Boundary Layer Meshing for Viscous Flows in Complex Domains

Rao V. Garimella and Mark S. Shephard
Scientific Computation Research Center
Rensselaer Polytechnic Institute, Troy NY 12180.
E-mail:garimell@scorec.rpi.edu, shephard@scorec.rpi.edu

Abstract.

High reynolds number flow simulations exhibit strong gradients normal to walls and across shear layers requiring much finer resolution of the solution in some directions compared to others. To keep mesh sizes manageable for such problems, meshes with highly anisotropic elements are necessary. In this paper, a method for generating boundary layer meshes for arbitrarily complex non-manifold geometric domains is discussed. A popular strategy for generation of boundary layers meshes is generalized to deal with non-manifold model topology as well as to generate better shaped elements. Results on complex configurations are presented to demonstrate the capabilities of the mesh generator.

1 Introduction

High Reynolds numbers flow simulations exhibit strong gradients near walls and in free shear layers requiring fine resolution of the solution normal to the boundary layer and across the free shear layers. Refining a finite element mesh isotropically to capture strong gradients in some directions results in excessive refinement along the other directions. In the interest of keeping the mesh size manageable anisotropic elements are needed in the critical regions.

The requirements on a mesh generator capable of producing such meshes are severe. Some of the characteristics of anisotropic meshes capable of capturing the solution in high Reynolds number flows are (i) elements with aspect ratios of 1000 to 100000 or more, (ii) elements of high quality, created by careful choice of node positions to control dihedral angles, and (iii) smooth gradation into the isotropic mesh.

A popular strategy for generating boundary layer meshes is the Advancing Layers Method (also called advancing normals method) [1, 2, 3, 4, 5]. Other techniques for generating anisotropic meshes can be found in [6, 7, 8, 9, 10, 11, 12, 13, 14]. The advancing layers method starts from a triangulation of the surfaces from which the boundary layer mesh is grown. From each surface node, nodes are generated along a single direction. These nodes form the basis for constructing layers of prisms on top of each surface triangle. The prisms are used directly as elements, or are subdivided into three tetrahedra. Care is taken to prevent crossover of the normal directions along which nodes are placed to avoid formation of invalid elements by smoothing of the directions or deletion of elements. Also, interference of the boundary layer mesh on different surfaces has been accounted for and eliminated by element deletion in some implementations. The restriction of growing a single set of nodes from each surface node constrains the advancing layers method to a subset of non-manifold models and also limits the quality of elements that can be created.

A generalization of the advancing layers is presented in this paper. This generalized advancing layers method has several improvements over the standard advancing layers techniques allowing the generation of quality boundary layer meshes for non-manifold geometric models of arbitrary complexity. Section 2 provides motivation for, and an outline of the Generalized Advancing Layers method. Section 3 discusses issues associated with point placement for boundary layer meshing of arbitrarily complex non-manifold geometric domains. Section 4 describes techniques to ensure that the boundary layer elements generated will be valid, while the process of boundary layer element creation is presented in Section 5. Section 6 discusses the method used to guarantee that the boundary layer mesh is not self intersecting. The paper concludes with results and discussion in Section 7.

2 Generalized Advancing Layers Overview

The Generalized Advancing Layers Method uses the surface mesh as the basic structure on which to grow the anisotropic boundary layer mesh. However, unlike other methods, the current approach allows for multiple sets of

nodes to emanate from each surface mesh vertex thereby facilitating the creation of valid meshes for non-manifold models and good quality element construction near boundaries with sharp corners or high curvature.

The presence of multiple sets of nodes originating from a single surface node allows the prisms on the mesh faces to be much better shaped naturally leading to good tetrahedral element quality. The gaps created between the prisms are abstracted as more general polyhedral shapes called *blends* which are tetrahedronized. Filling the gaps between the prisms is an important part of the algorithm since failure to do so will expose the highly anisotropic faces to the isotropic mesher. High aspect ratio faces may also get exposed if the number of layers between adjacent prisms is different. To hide these faces from the volume mesher, tetrahedral elements called *transition elements* are created on top of the prisms with fewer layers than any adjacent prism.

Curves along which boundary layer nodes must be placed are chosen for all surface mesh vertices (*growth curves*). The number of growth curves at mesh vertex is dependent on the topology and geometry of the mesh at the vertex. In non-manifold models, model faces may have regions on one or both sides of the face. Also, any number of model faces may be connected to a single model edge. Due to the generality of the non-manifold topology, two or more growth curves may be necessary at nodes at non-manifold boundaries to ensure that mesh edges will not penetrate model faces that locally divide the space into two halves [15].

Growth curves are first chosen at mesh vertices on model vertices. Mesh entities of growth curves lying on model edges are incorporated into the model edge discretization. Next growth curves are chosen for mesh vertices classified on model edges. Growth curves on model faces are smoothed, shrunk or pruned to avoid crossover and self-intersection. Boundary layer mesh vertices and edges are created for these growth curves and adjacent growth curves are joined to form abstracts *quadrilaterals* and *triangles*. These boundary polygonal constructs are triangulated and the triangles incorporated into the model face discretization. Growth curves are then constructed at mesh vertices classified on model faces and are smoothed, shrunk and pruned to ensure creation of valid elements. Mesh vertices and edges are created along these growth curves. Entities from adjacent growth curves connected up to form *triangular prisms* built on surface triangles and the prisms are tetrahedronized. If there is a difference in the number of layers between adjacent prisms, *transition elements* are created atop the prisms with fewer layers. As a final step in the creation of anisotropic elements, the gaps created between prisms due to the presence of multiple growth curves are abstracted as more general polyhedral shapes (*blends*) and tetrahedronized.

Once the anisotropic mesh is created, it is checked for any self intersections. Self intersections are fixed first by shrinking the layers locally and then by deletion of elements, if necessary. This completes the boundary layer mesh which is then handed over to isotropic mesher for completing the meshing process. The steps in the process are illustrated in Figure 1.

3 Finding Growth Curves

Point placement in the boundary layer mesh occurs along growth curves while respecting user mesh size specification for the boundary layers. Growth curves may be boundary, interior or both. The definition of growth curves allows them to start off with the nodes on the boundary of the model, then to separate from the model surface and grow into the interior of the model, and then reattach to the surface again. Reattachment of growth curves is not yet permitted in the implementation. Since the elements in the anisotropic mesh are created from triangular prisms and other blend polyhedra the quality of elements resulting is heavily influenced by the deviation of the growth curves from the normal direction to the base triangle. Therefore, nodes of growth curves growing from mesh vertices classified on model edges and vertices are allowed to lie on the boundary if the normal direction of the growth curve is close enough to the boundary. Figure 2 illustrates the types of growth curves.

Since multiple growth curves can go into a single region from any mesh vertex on the surface, each growth curve is associated with mesh face uses (mesh faces and their sides) using nodes of this growth curve to form elements. This information facilitates the connection of nodes of different growth curves in an unambiguous manner. Nodes of growth curves from mesh vertices of a mesh edge use are connected to form an abstraction called a boundary layer quadrilateral (Figure 3a) if they reference a common mesh face use. Nodes of growth curves from vertices of a mesh face are connected to form a boundary layer prism (Figure 3b) if they reference a common mesh face use. Nodes from multiple growth curves of a mesh vertex use are connected to form a boundary layer triangle (Figure 3c). The connectivity between nodes of a multiple boundary layer quads at mesh mesh edge uses and multiple growth curves at mesh vertex uses is more general. These constructs will be connected using more general procedures to create blend meshes (Figure 3d,e). Note that boundary layer quads, triangles, prisms and blends are abstract constructs that never actually exist in the mesh but are useful for algorithmic procedures and their discussion. In reality, triangles and tetrahedra of the individual layers are created directly in the mesh generator.

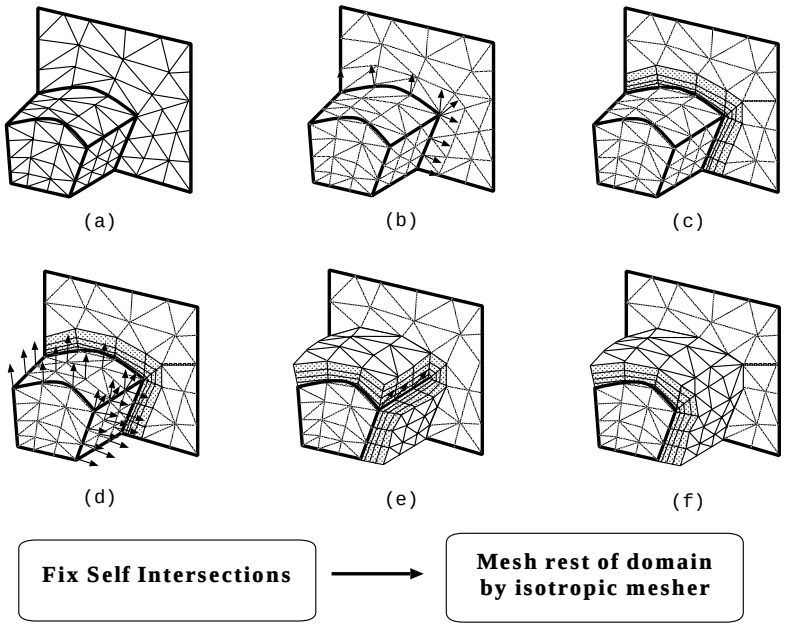

Figure 1: Steps in creation of boundary layer mesh (a) Surface mesh (b) Growth curves on model vertices and model edges (c) Boundary retriangulation (d) Growth curves on model faces (e) Prism creation (f) Blend creation.

3.1 Calculating the number of growth curves at each vertex

The number of growth curves at any mesh vertex with respect to a model face on which a boundary layer mesh is to be grown depends on the local mesh topology and geometry. The topological requirement for multiple growth curves at a mesh vertex with respect to a single face always arises for non-manifold faces with region to be meshed on both sides. At these boundaries at least two growth curves are necessity for a valid mesh. Figure 4 gives an example of such a situation. More complex interactions occur at model edges and vertices where, locally, the space may be subdivided into subspaces or manifolds, such that points from two different manifolds cannot be connected to each other without penetrating a model face [16, 17, 18, 19]. At such places, the minimum number of growth directions required depends on the number of manifolds.

Multiple growth directions may also be necessary at some mesh vertices due to the geometry of the model faces. Multiple growth directions are needed if the normals of the mesh faces vary so much that it is not possible to find valid common nodes that yields positive volume elements for the wedges of the connected mesh faces. Multiple growth directions are also desirable in cases where the normal variations are large enough to result in poorly shaped elements with large dihedral angles. Therefore, the procedures to find the growth curves first find the minimum number of growth curves required by the topology and increase this number further as dictated by the geometry.

3.2 Node placement and classification

The determination of growth curve starts with a procedure which assumes a single classification for all the nodes of the growth curve. If this causes problems with geometric or topological validity that cannot be resolved then a more general procedure is applied in which the nodes of a growth curve can have different classifications. In both cases the starting point is to place the nodes of the growth curve on the lowest order model entity possible.The basis for growing a boundary layer growth curve on a model face or edge is to use the average normal of the given mesh face uses of that direction and then find appropriate locations on the model entity close to the initial positions of the nodes. The specific checks to be satisfied for node placement and classification required to respect topological validity of the mesh, topological compatibility of the mesh with the model and estimated geometric validity of mesh are given in [15]. If creating a boundary growth curve violates any of these requirements, the growth curve is grown into the interior. Growth curves from the mesh vertices on a model face are always straight lines classified in the interior of

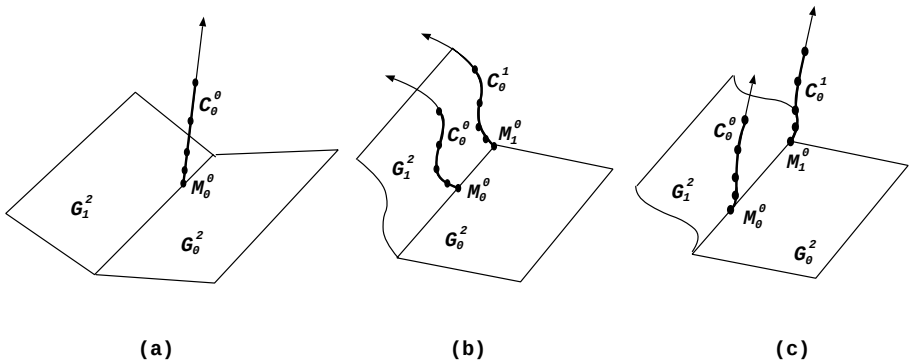

Figure 2: Types of growth curves (a) Interior Growth curve (b) Boundary growth curves (c) Partly boundary and partly interior growth curves

the model. Node spacing for growth curves may be *geometric*, *exponential* or *adaptive* [15]. In the adaptive method the growth curve at any location is terminated as soon as the layer thickness is approximately equal to the isotropic mesh size in the neighborhood.

4 Ensuring Element Validity

While growth curves are created with consideration for topological validity and topological compatibility, only preliminary consideration is given to geometric validity of elements during the point placement phase. This is because geometric invalidity arises mostly due to interactions between entities of neighboring growth curves connected to form tetrahedral elements. For this reason, it is only viable to check for geometric validity after creation of all growth curves. Invalidity of boundary layer elements occurs either due to crossover of growth curves or more than a 90° deviation of the growth curve from the mesh face normal.

4.1 Element Validity Checks

The individual triangles of the boundary layer quad are checked in real space for near zero area since negative area does not have any meaning for triangles on a general surface in 3-space. Then adjacent triangles are checked to see if the dihedral angle along their common edge is greater than an assumed tolerance α (taken to be 90°). This is to measure if the discretization of the surface is excessively distorted. Finally, the lateral edges of the boundary layer quad are checked for intersection by projection onto a common plane to verify that the growth curves are not crossed over. The validity of boundary layer prisms is done by checking the validity of all its component tetrahedra.

4.2 Correcting Invalid Elements

Correction of crossover considers three different methods, Smoothing, Shrinking and Pruning in that order. In the smoothing step, a weighted Laplacian smoothing procedure is applied to growth curves to eliminate crossover. Smoothing of boundary and interior growth curves is done separately on a model edge-by-edge and model face-by-face basis respectively for two reasons. Firstly, all boundary quads must be incorporated into the surface mesh before growth directions are determined at nodes on the model face. Secondly, boundary growth directions may be general curves on model surfaces preventing a direct application of the Laplacian smoothing technique. The shrinking procedure is based on the principle that crossover often occurs because the thickness of the boundary layer is high relative to the local curvature of the model face or the acuteness of the angle between model/mesh faces. Therefore, the shrinking process locally reduces the thickness of the boundary layers if it will make the affected elements valid. Shrinking of growth curves is followed by recursive adjustment of neighboring growth curve heights to ensure a smooth gradation of boundary layer thickness.

In some severe or degenerate case, neither smoothing nor shrinking can fix the invalid elements. In such a case the

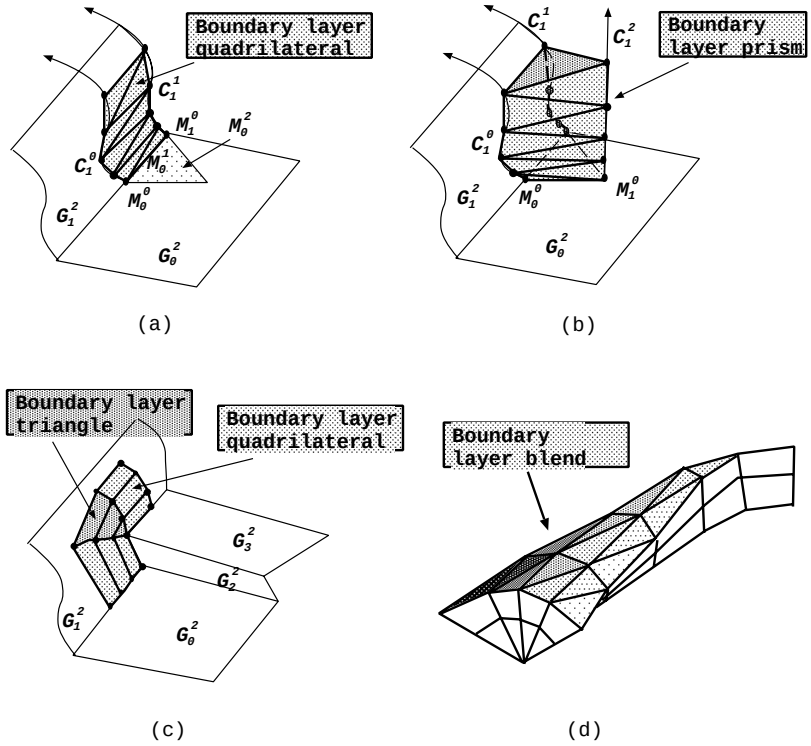

Figure 3: Abstract polygonal and polyhedral constructs in the boundary layer mesh.

growth curves of affected elements are pruned, i.e., some of their nodes are deleted, so that only valid elements are remaining. Pruning of growth curves is also accompanied by recursive pruning of adjacent growth curves to avoid sudden variations in the node distribution. When a simple prism is pruned, some of its high aspect ratio faces get exposed to the volume mesher which is not designed to properly deal with them. Therefore, transition elements are constructed on the top of prisms having a level difference with adjacent prisms. Figure 5 illustrates the three methods for fixing growth curve crossover in a 2D boundary layer mesh.

5 Boundary Layer Element Creation

Of the different constructs used to create anisotropic elements, the boundary layer quads and prism contribute the most elements, being grown on surface mesh edges and mesh faces respectively. The other constructs only serve to shield the isotropic mesher from the stretched mesh faces on the sides of prisms. The presence of multiple growth curves will yield adjacent prisms which are separated by gaps. The walls of these gaps are formed by the highly stretched faces of the anisotropic mesh which must be shielded from the isotropic mesh generator. These gaps are filled by boundary layer blends (Figure 6). In principle, boundary layer blends may or may not have fixed number of elements along the edges. Variable blend constructs typically occur at model edges where the dihedral angle between the connected mesh faces is varying along the edge. While boundary layer blends at mesh edges are easy to mesh using templates, boundary layer blends at mesh vertices are harder since an arbitrarily number of prisms and blends may be incident on the mesh vertex.

The last boundary layer construct used is the transition tetrahedron. When the growth curves of a mesh face forming a prism have different number of nodes, a step is formed in the boundary layer. To avoid leaving the highly stretched faces exposed, tetrahedra are created transitioning from the growth curves with larger number of nodes to those with smaller number of nodes (See Figure 7). Transition tetrahedra can span more than one layer (A similar concept has also been introduced by Connell and Braaten[1] for dealing with one layer difference).

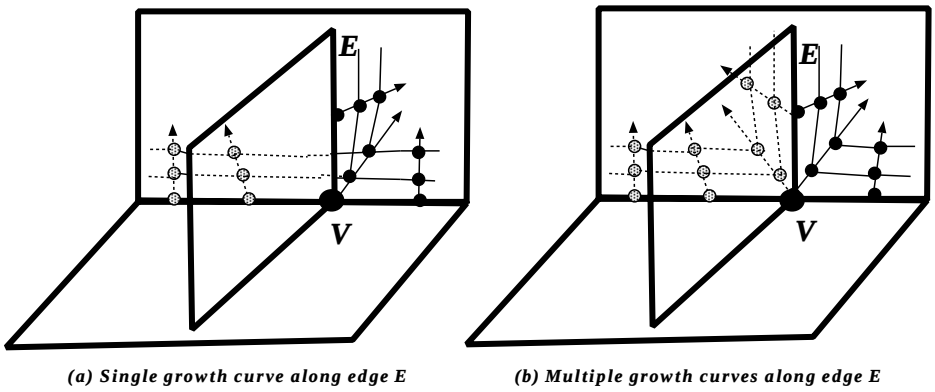

Figure 4: Need for multiple growth curves at non-manifold boundaries. In (a) a single set of growth curves originate from nodes on model vertex V and model edge E. Edges connecting the shaded and solid nodes then intersect model edge E. In (b) Multiple growth curves from nodes on vertex V and edge E result in valid connections of the nodes.

5.1 Model Edge Retriangulation

The insertion of boundary layer mesh edges and vertices classified on model edges is carried out through local mesh modification operators [20]. Given an edge to be inserted into the discretization of a model edge, existing mesh edges overlapping the edge to be inserted are identified by examining the one-dimensional parametric space of the model edge. The end vertices of the edge to be inserted are introduced into the model edge discretization by an edge split if coincident vertices do not already exist in the mesh. Then all edges overlapping the edge to be inserted are deleted.

5.2 Triangulation of Boundary Layer Quads and Blend Triangles

Triangulation of boundary layer quads is done by connecting nodes of adjacent growth curves *not* originating from the same mesh vertex. In converting these boundary layer quads to triangles the choice of the diagonal is dictated by the validity of the connected prisms. For reasons explained in the section describing prism tetrahedronization (Section 5.4 below), the diagonal is made by connecting lower node of the growth curve at the mesh vertex with a lower identifying number (vertex id) to the next higher node of the growth curve at the other vertex. Creation of boundary layer blend triangles is similar to the creation of boundary layer quads except that they establish connections between nodes of two growth curves originating from the same mesh vertex.

5.3 Model Face Retriangulation

Mesh entities classified on model faces are incorporated into the surface mesh by using local mesh modifications [20]. This method is chosen to avoid the use of the parametric space provided by the geometric modeler which can sometimes be highly distorted. This appears to be a particularly severe problem for surfaces formed by concatenation of triangular facets. The model face retriangulation procedure is done for each model face that the growth curves of a model edge affect. Given a model edge and a model face on which growth curves lie, the following steps are carried out to create and incorporate the boundary layer mesh into the surface mesh triangulation:

1. Boundary layer quads classified on the the model face are created as described above.
2. Each boundary layer mesh entity that forms the outer boundary of the set of boundary layer mesh faces classified on the model face is incorporated into the surface mesh by the edge recovery procedure briefly described below and discussed in full detail in [20]. Once all the necessary edges have been recovered, the outer boundary of the set of faces to be inserted into the mesh exactly matches the outer boundary of a set of faces in the underlying surface mesh.
3. Mesh faces of the existing surface triangulation overlapping the boundary layer mesh faces are deleted and the boundary layer faces incorporated into the mesh instead.

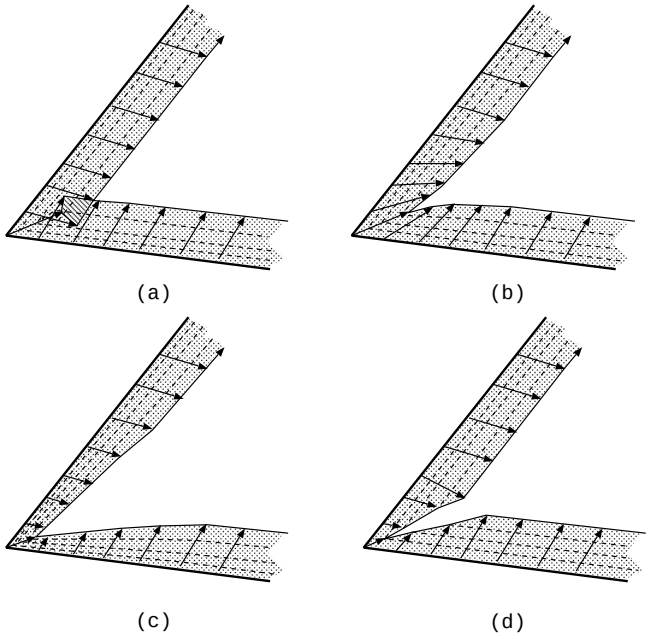

Figure 5: Fixing growth curve crossover in 2D (a) Mesh with crossover at corner (b) Mesh with crossover fixed by smoothing (c) Mesh with crossover fixed by shrinking (d) Mesh with crossover fixed by deletion

The edge recovery procedure inserts the vertices of the edge to be recovered into the existing surface mesh using projection along mesh face normals. A local mesh optimization is performed to eliminate any other poorly shaped faces created during the vertex insertion process. Next, a path of edge connected mesh faces is found from one vertex to another, again by projection of the edge to be recovered onto planes of the mesh faces. Mesh edges that cross the projected edge are successively swapped to recover the edge. After recovering the edges on the outer boundary of the set of quads to be introduced into the mesh, the mesh faces overlapping the boundary layer mesh faces are deleted and the boundary layer quads put in their place. The mesh resulting from face retriangulation is subjected to checks and remedies to ensure that it is not self intersecting [21].

5.4 Prisms, blends and transition elements

The bulk of the elements in the boundary layer are from the boundary layer prisms. Boundary layer prisms are grown on mesh face uses by connecting the three growth curves at the face vertices which share each mesh face use. The tetrahedronization of each boundary layer prism gives rise to three tetrahedra. Boundary layer prisms can be thought of as being formed by three quads that are grown from the edges of the triangular mesh face. There are eight possible combinations of diagonals for the quads of a prism. Of these only six are configurations which can be tetrahedronized without the insertion of any new points inside the prism. Therefore, in assigning directions for the diagonals of the quads in the boundary layer mesh, care must be taken not to assign directions such that some prisms cannot be tetrahedronized. This is done by a very simple algorithm based on numbering of the surface mesh vertices. Given a surface mesh with any arbitrary assignment of unique numbers (ids) for the mesh vertices, the ids of vertices of a face in either clockwise or counterclockwise direction cannot be strictly increasing or strictly decreasing. Using this notion, the diagonals of boundary layer quads are constrained to go from the lower node of the growth curves of vertices with a lower id to the upper node of growth curves of vertices with a higher id ensuring the validity of all prism tetrahedronizations. The tetrahedronization of boundary layer prisms is done using two templates.

Boundary layer blend polyhedra are created between the exposed sides of two prisms from adjacent faces. Only simple blends, in which the gap between the two quads is filled by elements without the creation of new points, are considered here. Two quads originating from the same edge may have separate growth curves at each end or share a common growth curve at one end giving rise to a total of three types of blend polyhedra [15]. In the more general situation, it is proposed that if the gap to be filled between two quads is too large then additional growth curves be

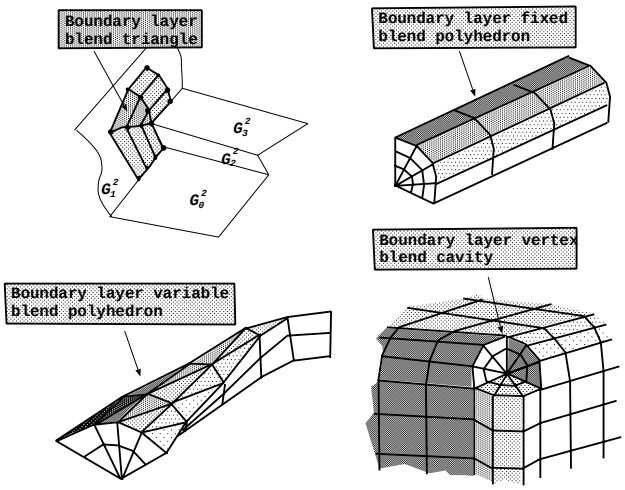

Figure 6: Blend elements

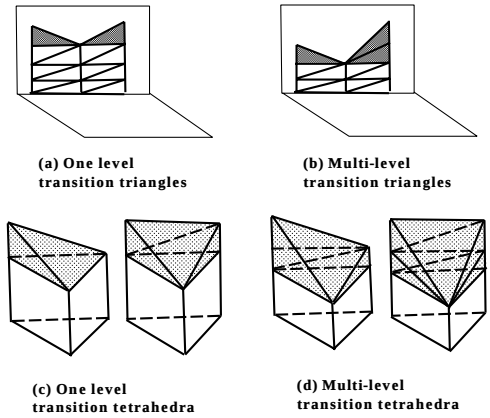

Figure 7: Transition elements

introduce to produce a blend mesh as shown in Figure 6.

Since boundary prisms can be made only by connecting corresponding nodes on the component growth curves, faces from boundary layer quads with more nodes than the others remain exposed. These stretched faces are closed off from the isotropic mesh generator meshing the rest of the volume by transition triangles for boundary layer quads and transition tetrahedra for boundary layer prisms. Transition triangles are formed atop boundary layer quads with a level difference between the two growth curves by connecting the top nodes of the two growth curves and forming an element similar to the lower triangle of a boundary layer quad. Atop a prism with one level difference between its component growth curves, there may be one or two transition tetrahedra depending on whether one or two growth curves have fewer nodes than the others (Figure 7). If the level difference is more than one then transition tetrahedra are stacked on top of each other.

6 Fixing Boundary Layer Intersections

When boundary layer elements are generated on model faces that are too close to each other the layers may run into each other in which case the polyhedral cavity that is presented to the isotropic mesher is self-intersecting. Since the isotropic mesher expects a polyhedral cavity with no self-intersections, this situation must rectified. Boundary layer intersection is fixed by local shrinking of layers and/or local pruning of growth curves leading to deletion of elements. Correction of self intersections are done after element creation as it is simpler to find the mesh faces forming the polyhedral cavity left to be meshed and the neighborhood used for intersection checks can be more localized.

The technique used to detect self intersections looks at mesh faces that have fewer regions connected to them than they should in a completed mesh (*exposed* faces). These faces may be classified on model faces with no boundary layer, and top and side faces of the boundary layer prisms, blends and transition polyhedra. An octree is built in the domain and the exposed faces attached to the terminal octants. The octree serves as a localization structure for the intersection checks. Each exposed face is intersected with the set of faces in its neighborhood. If an intersection is detected, elimination of the intersection is attempted by local shrinking of the prism connected to the face while being constrained to keep connected elements valid. Since the algorithm only checks and fixes the intersection between *exposed* faces and not of entire prisms, blends or transition polyhedra, shrinking of two boundary layer constructs to correct interference may result in new intersections of other faces in the neighborhood. Therefore, the algorithm is iterative and continues until all intersections are fixed. The algorithm is made more efficient by recognizing that if an intersection between two faces has been detected and fixed there is a good possibility that the fix has caused new intersections or that there are already more intersections in the neighborhood. Therefore, the algorithm employs locally recursive intersection detection and correction in the neighborhood of a located intersection. If all the intersections between front faces cannot be fixed by shrinking prisms, then the growth curves are pruned using the same algorithm as described in Section 4.2. As in the case of deletion of elements to fix growth curve crossover, transition elements are introduced if adjacent prisms have differing number of layers.

7 Results and Discussion

Figure 8 shows the boundary layer mesh for capturing the flow in an expanding pipe. boundary layers are generated on the walls of both pipe sections and also on both sides of a surface introduced in the geometry to capture the free shear layer. The boundary layer mesh thickness on all the surfaces increases in thickness from the inflow to the outflow surface. In addition the number of nodes in the boundary layer mesh are different on each surface.

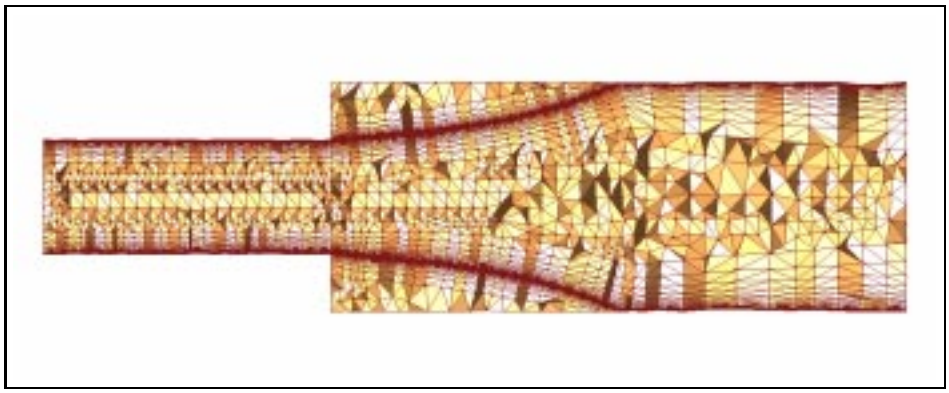

Figure 8: Anisotropic mesh for capturing flow in an expanding pipe including the free shear layers. The variation in the boundary layer thickness is clearly seen in the picture.

Figure 7 shows a cutaway of the boundary layer mesh for the space shuttle while Figure 10 shows the boundary layer mesh (in lighter shades) for the underbody of an automobile. The boundary layer mesh in these example was chosen to be thicker than normal for clarity of illustration. The mesh shown is valid, non-self intersecting and suitable for input to the isotropic mesh generator for completion of the mesh generation task. Using smoothing and compression of growth curves, the method has successfully resolved crossover of growth curves in areas of high curvature and interpenetration of the boundary layers in very confined spaces except in very few localized areas where elements had to be deleted.

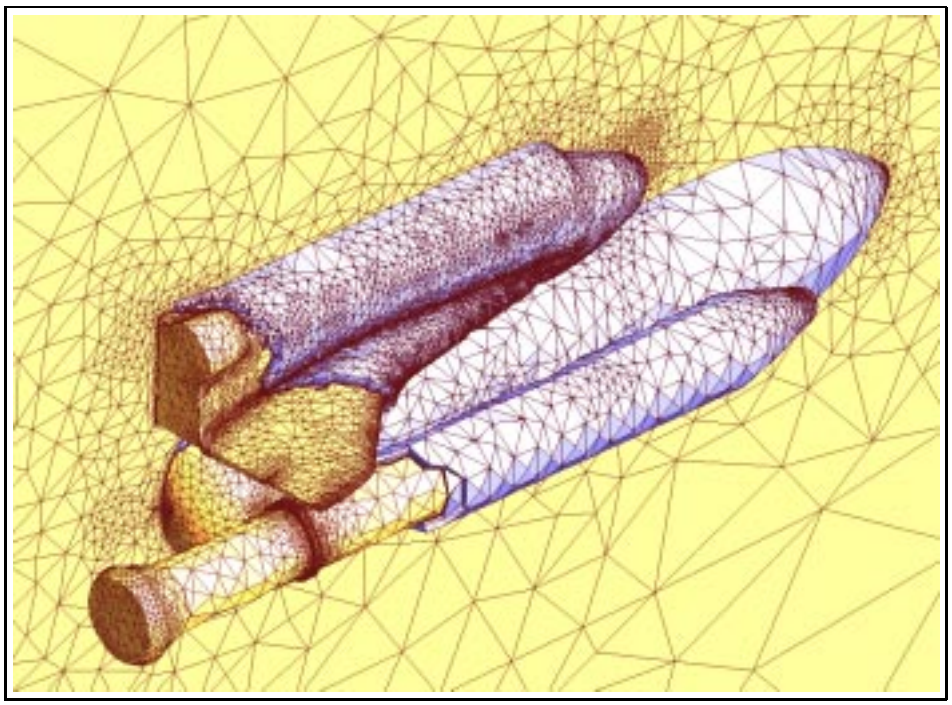

Figure 9: Cut-away of boundary layer mesh for a model of the space shuttle with center tank and booster rockets.

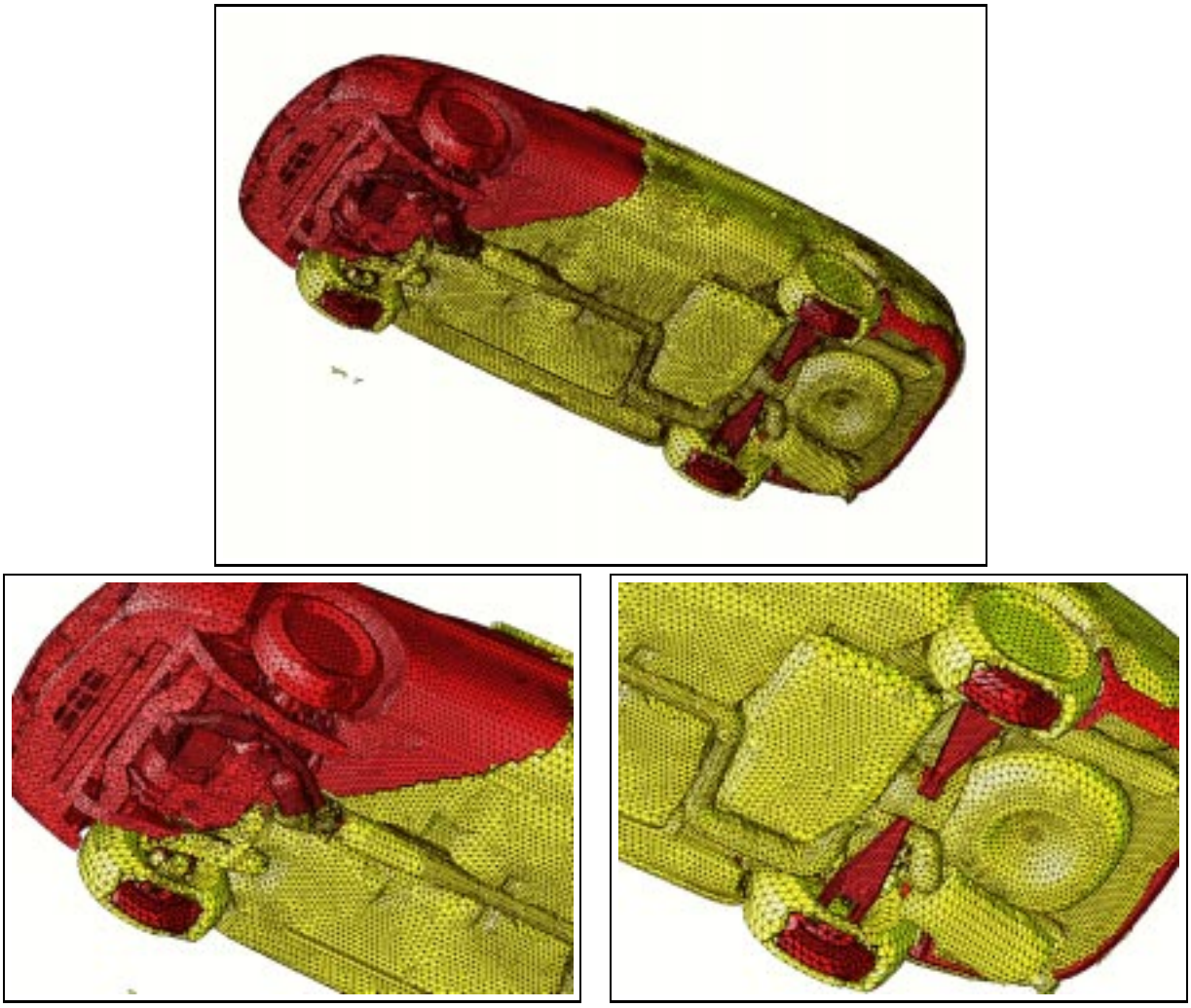

Figure 10: Underbody of a car showing cut-away of boundary layer mesh with zoomed-in views of the front and rear of the car.

The boundary layer meshes from the Generalized Advancing Layers procedure have been used for calculation of heat transfer coefficients for automobile configurations as well as for simulation of the Czochralski process of bulk crystal growth [22] (Figure 11).

References

[1] S.D. Connell and M.E. Braaten. Semistructured mesh generation for three dimensional navier-stokes calculations. *AIAA Journal*, 33(6), 1995.

[2] Y. Kallinderis and S. Ward. Hybrid prismatic/tetrahedral grid generation for complex 3-d geometries. In *AIAA-93-0669*, 1993.

[3] Y. Kallinderis, A. Khawaja, and H. McMorris. Hybrid prismatic/tetrahedral grid generation for complex 3-d geometries. In *AIAA-95-0211*, 1995.

[4] R. Löhner. Matching semi-structured and unstructured grids for navier-stokes calculations. In *AIAA-93-3348-CP*, 1995.

[5] S. Pirzadeh. Viscous unstructured three-dimension grids by the advancing-layers method. In *Proceedings of the 32nd Aerospace Sciences Meeting & Exhibit*, number AIAA-94-0417, Reno, NV, Jan 1994.

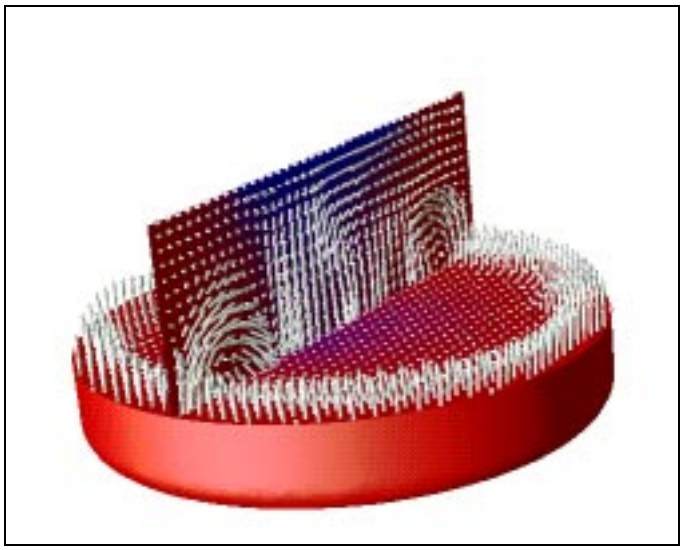

Figure 11: Velocity vectors horizontal and vertical-horizontal planes at t=50 seconds for growth of indium-phosphide crystal. The mesh has 6 anisotropic layers at the wall with a first layer thickness of 0.05 and a total of 634001 elements. (Courtesy: S. Adjerid et.al. [22])

[6] M.J. Castro-Diaz and et. al. New progress in anisotropic grid adaptation for inviscid and viscous flow simulations. In *4th International Meshing Roundtable*, Albuquerque, NM, Oct 1995. Sandia National Labs.

[7] M.J. Castro-Diaz and et. al. Anisotropic unstructured mesh adaptation for flow simulations. *Int. J. Numer. Meth. Engng.*, 25(4):475, 1997.

[8] O. Hassan and et. al. Mesh generation and adaptivity for the solution of compressible viscous high speed flows. *Int. J. Numer. Meth. Engng.*, 38(7):1123–1148, 1995.

[9] O. Hassan and et. al. Unstructured tetrahedral mesh generation for three-dimensional viscous flows. *Int. J. Numer. Meth. Engng.*, 39:549–567, 1996.

[10] M.J. Marchant and N.P. Weatherill. Unstructured grid generation for viscous flow simulations. In *4th International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, pages 151–162, Swansea, Apr 1994.

[11] D.J. Mavriplis. Adaptive mesh generation for viscous flows using delaunay triangulation. *Journal of Computational Physics*, 90:271–291, 1990.

[12] D.L. Marcum. Generation of unstructured grids for viscous flow applications. In *AIAA-95-0212*, 95.

[13] D.L. Marcum and N.P. Weatherill. Generation of unstructured grids for viscous flow applications. In *AIAA-95-1726*, 95.

[14] Dmitri Sharov and Kazuhiro Nakahashi. Hybrid prismatic/tetrahedral grid generation for viscous flow applications. *AIAA Journal*, 36(2):157–162, Feb 1998.

[15] Rao V Garimella. *Anisotropic Tetrahedral Mesh Generation.* PhD thesis, Rensselaer Polytechnic Institute, Troy, NY 12180, August 1998. in preparation.

[16] E. L. Gursoz, Y. Choi, and F. B. Prinz. Vertex-based representation of non-manifold boundaries. In M. J. Wozny, J. U. Turner, and K. Priess, editors, *Geometric Modeling Product Engineering*, pages 107–130. North Holland, 1990.

[17] K. J. Weiler. *Topological Structures for Geometric Modeling.* PhD thesis, Rensselaer Design Research Center, Rensselaer Polytechnic Institute, Troy, NY, May 1986.

[18] M. Mäntylä. *Introduction to Solid Modeling.* Computer Science Press, Rockville, Maryland, 1988.

[19] M. Mortenson. *Geometric Modeling.* J. Wiley and Sons, New York, 1985.

[20] B. Kaan Karamete, Rao Garimella, and Mark S. Shephard. Recovery of an arbitrary edge on an existing surface mesh using local mesh modifications. *IJNME*, 1998. in submission.

[21] Hugues L. de Cougny. *Parallel Unstructured Distributed Three Dimensional Mesh Generation.* PhD thesis, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, May 1998.

[22] S. Adjerid, J.E. Flaherty, K. Jansen, and M.S. Shephard. Parallel finite element simulations of czochralski melt flows. In S.N. Atluri, editor, *Proceedings of ICES98*, Atlant, GA, 1998. to be published.