequently, considerably higher computational cost. An alternative solution is to anisotropically refine the mesh such that the resulting elements are stretched in perpendicular directions to the thickness direction. Such a mesh generation procedure must be capable of dealing with domains which are not uniformly thin (such as models shown in Figure 1) and be able to produce a user specified number of elements through any thickness.

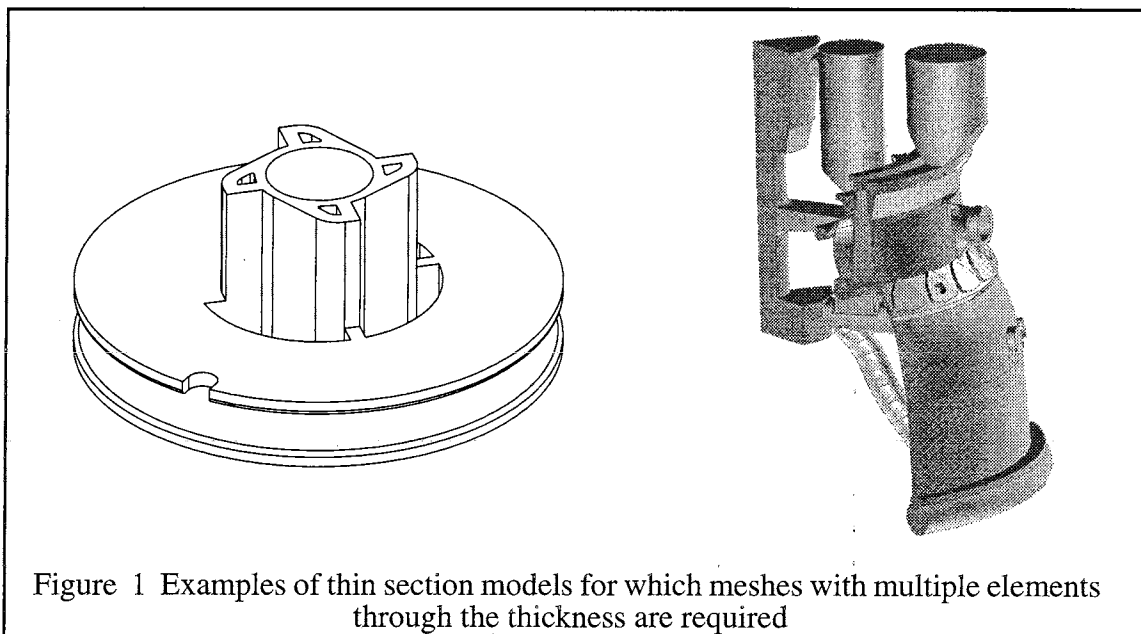


Figure 1 Examples of thin section models for which meshes with multiple elements through the thickness are required

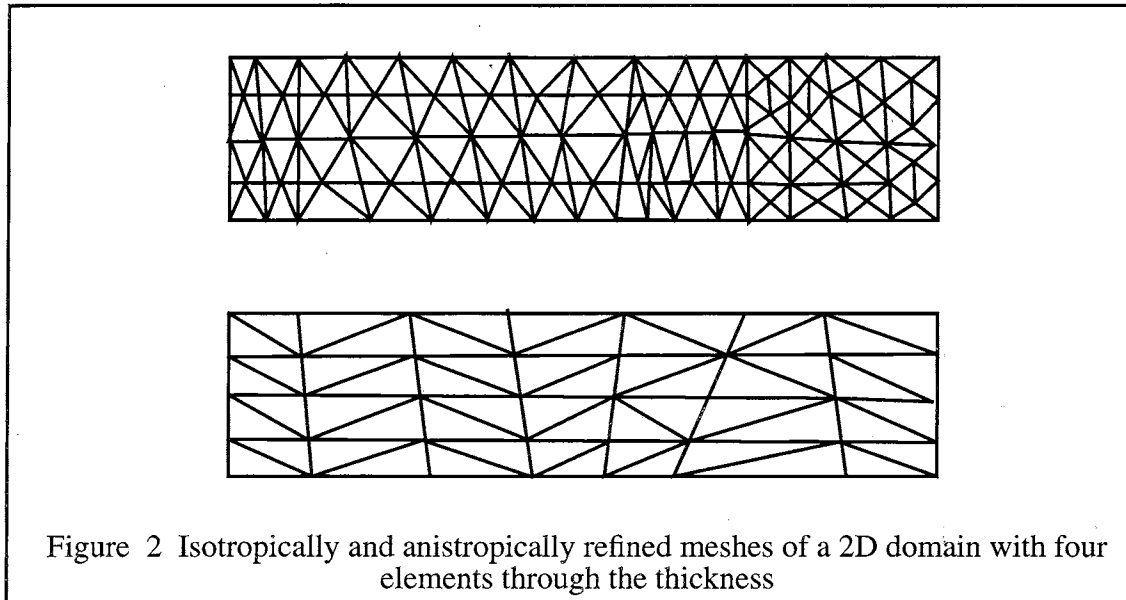


Figure 2 Isotropically and anisotropically refined meshes of a 2D domain with four elements through the thickness

Until now, the main thrust for automatic anisotropic mesh generation has come from computational fluid dynamics needs [1]. Many of the proposed techniques for anisotropic mesh generation create anisotropy in the mesh by transforming an isotropic mesh using a transformation tensor [3][4]. Typically, this transformation is constructed from error estimates in an adaptive analysis procedure. The difference between the above procedures lies in the variables used for the error estimates, construction of the transformation tensor and methods used for enriching the mesh. Another group of anisotropic mesh generation methods has concentrated on the generation of boundary layer meshes on flow boundaries, utilizing knowledge of the model geometry and topology [5][6][7]. The last category of anisotropic mesh generators is for triangular meshes of general surfaces.

This paper presents an alternate method designed to automatically generate tetrahedral meshes with multiple elements through the thickness of general domains. The method is based on enrichment of an initial mesh where there are less than a user specified number of elements in the thickness direction. It is designed to work in conjunction with automatic isotropic tetrahedral mesh generators to create multiple elements through the thickness for completely general geometric models. The initial meshes for the procedure are generated by Finite Octree [9] but they can be from any tetrahedral mesh generator that can provide the necessary input [2].

The paper presents a definition of what constitutes an insufficient number of elements locally through the thickness of a model. A method for detecting such portions of the mesh is described based on identifying shortest *edge paths* in the mesh between mesh vertices classified on locally opposite model faces. The mesh is considered to be locally deficient in the thickness direction if the path has less than a required number of edges.

The creation of the required number of elements through the thickness is done through a series of tetrahedral mesh modification procedures, in particular, edge splitting and edge swapping [3][8]. The necessary number of edges are added between pairs of opposite vertices by splitting path edges. Parts of the mesh affected by edge splitting are further processed to improve alignment of the newly created mesh edges along and perpendicular to the thickness direction. The creation of multiple elements through the thickness and the realignment of edges is done for one model face at a time.

NOTATION

${}_v T_i^d$	Topological entity i from model v of dimension d , $d = 0$ for a vertex, $d = 1$ for an edge, $d = 2$ for a face and $d = 3$ for a region
$\partial({}_v T_i^d)$	Boundary of topological entity, ${}_v T_i^d$
$\overline{{}_v T_i^d}$	Closure of topological entity, ${}_v T_i^d$, defined as $({}_v T_i^d \cup \partial({}_v T_i^d))$
\sqsubset	Classification symbol used to indicate association of one or more entities of a mesh, m , with an entity in the geometric model, g

DEFINITION OF "THIN" SECTIONS

Definition: An ordered set of mesh edges between two mesh vertices is defined as an *edge path*. An edge path with the least number of edges among all edge paths between a pair of vertices is called a *shortest edge path*.

Definition: A mesh is considered to be *locally deficient in the thickness direction* if the shortest edge path between mesh vertices on opposite model faces has less than a user requested number of edges. Such an edge path is referred to as a *deficient path*.

Figure 3 shows examples of locally deficient meshes in two types of domains. In Figure 3a, the domain has varying thickness and the mesh is uniform. Assuming that three elements are desired through the thickness, the mesh is locally deficient in the thickness direction in indicated portions of the domain. In Figure 3b, the domain is of uniform thickness but due to non-uniform refinement of the mesh, only some parts of the domain are considered to be locally deficient.

Definition: A mesh vertex, ${}_m T_j^0 \sqsubset \overline{{}_g T_n^2}$, is defined as an *opposite vertex* of another mesh vertex, ${}_m T_i^0 \sqsubset \overline{{}_g T_l^2}$, with respect to model face ${}_g T_l^2$ if

1. it is the closest to ${}_m T_i^0$ among all ${}_m T_k^0 \sqsubset \overline{{}_g T_n^2}$
2. all edges of the shortest edge path are classified on one model entity except $\overline{{}_g T_l^2}$

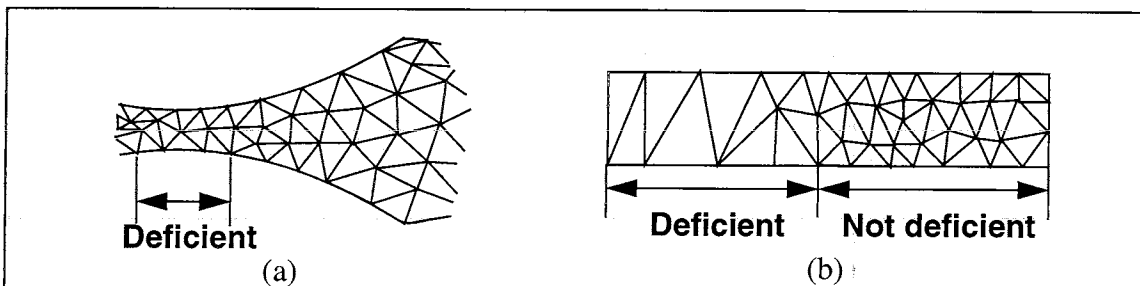
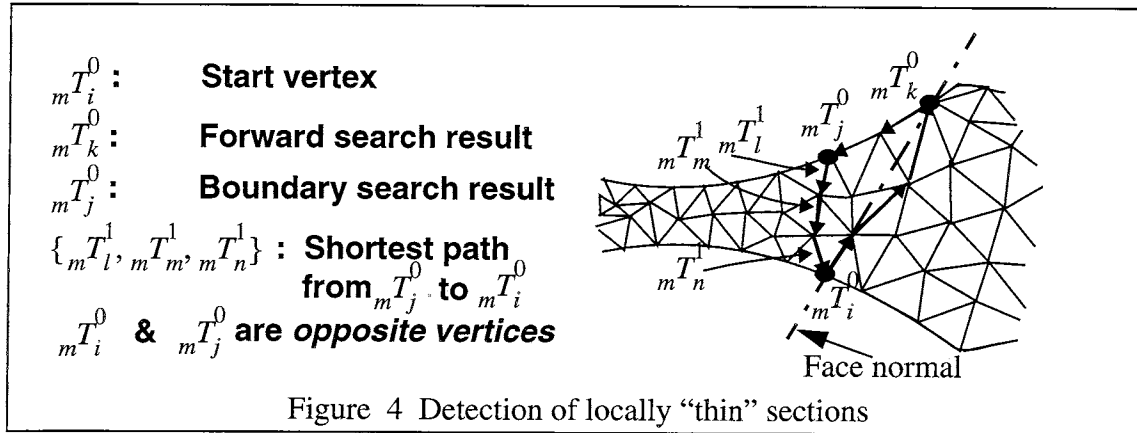


Figure 3 Examples of locally thin sections (3 elements requested through the thickness)



DETERMINATION OF OPPOSITE MESH VERTICES

The determination of the opposite vertex of a boundary mesh vertex, $mT_i^0 \in gT_l^2$, is a 3 step procedure (refer to Figure 4) consisting of:

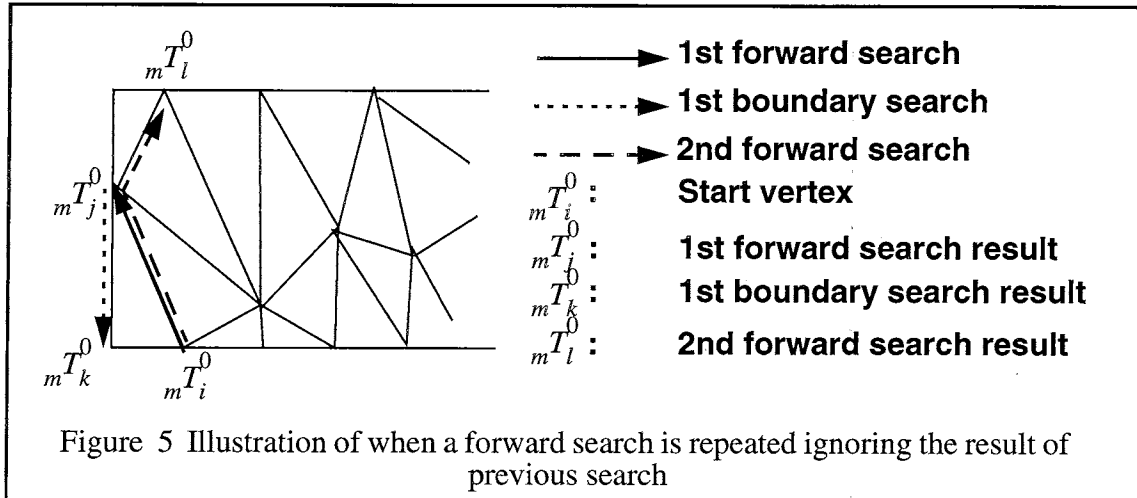
1. Forward search

The forward search assumes that the thickness direction at a mesh vertex mT_i^0 with respect to a model face gT_l^2 to be the normal to gT_l^2 at mT_i^0 . Under this assumption, the forward search attempts to find a potential opposite vertex, mT_k^0 , approximately in the direction of the normal and classified on the closure of a model face. Starting from mT_i^0 the forward search proceeds from one mesh vertex to another along mesh edges until a potential opposite vertex is reached. At a given vertex in the forward search, the mesh edge of the vertex that is best aligned with the normal direction is chosen for traveling to the next mesh vertex.

The forward search ends when a potential opposite mesh vertex is found or $(N_t - 1)$ edges have been traversed without reaching the closure of a model face (in which case at least N_t edges are present in the assumed thickness direction). If the forward search ends on a model edge or vertex, the model face whose normal at the opposite vertex is best opposed to the normal used in the forward search is taken as the opposite model face. On the other hand if the search terminates because of the second criterion, the vertex is labelled as not having an opposite vertex. As the forward search is ended when $(N_t - 1)$ edges have been traversed, its computational complexity is $O(N_t)$.

2. Boundary search

The boundary search is a discrete closest point search on the opposite model face, $gT_n^2 \neq gT_l^2$, with respect to mT_i^0 . The search starts at the vertex, mT_k^0 , found in the forward search. From the current vertex, the search goes to an adjacent (edge connected) vertex which is closest to mT_i^0 . The procedure is repeated until no adjacent vertex is closer to mT_i^0 than the current vertex, $mT_j^0 \in gT_n^2$ or N_t edges have been traversed. If the boundary search ends on a vertex classified on the boundary of the original model face, the forward search is repeated ignoring that vertex. The boundary search is then repeated with the new result of the forward search (see Figure 5). Since the boundary search has a limit on the number of iterations (N_t) , the computational cost of the boundary search is $O(N_t)$.



3. Reverse search

The reverse search is a greedy shortest path algorithm between the opposite vertex and the original vertex. The reverse search begins with mT_i^0 as the target and mT_j^0 as the current vertex. From the current vertex, the search goes to an adjacent vertex which minimizes the distance to the target vertex. This is repeated until the target vertex or a local minimum¹ is reached. If a local minimum is reached, the search is repeated going this time from the original vertex to the opposite vertex. If the shortest path still cannot be found, mT_i^0 is assumed to have no opposite vertex.

Since the boundary search is subject to a limit on the number of iterations, and the steps of the reverse search and the forward search are similar, the computational complexity of the reverse search may be assumed to be on the order of the forward search. Therefore, the computational cost of the reverse search can be taken as $O(N_t)$.

Thus, the total cost of finding an opposite vertex for any boundary mesh vertex may be considered to be $O(N_t)$ or a constant, since N_t is a constant for a given problem. If there are N_b boundary mesh vertices, the total cost of finding their opposite vertices is $O(N_b)$.

The algorithm to determine opposite vertices is designed to efficiently find opposite vertices for creation of multiple elements through the thickness. Although the algorithm is not guaranteed to always find the absolute best opposite vertex, it has been found to do quite well in finding the best or nearly the best opposite vertex.

CREATION OF MULTIPLE ELEMENTS

The introduction of the required number of elements in the mesh through the thickness is done by splitting edges of paths that have fewer edges than necessary (Figure 6). The process of splitting an edge in 3D (Figure 7) consists of the following steps [3][8]:

1. Add new vertex to edge being split.
2. Split each face connected to the original edge by adding an edge between the new vertex and the vertex of the face opposite the original edge.

1. A local minimum is reached when no adjacent vertex is closer to the target vertex than the current vertex.

3. Split each region connected to the original edge by creating an interior face between the new vertex and the vertices of the region opposite the original edge.
4. Update the classification of all newly created entities. New entities inherit classification from the appropriate original entities.

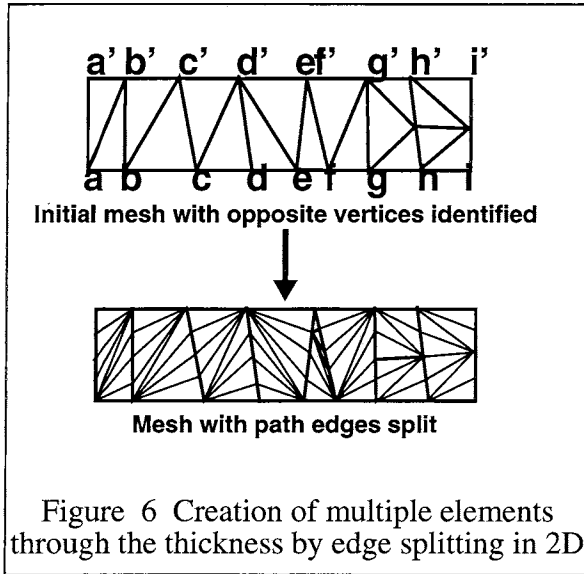


Figure 6 Creation of multiple elements through the thickness by edge splitting in 2D

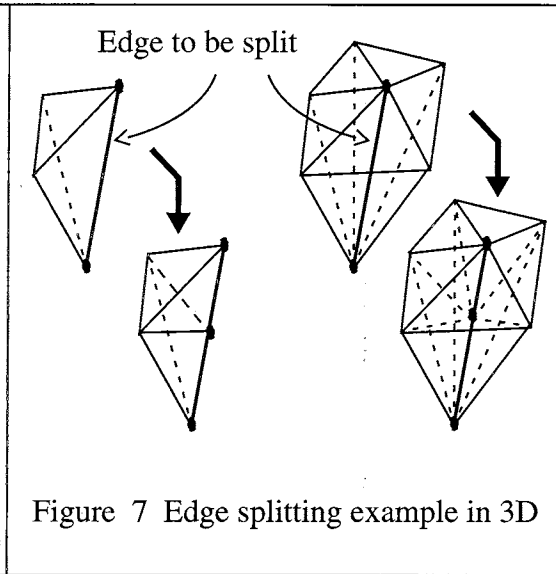


Figure 7 Edge splitting example in 3D

The edges of a path to be split are picked in decreasing order of their lengths. If there are fewer edges in the path than vertices to be added, edges may be split more than once. Edge split points are determined by bisection and subsequent snapping of the calculated point to the boundary, if the edge is a boundary edge. If snapping to the boundary results in the new regions being invalid, the edge is not split.

Since splitting of edges changes the paths determined in the initial opposite vertex search, the path information is updated as its edges are being split. This is more efficient than redoing the opposite vertex search.

REALIGNMENT OF EDGES

From Figure 6 it can be seen that splitting of path edges

1. creates new deficient paths through the thickness,
2. creates large number of connections at some vertices, and
3. results in large face angles (in 2D) and dihedral angles (in 3D).

Therefore, realignment of the diagonal mesh edges is necessary along and perpendicular to the thickness direction as shown in Figure 8. The realignment of edges is accomplished through the use of edge swapping [3][8]. The process of edge swapping is equivalent to retriangulation of the polyhedron formed by deletion of all regions connected to the edge such that the new triangulation does not contain any new points and does not contain the edge being swapped (Figure 9).

Definition: A mesh edge, ${}_m T_j^1 \sqsubset_g \overline{T_n^2}$, is defined as an *opposite edge* of another mesh edge, ${}_m T_i^1 \sqsubset_g \overline{T_i^2}$, if each of the vertices of ${}_m T_j^1$ are opposite to one of the vertices of ${}_m T_i^1$.

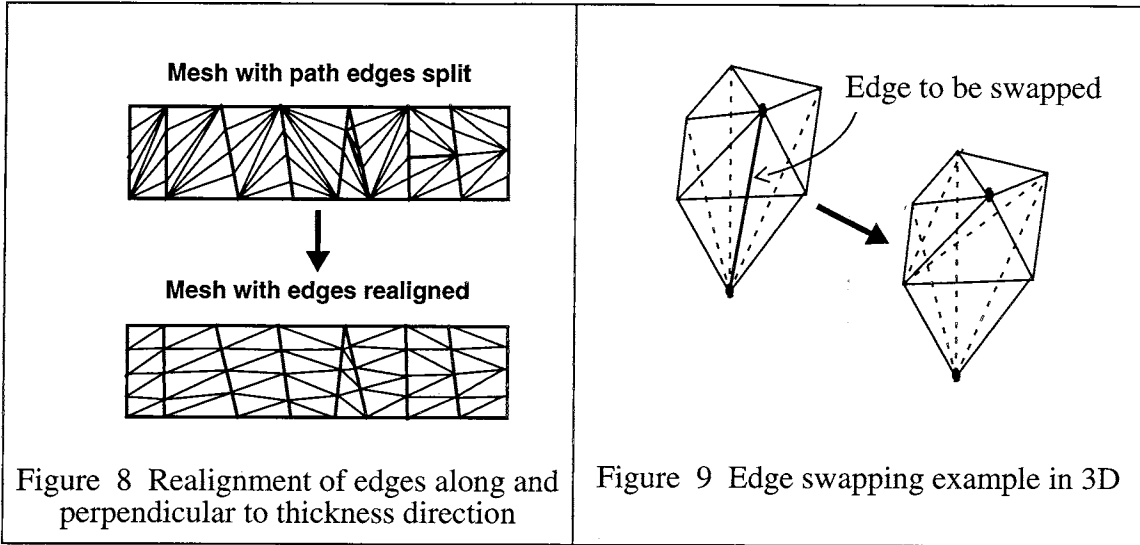


Figure 8 Realignment of edges along and perpendicular to thickness direction

Figure 9 Edge swapping example in 3D

Definition: A mesh face, $mT_j^2 \sqsubset_g T_n^2$, is defined as an *opposite face* of another mesh face, $mT_i^2 \sqsubset_g T_l^2$, if each of the edges of mT_j^2 is opposite to one of the edges of mT_i^2 .

The knowledge the special mesh topology created by splitting is used in the identification of mesh edges to swap and the sequence in which to swap them. The identification of edges to swap is facilitated by abstracting portions of the mesh between opposite faces as wedges (triangular prisms). The lateral faces of such wedges are abstracted as triangulated quadrilaterals. For example, shown in Figure 10 is a portion of a mesh in which multiple elements have been introduced through the thickness. In the figure, mT_1^2 and mT_2^2 are opposite faces of mT_4^2 and mT_3^2 respectively. One of the 5 abstracted quadrilaterals shown is formed by the vertex set $\{mT_2^0, mT_6^0, mT_8^0, mT_4^0\}$. The two wedges shown are formed by the vertex sets $\{mT_1^0, mT_2^0, mT_3^0, mT_5^0, mT_6^0, mT_7^0\}$ and $\{mT_2^0, mT_1^0, mT_4^0, mT_6^0, mT_5^0, mT_8^0\}$.

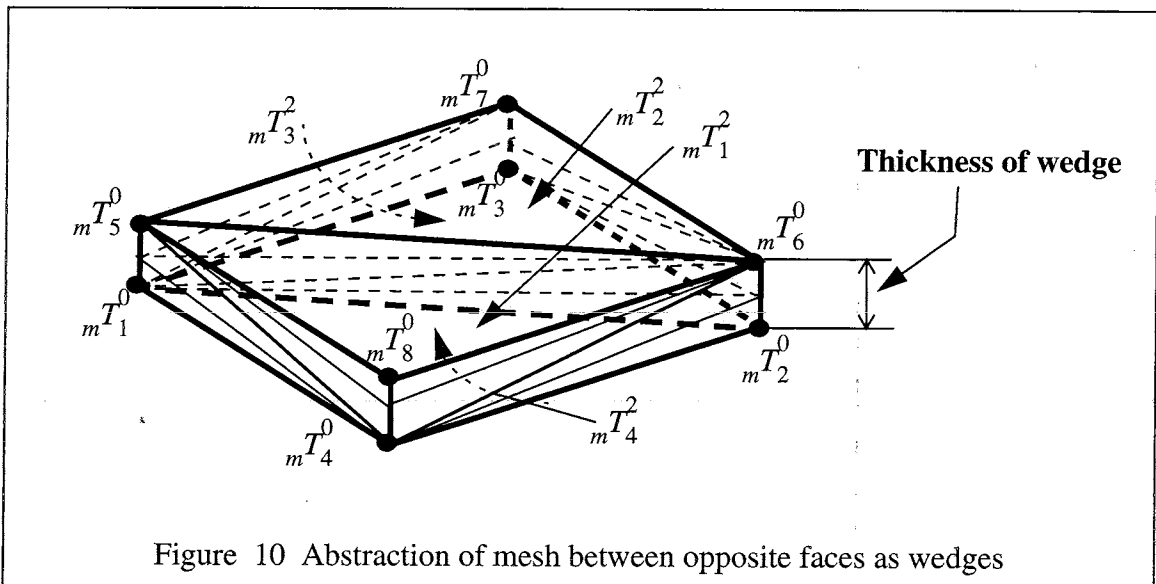


Figure 10 Abstraction of mesh between opposite faces as wedges

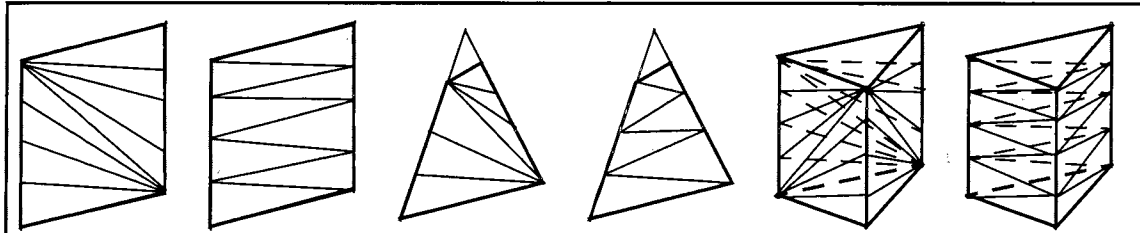


Figure 11 Diagonal and zigzag configurations for quadrilaterals, triangles and wedges

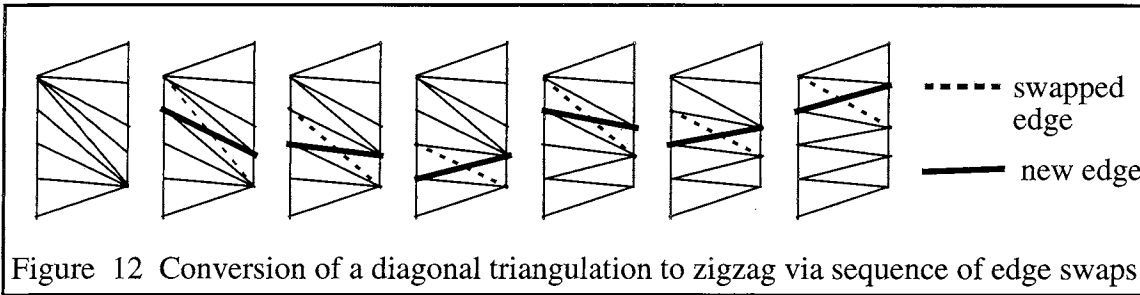


Figure 12 Conversion of a diagonal triangulation to zigzag via sequence of edge swaps

The triangulation on a quadrilateral resulting from edge splitting and the triangulation desired after realignment is shown in Figure 11. The former triangulation is called a *diagonal* triangulation while the latter a *zigzag* triangulation. A wedge has a diagonal or a zigzag configuration if all of its quadrilateral faces have a diagonal or a zigzag triangulation respectively (Figure 11). Note that the sides of the quadrilaterals in the thickness direction are edge paths between opposite vertices.

With this abstraction defined, the process of realigning edges along and perpendicular to the thickness direction can be viewed as a conversion of all triangulated quadrilaterals in the mesh from a diagonal to a zigzag configuration. This allows the edge realignment process to be driven largely by topological considerations and is therefore more efficient than using geometric criteria.

The conversion of the diagonal triangulation on each quadrilateral to zigzag is done in a templated sequence of swaps illustrated in Figure 12. This sequence can be easily generalized for a quadrilateral with any number of edges through its thickness.

If wedge topology is not present in a portion of the mesh, it is still possible to introduce multiple elements through the thickness and realign mesh edges to eliminate deficient paths through the thickness. For general mesh topology through the thickness, edges may be realigned using geometric criteria such as the thickness direction in the local neighborhood. A special situation dealt with in the algorithm is two adjacent mesh vertices having the same opposite mesh vertices (Figure 11). In this case, the edge in each path connected to the opposite vertex is ignored and the remaining edges used for the quadrilateral abstraction. Swapping of edges then proceeds as usual with this topology.

The description of the realignment procedure until now assumed that multiple elements were introduced into a mesh which had only one element through the thickness. If the initial mesh had more than one element through the thickness and additional elements were introduced, then the multiple wedges, stacked on top of each other, would exist between opposite faces. The procedures for recognition of quadrilaterals and wedges must account for this. Once the individual wedges through the thickness have been identified, the swapping sequence can be applied to the quadrilateral faces of the wedge as before.

When converting a quadrilateral triangulation from diagonal to zigzag, care must be

taken that wedges on either side of the quadrilateral are not forced into a configuration for which a tetrahedronization does not exist. The following rules may be stated about the validity of wedge configurations with 2 elements through the thickness:

Rule 1: If the directions¹ of triangulations of all 3 quadrilaterals of a wedge are the same, then the wedge cannot be tetrahedronized, regardless of the type of the triangulations.

Rule 2: If 2 of the triangulations are diagonal, their directions must be opposite for the wedge to have a valid tetrahedronization. The direction of triangulation of the third quadrilateral is immaterial.

Rule 3: If 2 of the triangulations are zigzag, at least their directions must be the same for the wedge to have a valid tetrahedronization. In addition, if the third triangulation is zigzag, its direction must not violate *Rule 1*; if it is diagonal, its direction is immaterial.

If the number of edges through the thickness of the wedge is more than 2, then no valid tetrahedronization is possible with 1 diagonal and 2 zigzag triangulations (i.e. *Rule 3* no longer holds). *Rule 1* and *Rule 2* are still valid for this case.

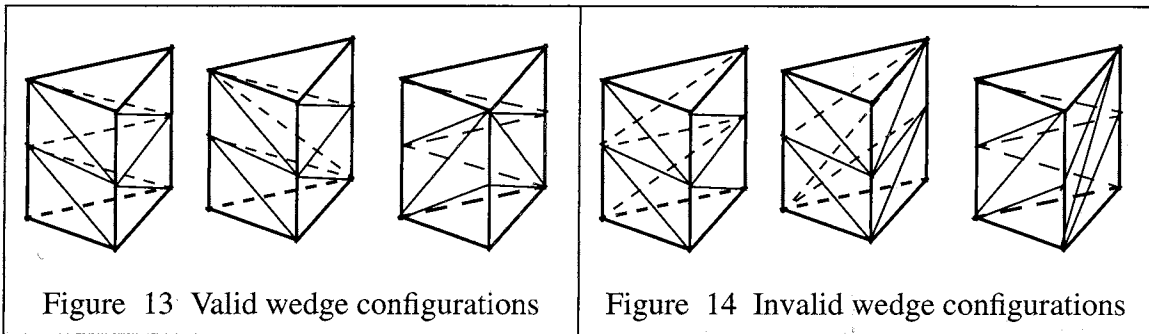


Figure 13 Valid wedge configurations

Figure 14 Invalid wedge configurations

The invalidity of all wedge configurations with two zigzag and one diagonal triangulation for wedges with more than two elements through their thickness places an important restriction on the mesh enrichment process. This restriction is that multiple elements must be introduced iteratively with edges through the thickness being split only once before the realignment procedure is applied to the modified mesh. Therefore, if an initial mesh has one element through the thickness everywhere and N_t elements have been requested through the thickness, the splitting and realignment process has to be performed $\log_2 N_t$ times rounded off to the next integer. To lift this restriction, the local nature of the diagonal to zigzag conversion process must be sacrificed and propagated out into the mesh.

If a quadrilateral triangulation could not be converted to zigzag due to one of its adjoining wedges becoming invalid, it is revisited later to account for the possibility that other quadrilaterals in the local neighborhood may have been changed to zigzag allowing for successful conversion. Still, in their current form, the realignment procedures may be prevented from converting all quadrilateral triangulations to zigzag by the invalidity of certain wedge configurations, even with two edges through the thickness. Work is in progress to allow conversion of any quadrilateral by modifying the triangulation of other quadrilaterals in the neighborhood. Work is also in progress to delete the elements forming a wedge and tetrahedronizing the wedge in its new configuration using templates, a more efficient process than swapping successively.

1. The direction of triangulation of a quadrilateral is defined with respect to a wedge using the quadrilateral. It is determined by the orientation of the diagonal edges of a triangulation when looking at a quadrilateral such that the wedge is behind the quadrilateral.

PRE- AND POST-PROCESSING

The success of the realignment procedures to identify wedge configurations partly depends on the initial mesh having matching faces through the thickness. Since a mesh obtained from an unstructured mesh generator is not guaranteed to have this topology, pre-processing procedures are necessary to obtain this form. These procedures currently rely on edge swapping to introduce structure in the mesh through the thickness, but use of edge splitting and collapsing is also proposed. From a geometric point of view, it is desirable to align opposite vertices along the average normal direction of the opposite model faces to improve the quality of elements generated by the main procedures. Node repositioning with iterations in parametric space of surfaces is proposed for this purpose.

A post processing step to improve the quality of the mesh is applied after the generation of multiple elements. This procedure is a generalized mesh optimization procedure which relies on local mesh modification to improve large dihedral angles. These procedures are prevented from collapsing edges which have been introduced into the thickness to eliminate deficient paths or from swapping edges which have been realigned.

RESULTS

Figure 15 shows the initial mesh and a mesh with 4 elements through the thickness for a model with thin sections. The close-up pictures show the interior of a portion of the initial and final meshes. It can be seen from the figure, that the final mesh has the required number of elements through the thickness nearly everywhere in the domain. Also, the swapping procedures did a good job of creating zigzag triangulations through the thin sections thereby eliminating most deficient paths through the thickness.

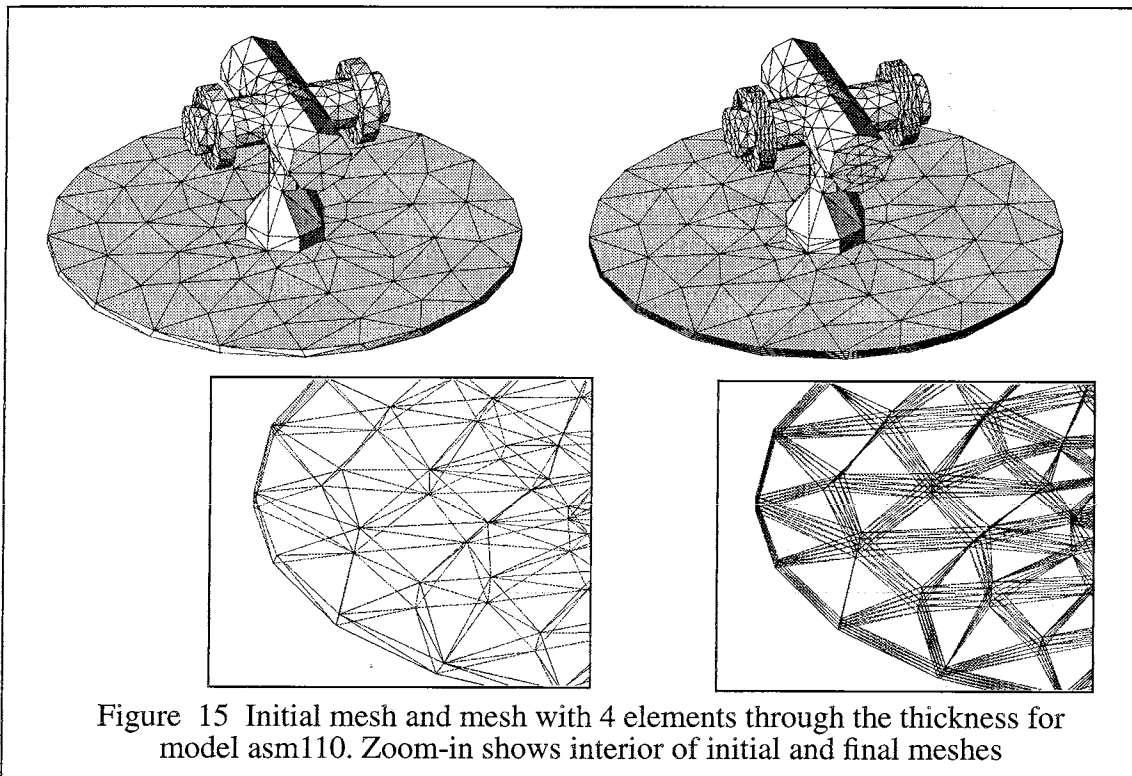


Figure 15 Initial mesh and mesh with 4 elements through the thickness for model asm110. Zoom-in shows interior of initial and final meshes

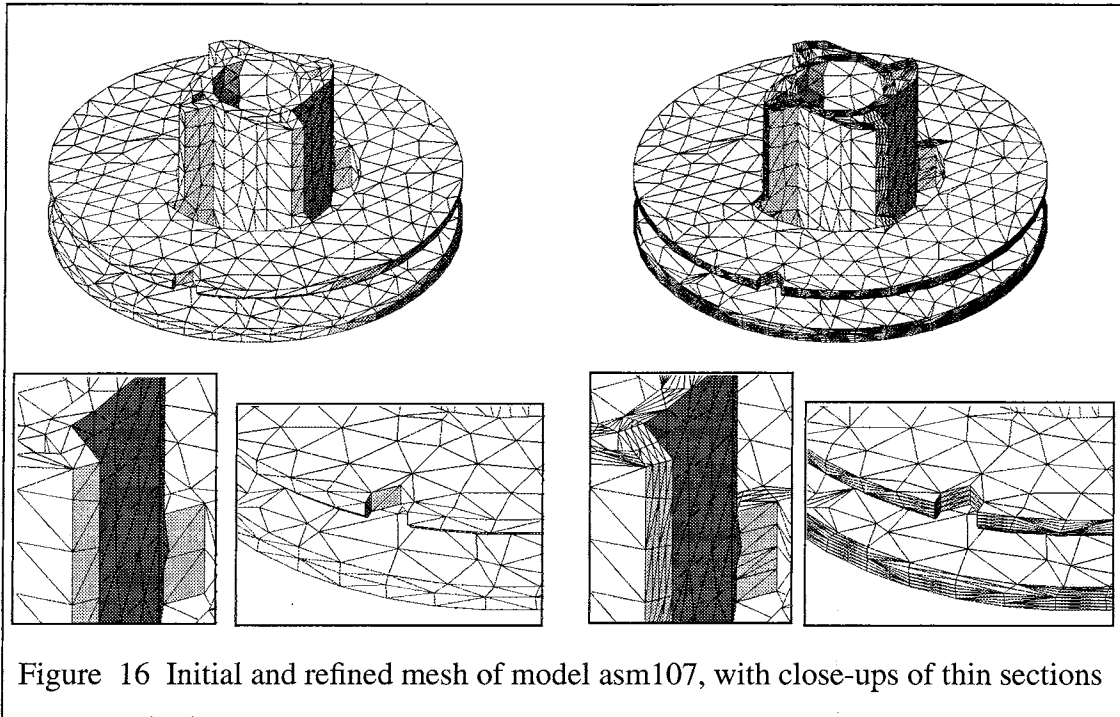


Figure 16 Initial and refined mesh of model asm107, with close-ups of thin sections

Figure 16 shows the initial and final meshes for another model with thin sections. As before, the locally deficient portions of the initial mesh have been identified very well and multiple edges created in paths through the thickness. Although the realignment procedures were also successful in converting a large number of the triangulations to the desired form, there are still some edges that could not be realigned.

Figure 17 shows a mesh with multiple elements through the thickness for a model with a number of thin sections and solid portions interacting in a complex manner. The model represents a sector of a combustor casing with gating attached for casting. Though the model is not of a real part, its complexity is comparable to the real parts developed by the same user. It can be seen that the procedures deal with this general domain quite well, although the performance of the realignment procedures needs to be improved.

CONCLUSIONS

This paper presented a method for anisotropically refining meshes to generate a user specified number of elements through the thickness. The method consisted mainly of identifying opposite vertices for mesh vertices on each model face, introducing the required number of edges between opposite vertices by edge splitting, and then swapping mesh edges to eliminate new deficient paths created by the splitting. A mechanism was proposed for realigning edges more efficiently and in a locally optimum manner by abstracting groups of elements through the thickness as wedges, whenever such a structure exists in the mesh. Using this mechanism, it was seen that the current method was constrained in the number of elements added through the thickness at a time and in the realignment of edges by the invalidity of some wedge triangulations.

Results were presented showing meshes for three complex models with multiple elements through the thickness. It was seen that the procedures to identify locally deficient parts of meshes and refine them in the thickness direction worked well for all the models.

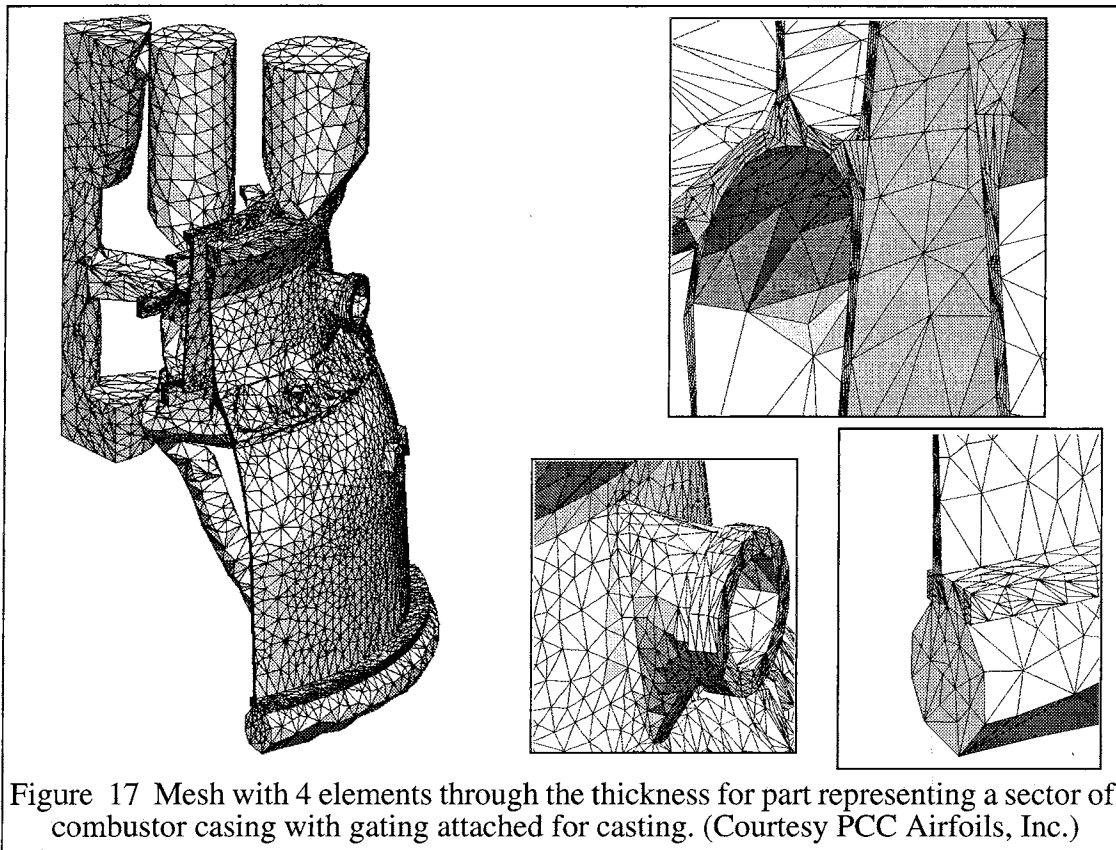


Figure 17 Mesh with 4 elements through the thickness for part representing a sector of combustor casing with gating attached for casting. (Courtesy PCC Airfoils, Inc.)

The realignment procedures also did a good job of converting diagonal triangulations to zigzag but were not successful all the time.

Future work with this method will focus largely on the issue of eliminating all deficient paths through the thickness in the mesh. Enhancements in progress include templated tetrahedronization instead of sequential edge swapping for conversion of wedges, enhancement of pre-processing algorithm and improvement of quality of mesh.

ACKNOWLEDGEMENTS

This work was supported by the ARPA Investment Casting Cooperative Arrangement.

REFERENCES

- 1 Simpson, R.B., "Anisotropic Mesh Transformations and Optimal Error Control," *Applied Numerical Mathematics* v 14, pp. 183-198, 1994.
- 2 Beall, M.W., *SCOREC Mesh Database, Version 2.3, Users Guide*, Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY 12180, 1994.
- 3 de l'Isle, E.B., George, P.L., "Optimization of Tetrahedral Meshes," INRIA, Domaine de Volceau, Rocquencourt, BP 105, Le Chesnay, France, 1993.
- 4 Mavriplis, D.J., "Adaptive Mesh Generation for Viscous Flows using Delaunay Triangulation," *Journal of Computational Physics* v 90, pp. 271-291, 1990.

- 5 Hassan, O., et.al., "Mesh Generation and Adaptivity for the Solution of Compressible Viscous High Speed Flows," *Int. J. Num. Meth. Engng.* v 38 n 7, pp. 1123-1148, 1995.
- 6 Kallinderis, Y., Khawaja, A., McMorris, H., "Hybrid Prismatic/Tetrahedral Grid Generation for Complex Geometries," *33rd Aerospace Sciences Meeting and Exhibit*, Reno, NV, Jan 95.
- 7 Lohner, R., "Matching Semi-Structured and Unstructured Grids for Navier-Stokes Calculations," *AIAA paper 93-3348-CP*, 1993.
- 8 de Cougny, H.L., Shephard, M.S., "Local Modification Tools for Adaptive Mesh Enrichment and Their Parallelization," *Unpublished report*, SCOREC, Rensselaer Polytechnic Institute, Troy, NY, 1995.
- 9 Shephard, M.S., Georges, M.K., "Automatic Three-Dimensional Mesh Generation by the Finite Octree Technique" *Int. J. Num. Meth. Engng.* v 32 n:4, pp. 709-749, 1991.
- 10 Shape Data Limited, Parker's House, 46, Regent Street, Cambridge CB2 1DB, England. *PARASOLID v6.0 Programming Reference Manual*, August 1994.
- 11 Weiler, K.J., "The Radial-Edge Structure: A Topological Representation for Non-Manifold Geometric Boundary Representations," in *Geometric Modeling for CAD Applications*, ed. Wozny, M.J., et. al., pp. 3-36, North Holland, 1988.