INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng. **47**, 169–190 (2000)

Meshing environment for geometry-based analysis

Mark S. Shephard*

Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, U.S.A.

SUMMARY

A geometry-based automatic mesh generation and control environment is presented. The environment consists of a set of tools that support the automatic creation and modification of simplex meshes directly from non-manifold solid models. The environment builds on three flexible object-oriented structures representing the geometric model, attribute information associated with the model, and the mesh. The mesh generation functions are layered and can be tailored to generate controlled simplex meshes for a wide range of problem types, including those that need curvilinear meshes or highly anisotropic meshes. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: mesh generation; meshing tools

1. INTRODUCTION

This paper was prepared in commemoration of Richard H. Gallagher, the author's thesis advisor and mentor. One lesson Professor Gallagher taught his students is the importance of examining a problem for both the technical challenges it presents, and the importance of its solution to the engineering community. The topic of this paper, automatic mesh generation, represents an area for which developments have been driven by both its technical challenges and need by the engineering community. It was clear in the late 1970's that the generation of finite element meshes represented an unacceptable bottleneck in the application of finite element methods in engineering practice. It was also clear that the approaches being used at that time were not capable of eliminating that bottleneck. What was not clear at that time was that it would take nearly 20 years to develop and mature alternative techniques [1–6] to the point that they would start being accepted in engineering practice.

Professor Gallagher also taught his students that a topic should be developed to a level of depth to ensure that the fundamental issues important to its completion were resolved. Automatic three-dimensional mesh generation again proved to be a topic that demonstrated the importance of examining a topic to this level of depth. In the early 1980s several authors presented papers which described the process of automatically creating three-dimensional meshes by the techniques that are at the heart of today's automatic mesh generators. What took another decade to resolve were the less obvious issues of meshing domains as defined by the CAD systems engineers use, and dealing with the numerical tolerance issues critical to ensuring the reliability of the implementation of an automatic mesh generator.

---

*Correspondence to: Mark S. Shephard, Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, U.S.A. E-mail: shephard@scorec.rpi.edu

CCC 0029-5981/2000/010169-22$17.50

Received 10 December 1998

This paper overviews a Meshing Environment for Geometry-based Analysis (MEGA). MEGA supports mesh generation starting from basic geometric definitions including toleranced non-manifold geometric models and discretized models. Key capabilities needed to support this process include: (i) general structures to define the geometric and mesh models and their interactions, (ii) a mechanism to directly interact with the geometric representations, and (iii) definition of what constitutes a valid finite element mesh for a given model. Section 2 overviews these issues.

Given an appropriate set of structures, it is possible to view the process of automatic mesh generation as a series of operations acting on the current state of the representations. Section 3 discusses the basic operations needed to support this approach. By basing MEGA on these structures and operators, a complete set of mesh generation and modification procedures can be constructed through the selection, ordering and control of those operations as dictated by the goals of the mesh generation process.

Section 4 overviews the algorithmic procedures used to generate meshes from solid models. This includes the surface mesh generator used to mesh the boundary of the model, the volume mesh generator used to fill the interior mesh, and the boundary layer mesh generator which can create highly anisotropic meshes in the vicinity of model faces.

Section 5 discusses procedures to perform modifications to a given mesh to satisfy specific criteria. The most basic of these is the ability to improve the quality of the elements in the mesh with respect to a given shape criteria. An extension to this procedure allows the mesh to override the adverse influence of small geometric model features while maintaining the validity of the mesh in the context outlined. Another procedure can modify any compatible surface triangulation to create surface triangles with the sizes and gradations desired, while maintaining proper representation of the geometric model features. An adaptive enrichment procedure can be used to refine and coarsen the mesh as dictated by the correction indicators provided by an adaptive analysis process. Another procedure curves the element geometries to properly support high order p-version analysis. Finally, a procedure which can modify a given mesh to insure there are multiple elements through the thickness of thin model sections is outlined.

Although not discussed in the current paper most of the MEGA procedures operate in parallel on distributed or shared memory computers using MPI for interprocessor communications. The interested reader is referred to in References [7–12].

2. GEOMETRY-BASED AUTOMATIC MESH GENERATION

Mesh generation is the controlled discretization of a geometric domain into a set of topologically simple entities which satisfy specific size and compatibility constraints. In today's engineering design environment the geometric domain is typically defined in terms of a solid model representation housed within a computer aided design system. The starting domain definition may also come from processed image data, or some other pre-discretized form. To be considered automatic, a mesh generator must be able to interact with the given domain representation to produce the resulting mesh.

2.1. Representing the geometric model

Since the process of generating a mesh is dominated by ensuring the mesh yields a valid discretization of the boundary of the domain, a boundary representation of the domain is a natural choice. Since the domains users wish to mesh for purposes of analysis can be combinations of volumes, surfaces and edges bounding each other in arbitrary manners, MEGA uses a non-manifold boundary representation [13] to describe the topological adjacencies of the boundary entities. The boundary representation is used to control the interactions between the mesh generation operations and geometric modeller, and to support the linkage to the mesh's topological entities to the model's topological entities. By using this representation, all components of MEGA are independent of the specifics of any one modeller implementation. In addition, the model representation can provide representational extensions that may not exist in the specific modeller, even though the modeller can support the underlying geometric operations required. The most common situation of this type is the case where the geometric modeller's base topological is a 2-manifold representation, but ad-hoc modelling functions are supported for specific, but important, non-manifold situations such as multiple material regions.

Another advantage of maintaining an independent topological representation is that it can support related multiple representations of the same geometry. The multiple levels of model representation are used for a variety of functions including conversion of assemblies into proper non-manifold models, support the elimination of small geometric features (see Section 5) and supporting mesh partition construction as used in parallel computations. The model abstraction also provides a source for additional non-geometric information in the form of attributes that are used to guide the meshing process.

2.2. *Representing meshes*

MEGA uses an object-oriented topological representation for the mesh. Geometrically mesh entity can be assigned any shape of the correct dimension. Since the number of topological entities in a mesh is very large compared to the geometric model, it is desirable to be able to focus only on the primary topological entities of regions, faces, edges and vertices, avoiding the need to include loop and shell topological entities, and, in the case of non-manifold models, entity uses for the vertices, edges, loops, and faces [14].

Under the assumption that each topological mesh entity of dimension d, M_i^d, is bounded by a set of topological mesh entities of dimension $d-1$, $M_i^d\{M^{d-1}\}$, the full set of mesh topological entities are:

$$T_M = \{M\{M^0\}, M\{M^1\}, M\{M^2\}, M\{M^3\}\}$$

where $M\{M^d\}$, $d = 0, 1, 2, 3$, are, respectively, the set of vertices, edges, faces and regions which define the primary topological elements of the mesh domain. It is possible to limit the mesh representation to just these entities under the following set of restrictions [14]:

1. Regions and faces have no interior holes.
2. Each entity of order d_i in a mesh, M^{d_i}, may use a particular entity of lower order, $M^{d_j}, d_j < d_i$, at most once.
3. For any entity $M_i^{d_i}$ there is a unique set of entities of order $d_i - 1$, $M_i^{d_i}\langle M^{d_{i-1}}\rangle$ that are on the boundary of $M_i^{d_i}$ if at least one member of $M_i^{d_i}\langle M^{d_{i-1}}\rangle$ classified on $G_j^{d_j}$ where $d_j \geqslant d_i$.

The first restriction means that regions may be directly represented by the faces that bound them, and faces may be represented by the edges that bound them. The second restriction allows the orientation of an entity to be defined in terms of its boundary entities (without the introduction of entity uses). For example, the orientation of an edge, M_i^l bounded by vertices M_j^0 and M_k^0 is uniquely defined as going from M_j^0 to M_k^0 only if $j \neq k$.

The third restriction means that an interior entity (defined as $M_i^{d_i} \sqsubset G_i^{d_j}$ where, $d_j \geqslant d_i$ and at least one of $\partial(M_i^{d_i}) \sqsubset G_i^{d_j}$) is uniquely specified by its bounding entities. This condition only applies to interior entities; entities on the boundary of the model may have a non-unique set of boundary entities as illustrated with a model and a coarse mesh of a plate with a hole in Figure 1. Here, the mesh is sufficiently coarse yielding the mesh and model topology identical on the hole boundary. The two mesh edges, M_1^1 and M_2^1, on the hole boundary have the same set of vertices, M_l^0 and M_l^0.

2.3. Relating meshes to models

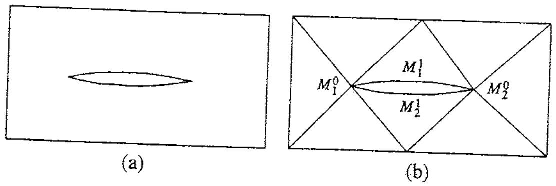
A key component of automatic mesh generation directly from a solid model representation is ensuring the validity of the mesh with respect to the geometric model [14–16]. The procedures that have been devised to address this issue focus their attention on understanding the association of the mesh to the geometric model. This association is referred to as classification in which the mesh topological entities are classified with respect to the geometric model topological entities upon which they lie.

Definition (Classification). The unique association of mesh topological entities of dimension d_i, $M_i^{d_i}$ to the topological entity of the geometric model of dimension d_j, $G_j^{d_j}$ where $d_i \leqslant d_j$, on which it lies is termed classification and is denoted $M_i^{d_i} \sqsubset G_j^{d_j}$ where the classification symbol, $\sqsubset$, indicates that the left hand entity, or set, is classified on the right hand entity.

The concept of classification is used by all mesh generation and modification procedures to ensure the robustness of the procedures. It is a key ingredient to the definition of a valid mesh in terms of a *geometric triangulation* which can be obtained through an assurance algorithm. References 15, 16 describe the technical details of this process.

3. BASE OPERATIONS USED BY MEGA

Over the years papers have presented what appears to be a number of alternative approaches to automatic mesh generation. An examination of the functional components of these alternative procedures shows that they all rely on a reasonably small number of basic operations and the differences in approaches relate to criteria driving decisions, the order in which operations are applied, and underlying structures to support ensuring the computational efficiency of the method. To take advantage of this, MEGA has attempted to construct all mesh generation and



(a) (b)

Figure 1. Example of mesh entities on the boundary having non-unique boundary entities: (a) geometric model; and (b) mesh

mesh modification operations in terms of the base operations. This strategy allows MEGA to effectively implement various combinations of meshing approaches to meet the needs of the applications. Although this approach does yield a program that uses a bit more memory and is slightly slower that a tailored single method, the ability to effectively respond to a wide variety of meshing requirements as given by various user communities is critical.

Given the topological structures of the geometry model and mesh, and the ability to perform needed geometric operations, the mesh generation and modification procedures can be constructed in terms of operator sets that:

(1) determine topological adjacencies and mesh classification,
(2) interrogate the geometric model for shape parameters,
(3) modify the geometric model topology,
(4) create/delete mesh entities and assign/modify shape information,
(5) evaluate/assign mesh entity attributes (including constraints),
(6) modify the mesh entities.

The operators in the first four sets can be easily constructed in a manner independent of the mesh entities topology, so dealing with various combinations of hexahedra, wedges, pyramids, and tetrahedra can be supported. However, key aspects of the remaining procedures do not generalize as easily. Therefore, the majority of them implemented in MEGA are valid only for simplex types of mesh entities (tetrahedra and triangles).

3.1. Determine topological adjacencies and mesh classification

The model and mesh data structures directly store the adjacency information to both the higher- and lower-order entities. For example, an edge stores its two vertices and the list of faces that it bounds. With this information it is easy to demonstrate that any other adjacency can be determined by traversing the adjacencies and local sorts [13, 14]. It is also easy to show that these structures can be used to determine second-order adjacencies that are often used by mesh generation procedures [14]. An example of a second-order adjacency is the list of regions that share a face with those of a specific region. Whenever a mesh entity is created, its classification with respect to the geometric model is known and maintained.

3.2. Interrogate the geometric model for a shape parameter

Since a topological model maintains no information on the geometric shape of the entities, a means is needed to provide the mesh generator with the required geometric shape information. This need is addressed by taking advantage of the structure of today's commercial CAD systems which are built upon solid modelling kernels, such as ACIS [17], SHAPES [18], and Parasolids [19]. The solid modelling kernels support the geometric modelling system through a set of interface operators, keyed via topological entities, that can be exposed to geometry-based applications to meet their application needs. In addition to providing a superior software development environment, building geometry-based applications through the same operator set that is used to build the model provides a means to deal with geometric modelling tolerances in a consistent manner. The reliability of an automatic mesh generator is directly related to its ability to apply geometric model tolerances in a manner consistent with that used in the definition of the geometric model [16].

Examination of the shape information needed by mesh generation and modification operations shows that they can be effectively implemented using only shape parameters evaluated on a pointwise basis. Common interrogations of this type include: determine the normal to a point on the surface, provide the co-ordinates of a point on the surface given its parametric values, determine the closest point on a model entity to a given point, etc. The ability to limit the interactions of the meshing procedures to only pointwise interrogations allows the mesh generation software to be written in a manner that is independent to the specific geometric shape forms used in the geometric modelling package.

3.3. *Modify geometric model topology*

Although the base geometric information is fixed before being given to the mesh generator, there are times where it is desirable to modify the topology of the geometric model to be used by the mesh generation application without modifying the shape information associated with the original geometric model. Figure 2 depicts an approach to address this issue. The central idea is one in which the topological representation of the original model (linked back to the solid modeller) is maintained, as well as copies of each of the modified model topologies for which one wants to be able to explicitly interrogate that topology. Since the size of the topological representation is small compared to the mesh or the shape information in the model, this approach does not dramatically increase the storage requirements.

In addition to the model topologies, Figure 2 also depicts an attribute structure. This structure is needed to store the process information used to map between the models. For example, in the case of a dimensional reduction it may require storage of a functional form for the 'thickness' of the portion of the model that has been reduced.

3.4. *Create/delete mesh entity and assign/modify shape information*

The most basic mesh entity creation operator is to create a vertex. Vertices are typically created at specific locations determined based on a geometric interrogation of the model or the mesh. For example, defining a vertex given its parameter value on an edge, or at the closest point. Information to specify the classification of newly created vertices is known at the time of creation based on the method of creation. Higher-order mesh entities are created by defining the lower-order mesh entities that bound them. An initial classification of the higher-order mesh entities can

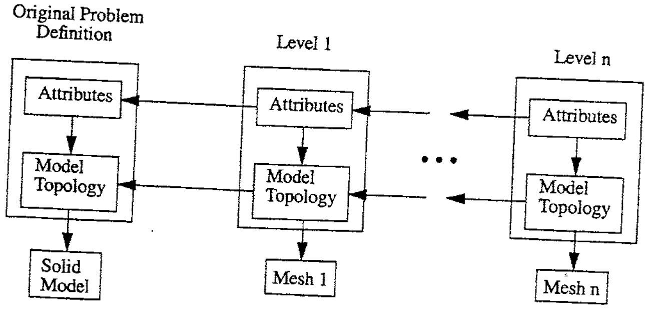


Figure 2. Basic structures for analysis idealization

be determined by the classification of its bounding entities. Since the classification implied by the classification of the boundary entities is usually correct, and the procedures that determine a classification should be modified are often carried out in a later step in the mesh generation process, the initial classification is set based on the classification of the boundary entities.

Whenever mesh entities are created or deleted, the appropriate associativities must be updated. Most meshing procedures assume newly created mesh edges are straight lines and new faces are bounded by straight edges. The classification information is central if higher-order shape information is to be assigned since that assignment should be consistent. For example, if a mesh edge on a model face is to be assigned the shape of an interpolating quadratic, the point introduced along that mesh edge should lie on the model face in a 'reasonable location between' the two vertices of the edge.

3.5. *Evaluate/assign mesh entity attribute*

Throughout a mesh generation process, decisions are made based on attributes of mesh entities. These attributes range from geometric measures (like dihedral angles, aspect ratio, etc.) to constraints (this mesh entity must be maintained, etc.). The data structures support the association of such attributes to the mesh entities and there are operators available to evaluate and retrieve specific attributes.

3.6. *Modify mesh*

Mesh generation and modification procedures rely heavily on a set of operators that perform local mesh modifications. In the case of simplex elements a complete set of operations to locally modify a mesh based on swapping, splitting and collapsing has been developed.

The most local of swapping procedures for simplex elements relies on the basic property that $n_{sd} + 2$ points in $\Re^{n_{sd}}$ may be triangulated in, at most, two ways. Although it has been shown that these two options are sufficient to convert a triangulation into a Delaunay triangulation, more extensive swapping operations are desired to allow the effective application of alternative mesh optimization criteria. Two more extensive local swapping tools developed for tetrahedral meshes are edge removal and multi-face removal [10].

In edge removal a mesh edge $M_i^1 \sqsubset G_j^3$ which is bounded by vertices M_k^0 and M_l^0 can be eliminated by retriangulating the polyhedron of all connected tetrahedrons. The polyhedron is retriangulated by: (i) triangulating the polygon of all mesh vertices of the polyhedron which are neither M_k^0 nor M_l^0, and (ii) connecting the new mesh faces to M_k^0 and M_l^0 (Figure 3).

Multi-face removal is a procedure that reverses edge removal, that is, it considers a configuration that could have resulted from edge removal and obtains the starting configuration. Methods for the valid application of multi-face removal are given in [10].

Additional vertices can be introduced into an existing mesh by splitting selected mesh entities. In three dimensions, a mesh region can be split into four new regions, a mesh face into three new faces, and a mesh edge into two new edges. Mesh entity splitting adds a vertex and then connects the boundary vertices of the polyhedron to the new vertex. In the case of a mesh region, the polyhedron is the mesh region itself. In the case of a mesh face or a mesh edge, the polyhedron is built from the union of all mesh regions connected to the face or edge, respectively. Figure 4 displays the three types of splits in three dimensions and indicates for each one of them the change in the number of mesh regions. If the mesh entity to be split is classified in a model region and the

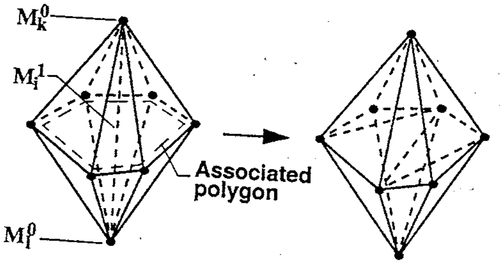


Figure 3. General edge swap

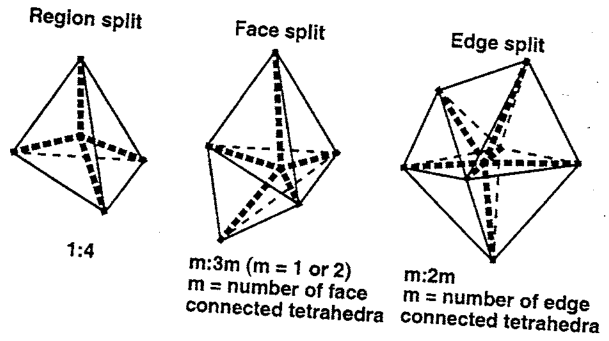


Figure 4. Split operations

mesh edges of the polygon are all straight, all boundary faces of the polyhedron are fully visible from the split vertex (no matter where it is located) and splitting is therefore always possible. If the mesh entity is classified on a curved model boundary, and the new vertex has to be snapped to the model entity it is classified on, splitting may not be geometrically possible.

Edge-splitting is a flexible refinement procedure. When multiple edges are marked for splitting, it is advantageous to be able to quickly determine how tetrahedron with more than one edge marked for splitting should be subdivided. To effectively support this a set of forty-two surface triangulation options for the subdivision of a tetrahedron, depending on the number of marked edges and the desired surface triangulations, have been implemented (Figure 5). Since the creation of interior edges has no influence on neighbouring elements, one is free to select among the options in any manner desired. For a given surface triangulation it is not always possible to create a valid set of tetrahedra without the introduction of an interior vertex. Of the forty-two possible surface triangulation options, four of them require the introduction of a vertex interior to the element.

Although a number of collapse operators can be defined, many procedures rely on edge collapsing. Edge collapsing retriangulates the polyhedron of all mesh regions connected to the one end vertex. The polyhedron is retriangulated by connecting the boundary faces of the polyhedron to the other end vertex (Figure 6). Edge collapsing cannot be performed if at least one of the new mesh regions is geometrically invalid (negative volume) [10].

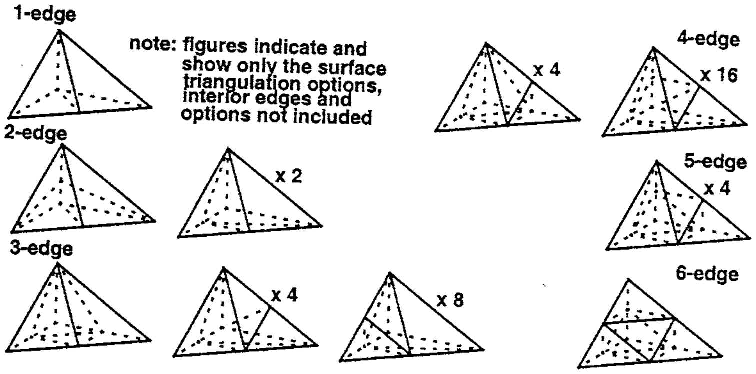


Figure 5. Tetrahedral refinement patterns based on numbers of marked edges

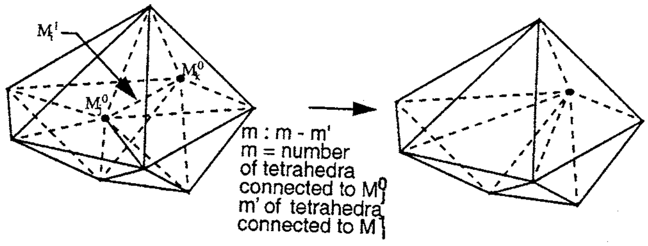


Figure 6. Edge collapsing in three dimensions

4. MESH GENERATION PROCEDURES

The mesh generation procedures employ the basic operators within an overall mesh entity creation strategy. Mesh entity creation strategies employ criteria based on properties and meshing heuristics in the process of mesh entity creation. In addition, they may employ some form of secondary structures to enhance the efficiency of the process.

The most commonly applied property upon which automatic mesh generators have been built is the Delaunay circumsphere property [2, 3]. The key attributes of this property are that it is the only known generally useful global property, and there are computationally effective methods to employ it in the meshing process. On the other hand, it is known that meshes satisfying the Delaunay criteria do not necessarily yield meshes which will provide the best solution results [20, 21]. Therefore, alternative criteria which are only local in nature are often considered. For example, it is common to drive local mesh modifications procedures with a criteria to minimize the maximum dihedral angle.

All mesh generation procedures employ criteria which are based on some level of heuristics. In some cases, the heuristics are related to ensuring that the methods of mesh entity creation yield a mesh satisfying some given mesh entity size control information. In other cases they are used in the creation of mesh entities. In those situations where new mesh vertices are created to fill unmeshed portions of the domain, it is necessary to employ some criteria to define the location of the point. The element removal operations used in advancing front procedures rely heavily on such criteria.

Since no one set of mesh entity creation strategies are capable of yielding meshes best suited for all situations, MEGA includes multiple procedures for mesh entity creation and modification. Mega also uses multiple structures to maintain the efficiency of the mesh generation process. Almost all procedures take advantage of the mesh entity adjacency and classification information. Specific procedures take advantage of an octree structure [22] as indicated below.

In those cases where the input is a geometric model, MEGA begins by generating a surface mesh using the procedure outlined in Section 4.1. This surface mesh may be modified by subsequent procedures as the mesh generation process continues. Volume meshes for isotropic mesh refinements are generated by a procedure which employs an octree structure and then creates mesh entities by a combination of templates, element removal, and Delaunay point insertion (Section 4.2). Anisotropic meshes can be created in boundary layers using the procedure outlined in Section 4.3.

4.1. Surface mesh generation

The surface mesh generator is responsible for generating a valid mesh of the boundary of the geometric model. A valid boundary mesh is a geometric triangulation (topologically compatible and geometrically similar with each model boundary entity [15, 16]) of the boundary of the model that is also not self-intersecting in real space. (The requirement of not self-intersecting in real space is needed to ensure that the volume mesh generation procedures cannot be confused.) The model faces are meshed using a 2-D constrained Delaunay procedure employing the parametric space of the face and mapping metrics between the parametric and real space [8, 23]. The mesh face sizes and gradation created during surface mesh generation are controlled by user defined parameters indicating element sizes associated with model entities and curvature-based refinement to control geometric approximation of straight-edged mesh entities.

An octree is constructed during the surface meshing process. Within this octree, the octant sizes are set to be approximately the size of mesh entities requested in that area. A one level difference between face and edge neighbours in the octree is used to control mesh gradations. The octree is used during the surface meshing process to aid in controlling element sizes and gradations and to speed the process of determining self-intersections.

The steps performed during surface mesh generation of a model face are:

1. Discretizing the model edges bounding the face into mesh edges of the size and gradation indicated in the mesh control information.
2. Create an initial triangulation in terms of a rectangle, split it into two triangles that contain all the points on the boundary of the face.
3. Insertion of boundary vertices into the triangulation.
4. Recovery of any mesh edges on the model edges bounding the face lost during insertion of the boundary vertices.
5. Determination and deletion of all mesh faces created in the steps above that are exterior to the model face being meshed.
6. Insertion of interior vertices to obtain the mesh face sizes and gradations dictated by the mesh control information.

The vertices placed on the edges bounding the face are inserted in the parametric space of the face accounting for the shape distortion introduced by the mapping to real space. Mesh faces created during vertex insertion are required to have positive area in the parametric space. Edge swapping is used to recover any mesh edges classified on the model edges that are lost during the boundary point insertion process. Edge recovery can require the introduction of mesh edge splits.

After the mesh edges defining the boundary of the face are recovered, all mesh faces using the four vertices of the initial triangulation are deleted since they are outside the model face. The remaining mesh faces outside the model face are detected by considering the mesh faces using mesh edges that are on the face boundary deleting additional ones connected to them. If additional exterior mesh faces exist, they are detected by propagation out from the exterior model faces with edges on the boundary of the model face.

Interior points are inserted into the mesh by examining the lengths of interior mesh edges and splitting them until their length matches that indicated by the local mesh control information. To maintain the quality of the mesh faces created, care is taken to ensure that newly inserted vertices are not too close to previously inserted vertices.

After the surface meshes are created for all the model faces, the assembled surface triangulation is checked for self-intersections between mesh faces classified on different model faces. Any self-intersections must be eliminated so that the surface mesh strictly encloses the regions to be meshed. Self intersections are eliminated through local edge refinement. Before the self-intersection process begins, the entire surface mesh has been inserted into the octree. Therefore, the octree can be effectively used to determine if there are any self intersections.

4.2. Volume mesh generation

The tetrahedral mesh is created from a pretriangulated surface mesh employing an octree structure [8, 9]. The octree is created such that the octants containing the surface triangles are sized to have edge lengths equivalent to that of the edges of the surface triangulation that is partly or completely interior to them. The interior octants are created such that they satisfy the one level difference rule and any local mesh size specification. To avoid the poorly shaped elements caused by close interaction of surface triangles and interior octants the additional cell type of boundary-like interior cells is introduced. These are interior cells which are closer than some fraction of the surface triangle edge length to the surface triangulation. Using one half an edge length of the near-by surface triangle as the distance criterion works well for this purpose.

Figure 7 demonstrates the basics of this mesh generation process for a simple two-dimensional domain. The left image shows the set of domain boundary segments and the quadrants generated based on them. The image shows the boundary quadrants which contain portions of the boundary segments, the interior quadrants which are more than half an edge length from the boundary segments, and the boundary-like interior quadrants which are interior octants within one half an edge length of the boundary segments. The interior cells are meshed using templates and are indicated by the shaded triangles in the right image of Figure 7.

After the interior octants are meshed, the remaining portion of the domain to be meshed is the region lying between the outer faces of the meshed interior octants and the surface triangulation. This region is meshed employing element removal operations in a manner similar to that used in the current advancing front mesh generators. Since the completion of the meshing process requires connecting tetrahedra to one or, in multi-region problems, up to two sides of these faces,

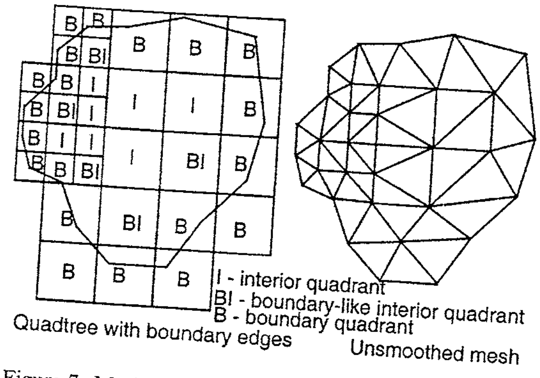


Figure 7. Mesh generation given a discretized boundary

they are referred to as partly connected faces. The mesh generation process is complete when all mesh faces are fully connected.

The element creation process to connect partly connected faces is not constrained to following cell boundaries. It is guided solely by the creation of elements to fill the region between partly connected faces. This is depicted in the right image of Figure 7 by the unshaded triangles created during this step. The tree structure is used during this process to efficiently locate neighbour information. Each partly connected face is associated with one or more octants, thus allowing the tree neighbour finding procedure to be used to locate neighbouring partly connected faces that a current face can be connected to.

Given a partially connected mesh face, face removal connects it to a mesh vertex of a nearby partly connected face. Since the volume to be meshed consists of the region between the given surface triangulation and the interior octree, the vertex used is usually an existing one. In some situations it is desirable to create a new vertex. The choice of this vertex must be such that the created element is of good quality and its creation does not lead to poor (in terms of shape) subsequent face removals in that neighbourhood.

Element removal procedures will on occasion run into difficulty when the criteria used during element removal do not give sufficient consideration for the situation remaining for subsequent face removals. Consideration of the influence of potential subsequent face removals can be considered through a set of 'forward' checks to make sure that any element creation does not make new mesh entities too close (relatively) to existing mesh entities. Since execution of a full set of these checks is expensive, and it does not necessarily indicate the best solution, a full set of checks is not performed.

An alternative method is to use a more efficient criterion which indirectly accounts for the various situations that can arise. Due to the global nature of its properties, the Delaunay circumsphere criteria is the one criteria that meets this need, and has in fact been used with success in element removal mesh generators. Although the Delaunay circumsphere criteria does yield quality isotropic meshes when given a well distributed set of points that avoids the creation of flat elements, ensuring that this is the case raises similar issues to those faced in element removal when ensuring that new mesh entities are not too close to other remaining entities.

The approach currently taken to address these issues is to employ the element removal procedure, with only minimal forward checks, while it is creating good elements. In areas where difficulties arise, specific care is taken to first delete what is needed to create a cavity to be meshed that has sufficient interior to place points for a Delaunay insertion process, which is what is used to mesh the cavity.

4.3. Boundary layer mesh generation

In many cases the most effective mesh configuration is one in which the mesh faces and regions in specific portions of the domain are highly anisotropic. Particularly high levels of anisotropy are desired in boundary layers where aspects of 1-to-10 000 or greater can be requested. The standard isotropic mesh generation procedures are not capable of effectively dealing with these shapes. Therefore specific procedures for anisotropic mesh generation are needed. In those cases where the high levels of anisotropy are near the boundary of the model, procedures that focus attention on generating anisotropic meshes in these thin regions, coupled with more standard isotropic procedures for the rest of the domain, are possible. The boundary layer mesh generation procedure in MEGA accepts a surface triangulation as input, generates the boundary layer mesh [24], and meshes the remainder of the domain using the volume mesher just described.

The basic steps in the boundary layer mesh are:

1. Define the growth directions from the mesh vertices on the model edges bounding faces which will receive boundary layers (Figure 8(a)). Note that based on topological and geometric considerations there can be multiple growth directions defined at a mesh vertex.
2. Update the surface triangulation on any model face affected by the boundary layer coming off the edges (Figure 8(b)).
3. Define the boundary layer growth directions on the model faces bounding faces which will receive boundary layers (Figure 8(c)).
4. Grow the boundary layer prism's on those where possible (Figure 8(d)).
5. Fill in the remaining boundary layer regions with blends (Figure 8(e)).
6. Check for self-intersection of the boundary layer with itself and the remaining unmeshed surface triangulation (Figure 8(f)).

Specific consideration was taken in the design and implementation of the boundary layer meshing procedure to be able to deal with geometrically complex non-manifold models. Figure 9 gives an example of the type of model the boundary layer mesh generator must be able to mesh.

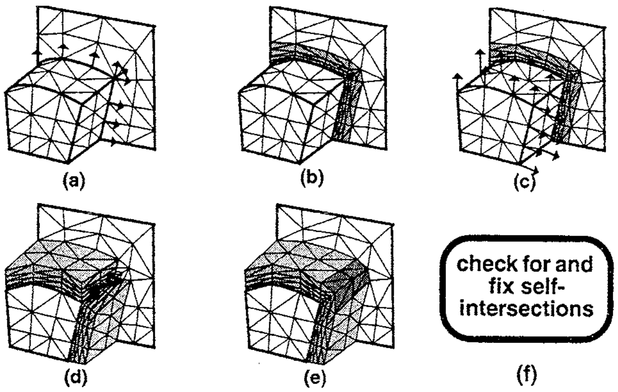


Figure 8. Overall steps in boundary layer meshing

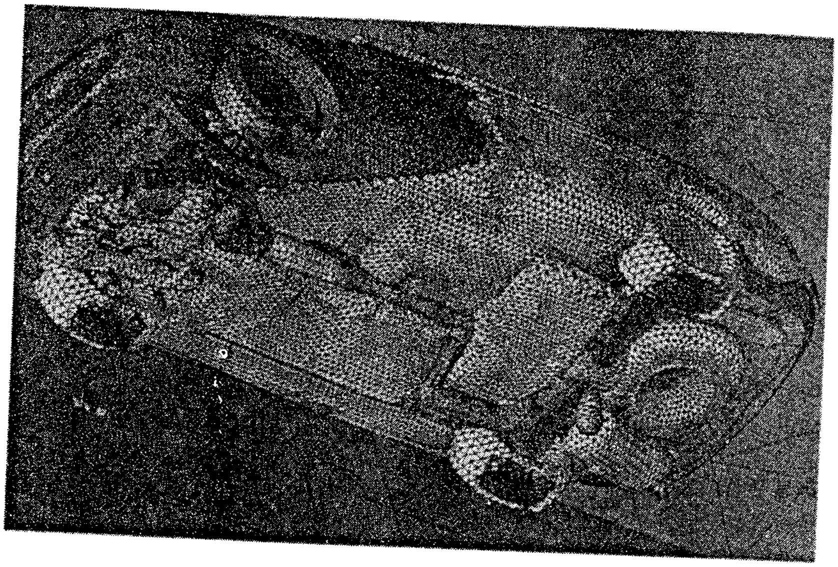


Figure 9. Example of a boundary layer mesh

4.4. *Mesh control attributes*

The ability to control the meshes generated through both a priori and a posteriori mesh control is a critical aspect of an automatic mesh generator. A priori mesh control information is used by the mesh generation procedures in the creation of the initial mesh to be analysed. This information is associated with the topological entities in the geometric model and can have general spatial variations. Currently supported a priori mesh controls include:

1. Mesh entity size—sets the initial size of mesh entities generated on model entities.
2. Boundary layer information—controls the creation of boundary layers in the meshing process.
3. Curvature-based refinement [25]—this attribute controls the geometric approximation between the model and the straight-edged mesh. The smaller the value, the more accurately the mesh approximates curved boundaries.
4. Mesh shape control—when a model feature size is smaller than the requested mesh size, there are three possible options:
    - (a) Mesh the model feature without any special processing. This can result in poorly shaped mesh elements.
    - (b) Mesh the model feature and then refine those mesh entities classified on it in order to improve the mesh shape.
    - (c) 'Eliminate' the adverse influence of the small model features.

Figure 10 demonstrates the application of mesh controls associated with the entities of the geometric model. Figure 10(a) shows the original model. Figure 10(b) shows a mesh of the model without any meshing attributes applied. Figure 10(c) shows a uniform mesh in which the element

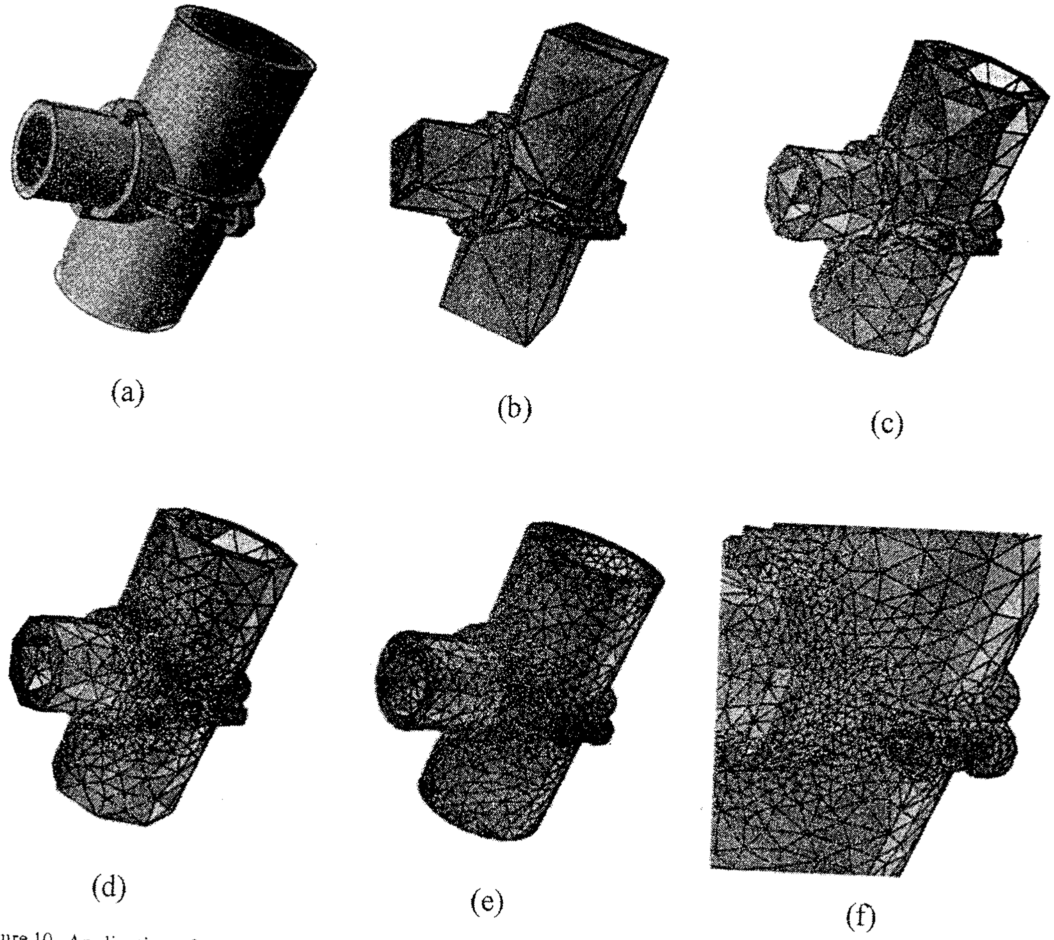


Figure 10. Application of mesh size controls: (a) geometric model; (b) mesh resulting from no meshing control; (c) global size control attribute applied; (d) local size attributes set; (e) local curvature refinement applied; and (f) close up of mesh in (e)

sizes requested were the same throughout the model. In the mesh shown in Figure 10(d), the element size requested for the edge connecting the vertical and horizontal pipes was set to be smaller than the rest of the model faces (Figure 10(c)). The mesh shown in Figure 10(e) employs a procedure which sets the element sizes associated with the model faces of the larger vertical pipe and the edges of the cylinders based on the local curvature of the face. Figure 10(f) shows a close up of Figure 10(e).

5. MESH MODIFICATION PROCEDURES

There are a variety of reasons that one may wish to alter a given mesh to satisfy specific criteria. They range from simple element shape improvement, to procedures aimed at forcing the mesh to satisfy specific requirements driven by the needs of the analysis. Criteria driven mesh modification procedures implemented in MEGA include:

- mesh entity shape improvement,
- elimination of the adverse effects of small model features,
- mesh curving,
- mesh surface sizing and optimization,
- adaptive mesh enrichment,
- multiple elements through the thickness.

5.1. *Element entity shape improvement*

It is common after the initial mesh satisfying the specified mesh gradations is created to apply a procedure aimed at improving the shapes of the mesh entities. Element shapes can be improved either by repositioning the mesh vertices, performing local mesh modifications, or some combination of both. It is not uncommon to apply such procedures to the surface mesh before the volume mesh is created, and to the volume mesh after it is created. Experience indicates that node point reposition is most useful for smoothing jumps in mesh gradation. Since the criteria found most effective for this process do not directly consider the shape of the resulting mesh entities, they are not the most effective in improving the shape of the worst shaped elements. In these cases procedures which reposition the nodes to explicitly improve the shapes of connected elements are required [26].

Element shapes can also be improved by the application of swapping, splitting and collapsing. These mesh modification operations are applied in MEGA to improve the shapes of only the elements that do not satisfy minimum shape requirements. The application of the specific mesh modification procedure is based on the ability of the procedure to deal with the specific local situation [10]. Care must be exercised in applying splitting or collapsing to ensure that resulting mesh entity sizes do not vary greatly from that requested by the mesh control information.

5.2. *Elimination of the adverse effects of small model features*

It is common for geometric models to have model features (edges and faces) which are substantially smaller that the mesh entity sizes desired. Their existence is due to either some stitching process the modeller has performed to close the model, or they are real model features substantially smaller that the mesh sizes needed in the current analysis. If the mesh entity sizes created during meshing are kept to those requested, mesh faces and regions in the vicinity of the small model entities will be poorly shaped. To improve the shape of the mesh entities without excessive refinement requires a mesh modification that will violate the topology of the original model. A procedure has been developed to allow such a modification to be performed [27, 28]. The basic steps in the procedure are:

1. Generate a mesh employing the user specified mesh entity size distributions.
2. Identify and flag mesh entities that need to be deleted in order to eliminate the mesh entities with poor quality metrics.
3. Ensure that deletion of the flagged mesh entities results in geometrically valid mesh entities and their elimination does not introduce undesired dimensional reductions.

4. Apply the mesh modification operator(s) to delete the desired mesh entity and update the model and pointwise geometric information as required to ensure mesh validity.

The last step in the process requires that the topology of the model be modified as the process continues.

Figure 11 demonstrates a typical situation where the application of the procedure eliminated elements with extremely small angles caused by the small model edges used to stitch the model, increasing the value of the worst angle by over an order of magnitude. An example of the application of this procedure to eliminate large numbers of local model features deemed unimportant is shown in Figure 12. Figure 12(a) shows a close-up of a portion of a geometric model with a large number of geometric features, while Figure 12(b) shows a mesh generated that explicitly represents all the model features. Figure 12(c) shows a coarser mesh created which includes the application of mesh modifications to eliminate the explicit representation of the smaller model features. A still coarser mesh in which nearly all the local model features are eliminated is shown in Figure 12(d).

5.3. Mesh curving

MEGA provides a procedure that takes straight-edged mesh entities and curves them to better approximate curved model geometry and to create meshes suitable for p-version analysis [29, 30]. The current version of the curving procedure is to limited quadratic edge geometries. The first step in generating a curved solid mesh is to curve the surface mesh with respect to the model faces and edges. For a quadratic Lagrangian this is done by 'snapping' the mid-point of the mesh edges to a point on the model entity that it is classified on. The current approach attempts to use the average parametric location of the mesh vertices bounding the mesh edge. While curving the surface mesh, invalid mesh faces can result, typically when dealing with trimmed concave surfaces. These elements are identified and corrected by modifying the curving of mesh edges classified on model faces. This correction may need to be propagated through the surface in order to create a valid curved surface mesh.

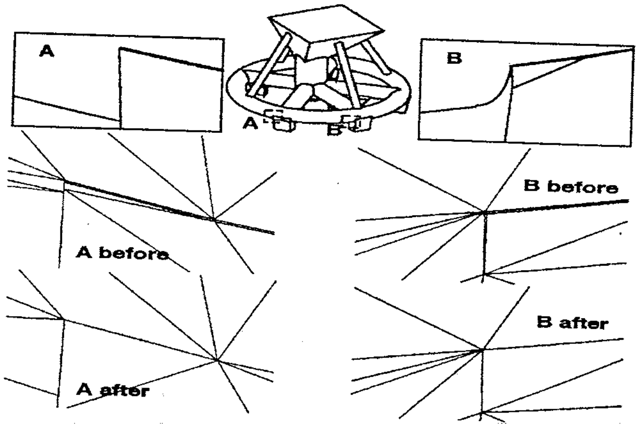


Figure 11. Mesh improvement due to elimination of small model features

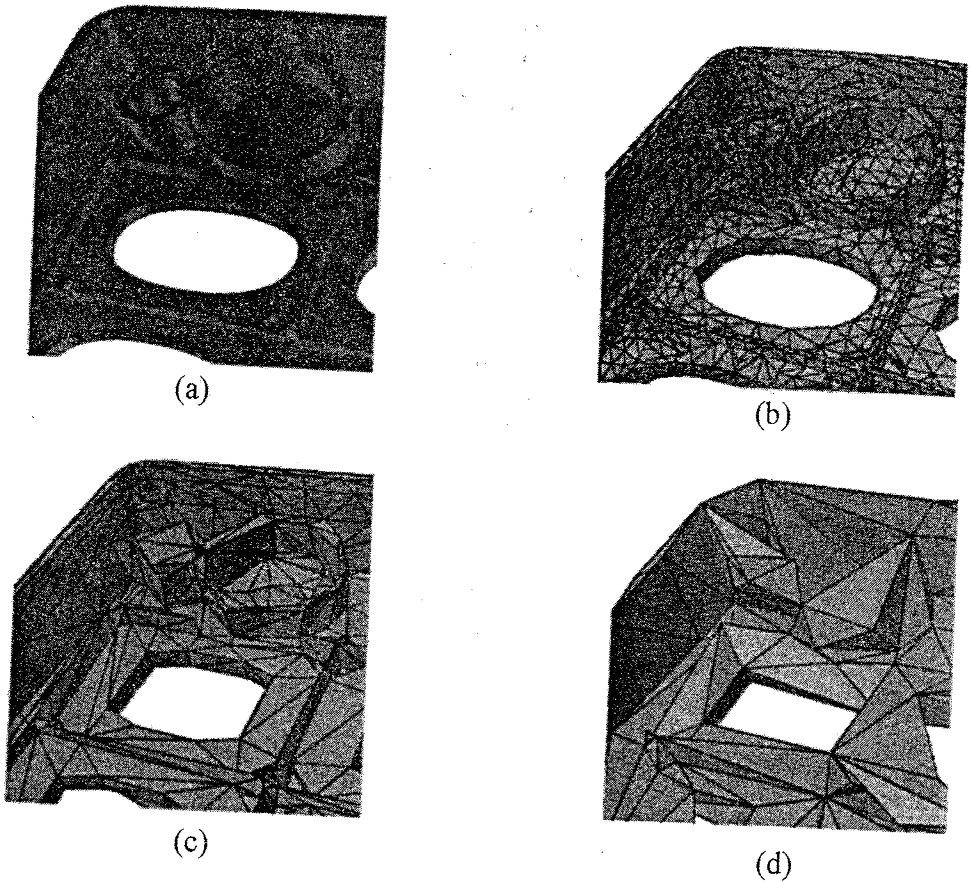

(a)
(b)
(c)
(d)

Figure 12. Mesh coarsening by the mesh modification with model modification tracking: (a) geometric model; (b) mesh with model features represented; (c) coarse mesh after mesh modification; and (d) coarser mesh after mesh modification

Once a valid surface mesh is created, the mesh regions that have curved faces and/or edges are then checked for element shape by evaluating the Jacobian at a set of integration points. If the region is invalid, mesh modification tools can be applied to correct it. These include: edge swapping, edge collapsing, face swapping, vertex repositioning, and interior edge curving. Figure 13 shows the results of applying MEGA's curving tools to a straight-edged mesh.

5.4. Mesh surface sizing and optimization

Often the geometric domain is defined in terms of a surface triangulation where that triangulation was derived for some other purpose, typically graphical display. For these situations MEGA has a procedure that accepts a surface mesh, with no knowledge of its relationship to the original geometric model, as input and converts it to a surface triangulation satisfying the given mesh control parameters in terms of mesh entity size, distribution and shape. This is done by applying the mesh modification operations of swapping, splitting and collapsing. Since the given

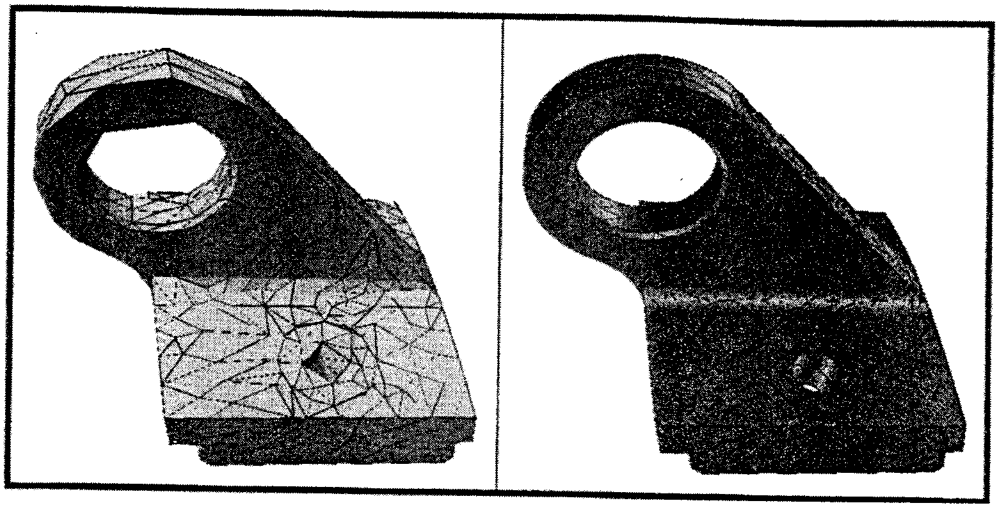


Figure 13. Curved mesh through mesh modification

triangulations do not provide specific classification information, knowledge of the locations of model edges and vertices is not known. To ensure that in those cases where the original model edges and vertices were at locations of large changes in surface slope, the procedure includes criteria to ascertain where these geometric features exist and ensures they are maintained [8, 31].

5.5. *Adaptive mesh enrichment*

The MEGA adaptive enrichment procedures employ the enrichment indicators provided by an analysis to refine and coarsen the mesh. Since these processes must deal with the issues caused by curved geometry, the complete set of mesh modification operators, including swapping, is used during these processes. Since refinement and coarsening can be edge driven, it is straight forward to apply any of the various criteria that have been published to control the changes in element shapes during the enrichment process. References 10, 11, 32 provide descriptions of the procedures used in MEGA.

5.6. *Multiple elements through the thickness*

Another situation where elements with high aspect ratios are desired is in three-dimensional models where there are portions of the model which have one geometric dimension that is much smaller than the others. In many cases the mesh entity sizes requested in the 'surface' of these 'thin sections' is such that an isotropic mesh generator produces only a single element through the thin direction. In some situations, such as casting simulations using low-order finite elements or finite volumes, such a mesh is useless. What is desired in these cases is the ability to have multiple elements edges through the thin direction while not affecting the element sizes in the 'surface'. A procedure that employs extensive mesh modification has been developed to address this issue

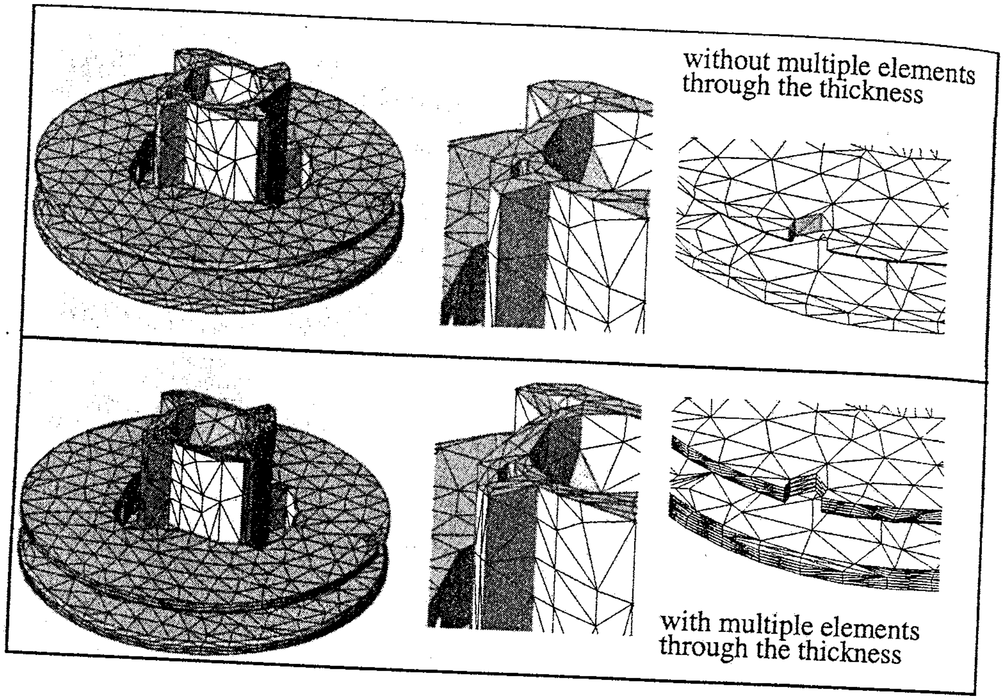


Figure 14. Multiple elements through the thickness example

in cases where a few element edges are desired through the thickness [33]. The basic steps of the procedure are:

1. Determine the portion of the models that need more 'edges through the thickness'.
2. Determine the mesh entities on the opposite surfaces to be connected. During this process perform mesh modification operations to maximize the number of surface triangles that align to form a wedge.
3. Subdivide the wedges found in the above step.
4. Perform more general mesh modification to get the desired number of element edges through the thickness in the remaining portion.

Figure 14 shows an example of multiple elements through the thickness where the top portion model has been assigned two elements through the thickness and the bottom portion four elements through the thickness.

6. CLOSING REMARKS

The automatic generation of finite element models starting from base domain definitions requires more than an effective triangulation algorithm. The technical issues associated with interacting with the domain definitions and ensuring the meshes generated are valid discretizations of those domains dominates the mesh generation process. The ability to produce the forms of meshes needed for various problem types appears to be best addressed by providing multiple mesh generation capabilities which can be easily combined.

The Meshing Environment for Geometry-based Analysis presented in this paper is one approach to addressing these issues. By building on a common set of structures and operators, MEGA supports a wide variety of mesh generation tools and allows the easy addition of new procedures without the need to reproduce all the base operations needed.

ACKNOWLEDGMENTS

As indicated in the introduction, the development of MEGA followed a long path through research and implementation of automatic mesh generation technologies carried out by a number of students and colleagues. Without their intellectual contributions, and extensive efforts, none of the capabilities outlined above would have been developed. Close to their order of primary contribution, they are: Mark A. Yerry, Peggy L. Baehmann, William J. Schroeder, Marcel K. Georges, Hugues L. de Cougny, Saikat Dey, Ramamoorthy Ravichandran, Mark W. Beall, Rao V. Garimella, Robert M. O'Bara and B. Kaan Karamete. The author would like to gratefully acknowledge their contributions.

REFERENCES

1. Armstrong, CG. (ed). *Advances in Engineering Software*, vol 13(5/6). Computational Mechanics Publication: Southampton, England, 1991.
2. George PL, *Automatic Mesh Generation*. Wiley: Chichester, 1991.
3. George PL, Borouchaki H. *Delaunay Triangulation and Meshing: Applications to Finite Elements*. Hermes: Paris, 1998.
4. Shephard MS, Weatherill NP (eds). *International Journal for Numerical Methods in Engineering*, vol 32. Wiley-Interscience: Chichester, England, 1991.
5. Shephard MS. Approaches to the automatic generation and control of finite element meshes. *Applied Mechanics Reviews* 1988; **41**(4):169–185.
6. Shephard MS. Update to: approaches to the automatic generation and control of finite element meshes. *Applied Mechanics Reviews* 1996; **49**(10-2):S5–S14.
7. de Cougny HL, Shephard MS. Parallel unstructured grid generation. *CRC Handbook of Grid Generation*, CRC Press Inc.: Boca Raton, 1997.
8. de Cougny HL. Distributed parallel mesh generation. *Ph.D. Thesis*, Scientific Computation Research Center, RPI, Troy, NY, 1998.
9. de Cougny HL, Shephard MS, Ozturan C. Parallel three-dimensional mesh generation on distributed memory MIMD computers. *Engineering with Computers* 1996; **12**(2):94–106.
10. de Cougny HL, Shephard MS. Parallel mesh adaptation by local mesh modification. SCOREC #21-1995, Scientific Computation Research Center, RPI, Troy, NY, 1998, submitted for publication.
11. Shephard MS, Flaherty JE, Bottasso CL, de Cougny HL, Ozturan C, Simone ML. Parallel automated adaptive analysis. *Parallel Computing* 1997; **23**:1327–1347.
12. Teresco JD, Beall MW, Flaherty JE, Shephard MS. A hierarchical partition model for adaptive element computation. *Computer Methods in Applied Mechanics and Engineering* 1999, to appear.
13. 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.
14. Beall MW, Shephard MS. A general topology-based mesh data structure. *International Journal for Numerical Methods in Engineering* 1997; **40**(9):1573–1596.
15. Schroeder WJ, Shephard MS. On rigorous conditions for automatically generated finite element meshes. In *Product Modeling for Computer-Aided Design and Manufacturing*, Turner JU, Pegna J, Wozny MJ (eds). North-Holland: Amsterdam, 1991; 267–291.
16. Shephard MS, Georges MK. Reliability of automatic 3-D mesh generation. *Computer Methods in Applied Mechanics and Engineering* 1992; **101**:443–462.
17. ACIS Interface Guide and ACIS API Guide. Spatial Technology, Inc., 2425 25th St., Building A, Boulder, CO, December 1992.

18. SHAPES Reference Manual. Release 2.0.8, XOX Corporation, Two Appletree Square, Suite 334, Minneapolis, Minnesota 55425, July 1993.
19. Parasolid Version 6 KI Programming Reference Manual. Electronic Data Systems Corporation, Parker's House, 46 Regent Street, Cambridge CB2 1DB England, October 1994.
20. Barth, TJ. Aspects of unstructured grids for finite-volume solvers for the Euler and Navier–Stokes equations. *Unstructured Grid Methods for Advection Dominated Flows*, Agard-R-787, Neuilly Sur Seine, France, 1992; 6.1–6.61.
21. Joe B. Delaunay versus max–min solid angle triangulations for 3-D mesh generation. *International Journal for Numerical Methods in Engineering*, 1991; **31**:987–997.
22. Samet H. *Application of Spatial Data Structures: Computer Graphics, Image Processing and GIS*. Addison-Wesley: New York, 1989.
23. de Cougny HL, Shephard MS. Surface meshing using vertex insertion. *Fifth International Meshing Roundtable '96*, SAND96-2301, Sandia Corp., 1996; 243–256.
24. Garimella R, Shephard MS. Boundary layer meshing for viscous flows in complex domains. *Proceedings of the Seventh International Meshing Roundtable*, SAND 98-2250, Sandia, Nat. Lab, Albuquerque, NM, 1998; 107–118.
25. Shephard MS, Dey S, Georges MK. Automatic meshing of curved three-dimensional domains: curving finite elements and curvature-based refinement. In *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations*, Babuska I, Flaherty JE, Hopcroft JE, Henshaw WD, Oliger JE, Tezduyar T (eds). IMA Volumes in Mathematics and its Applications, Springer: Berlin, 1995; **75**:67–96.
26. de Cougny HL, Shephard MS. Explicit node point mesh smoothing within the octree mesh generator. SCOREC #10-1990, Scientific Computation Research Center, RPI, Troy, NY, 1990.
27. Dey S, Shephard MS, Georges MK. Elimination of the adverse effects of small model features by local modifications of automatically generated meshes. *Engineering with Computers* 1997; **13**(3):134–152.
28. Shephard MS, Beall MW, O'Bara RM. Revisiting the elimination of the adverse effects of small model features in automatically generated meshes. *Proceedings of the Seventh International Meshing Roundtable*, SAND 98-2250, Sandia National Laboratories, Albuquerque, NM, 1988; 119–131.
29. Dey S, Shephard MS, Flaherty JE. Geometry-based issues associated with p-version finite element computations. *Computer Methods in Applied Mechanics and Engineering* 1997; **150**:39–55.
30. Shephard MS, Dey S, Flaherty JE. A straight forward structure to construct shape functions for variable p-order meshes. *Computer Methods in Applied Mechanics and Engineering* 1997; **147**:209–233.
31. de Cougny, HL. Refinement and coarsening of surface meshes. *Engineering with Computers* 1998; **14**:214–222.
32. Shephard MS, Flaherty JE, de Cougny HL, Bottasso CL, Ozturan C. Parallel automatic mesh generation and adaptive mesh control. In *Solving Large Scale Problems in Mechanics: Parallel and Distributed Computer Applications*, Papadrakakis M (ed.). Wiley: Chichester, 1997; 459–493.
33. Garimella R, Shephard MS. Tetrahedral mesh generation with multiple elements through the thickness. *Proceedings of the Fourth International Meshing Roundtable 1995*, Albuquerque, NM, SAND95-2130, 1995; 321–333.
34. Shephard MS, de Cougny HL, O'Bara RM, Beall MW. Automatic grid generation using spatially-based trees. *CRC Handbook of Grid Generation*, CRC Press Inc: Boca Raton: 1998.