INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2000; **49**:193–218

Boundary layer mesh generation for viscous flow simulations

Rao V. Garimella*,† and Mark S. Shephard‡

Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy NY 12180, U.S.A.

SUMMARY

Viscous flow problems exhibit boundary layers and free shear layers in which the solution gradients, normal and tangential to the flow, differ by orders of magnitude. The generalized advancing layers method is presented here as a method of generating meshes suitable for capturing such flows. The method includes several new technical advances allowing it to mesh complex geometric domains that cannot be handled by other techniques. It is currently being used for simulations in the automotive industry. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: anisotropic mesh generation; boundary layer meshes; viscous flow simulations

1. INTRODUCTION

Many physical problems exhibit relatively strong gradients in certain local directions compared to the other directions. Some examples of such situations are thermal and fluid boundary layers, and non-linear solutions in domains with very thin sections. A minimum element size along these directions is necessary to capture the solution in these regions. Anisotropic meshes with small element sizes in the directions of strong gradients and large sizes along the others leads to significant savings in mesh size and solution costs.

High Reynolds number fluid flow simulations have boundary layers at the wall and also free shear layers not attached to any model boundary. The relative rates at which the solution variables change in boundary and shear layers, normal and tangential to the flow, differ by orders of magnitude in such problems. Use of properly aligned anisotropic meshes in these cases is essential.

A generalization of the advancing layers method [1–4] is presented here for generating boundary layer meshes. The method is designed to efficiently and reliably generate good quality anisotropic tetrahedra near the boundary layer surfaces for arbitrarily complex non-manifold domains starting from a surface mesh. The method has several improvements over the previous advancing layers techniques. It is demonstrated that the common strategy of inflating the surface mesh *as is* to

* Correspondence to: R. V. Garimella, Los Alamos National Laboratory, EES-5, MS C306, Los Alamos, NM 87545, U.S.A.
†E-mail: raogarimella@alum.rpi.edu
‡E-mail: shephard@scorec.rpi.edu

Received 15 April 1999
Revised 4 August 1999

form the boundary layer leads to invalid meshes for some non-manifold models and poor quality elements at sharp corners in 2-manifold models. Various procedures are described to make the boundary layer elements valid and to ensure that the mesh is not self-intersecting. The improvements incorporated into the method has enabled it to be used successfully to generate boundary layer meshes for geometrically complex industrial models.

The rest of this paper is organized in the following manner. A review of the previous efforts in anisotropic mesh generation is presented in Section 2. Definitions and notations are described in Section 3. Section 4 presents an overview of the generalized advancing layers method used here. Section 5 discusses point placement for boundary layer meshing of arbitrarily complex non-manifold geometric domains. Section 6 describes techniques to ensure that the boundary layer elements generated will be valid while the creation of boundary layer elements is presented in Section 7. Section 8 discusses the method used to guarantee that the boundary layer mesh is not self-intersecting.

2. REVIEW OF MESH GENERATION FOR VISCOUS FLOW SIMULATIONS

Direct generation of unstructured anisotropic meshes has been attempted with both Delaunay [5–8] and advancing front methods [9–11]. The Delaunay criterion itself will always define as isotropic a mesh as possible for a given set of points within the space in which they are defined. Therefore, efforts on generating anisotropic meshes using the Delaunay method have focused on meshing in a transformed space using metrics which will yield an anisotropic mesh in the real space.

Mavripilis [12] presented a method for anisotropic adaptation of triangular meshes constructing a metric based on two independent stretch vectors at each point. Using this metric the local space is mapped to a control surface in a transformed higher dimension space in which a Delaunay triangulation is performed.

Vallet *et al.* [13] have proposed a similar idea for the initial mesh generation process as well as adaptation. George *et al.* [5, 6, 14] have generalized the ideas of generating anisotropic mesh generation by the Delaunay method using metric specifications. Also, the metrics are modified near viscous walls to keep the mesh as orthogonal to the wall as possible and maintain a certain minimum distance of the first node from the wall.

Hassan *et al.* [15] have used a modified advancing front method to generate anisotropic meshes where a layer of elements is generated from a front using isotropic criteria and compressed to the desired thickness. While this method worked well in 2D, it is prone to problems in 3D [16]. Hassan *et al.* [16] have also devised a variation of the advancing front method for boundary layer mesh generation. In this method, the standard advancing front procedure is adapted to place new vertices at the offsets required to generate anisotropic elements.

Marcum and Weatherill [17] have described an approach for unstructured grid generation for viscous flows using iterative point insertion followed by local reconnection subject to a quality criteria. The point distribution for the anisotropic mesh is generated along 'normals' to surfaces according to user specifications or error estimates. The most interesting aspect of this work is that they account for sharp 'discontinuities' at edges and vertices and generate points along additional directions in such cases.

Most of the work in generating meshes for viscous flow simulations has been in the direction of generating an anisotropic mesh next to surfaces where a boundary layer is expected and then

filling the rest of the domain by an isotropic mesh generator. The advancing layers method starts from a triangulation of the surfaces on which the boundary layer mesh must be grown. From each surface node a direction is picked for placing the nodes of the anisotropic mesh. These nodes are connected to form layers of prisms (if necessary, subdivided into tetrahedra) on top of each surface triangle.

Löhner [3] described one of the early efforts for combining layers of anisotropic tetrahedronized prisms grown on some model boundaries with an unstructured isotropic mesh generated by an advancing front method in the rest of the domain. The procedure detects poorly shaped, improperly sized and intersecting elements, and deletes them. A recent paper by Löhner [18] advocates the use of anisotropic refinement of an isotropic mesh using the Delaunay criterion to generate boundary layer meshes.

Kallinderis *et al.* [2, 19] have developed a hybrid prismatic/tetrahedral mesh generator by enclosing the body around which the flow is to be simulated with layers of prisms and then filling the rest of the domain using a combination of octree and advancing front methods. The procedure incorporates an algorithm to ensure that the interior nodes of the prisms are 'visible' from all the relevant faces of the previous layer [2]. Included in this method is a procedure to automatically recede and smoothly grade layers in confined regions of the model based on ray tracing methods [19]. Sharov and Nakahashi [20] have described a similar method with some modifications for generating better elements and for generating all tetrahedra.

Pirzadeh [4] describes a similar approach called the advancing layers method (ALM) for the generation of anisotropic meshes for viscous flow calculations. The significant features of this work are: (1) introduction of prism templates, (2) a non-iterative procedure for obtaining valid diagonals for the prisms, (3) an iterative procedure for obtaining valid directions for placement of points and (4) a procedure for avoiding interference between layers.

Connell and Braaten [1] described an implementation of the advancing layers procedure with enhancements to deal with general domains. Their work discusses many of the fundamental issues with mesh generation for viscous flow simulations using the advancing layers methods. The paper details an algorithm to ensure that all prisms have a valid set of diagonals. Also, discussed is a technique, for grading the boundary layer mesh to avoid exposing highly stretched faces to the isotropic mesh generator when elements are deleted. They also discuss the interference of layers, varying thickness boundary layers and resolution of wakes.

The advancing layers algorithms reviewed above posses the following complexities:

1. They cannot deal with general non-manifold situations.
2. They do not account for general interactions of the boundary layer mesh with adjacent surfaces.
3. They may produce poor-quality meshes in the presence of sharp discontinuities in the surface normals.
4. They do not sufficiently address the issue of interaction of anisotropic faces of the boundary layer mesh with the isotropic mesh.
5. They do not provide assurance algorithm for non-interference of boundary layers.

The research described herein is a generalization of the advancing layers method mentioned above combined with an isotropic mesh generator based on a combination of advancing front and Delaunay methods [21, 33]. It addresses many of the issues that arise for complex non-manifold models enabling it to reliably mesh these domains.

3. DEFINITIONS AND NOTATIONS

3.1. Geometric model definitions and concepts

Geometric models may be *2-manifold* or *non-manifold*. Informally, non-manifold models are general combinations of solids, surfaces and wires [22, 23]. Geometric model entities are denoted here by G_i^d, representing the ith geometric model entity of order d ($d = 0, 1, 2, 3$ for vertices, edges, faces and regions, respectively).

The data structure used to represent the model in this work is based on the *radial edge data structure* [23] which presents the idea of *uses* to represent how topological entities are used by others in a non-manifold model. Every face in the model has two face uses, one on each side of the face. An edge carries as many pairs of uses as there are pairs of face uses coming into it. A vertex carries as many uses as there are edge uses coming into it. The radial edge data structure is more detailed than the minimum amount of information required to represent non-manifold models. The representation can be reduced by fusing edge uses together to form a single 'edge use' connected to two face uses. Similarly, vertex uses are condensed so that the minimum number of uses are present at any vertex. Such a data structure is referred to as the *minimal use data structure* [24].

3.2. Mesh definitions and concepts

The representation for the mesh [25–27] used here consists of mesh vertices, edges, faces and regions (and if necessary, their uses). Mesh entities are denoted by M_i^d, referring to the ith mesh entity of order d ($d = 0, 1, 2, 3$ for vertices, edges, faces and regions, respectively). Each entity in the mesh has a unique *classification* with respect to the model.

Definition 3.1. Classification is the unique association of a mesh entity, $M_i^{d_i}$, to a geometric model entity, $G_j^{d_j}$ ($d_i \leqslant d_j$) to indicate that $M_i^{d_i}$ forms part or all of the discretization of $G_j^{d_j}$ but not its boundary. The classification operator is denoted by $\sqsubset$ and $M_i^{d_i} \sqsubset G_j^{d_j}$ is used to denote the classification of $M_i^{d_i}$ on $G_j^{d_j}$.

Definition 3.2. A *mesh manifold* is a set of mesh face uses around a vertex, connected by edge uses, that locally separate the three-dimensional space into two halves.

Some examples of mesh face use manifolds are shown in Figure 1. In Figure 1(a), mesh manifolds for a mesh vertex classified on a model face, $M_v^0 \sqsubset G_0^2$, are shown. In Figure 1(b), mesh manifolds are shown for two vertices in a non-manifold model. In the figure, G_1^2 is an embedded face[§] making edge contact with two model faces G_0^2 and G_2^2. The local topology at M_a^0 is non-manifold and two mesh manifolds exist at the vertex with respect to just one side of the model faces G_0^2 and G_2^2. At M_b^0, only one mesh manifold exists in the model region under consideration. The concept of mesh manifolds is used to conceptually reduce a complex non-manifold boundary to a set of topologically simple 2-manifold boundaries.

[§]Embedded face – face with the same model region on both sides.

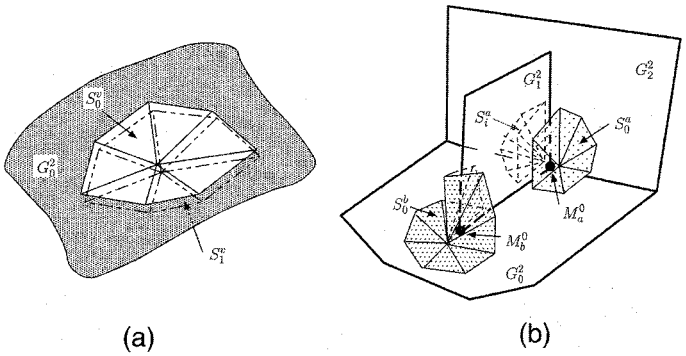

Figure 1. Examples of mesh face use manifolds.

4. OVERVIEW OF GENERALIZED ADVANCING LAYERS METHOD

The boundary layer meshing approach described here employs the advancing layers approach as its basis and generalizes it for meshing arbitrarily complex non-manifold geometric domains with good quality anisotropic elements near the surface. The technique is therefore referred to as the *generalized advancing layers method*. Like the advancing layers method, the procedure takes an input surface mesh, grows the anisotropic boundary layer mesh on it and then hands it over to the isotropic mesher to finish meshing the domain. Nodes of the boundary layer mesh are placed on curves (called *growth curves*) originating from surface mesh nodes. These boundary layer nodes are connected to form the anisotropic elements of the boundary layer mesh.

However, unlike other methods, the generalized advancing layers method allows multiple growth curves (i.e. multiple sets of boundary layer nodes) to emanate from each surface node. Therefore, the anisotropic mesh is not constrained to be an inflation of the surface triangles into triangular prisms and their tetrahedronization. The flexibility of introducing multiple growth curves eliminates the restriction that boundary layer prisms sharing a surface mesh edge or vertex must be joined along their sides. The procedure incorporates techniques to fill the gaps between prisms caused by multiple growth curves. This is important since failure to do so will expose the highly anisotropic faces to the isotropic mesher.

The basic steps of the generalized advancing layers method are as follows (refer Figure 2):

1. Growth curves are first determined at mesh vertices classified on model vertices.
2. If any of these growth curves lie partly or fully on a model edge, the boundary layer entities (mesh vertices and edges) classified on the model edges are created.
3. Boundary layer mesh entities classified on model edges are incorporated into the model edge discretization.
4. Growth curves are determined at mesh vertices classified on model edges (Figure 2(b)).
5. The growth curves that lie on model boundaries are smoothed, shrunk or pruned to avoid crossover and self-intersection.
6. Growth curves on the model boundary are combined to form three types of abstract boundary layer constructs—quads, transitions and blends. These constructs are triangulated resulting in boundary layer triangles classified on model faces.

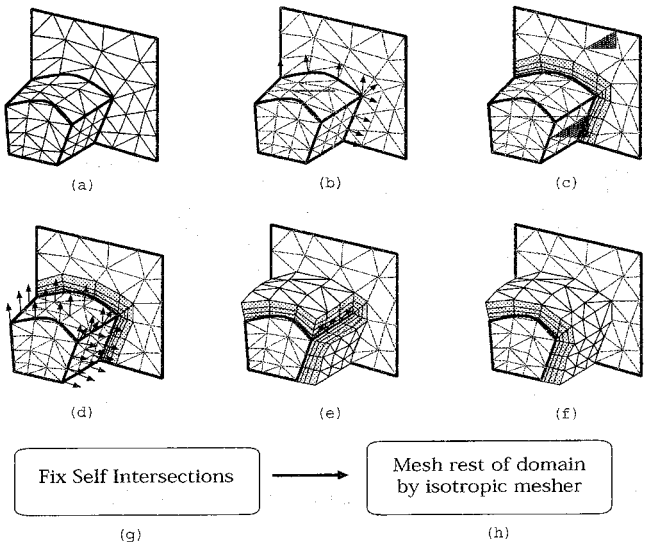

Figure 2. Steps of boundary layer meshing: (a) surface mesh; (b) growth curves on model vertices and model edges; (c) boundary retriangulation; (d) growth curves on model faces; (e) prism creation; (f) blend creation; (g) fixing self-intersection; (h) meshing remaining portion of domain by an isotropic mesher.

7. Boundary layer triangles lying on model faces are incorporated into the surface triangulation (Figure 2(c)).
8. Growth curves are determined at mesh vertices classified on model faces (Figure 2(d)).
9. These growth curves are smoothed, shrunk and pruned to ensure creation of valid elements.
10. Growth curves are connected up in the interior to form three more types of abstract boundary layer constructs—prisms, blends and transition elements (Figure 2(e) and 2(f)). The component tetrahedra of these abstractions are directly created to form the solid elements of the boundary layer mesh.
11. The inner boundary of the boundary layer mesh is checked for self-intersection so as to provide valid input to the isotropic mesher. Self intersections are fixed by local shrinking of the layers locally and then by deletion of elements, if necessary (Figure 2(g)).
12. The rest of the domain is meshed by the isotropic mesher (Figure 2(h)).

5. GROWTH CURVES

5.1. *Introduction*

Points in the boundary layer mesh are placed along boundary and interior growth curves while respecting user-requested layer sizes. All nodes of an interior growth curve except the first are classified in a region of the model. Interior growth curves are straight lines with present capabilities of the mesher. All nodes of a boundary growth curve are classified on the boundary of the model. Boundary growth curves may take an arbitrary shape defined by the surface that the nodes of the growth curves are classified on.

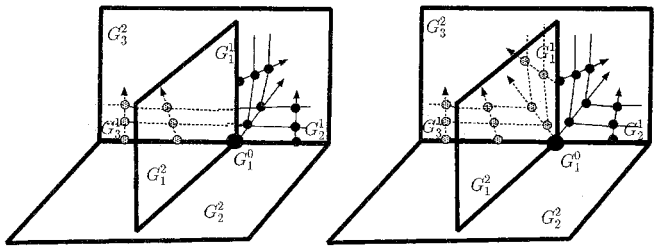

Figure 3. Need for multiple growth curves at non-manifold boundaries: (a) single growth curve along G_1^1; (b) two growth curves along G_1^1.

The quality of tetrahedra resulting from prisms in the advancing layers method is heavily influenced by the deviation of the sides of the prism 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 to the adjacent model surfaces and if the quality of the elements will be good with the nodes on the boundary.

The generalized advancing layers method permits multiple growth curves to originate into a single region from any mesh vertex classified on the model boundary. The number of growth curves at any mesh vertex with respect to a model face use depends on the local model topology and geometry. The topological requirement for multiple growth curves at a mesh vertex with respect to a single face use arises at some non-manifold boundaries. At these boundaries, multiple growth curves are necessary for generating a valid mesh.

Axiom 5.1. The minimum number of growth curves at any boundary mesh vertex required to produce a topologically valid mesh is equal to the number of mesh manifolds at the vertex that include at least one mesh face use classified on a model face with a boundary layer.

The above assertion can be easily demonstrated by the example shown in Figures 3(a) and 3(b). Here, the embedded face G_1^2 is incident on vertex G_1^0 along with two other faces, G_2^2 and G_3^2. It is assumed that a boundary layer mesh is being grown on G_2^2 and on both sides of G_1^2. It can be seen from Figure 3(a) that use of only one growth curve at $M_i^0 \sqsubset G_1^0$ and $M_i^0 \sqsubset G_1^1$ will lead to intersection of some quads with G_1^1 or penetration of G_1^2. Two growth curves at the vertex, one for each mesh manifolds at the vertex is the minimum acceptable number. Also, the nodes of each of these growth curves must lie within the respective mesh manifold (Figure 3(b)). Similarly, in 3D, interior edges may penetrate model faces if the minimum number of growth curves are not present at each vertex.

At some mesh vertices, multiple growth curves may become necessary due to the geometry of the model faces and the coarseness of their discretization. This is because creation of valid prisms requires that the nodes of a growth curve at any mesh vertex be 'visible' from any mesh face connected to the mesh vertex. Nodal visibility ensures that an element formed by connecting the mesh face to the node has positive volume. If the surface discretization is very coarse or the

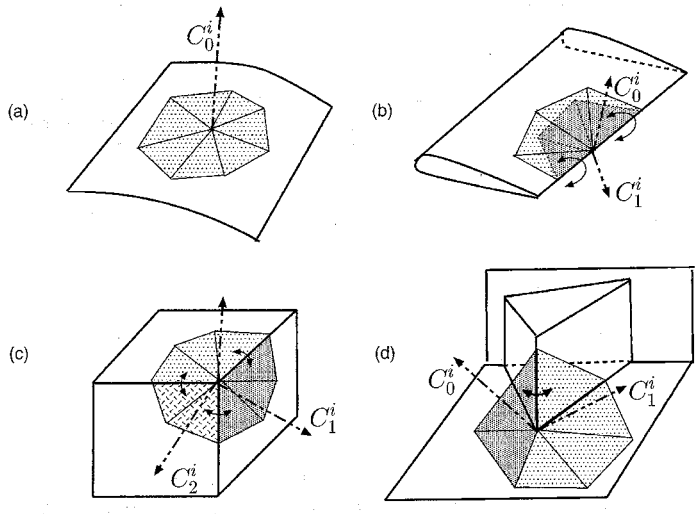

Figure 4. Mesh face use subsets in mesh manifolds: (a) all mesh faces share common growth curve; (b) two convex edges, shown by curved double-headed arrows, in mesh manifold; (c) three convex edges in mesh manifold; (d) only one convex edge in mesh manifold which is subdivided into two subsets.

model geometry itself changes enough, the normals of the mesh faces may vary so much that it may not be possible to find a valid common node that is visible from all the faces (even with methods described in References [4, 28]). Such impossible situations are the limit of the case where the growth curve deviates greatly from the mesh face normal leading to large dihedral angles in elements. Therefore, in general, it is desirable to have multiple growth curves at mesh vertices where the normals of the connected mesh faces change too much.

In keeping with the necessity of creating a valid mesh and desirability of creating well-shaped prisms, mesh manifolds are first found at each vertex and these are then divided up into subsets of mesh face uses. Each of these subsets of mesh face uses then share a common growth curve to be used in their prisms. The procedures to find these subsets works with face/side pairs in the mesh instead of requiring face uses to be represented.

The determination of subsets of mesh face uses in a mesh manifold sharing a common growth curve is based on the dihedral angle between pairs of mesh face uses. Figure 4 shows some examples of mesh face use subsets. In Figure 4(a), the mesh face uses (shown shaded) form a single subset sharing one growth curve. In Figures 4(b) and 4(c) some pairs of mesh face uses have a large dihedral angle between them and therefore they are split up into multiple face use sets. In Figure 4(d), the mesh face uses are split up into two subsets since there is only one pair of face uses with a large dihedral angle and using only one growth curve for this manifold will result in flat elements.

5.2. *Calculation of growth curves*

Growth curves from mesh vertices classified on model vertices and model edges are first attempted to be grown as boundary growth curves. In doing so, the growth curves must respect topological

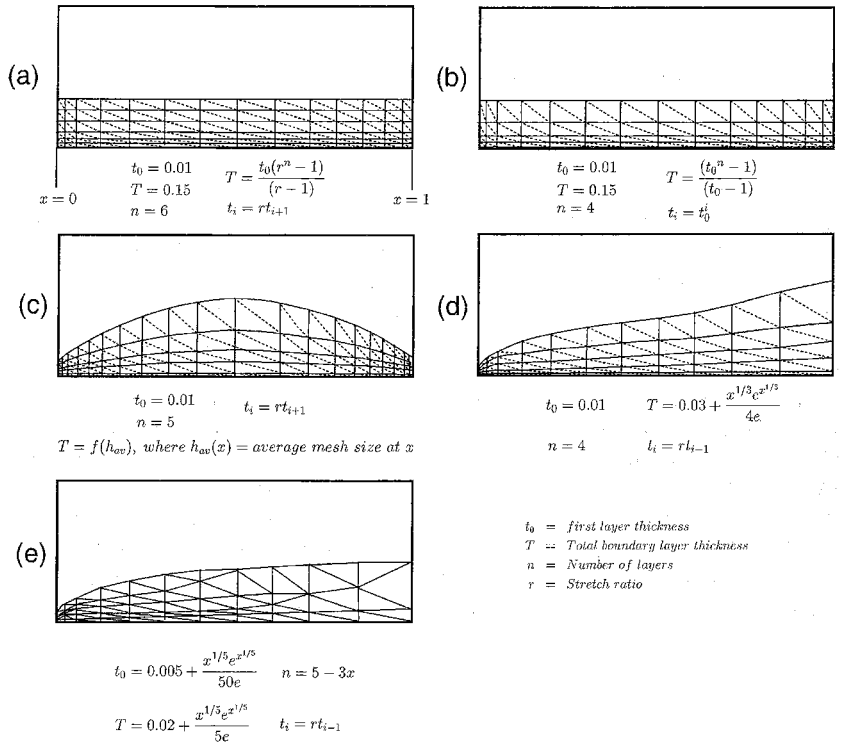

Figure 5. Methods of specifying boundary layers: (a) geometric variation of layer thickness; (b) exponential variation of layer thickness; (c) adaptively varying boundary layer thickness; (d) prescribed variation in boundary layer thickness; (e) prescribed variation of boundary layer thickness and number of layers.

compatibility of the mesh with the model and *estimated* geometric validity of mesh. If creating a boundary growth curve violates any of these requirements, the growth curve is grown into the interior.

In computing growth curves, it is assumed that all nodes of the growth curves except the first have a single classification on the lowest order model entity possible. For example, when constructing a growth curve from a mesh vertex classified on a model vertex, the lowest order model entity that can carry the growth curve is a connected model edge. Since model edges and faces may be curved, a straight line approximation of the growth curve (obtained from an average normal of the given mesh face uses) is used to find locations on the model entity close to the initial positions of the nodes.

An extensive set of checks is performed to ensure that the computed growth curve satisfies validity and quality requirements of the mesh. Checks are performed to ensure that future connections (mesh edges and faces) between the growth curve and any adjacent boundary growth curves will not violate topological compatibility. Also, dihedral angles of future elements resulting from the growth curve are estimated to ensure element quality. If two growth curves from a mesh vertex in a non-manifold model lie on the same model face, they are checked to see if they are

coincident and merged. If not, they are checked to ensure that boundary layer quads to be formed with them will not intersect each other. In case of intersection, the growth curve is not created and the other growth curve is used instead.

5.3. *Node spacing along the growth curves*

Node spacing for growth curves may be specified in one of three ways—*geometric*, *exponential* or *adaptive*. In the geometric method, the first layer thickness, the number of layers and the total thickness of the boundary layer mesh are prescribed. Using this, the thickness of the individual layers is calculated to grow by geometric progression (Figure 5(a)).

For exponential growth, only the first layer thickness and number of layers is specified for calculation of the node spacing (Figure 5(b)). The growth of the layer thicknesses is exponential.

In the adaptive method of boundary layer thickness specification, the first layer thickness t_0 and the number of layers, n, are specified. The growth of the boundary layer thickness is still geometric but the layer thickness growth factor r is calculated to ensure a smooth gradation of the boundary layer mesh into the isotropic mesh (Figure 5(c)). This is done by assuming the last layer thickness to be α times the isotropic mesh size, $0.5 < \alpha < 1.0$.

The attribute specification system used for prescribing boundary layer mesh parameters allows spatial variation of all the variables, t_0, T and n while maintaining the geometric growth rate of layer thicknesses (Figure 5(d)). Figure 5(e) shows the boundary layers when the boundary layer thickness and the number of layers both vary on a model entity.

6. ENSURING ELEMENT VALIDITY

Invalidity of elements in the generalized advancing layers method occurs due to invisibility of growth curve nodes from a mesh face and due to crossover of growth curves (Figure 6(a)). The former is dealt with during growth curve creation and the latter is dealt with after the creation of all growth curves. Growth curve crossover is addressed here by smoothing, shrinking and pruning applied in that order.

In the smoothing step (Figure 6(b)), a weighted Laplacian smoothing procedure is applied to growth curves to eliminate crossover. It is the preferred method of eliminating crossover since it respects the original spacing of nodes along the growth curves. Although smoothing distorts previously well shaped elements, it also corrects crossover in many cases and evens out shape and size variations in the boundary layer mesh. Smoothing of interior growth curves is done by reorienting each growth curve to the average of its adjacent growth curves. Smoothing of boundary growth curves is done by a modified procedure that accounts for their general shape. In this procedure, straight line approximations of the growth curve and its adjacent boundary growth curves are used for computing a smoothed direction and closest point searches done to locate the nodes of the growth curve onto the model boundary. Multiple passes of smoothing are used over each entity and over all the entities.

The shrinking procedure is based on the principle that crossover often occurs because the boundary layer is too thick relative to the 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 (Figure 6(c)[i]). This is accomplished by progressively reducing the node spacing of the boundary and interior growth curves which are

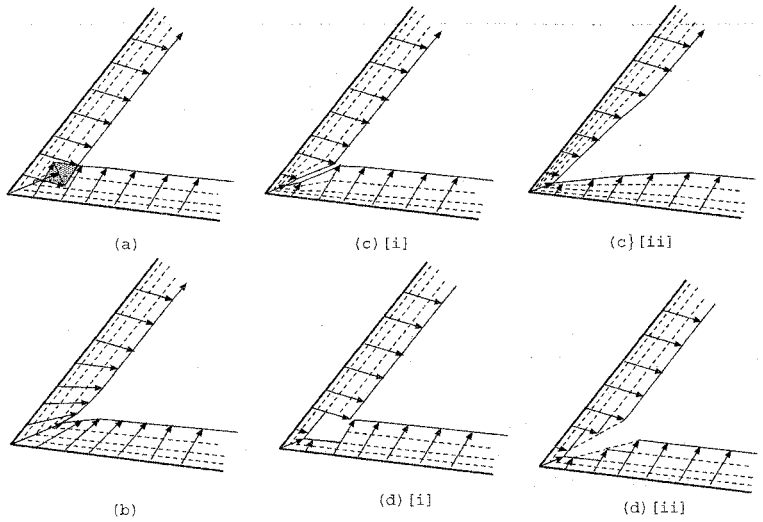

Figure 6. Fixing growth curve crossover in 2D mesh: (a) invalid mesh; (b) mesh fixed by smoothing; (c) [i] mesh fixed by shrinking, [ii] neighbouring growth curves recursively shrunk for smooth gradation; (d) [i] mesh fixed by pruning, [ii] neighbouring growth curves recursively pruned and the steps bridged by transition elements.

connected to invalid quads and prisms, respectively. The reduction in the height of the growth curve is always accompanied by a recursive adjustment of neighbouring growth curve heights to ensure a smooth gradation of boundary layer thickness (Figure 6(c)[ii]). Also, shrinking does not allow previously valid elements to become invalid. Multiple passes of the shrinking procedure are carried out on the boundary layer mesh to maximize the possibility of the fixing invalid elements. Shrinking growth curves on the boundary is similar to the interior shrinking with the modifications required to deal with the boundary.

If neither smoothing nor shrinking can fix the invalid elements, the growth curves of affected elements are pruned, i.e. some of their nodes are deleted. The process of pruning growth curves eliminates as many nodes as necessary starting from the top of the growth curve (Figure 6(d)[i]). As with shrinking, adjacent growth curves are also recursively pruned so that adjacent growth curves are allowed to differ by only one node (Figure 6(d)[ii]). This is done to prevent large steps in the boundary layer mesh which can affect mesh quality and gradation. At the end of the pruning procedure, the boundary layer mesh has no invalid elements.

The validity of boundary layer quadrilaterals is checked by a series of tests performed in real and parametric[¶] spaces. First, the individual triangles of the boundary layer quad are checked in real space for zero area[‖]. Then adjacent triangles are checked in real space 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, as a conservative

[¶]The parametric space is obtained from the geometric modeller.

[‖]Note that negative area does not have any meaning for triangles on a general surface in 3-space.

measure the lateral edges of the boundary layer quad are checked for intersection in parametric space to verify that the growth curves are not crossed over.

The validity of boundary layer prisms, solid transition elements and blends is done by checking the validity of all its component tetrahedra, i.e. all tetrahedra are checked to ensure that the worst dihedral angle is less than some angle, α. While minimum element validity is satisfied if $\alpha < \pi$, α is taken to be less than that value $(\pi - \pi/30)$ in the interest of creating elements of good quality.

7. ELEMENT CREATION

The primary construct in the creation of the boundary layer mesh is the triangular prism formed by connecting the nodes of three growth curves from the vertices of a mesh face. Other constructs are boundary layer transitions and blends. Any of these constructs may abut a model face and modify the surface triangulation. The equivalent constructs in the surface triangulation are boundary layer quads, blend triangles and transition triangles. All of these constructs are abstract mechanisms for generating the triangles and tetrahedra of the boundary layer mesh.

7.1. Triangulation of boundary layer quads

Boundary layer quads are formed 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 future validity of the connected prisms. For reasons discussed in prism tetrahedronization (Section 7.5 below), the diagonal is made so that it connects node l of the growth curve at the mesh vertex with a lower identifying number (vertex ID) to node $l+1$ of the growth curve at the other vertex. If one of the growth curves has more nodes than the other, transition triangles are formed on top of the quads. The procedure incorporates well-defined checks to deduce the classification of entities and to ensure the correct orientation of mesh edges and faces is present.

7.2. Creation of boundary layer transition triangles

Transition triangles are formed atop boundary layer quads with a level difference (difference in the number of nodes) between the two growth curves. This is done by simply connecting the top node of the growth curve with fewer nodes with nodes of the other growth curve which are at a higher level. Note that the mesh vertex with the lower ID may have the growth curve with more nodes and therefore, the diagonal edge of the transition triangle may go in an opposite direction to the diagonals of the boundary layer quad.

7.3. Creation of boundary layer blend triangles

Creation of boundary layer blend triangles is similar to the creation of boundary layer quads. The difference is that boundary layer blend triangles establish connections between nodes of two growth curves originating from the same mesh vertex. The first layer of a blend triangle is made up of a single triangular mesh face and the rest of the layers contain triangulated quads as in the case of the boundary layer quad. The direction of the diagonal for the quads in a boundary layer triangle is arbitrary and may be based solely on the quality of the triangles.

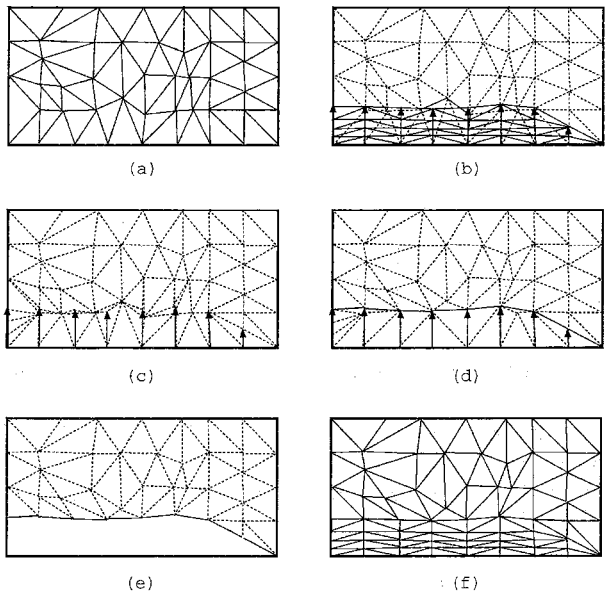

Figure 7. Model face retriangulation by local mesh modifications: (a) initial surface mesh; (b) surface mesh with boundary layer elements overlayed; (c) insertion of outermost boundary layer vertices into surface mesh; (d) recovery of outermost boundary layer edges by edge swapping and edge collapsing; (e) deletion of surface mesh triangles overlapping the boundary layer mesh; (f) incorporation of boundary layer mesh into surface mesh.

7.4. Model face retriangulation

When the boundary layer mesh interacts with a model face, the boundary layer triangulation must be incorporated into the appropriate portion of the face triangulation. The approach adopted here uses local mesh modification operators combined with checks for the smoothness of the surface discretization. This is done to avoid meshing by an advancing front method for highly distorted parameterizations since they may give incorrect results for checks for intersection of the front.

Given a model edge and a model face on which growth curves from mesh vertices of the model edge lie, the following steps are carried out to create and incorporate the boundary layer mesh into the surface mesh triangulation (Figure 7):

1. Boundary layer quads and triangles classified on the the model face are created (Figure 7(b).
2. Each boundary layer mesh entity that forms the outer periphery of the boundary layer mesh faces classified on the model face is incorporated into the surface mesh by the edge recovery procedure ([29]).

   If the edge cannot be recovered one or both growth curves of the appropriate quad are deleted and replaced with an interior growth curve(s). This has the effect of peeling the boundary layer away from the adjacent wall in the neighbourhood.

   At the end of the recovery process, the periphery of the set of boundary layer faces matches the outer boundary of a set of faces in the underlying surface mesh (Figures 7(c) and 7(d)).

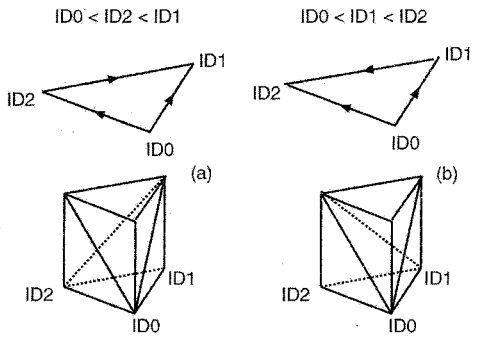

Figure 8. The two templates for prism tetrahedronization with node ID-based diagonal choice.

3. Mesh faces of the existing surface triangulation overlapping the boundary layer mesh faces are deleted (Figure 7(e)).
4. The boundary layer faces are incorporated into the surface mesh in place of the deleted elements (Figure 7(f)).

7.5. *Creation of boundary layer prisms*

The bulk of the elements in the boundary layer mesh are comprised of tetrahedronized layers of boundary layer prisms. Boundary layer prisms are grown on mesh face uses by connecting three suitable growth curves at the vertices of the face. The tetrahedronization of each boundary layer prism in a layer gives rise to three tetrahedra.

The tetrahedronization of triangular prisms depends on the diagonals on its lateral faces. There are eight possible combinations of diagonals for the lateral faces of a prism of which only six can be tetrahedronized without additional point insertions. The six templates can be further reduced to two by permutation of the prism faces as shown in Figure 8. Therefore, care must be taken to only assign diagonal directions such that all prisms can be tetrahedronized. This is done by a 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. Therefore, the diagonal of a boundary layer quad is always created between the lower node of the growth curve whose base vertex (in the surface mesh) has a lower ID to the upper node of the growth curve whose base vertex has a higher ID (shown in Figure 8).

7.6. *Creation of transition tetrahedra*

When the growth curves of a mesh face forming a prism have different number of nodes, a step (or a level difference) is formed in the boundary layer mesh exposing stretched faces to the isotropic mesh generator. This difference may come from user-requested variations in the number of nodes or due to pruning of growth curves. To avoid leaving highly stretched faces of the step exposed to the volume mesher, transition tetrahedra are created to bridge the one or more levels of difference in growth curves (see Figure 9). 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. If the level difference between the growth curves

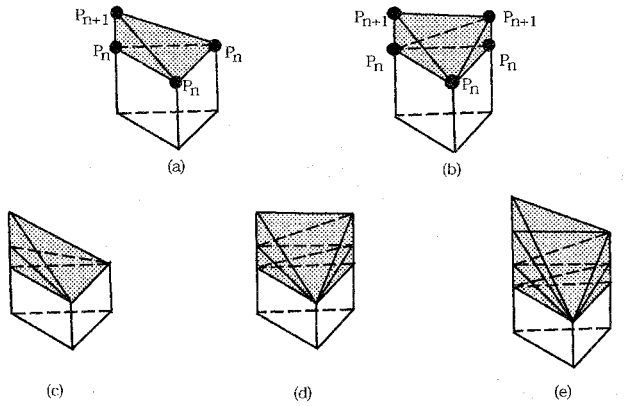

Figure 9. Transition elements: (a) one level type I transition layer; (b) one level type II transition layer; (c) multi-layer type I transition layers; (d) multi type II transition layers; (e) multi-level type II/I transition layers.

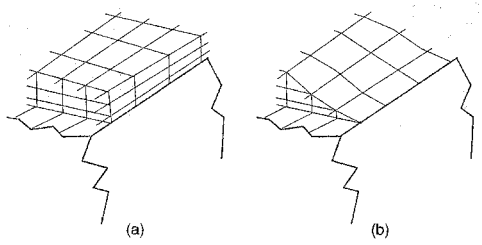

Figure 10. Transitioning of boundary layers at model edge: (a) boundary layer without elimination of exposed faces; (b) boundary layer with elimination of exposed faces by transitioning.

is more than one, layers of transition elements are created on top of the prism. One-level transition elements are better shaped than multiple-level one although the creation of either is preferable to leaving the stretched faces exposed. This is the reason why recursive pruning procedures are in place to create a one-level difference between growth curves as much as possible. The idea of a one-level transition element is similar to the procedure used in the work of Connell and Braaten [1] to phase out the boundary layer at some edges.

Transition elements are also useful in situations where the boundary layer mesh ends abruptly at a sharp corner. To prevent the isotropic volume mesher from seeing the stretched faces of the boundary layer mesh, the number of nodes along the sharp corner edge are reduced to zero and the boundary layer transitioned out from the edge (as described in Reference [1]). This is shown in Figure 10.

7.7. Creation of boundary layer blend polyhedra

The introduction of multiple growth curves at mesh vertices due to surface mesh geometry introduces gaps between adjacent prisms. These gaps are made up of highly stretched faces present on the sides of prisms. If left as they are, the highly anisotropic faces of the gaps cause problems for

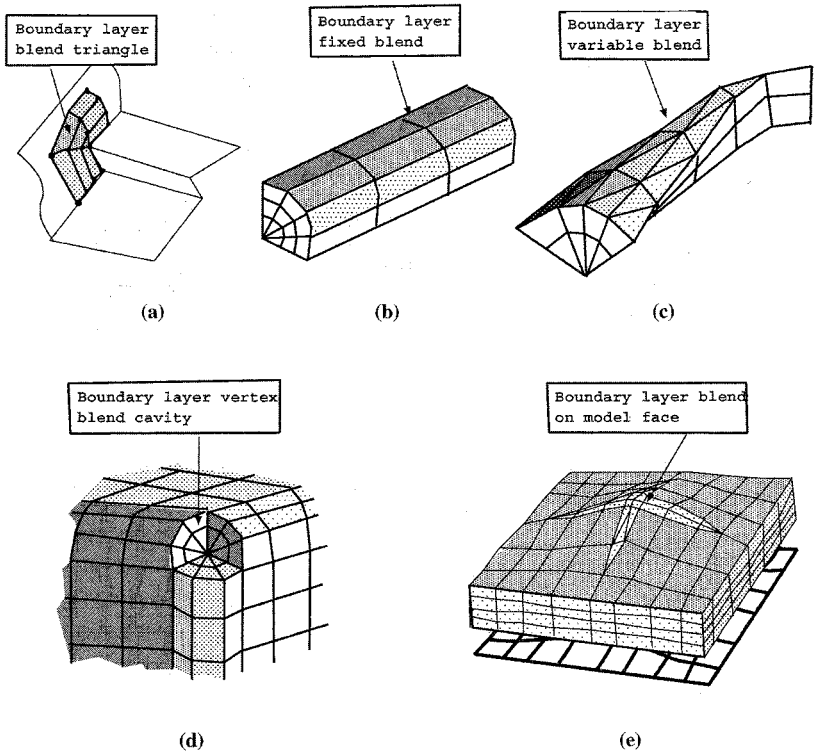

Figure 11. Boundary layer blend elements.

the isotropic volume mesher. Therefore, the concept of blend meshes is introduced in the generalized advancing layers procedure to fill the gaps between prisms while maintaining a good mesh gradation in the boundary layer mesh.

Consider a situation in two dimensions where there are two growth curves at a model vertex representing a convex corner (Figures 12(a) and 12(b)). The gap in the two-dimensional boundary layer mesh formed at this vertex must be filled by blend triangles (Figure 12(c)). Also, to maintain a good mesh gradation on this outer surface, additional growth curves may have to be introduced at a mesh vertex in between the growth curves used by the standard boundary layer elements as shown in Figure 12(d). The number of additional growth curves required at a mesh vertex can be calculated using local information and the assumption that the outer faces of the blends must be nearly isotropic.

In three dimensions, gaps between prisms may occur at model faces, edges and vertices. At model edges, the dihedral angle between two model faces connected to the model edge may be constant or vary continuously. If the dihedral angle is constant, then the same number of growth curves are created at each mesh vertex classified on the model edge. Such blend mesh with a fixed topology along a model edge is referred to as a *fixed blend* (Figure 11(b)). If the dihedral angle varies along the model edge then there may be more growth curves at some mesh vertices

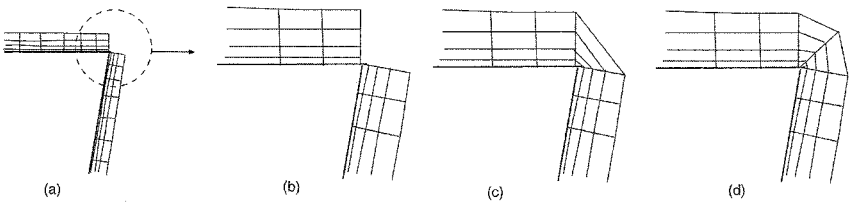

Figure 12. Two-dimensional illustration of the need for blends and multiple growth curves within blends: (a) boundary layer mesh on two surfaces with convex corner; (b) gap between the corners shown in greater detail; (c) blend mesh directly bridging the two existing growth curves; (d) blend mesh with introduction of additional growth curve.

than others. In this case the topology of the blend mesh must also change along the model edge resulting in *variable blends* (Figure 11(c)).

Gaps between the various boundary layer constructs at model vertices are more general since an arbitrary number of prisms and blends can contribute to them (Figure 11(e)). However, since their anisotropy is much lesser than edge blends, it is proposed that vertex blends will be created using general mesh generation techniques. Also, blends may also occur at model faces wherever multiple growth curves are present but they can be dealt with similarly (Figure 11(e)). Blend meshes are not yet present in the current implementation but will be introduced as described above.

8. FIXING BOUNDARY LAYER INTERSECTIONS

When boundary layer elements are generated on model faces that are too close they may overlap and the polyhedral cavity that remains to be meshed may self-intersect. These self-intersections must be fixed before handing the mesh over to the volume mesh generator.

After creation of the boundary layer elements, mesh faces that have fewer regions connected to them than necessary are considered to be exposed to the isotropic mesher. Exposed faces may be faces of the boundary layer mesh or faces on model faces. Exposed faces are checked for intersection with other exposed faces in the neighbourhood. If an intersection is found, its connected prisms are shrunk to fix the intersection if the connected elements are valid in the new configuration. Also, a recursive adjustment procedure is applied to the heights of other growth curves in the neighbourhood. Intersections that cannot be fixed after multiple shrinking iterations are fixed by pruning of growth curves. If the pruning step creates level differences between growth curves, transition elements are created to bridge the step. The process of fixing boundary layer run-in is illustrated in Figure 13 using a 2D example. Correction of self intersections is done after element creation since detection of exposed faces is simpler and the checks for self-intersections can be more localized. The intersection checks are further localized with the help of an octree.

9. BOUNDARY LAYER MESHING RESULTS

9.1. Example meshes for general models

The generalized advancing layers method described here has been used extensively to generate boundary layer meshes for thermal management simulations of complex automobile configurations

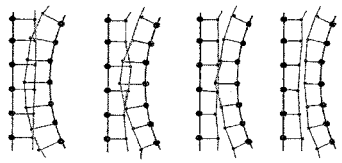

Figure 13. Iterative procedure for fixing boundary layer intersections—2D example.

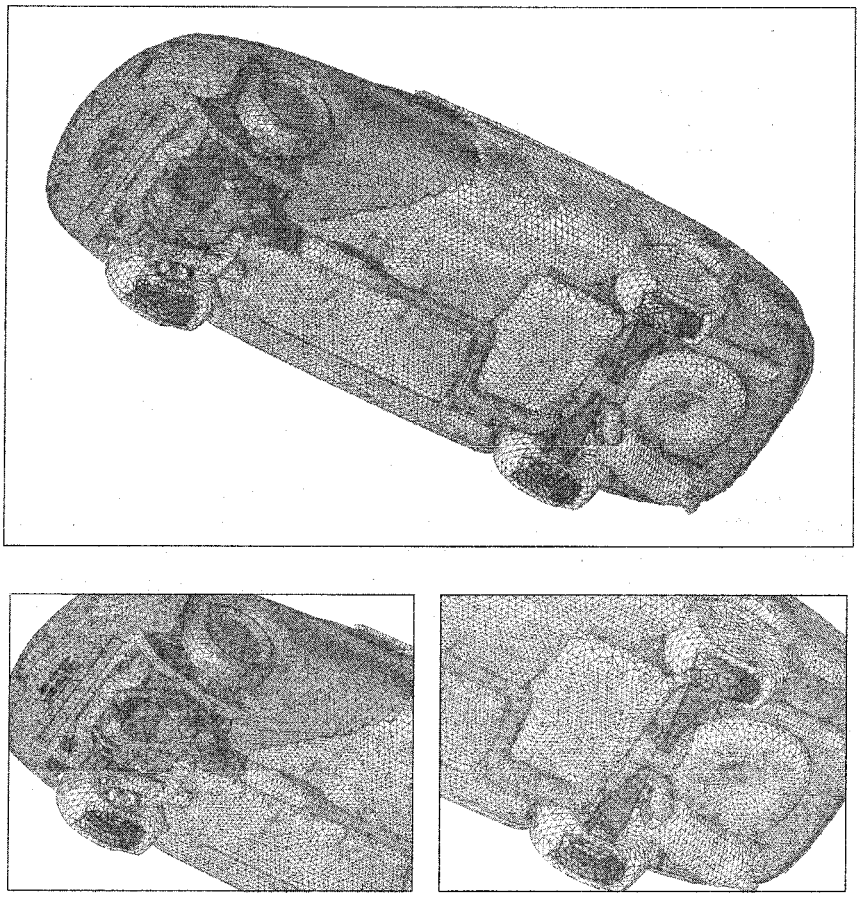

Figure 14. Boundary layer mesh for under-carriage of car: (a) complete boundary layer mesh on all surfaces of car; (b) cut away of boundary layer mesh revealing under-the-hood detail; (c) zoom in of front end of under-carriage; (d) zoom in of rear end of car.

with complete under-the-hood and under-carriage detail. The boundary layer mesh on the underbody of one such vehicle** is shown in Figure 14(a) with part of the boundary layer cut away to show the complexity of the surface. The model is a non-manifold model with 11 model regions and 1236 model faces of which 71 are embedded faces. In the mesh shown here, the boundary layer thickness was increased by 2 orders of magnitude for clarity of visualization. The boundary layer mesh and the full solid mesh have 1.7 and 3.1 million tetrahedra, respectively. The *aspect ratio* (i.e. the longest edge length to shortest height ratio) of elements in the first layer are approximately 2500 on the most coarsely refined surfaces of the automobile. Figure 14(b) shows a close-up of the surface mesh and boundary layer mesh under the hood and near the front wheels while Figure 14(c) shows the under-carriage at the rear. The largest meshes generated for these types of models have been of the order of 4.5 million elements.

The next example shows the use of the boundary layer mesh in simulations of flow in blood vessels for surgical planning [30]. Figure 15(a) shows the model†† of the arteries while Figure 15(b) shows a zoom-in of the surface mesh. Figures 15(c) and 15(d) display various cuts through the mesh showing the boundary layer and volume mesh inside the arteries (650 000 tetrahedra in the boundary layer, 800 000 total). The total boundary layer thickness is determined adaptively in the mesh based on the surface mesh size.

The example shown next is a model of the space shuttle with centre tank and booster rockets. Figure 16(a) shows the geometric model (without the boundaries of the enclosing domain). Shown in Figures 16(b), 16(c) and 16(d) are the retriangulated surface mesh on the symmetry plane, a cut-way of the boundary layer mesh and a close-up of the boundary layers showing the element anisotropy. The boundary layer mesh in this model has 810 000 elements while the complete mesh has 1 million elements. With the requested surface mesh sizes aspect ratio of the boundary layer elements is of the order of 20 000.

9.2. *Validation*

9.2.1. Laminar flow over flat plate. Figure 17(a) shows a schematic for simulation of laminar flow of an incompressible viscous fluid over a semi-infinite flat plate. The flow at the inlet is uniform ($Re = 10\,000$) in the direction of the x-axis. The domain starts ahead of the plate to capture the flow characteristics around the singular point at the leading edge of the plate. The expected solution in the domain is shown in Figure 17(b) [31]. The boundary layer has a first layer thickness of $\max(3\times10^{-4}, 0.00015\sqrt{x})$, total boundary layer thickness of $\max(0.006, 0.052\sqrt{x})$ and 20 layers of elements. The boundary layer mesh and a zoom-in of the mesh around the singular point is shown in Figure 18. The solution obtained for this problem is shown in Figure 19. Figure 19(a)(i) shows the constant u-velocity contours on the front face while Figure 19(a)(ii) shows a zoom-in of the domain near the singularity. In Figure 19(c) a close-up of the outflow boundary along with u-velocity profile at $x = 1.0$ is shown and is in good agreement to the expected profile. The results of the simulation have been validated using the similarity solution [31] (see [32] for detailed results).

9.2.2. Turbulent flow in sharply expanding pipe. The schematic for the simulation of turbulent flow in a sharply expanding pipe is shown in Figure 20(a). Fluid enters the narrow pipe which

**Courtesy: Simmetrix Inc.

††Courtesy: Dr. Charles Taylor, Assistant Professor, Department of Surgery and Department of Mechanical Engineering, Stanford University.

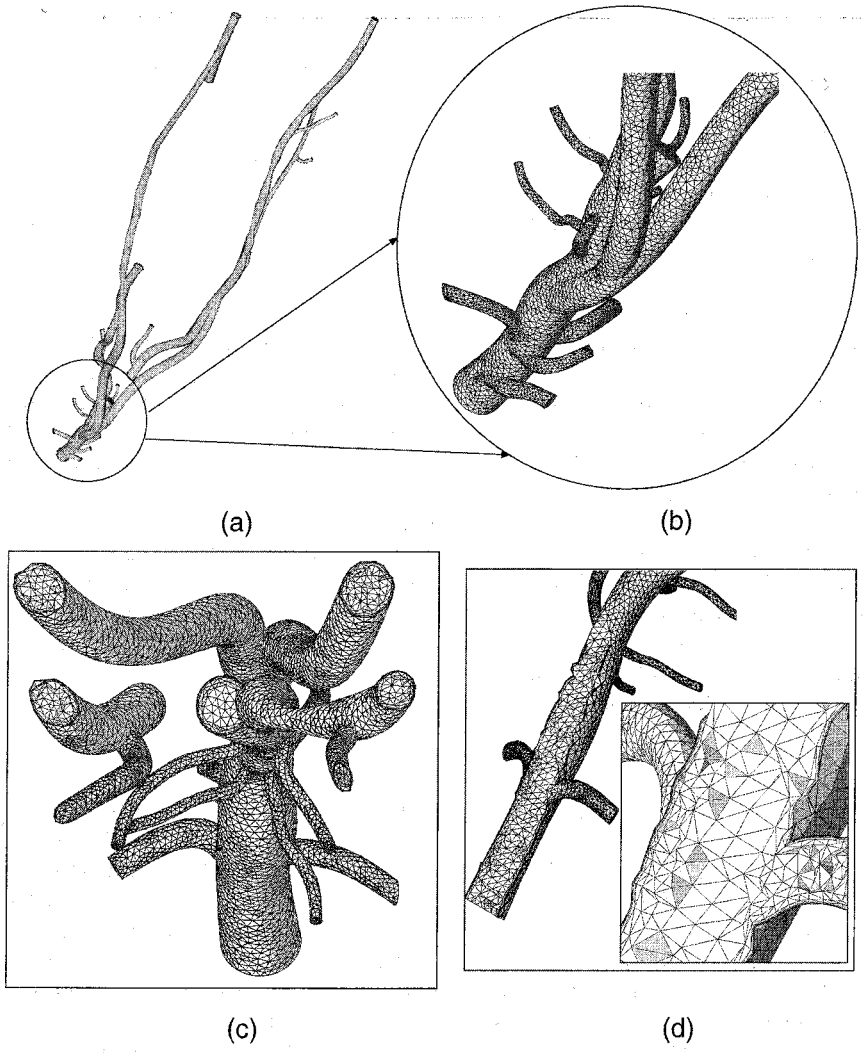

Figure 15. Boundary layer mesh for simulation of flow in blood vessels: (a) geometric model; (b) zoom in of surface mesh in the encircled region; (c);(d) cross-sections showing the boundary layer and isotropic meshes.

is connected to a large pipe without transition. In addition to the boundary layers on the pipe walls, a free shear layer is expected in the flow leaving the walls at the junction of the two pipes and reattaching to the walls of the large pipe further downstream. A recirculation region is expected behind the shear layers as shown in the figure. A schematic of a vertical cross-section of the geometric model is shown in Figure 20(b). Since the generalized advancing layers can build boundary layer meshes only on model boundaries, an artificial surface is defined in the larger pipe

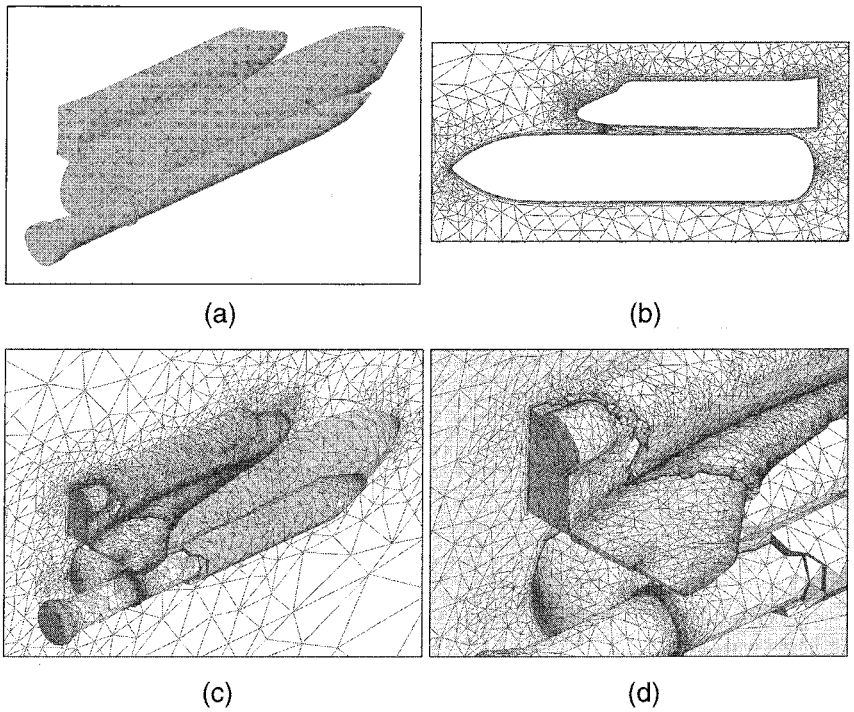

Figure 16. Boundary layer mesh for space shuttle: (a) model geometry; (b) retriangulated surface mesh; (c) cut away of boundary layer mesh; (d) close-up of boundary layer showing anisotropic elements.

based on an estimate of the shear layer path. The large pipe in the geometric model is only a third of the required length and the mesh that is generated in this part is stretched to match the original domain definition. This is done to keep the size of the mesh low and make use of the inherent anisotropy of the whole solution.

The surface mesh and the streamwise velocity contours are shown in Figure 21. Boundary layer elements are created on walls of the small pipe, on both sides of the shear layer surface and on the wall of the large pipe downstream of the reattachment point. The boundary layer thickness and number of nodes are different on the various model faces (and on each side of the shear layer face). The thickness and number of the layers also vary as a function of the x coordinate. The boundary layer mesh has 1.6 million elements.

9.3. Timing statistics

The generalized advancing layers method has been observed to produce elements at an average rate of 1000 elements per second or 3.6 million elements an hour on SUN Ultra Sparc 2 workstation. The maximum obtained rate of mesh generation is 2200 tetrahedra per second. The growth rate of the algorithm with respect to the number of surface triangles and the number of layers has been observed to be $O(N \log N)$ as expected due to the use of a search tree to resolve intersections [32].

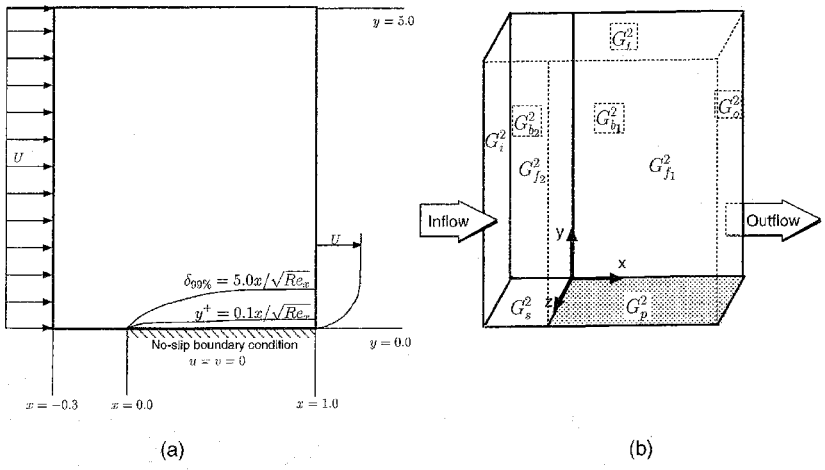

Figure 17. Schematic diagrams of set-up for simulation of laminar flow over flat plate: (a) schematic description of domain and important boundary conditions; (b) schematic diagram of geometric model (not to scale).

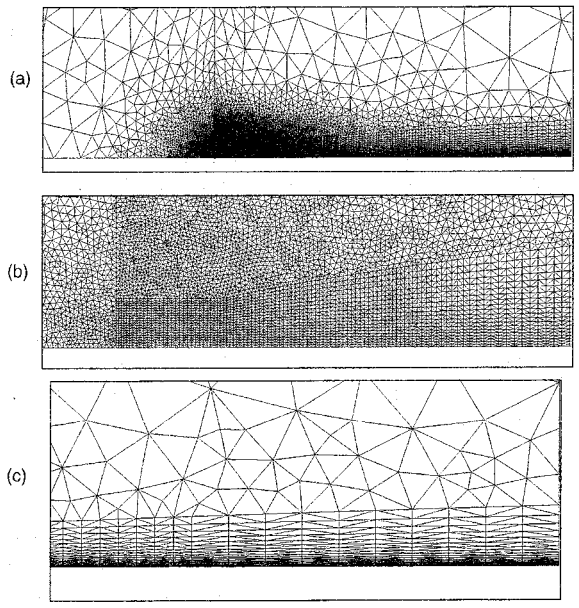

Figure 18. Close-up views of boundary layer mesh for laminar flow over flat plate simulation.

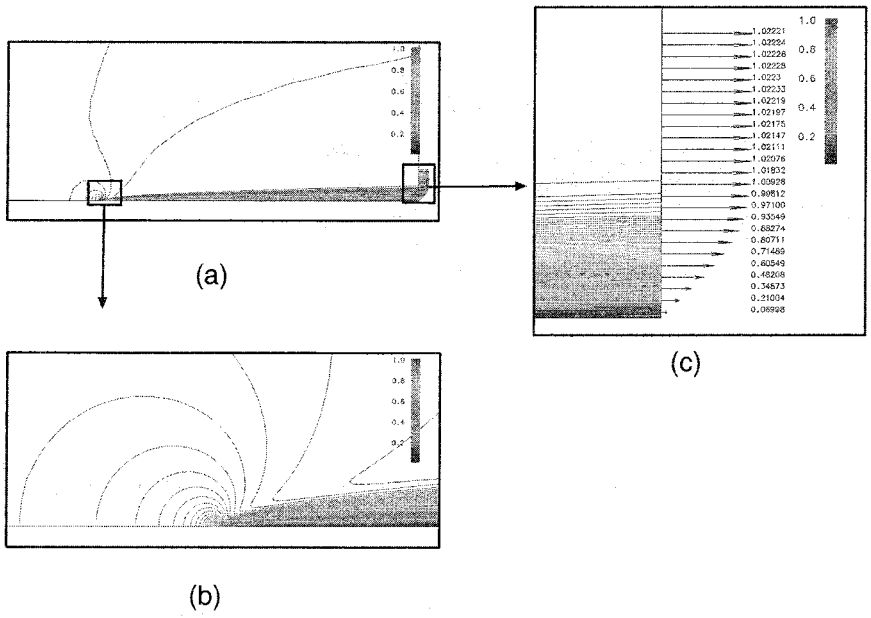

Figure 19. u-velocity contours and profile for laminar flow over flat plate: (a) u-velocity contours; (b) close-up view at singular point; (c) profile of u-velocity at outflow.

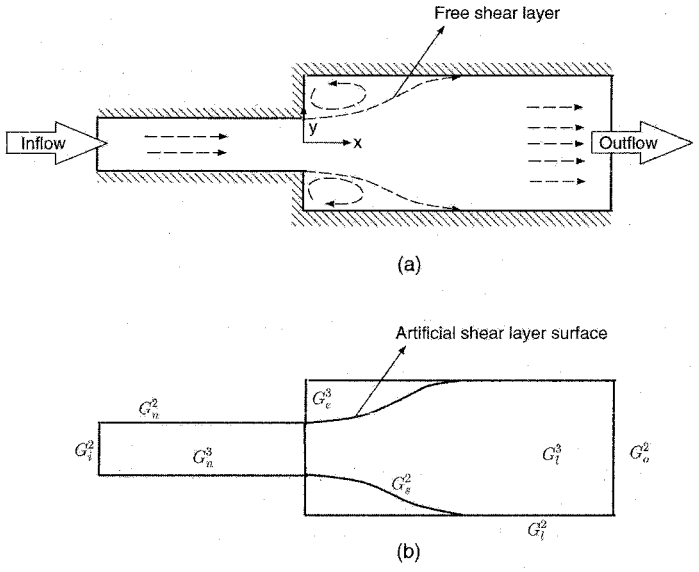

Figure 20. Schematic diagram of expanding pipe model: (a) problem domain; (b) geometric model cross-section.

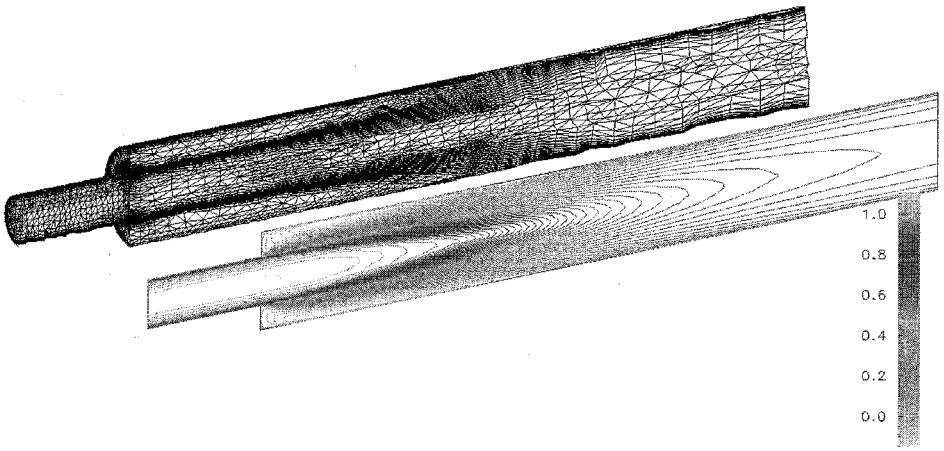

Figure 21. Cut view of mesh and streamwise velocity contours for simulation of flow in expanding pipe.

10. CLOSING REMARKS AND FUTURE WORK

10.1. Concluding remarks

The generalized advancing layers method has been presented as a way of generating anisotropic tetrahedral meshes for capturing viscous flows. The method, is designed for reliable generation of valid, good-quality meshes for arbitrarily complex non-manifold geometric models. It includes several technical advances to be able to handle complex domains. It provides control and flexibility in the creation of meshes suitable for fluid flow simulations.

The Generalized Advancing Layers Method creates valid meshes for general models by:

1. Ensuring that a minimum number of growth curves are present at non-manifold interfaces.
2. Guaranteeing all elements in the mesh are geometrically valid.
3. Ensuring that the boundary layer mesh is topologically compatible with the geometric model.
4. Using a robust procedure for surface retriangulation avoiding the use of the parametric space.
5. Using a guaranteed algorithm for resolving intersections of boundary layers.

The quality (mesh gradation, element dihedral angles) of the mesh is controlled in a comprehensive manner as listed below:

1. Multiple growth curves to allow better-shaped prisms at sharp corners.
2. Smoothing of growth curves to improve the overall quality of boundary layer elements.
3. Recursive adjustment of growth curve heights for better mesh gradation and quality.
4. Recursive pruning of growth curves to maintain one-level difference minimizing the use of multi-level transition elements.
5. Incorporation of boundary layer mesh into adjacent surface triangulation allowing prisms to remain close to their optimal shape.
6. Transition and blend elements to shield the isotropic mesher from highly anisotropic faces.

Results have been presented to demonstrate the capability of the mesh generator to mesh complex non-manifold models. Results of two classical problems in fluid mechanics were presented to demonstrate the suitability of the mesh for viscous flow simulations and its ability to capture the solution accurately. The generalized advancing layers method has successfully generated meshes of the order of 3–4 million elements for other large complex geometric models and is currently being used for simulations on real automobile configurations in industry.

REFERENCES

1. Connell SD, Braaten ME. Semistructured mesh generation for three dimensional Navier–Stokes calculations. *AIAA Journal* 1995; **33**(6):1017–1024.
2. Kallinderis Y, Ward S. Prismatic grid generation for three-dimensional complex geometries. *AIAA Journal*, 1993; **31**(10):1850–1856.
3. Löhner R. Matching semi-structured and unstructured grids for Navier–Stokes calculations. In *AIAA-93-3348-CP*, 1995.
4. Pirzadeh S. Viscous unstructured three-dimensional grids by the advancing-layers method. In *Proceedings of the 32nd Aerospace Sciences Meeting and Exhibit*, AIAA-94-0417, Reno, NV, January 1994.
5. Borouchaki H, George PL, Hecht F. Laug P, Saltel E. Delaunay mesh generation governed by metric specifications. Part I. Algorithms. *Finite Elements in Analysis and Design* 1997; **25**:61–83.
6. George P-L, Borouchaki H. *Delaunay Triangulation and Meshing—Application to Finite Elements*. 8, quai du Marché-Neuf, 7500, Hermes, Paris, 1998.
7. Baker TJ. Automatic mesh generation for complex three-dimensional regions using a constrained Delaunay triangulation. *Engineering with Computers* 1989; **5**:161–175.
8. Joe B. Delaunay versus max-min solid angle triangulations for three-dimensional mesh generation. *International Journal for Numerical Methods in Engineering* 1991; **31**:987–997.
9. Löhner R, Parikh P. Three-dimensional grid generation by the advancing front method. *International Journal for Numerical Methods in Engineering* 1988; **8**:1135–1149.
10. Jin H, Tanner RI. Generation of unstructured tetrahedra by advancing front technique. *International Journal for Numerical Methods in Engineering* 1993; **36**:1805–1823.
11. Möller P, Hansbo P. On advancing front mesh generation in three dimensions. *International Journal for Numerical Methods in Engineering* 1995; **38**:3551–3569.
12. Mavriplis DJ. Adaptive mesh generation for viscous flows using delaunay triangulation. *Journal of Computational Physics* 1990; **90**:271–291.
13. Vallet MG, Hecht F, Mantel B. Anisotropic control of mesh generation based upon a Voronoi type method. In *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, Arcilla AS, Häuser J, Eiseman PR, Thompson JF (eds). Elsevier Science Publishers B.V. (North-Holland): Amsterdam, 1991; 93–103.
14. Castro-Diaz MJ, Hecht F, Mohammadi B, Pironneau O. Anisotropic unstructured mesh adaptation for flow simulations. *International Journal for Numerical Methods in Engineering* 1997; **25**(4):475.
15. Hassan O, Probert EJ, Morgan K, Peraire J. Mesh generation and adaptivity for the solution of compressible viscous high speed flows. *International Journal for Numerical Methods in Engineering* 1995; **38**(7):1123–1148.
16. Hassan O, Morgan K, Probert EJ, Peraire J. Unstructured tetrahedral mesh generation for three-dimensional viscous flows. *International Journal for Numerical Methods in Engineering* 1996; **39**:549–567.
17. Marcum DL. Generation of unstructured grids for viscous flow applications. *AIAA-95-0212*, 1995.
18. Löhner R. Generation of unstructured grids suitable for rans calculations. In *Computational Mechanics – New Trends and Applications*, Idelsohn S, Oñate E, Dvorkin E (eds). CIMNE: Barcelona, Spain, 1998; 1–11.
19. Khawaja A, McMorris H, Kallinderis Y. Hybrid grids for viscous flows around complex 3-D geometries including multiple bodies. *AIAA-95-1685*, 1995; 424–441.
20. Sharov D, Nakahashi K. Hybrid prismatic/tetrahedral grid generation for viscous flow applications. *AIAA Journal* 1998; **36**(2):157–162.
21. de Cougny HL, Shephard MS. Parallel unstructured grid generation. In *CRC Handbook of Grid Generation*. CRC Press: Boca Raton, FL, 1999; 24.10–24.16.
22. Mäntylä M. *Introduction to Solid Modeling*. Computer Science Press: Rockville, Maryland, 1988.
23. Weiler KJ. The radial-edge structure: a topological representation for non-manifold geometric boundary representations. In *Geometric Modeling for CAD Applications*, Wozny MJ, McLaughlin HW, Encarnacao JL (eds). North-Holland, Amsterdam, 1988; 3–36.
24. Beall MW. Framework for the reliable automated solution of problems in mathematical physics over aribitrary domains using scalable parallel adaptive techniques. *Ph.D. Thesis*, Rensselaer Polytechnic Institute, Troy, NY 12180, 1999.
25. Schroeder WJ, Shephard MS. On rigorous conditions for automatically generated finite element meshes. In *Product Modeling for Computer-Aided Design and Manufacturing*, Turner J, Pegna J, Wozny M (eds). North-Holland: Amsterdam, 1991; 267–281.

26. Shephard MS, Georges MK. Reliability of automatic 3-D mesh generation. *Computer Methods in Applied Mechanics and Engineering* 1992; **101**:443–462.
27. Beall MW, Shephard MS. A general topology-based mesh data structure. *International Journal for Numerical Methods in Engineering* 1997; **40**(9):1573–1596.
28. Kallinderis Y, Khawaja A, McMorris H. Hybrid prismatic/tetrahedral grid generation for complex 3-D geometries. *AIAA-95-0211*, 1995.
29. Karamete BK, Garimella R, Shephard MS. Recovery of an arbitrary edge on an existing surface mesh using local mesh modifications. *SCOREC Technical Report* #19-1998.
30. Taylor CA, Hughes TJR, Zarins CK. Finite element modeling of blood flow in arteries. *Computer Methods in Applied Mechanics and Engineering* 1998; **158**(1-2):155–196.
31. White FM. *Viscous Fluid Flow* (2nd edn). McGraw Hill, Inc: New York, 1991.
32. Garimella RV. Anisotropic tetrahedral mesh generation. *Ph.D. Thesis*, Rensselaer Polytechnic Institute, Troy, NY 12180, December 1998.
33. de Cougny HL. Parallel unstructured distributed three dimensional mesh generation. *Ph.D. Thesis*, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, Dec. 1997.

