International Journal of Computational Geometry & Applications
Vol. 10, No. 3 (2000) 309–331

VIRTUAL TOPOLOGY OPERATORS FOR MESHING

ALLA SHEFFER* and MICHEL BERCOVIER
Institute of Computer Science, The Hebrew University, Jerusalem 91904, Israel
**E-mail*: *sheffa@cs.huji.ac.il*

TED BLACKER† and JAN CLEMENTS
Fluent Inc., 500 Davis St., Suite 600, Evanston, Illinois 60201, USA
†*E-mail*: *ted@fluent.com*

Received 15 June 1998
Revised 1 February 1999
Communicated by S. H. Teng

ABSTRACT

In recent years several automatic 3D meshing algorithms have emerged. However direct analysis of CAD models is still elusive. Among the major obstacles preventing automation is the necessity to edit the CAD models to be suitable both for the analysis objectives and the available meshing algorithms. Such editing includes topology correction and validation, detail suppression and decomposition. Editing the geometry directly (e.g. surface redefinitions) is cumbersome, tedious, and expensive. Introducing virtual topology allows such operations as modifications to the topology only. In this work the concept and operators of virtual topology are described, along with their use in performing the required editing of the model. A set of automatic and semi-automatic tools for the various editing operations are introduced.

Keywords: Geometry clean-up, simplification, decomposition, B-Rep, CAD and analysis integration.

1. Introduction

A focus of research and development effort on new algorithms for meshing has resulted in several automatic 3D algorithms as reviewed by Field.[6] Among them fully automated tetrahedral meshing algorithms[7] and hexahedral meshing algorithms.[2,15,8,16,3,21]

However, attempts at direct analysis of models obtained from CAD geometry sources have unveiled numerous obstacles preventing true automation of the analysis process. CAD and Finite Element (FE) analysis are two significantly different disciplines, and hence the demands and emphases they put on the model representation differ substantially. As a result models generated by CAD systems are often unsuitable for analysis needs, requiring multiple editing and adjustment before the meshing can proceed. The difficulty increases when the design and analysis are done in different software systems, as often is the case.

Among the problems requiring editing are:

- Incomplete and inconsistent topology and geometry descriptions.
- Irrelevant minor details which severely complicate the meshing.
- Complex models that need decomposition into parts, either to be meshable by the tools available to the user or to allow needed control of local mesh density.

In view of the advances in mesh generation automation, integration of CAD and analysis is quickly becoming a major focus in providing a fully automated model analysis.[13]

One common approach to provide the necessary geometric editing is through direct geometric adjustments, i.e. the underlying definitions of surfaces and edges are modified to accomplish the required changes.[11,4] This requires the use of sophisticated surface definitions, is expensive, approximates the true geometry, and usually destructively replaces the old geometry with new definitions. Such techniques require extensive user interaction and while dealing with most of the editing problems, lack generality. Using just topological modifications for detail suppression was suggested by Armstrong,[1] but was not extended further to other editing operations.

In this work the problems of CAD and analysis integration are discussed, together with a comprehensive scheme for performing the required adjustments. The suggested scheme is based on the virtual topology concept introduced in Ref. [18]. The virtual topology provides an alternative model representation along with a set of operators to perform the editing operations commonly required for the meshing pre-processing. The operators are applied only on the topological structure or connectivity of the model. This significantly simplifies the modeling process as geometric descriptions are mainly left intact. It is thus relatively inexpensive. Since the existing geometry is maintained, the original model can be easily restored. This paper presents the use of virtual topology to perform high level editing operations. Algorithms for automation of the different editing tasks are described, substantially reducing the required user interaction.

An additional benefit of virtual topology arises from its direct use during meshing. The virtual structure often allows easy geometric modification (e.g. sliding a virtual edge on a face), making it very flexible or "rubberized" to the user. Since the mesh is "owned by" the virtual structure, it must follow the topology movement. Moving virtual entities then allows whole sections of the mesh to be dynamically adjusted, providing direct control of the final mesh.

The paper is organized as follows: The information content and description formats used in different CAD systems and their associated drawbacks are discussed in Section 2. In Section 3 the virtual topology concept and definitions are reviewed. The virtual topology operators, along with a set of new high level operators (for commonly required editing operations) are described in Section 4. Section 5 explains the use of the virtual topology operators at the different steps of model editing and suggests several automation techniques for model adjustment. The benefits of using the virtual topology approach are discussed. Section 6 provides a few complex model examples, implementation details and results. Section 7 concludes the work presented, and suggests further areas of research.

2. CAD and Analysis Integration Problems

The integration of design and analysis has been, and continues to be the subject of much development. As stated earlier, the objectives of the two disciplines differ significantly and therefore large differences in both the information content and representation format arise. This chapter presents a brief overview of the information content and representation formats available within a CAD environment and those required to perform FE analysis. The problems arising when using CAD data for analysis are discussed.

The discussion here is limited to the shape and geometry aspects of the model relevant to the analysis and affecting the mesh generation, although other factors like correct description of physical properties of the object, loading and boundary conditions are not less important to correct behavior simulation.

2.1. CAD Data Representation

CAD systems offer a variety of techniques which can be categorized according to the data representation, i.e. 2D drawings and 3D models including wireframe, surface or solid. The main use of CAD data is for visualization and later manufacturing of the modeled objects. Its origins as an automated drafting system are still evident in many of the representations.

2D drawings (drafting) are essentially pictures of a product's shape annotated by symbols and text. Such drawings are unreliable representations of the object since often size parameters might be adjusted using the annotations without corresponding geometric changes. Drawings can legitimately contain incomplete or schematic geometric definitions. This description relies on the annotated dimensioning and size specifications, with the geometric definition used mainly for visualization.

A 3D wireframe is the simplest but least complete form of solid representation. It consists of a set of vertices on the object exterior together with the curves/edges connecting them. It is better suited for the needs of visualization, but lacks surface information. It may also be ambiguous where actual surfaces cannot be uniquely deduced from the wireframe. Often, the wireframe itself may contain inconsistent or incomplete topological connectivity information. In many ways this representation is merely an electronic version of a drawing with similar shortcomings.

A 3D surface model generally consists of a set of surface patches which fit together to provide a complete representation of the outer surface of the object. Some surface models include topological information on the connectivity between the patches. A surface model provides information content generally sufficient for such applications as shaded visualization, NC machining and FE meshing.

3D solids provide a complete representation of the topology and geometry of an object. The two most commonly used representations are boundary representation (B-Rep), which provides the geometric and topological connectivity description of the object envelope; and constructive solid geometry (CSG) providing a procedural representation of the object. Both contain full topological and geometric descriptions of the model (up to the system defined precision), and when used correctly guarantee model consistency.

2.2. Transfer from CAD

Access to information stored in a CAD system can be provided either through export of data in a format readable to the target system or through a set of access routines.

While providing access routines can solve many ambiguities that might arise in exported data, it requires that both the CAD and the FE system are locally available. It also requires strict protocol definitions, which are problematic when dealing with systems from multiple vendors. If a common set of tools for accessing information from any system could be agreed upon and implemented, this would be of significant benefit to users and vendors.

Export of data requires an exact definition of format agreed upon between the source and target systems. One of the more commonly used protocols is IGES.[10] But using it still results in problems for the receiving system. For example IGES usually does not provide topological connectivity information, leaving it to the target system, to reconstruct it from the geometry. Further more, IGES data comes in many flavors and can be highly subject to the export options chosen by the source system.

Export of data in system specific format can be used more effectively to overcome these problems, but, obviously, it imposes limitation on source and target system combinations. An ambitious standards program is underway to establish common techniques for sharing and exchange of information at all stages of a product's life-cycle. The standard ISO 1303,[9] is known as STEP (the STandard for the Exchange of Product Model Data) and is being released as a series of parts. It should provide greater standardization and more complete and concise model descriptions, but it is not widely used and is still under development. However, even it will not solve all the transfer related problems.

2.3. Analysis Requirements

The purpose of analysis, when used as part of the design process, is generally to predict the functional aspects of the design rather than other practical or esthetic ones. Hence, there needs to be a close coupling between the analysis results and the design information.

Mesh generation is an essential preprocessing step for generating analysis results of prescribed accuracy for the given computational domain. The quality of the mesh can play an essential factor in both the accuracy of the FE analysis and the speed of the solution. Hence the initial and most basic requirement from the models prepared for analysis is to be meshable using the tools available to the user. Almost all the automatic mesh generation tools require a full geometric and topological description of the model.

Most volume meshing tools require the object be presented as a valid closed solid, demanding a complete and correct description. Some meshing tools[20] can operate on less complete definitions of geometry (e.g allowing a group of faces with overlapping edges etc.) provided the inconsistencies are small with respect to the element size, but these are the exception rather than the rule.

The object model created by the design system is often not the model required for the analysis. For instance the cavity of a mold is an offset volume off the original object, taking into account retraction rules. On one hand, in order to capture local behavior the geometry must be represented accurately. On the other hand, when only global results are required, as often is the case, local detail may be removed. Structural analysis may require a reduction of 3D details to 2D (mid)surface idealizations. Sometimes idealization of the model to basic elements like beams, shells or plates is essential to capture correctly it's physical behavior.

2.4. Pre-Processing Requirements

Based on the types of the provided CAD data and the analysis requirements as well as the transfer techniques available, as described above, the following problems are likely to arise when attempting to mesh a model translated from a CAD source.

- Incomplete Models - When the object is represented as a surface model, the topological connectivity might not be provided. For a wireframe the surface geometry description will be lacking as well.

- Incorrect/Inconsistent Models - inconsistencies in models can arise from three major sources:

  - inconsistencies in the export file format (such inconsistencies should be removed with the use of the STEP format);
  - source CAD systems where topological correctness is not essential;
  - CAD data precision lower then that used by the analysis system.

- Undefined Geometry - The representation used in the source system may not be understandable by the target system. For example some systems use the representation of a surface of revolution by an axis and a section of revolution, while others do not.

- Over-Detailed models - often the CAD models provided include multiple minor details relevant for the design and manufacturing but irrelevant and even harmful for the analysis. Irrelevant details, while contributing nothing to the analysis results, can complicate and even prevent automatic meshing of the model and can affect the quality of the mesh causing irregularities around the details. Generally the term minor details refers to small volumetric details (features), small faces and short edges, but the decision on which details are irrelevant for a specific analysis procedure may depend on the exact procedure requirements.

- Complex Models - Even after all the insignificant details are suppressed/removed and all possible idealization of the model performed, the models may be still too complex for the meshing tools available. In such cases the common solution is decomposition of the model into parts simple enough to be handled by the available techniques, while preserving mesh conformity on the common boundary. Decomposition is also often required to achieve desired mesh properties such as local density control, orthogonal meshes, etc.

To overcome the problems above, one must develop tools as independent as possible from the underlying CAD system. The concept of virtual topology is introduced to accomplish these objectives.

3. Virtual Topology Definition

Among several representations of 3D models that are used in CAD and the subsequent analysis, B-Rep has been accepted as the most common. It provides a complete and unambiguous definition of the solid through a concise representation of the bounding topology and a robust definition of the underlying geometry.

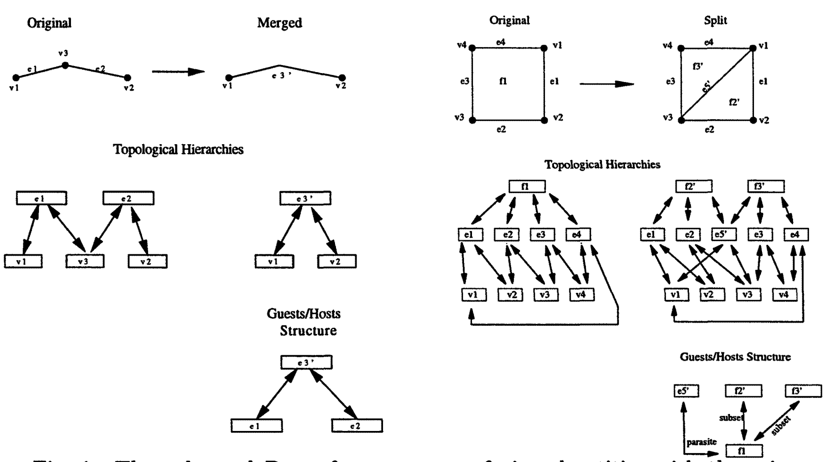


Fig. 1. The enhanced B-rep for two types of virtual entities with the original and resulting topological hierarchies and the resulting virtual hierarchy: (Left) A superset virtual edge; (Right) Two subset virtual faces, resulting from splitting a face with a virtual parasite edge

The B-Rep describes the topology and the geometry of a solid boundary. It uses vertices, edges, faces and volumes as the basic topological entities. The B-Rep provides a full description of the topological connectivity between the entities, such as which edges bound a face, which faces form an envelope of a volume, and so on. The connectivity information of a solid can be viewed as a hierarchical structure with a volume at the top and vertices at the bottom and with links between each higher dimension entity and lower dimension entities that bound it (as in Fig. 1). When referring to entities in the hierarchy, above the current entity, the term "upper topology" is commonly used. Entities below the entity in the hierarchy are referred to as it's "lower topology".

Each topological entity has a full geometric description such as a curve for an edge and a surface for a face. The geometric description is usually encapsulated as a separate entity.

This approach creates a one-to-one correspondence between topology and geometry, such that any adjustment of the topology requires changing the geometry as well, in order to preserve the unique correspondence of topology and geometry.

The virtual topology enhancement of the B-rep uncouples the topology from the geometry. While in the standard B-Rep each entity has a mathematical (geometric) description attached, in the enhanced B-Rep an entity can have a virtual description instead. Thus two types of topological entities are defined, based on the geometry description attached:

- *Real*, containing as the description the exact, mathematical definition of the entity geometry (curve, surface, etc...)
- *Virtual*, referencing topological entities from which the entity geometric description is to be derived and the type of relationship with those entities as described below.

The entities on which the virtual topological entity relies are defined as *hosts* of the entity. A virtual entity provides all the geometric properties of a real entity by accessing the hosts' data. An entity that relies on another entity is called a *guest* of that entity.

The structures of the guest/host relationships vary depending on the editing operations applied, and the nature of the topology being edited. Virtual topological entities are classified as one of the following types based on the relationship with their host(s):

- Supersets. These entities reference several host entities, representing their union (e.g. two edges combined into one as in Fig. 1 (Left)).
- Subsets. These entities reference a single host, representing a portion of the host entity (e.g. a part of a host face as the two triangular faces *f2'* and *f3'* on Fig. 1 (Right)).
- Interpolants. These entities reference several host entities and form an average entity based on the host definitions (e.g. a single edge based on a set of nearly overlying edges).
- Parasites. These entities reference a single higher dimensional host and rely on it to provide geometry (e.g. an edge constrained to lie on a face as is the case for *e5'* on Fig. 1 (Right)).
- Orphans. These entities have no host entities, and hence no actual geometric description and rely on a very simple geometry derived from their boundary (e.g. a face constructed from a set of edges with no explicit surface definition).

This way in the enhanced virtual B-Rep two hierarchical structures are provided, the topological connectivity structure and a hosts/guests structure which holds in it's leafs *real* entities, and through which the virtual entities derive their geometric properties. The *host* topological entities in most cases are no longer part of the B-Rep of the object, but only part of the virtual hierarchy (as in the examples in Fig. 1).

Two examples of the enhanced B-rep structure are shown in Fig. 1. On the left figure a *superset* edge based on two straight *host* edges is displayed. On the right a virtual *parasite* edge is created on top of a face, and two *subset* virtual faces are created on the two resulting face parts.

Virtual entities themselves can be edited and thus the guest/host relationship can be recursive. In other words, the newly created virtual layer is an actual topology which can be edited similarly to the initial real one. Thus a nesting of virtual entities is allowed whereby a guest of one entity may itself be a host of another entity.

The separation of topology and geometry allows a common simplified interface for both real and virtual topological entities. The user is not exposed to the type of virtual entity and the guest/host relationship is hidden to simplify user interactions.

4. Virtual Topology Operators

Construction of a suitable topology is achieved through the use of a set of virtual topology operators. The operators largely correspond to the Euler operators defined by Mäntilä,[14] and were defined to perform the operations necessary when editing CAD models for analysis. All the operators are applied *only* on the model topology, leaving the geometry unchanged as opposed to standard CAD or solid modeling systems.

They include low level operators performing all the necessary editing operations, as well as several high level operators performing commonly repeated sets of operations. The editing process commonly starts with a standard B-Rep with no virtual entities. Necessary adjustments are then done using these operators.
The low level virtual topology operators include:

- Merging of adjacent entities of similar dimension, forming a *superset* entity.
- Splitting of entities into parts and forming *subset* entities.
- Connection of a set of entities into an *interpolant* entity which interpolates the host entities.
- Construction of individual virtual entities, of either the *parasite* or *orphan* type, i.e. with a higher dimensional host, or with no hosts.

The high level operators introduced in the present work perform common sequences of low level operators. They include: collapse of faces to an edge or a vertex; removal of through holes; complex connect operations.

The operators provide the user the tools to edit the model, but do not replace the decision making process. The possibilities of automation of the editing process are discussed in the sections below.

The operators are described in more detail in the following subsections.

4.1. Low Level Operators

4.1.1. Merge

The merge operator allows merging adjacent topologies of similar dimension into one by removing the common boundary. Entities of similar dimension can be merged if they share a common boundary and have common upper topology. Given a set of adjacent entities a virtual *superset* entity is created with the original entities as it's hosts. The lower topology of the new entity is the unshared lower topology of the hosts.

Fig. 2. (a) A rounded octagonal prism. (b) The model after merger of the vertical faces. The geometry is unchanged, but the topology is now that of a cylinder.

For example, two edges can be merged if they have a common vertex and belong to the same set of faces. In this case a new *superset* virtual edge is created with the two original edges as hosts and the unshared vertices of the host edges as end vertices.

The new entity resulting from a merge replaces the host entities in the upper topology definition (e.g. when merging two faces belonging to a single volume, the new face replaces the two host faces in the bounding faces list of the volume). An example of face merge is shown in Fig. 2, where all the side faces of the prism are merged into a single face.

4.1.2. Split

A single topological entity can be split into two or more pieces by introducing new boundaries. An entity used as a splitting boundary can be an existing virtual/real entity, but more commonly a virtual *parasite* entity is created especially for this use. The parasite entity is usually constructed as a guest of the entity that is being split. After the construction of the boundary, virtual *subset* entities are created on parts of the original entity.

For example the procedure to split a face is to first create a *parasite* edge lying

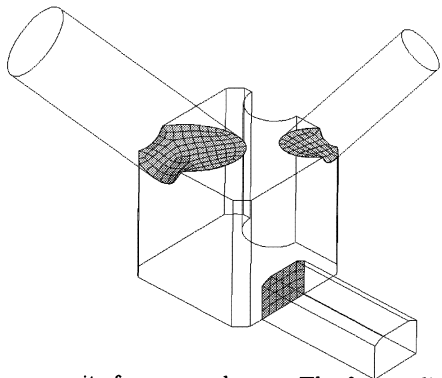


Fig. 3. Mesh on three *parasite* faces on volumes. The faces split the volume into four parts.

across a face and then to create two *subset* faces on the two resulting face parts which use the original face as a host and use as bounding edges the resulting edge loops.

Similar to the merge operation the entities resulting from a split replace the host entity in any upper topology definitions (e.g. when splitting an edge belonging to a number of faces, the two new edges replace the split edge in the edge lists of the faces). A single volume split into four connected volumes using three virtual *parasite* faces is shown in Fig. 3.

4.1.3. Construction Operator

Sometimes virtual entities have to be created independently, and not as a result of editing existing entities. Usually such entities are later used as tools for other operators such as a split. A construction operator is introduced allowing construction of a virtual entity based on the definition of it's lower topology and it's host.

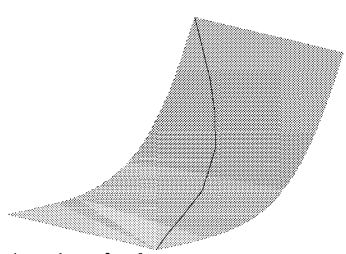


Fig. 4. A *parasite* virtual edge on a face. Since the exact NURB edge projection is not required, the display is a rough projection of a straight line between the end vertices.

The virtual type of the entity constructed depends on the host argument.

- *Subset* entities are constructed given a host of similar dimension (e.g. a partial face).
- *Parasite* entities are constructed given a host of higher dimension (e.g. a virtual vertex on an edge). Since the higher dimension entity can not define

it's guest's geometry, the *parasite* virtual entities have a minimal geometry of their own based on their lower topology combined with a constraint to be on the host entity. For example a *parasite* edge on a face host is regarded as a projection on the face of a straight line between its end vertices. Examples of parasite entities can be seen in Fig. 4 where a *parasite* edge on a face is shown, and in Fig. 3 where three *parasite* faces are used to split a volume.

- *Orphan* entities are constructed when no host is given. They have a minimal geometry based solely on their lower topology. For example an *orphan* face is regarded as a least square fit of a plane to the edges that bound it. The definition is sufficient to answer the geometric queries about the face. During meshing and smoothing nodes are usually allowed to migrate off the plane to produce a "spider web" like effect for the mesh on the face. An example of a meshed orphan face is given in Fig. 10.

Entities created through the use of the construct operators do not initially form a part of the higher topological hierarchy.

4.1.4. Connect

The connect operator is used to combine nearby entities into one, constructing an *interpolant* entity. For a set of entities of similar dimension to be connected they have to share common lower topology (i.e. for a set of edges to be connected they have to have the same end vertices). The geometry of the new entity is derived from the host entities. For example an *interpolant* edge when queried returns geometric values based on interpolating the hosts' values (Fig. 5).

(a) (b)

Fig. 5. (a) Original topology, with mismatched edges and vertices. (b) The topology after the vertices and edges were connected. The faces are replaced by virtual faces that take into account the connected edge. Likewise the edges at the connected vertices are replaced by virtual edges.

The new *interpolant* entity replaces all it's hosts in their upper topology definitions, resulting in connected upper topology. This can be seen in Fig. 5 where two unconnected faces are connected by sharing the new virtual edge.

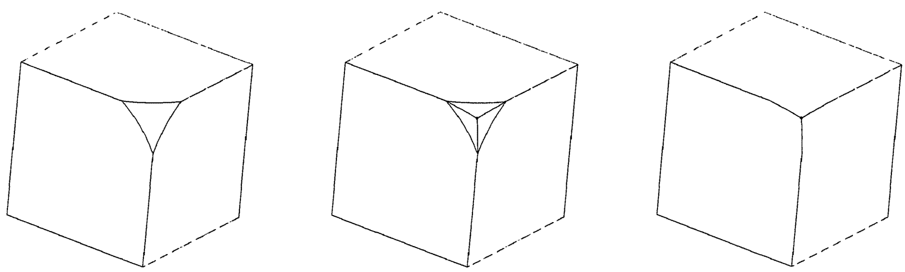
When connecting several entities into one, it is necessary to introduce virtual upper topology in order to preserve geometric consistency. For example when merging two edges of adjacent faces the adjacent faces (the upper topology) must become virtual since their geometry must also incorporate the change of borders as a result of the new virtual edge. Notice the change of the geometric description of the edges sharing the connected vertices in Fig. 5.

4.2. High Level Operators

When editing a model to suit the meshing needs, several often repeated sequences of operators exist. Some of those sequences can be regarded as high level editing operations, like removal of through holes, collapse of a face to a vertex or an edge, and complex connect operations. High level operators are provided to automate the performance of these sequences, and reduce the required user interaction.

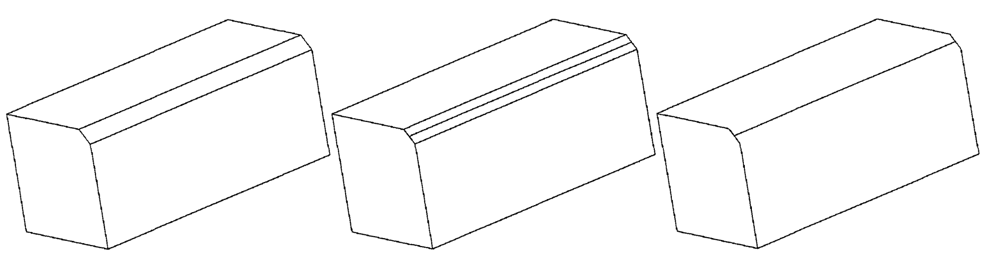
4.2.1. Collapse

The collapse operator is used to provide a symmetric "dispersal" of faces between neighboring faces. This is useful in the suppression/removal of small details such as sliver faces. Merging such a face with only one of it's neighbor will often lead to an asymmetric structure, which is not desirable. The best solution is often the "collapse" of the face to a vertex or an edge such that it's area is divided equally between a set of it's neighbor faces like in Fig. 6 and 7.



(a) (b) (c)

Fig. 6. The stages of a face collapse to a vertex: (a)A brick with a cut corner. (b) Splitting the corner face into three parts. (c) The final structure after the merge of the parts with adjacent faces.



(a) (b) (c)

Fig. 7. The stages of a face collapse to an edge: (a) A brick with a chamfered face. (b) Intermediate state after splitting the chamfer into halves. (b) The brick after the collapse of the chamfer between two neighbor faces.

The stages of a collapse operator applied on face f and a subset of it's neighbor faces S_f are:

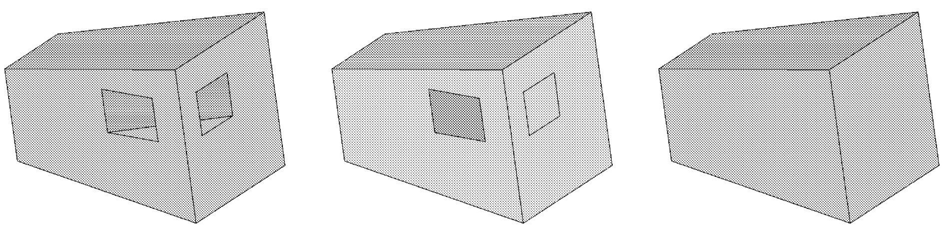
- Split the edges of face f unshared with S_f faces, to provide a symmetric face split.

- Create a set of virtual *parasite* edges on f using as end points the center of face f and the boundary vertices common to S_f faces or the splitting vertices from the previous stage.
- Split face f into subfaces using the *parasite* edges. (Figures 6 and 7 (b).)
- Merge each subface with the adjacent S_f face (since each subface has only one adjacent S_f face there is no ambiguity).

Fig. 6 shows a collapse of a corner face between all of it's neighbors. In Fig. 7 a collapse of a sliver face to an edge is described. Only two of the neighbor faces participate in this collapse.

4.2.2. Hole Removal

Among minor details highly complicating the mesh generation process, through holes are often one of the most "troublesome". A high level virtual operator using several lower level operators can be used to remove the holes from the model topology.



(a) (b) (c)

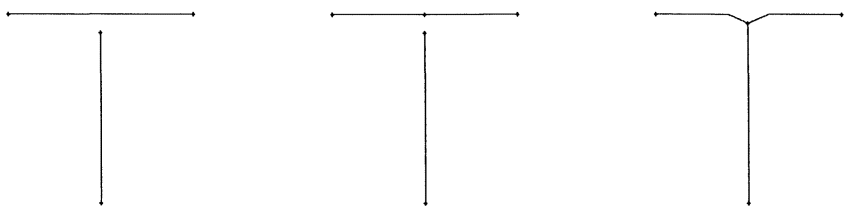
Fig. 8. The stages of a through hole removal: (a) The original model. (b) Virtual faces created on the hole caps, and the envelope faces are stitched into a new volume. (c) Each virtual face is merged with it's surrounding face.

The stages of the removal (demonstrated in Fig. 8) are:

- Virtual *orphan* faces are constructed from the edge loops on the two hole caps as in Fig. 8(b) (note that providing this operation using geometric tools might be very complicated, based on the edge loop geometry).
- A virtual volume is stitched from the faces of the original volume not including the hole faces plus the two virtual cap faces. The new virtual volume replaces the original in the access/display lists.
- Finally the caps are merged with their surrounding faces (Fig. 8 (c)).

4.2.3. "T"-connect

A "T"-intersection connect is used to handle situations as shown in Fig. 9 where a vertex is geometrically on/near an edge but is not connected to it. A high level operator introducing a connected topology for the intersection was defined using the following sequence of low level operators, demonstrated in Fig. 9:



(a) (b) (c)

Fig. 9. The stages of a simple "T"-connect (a) The original disconnected "T". (b) Splitting the "T" edge with a projection of the connect vertex. (c) The connected structure after connecting the original and new vertices. (The edges display is affected by the change of end vertex location)

- A virtual *parasite*.vertex is constructed on the edge, using as it's coordinates the projection to the edge of the old vertex.
- The edge is split using the new vertex (Fig. 9(b)).
- The old and new vertices are connected as shown in Fig. 9(c).

5. Use of Virtual Topology for Modeling

As described in the introduction, models acquired from CAD sources are usually not suitable to be used "as is" by the existing range of mesh generation tools. Hence a pre-processing phase usually exists where the model is edited to make it meshable. The editing usually includes completion and correction of the model. Once a model is well defined it is usually simplified and when necessary decomposed into several parts.

The virtual operators provide the tools to perform the editing operations required by the different tasks avoiding the costly direct geometric adjustments.

Even with introduction of easy to use and fast operators for the pre-processing, manual correction of each inconsistency problem or removal of each minor detail can be extremely tedious and time consuming. Hence full or partial automation of different pre-processing tasks is essential.

5.1. Geometry Completion and Correction

As explained above the first step in incorporating a CAD model into an analysis system usually includes completion of the geometric description and multiple adjustments of the model to ensure consistency and correctness.

Commonly occurring problems include:

- Faces with no surface description resulting from incomplete models or undefined geometry in the export files.
- Face with misplaced edges - often the description of the face surface and the curve description of the edges embedding the surface, do not match - i.e. the edges as defined by the curves are not on the surface.

- Duplicate entities - a common problem is that a lower topology entity referenced by several higher topology entities, has different geometric definitions, as derived from the upper topology, and hence ends up defined as several separate topological entities.
- "T"-intersections - in CAD representation of objects using 3D surface models, the surface patches may sometimes create "T"- intersections (i.e. situations where a vertex of one patch is lying on an edge of another as demonstrated in Fig. 13).

More complex problems may occur, but most can be subdivided into a sequence of the ones described above. Virtual topology provides the means to handle those problems as described in the following subsections.

The vast majority of those problems can be automatically detected based on an error tolerance. This tolerance is usually based on the source CAD system tolerance derived from the input file. as explained below, and the suitable fix operation be initiated.

5.1.1. Face With No Surface Definition

Finding a fitting surface for a face based on the edges is non-trivial and might often result in surfaces with very high curvature, deviating far from the surrounding edges. Virtual topology provides a simpler and often more "intuitive" (low-curvature) solution via construction of an *orphan* virtual face based on the lower topology description (edges) instead. While it is clear that geometric accuracy may be lost, in most cases the description provided by an *orphan* entity based on the lower topology provides a good enough basis for the generated mesh like in Fig. 10, where an orphan face is constructed over a semi-circular loop of edges.

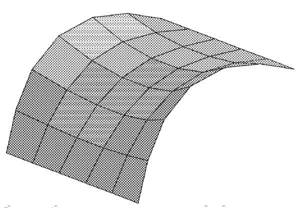


Fig. 10. Mesh of an orphan face constructed from semi-circular loop of edges. Note that although no surface is given the mesh has an almost cylindrical shape.

Unrecognized surface types are detected when loading a model (e.g. reading an IGES file) and the construction operator can be called automatically. In the case of wireframe models, user intervention is often required to distinguish which edge loops form a face and which are only interior loops.

5.1.2. Face With Misplaced Edges

A face's underlying surface and the edge's underlying curve are separately defined in most formats. Obviously, the edge definition should be such that the edge actually lies on the face it is bounding. However, an edge and face definition may not match when a low-tolerance definition is used in a higher-tolerance system as shown in Fig. 11.

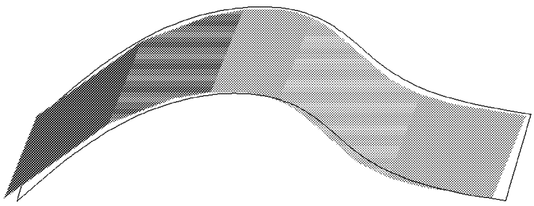


Fig. 11. Mismatched face bounding edges, lying off the face surface.

Correcting these discrepancies geometrically (i.e. with new definitions) is difficult and costly. Constructing a new face using only the edge definitions ignores any interior face curvature. However, the virtual construction operator can be used to create a virtual *subset* face with the edges as defined and a host of the existing face. The edges need not be on the host geometry, yet the surface definition will be preserved in the generated mesh.

Misplaced edges can be automatically spotted when loading the model into the system. If the distance between the edges and the face is within a prescribed tolerance (to prevent completely erroneous inputs), the construction operator is called.

5.1.3. Mismatched/Duplicate Entities

The problem of duplicate entity definitions can arise both due to differences in model representation and different tolerance requirements on source and target systems.

(a) (b)

Fig. 12. (a) Original topology, with mismatched edges and vertices. (b) The topology after the mismatched entities were connected.

This problem can be corrected using the virtual connect operator. Given a tolerance value (usually derived from the source system tolerance - if known), the algorithm proceeds from the lowest entities upward, correcting mismatched vertices, then edges and finally faces. Groups of entities that are within the prescribed tolerance are connected using the virtual connect operator, which replaces the set of entities by a single *interpolant* entity. An example of mismatched edges and vertices along with the corrected topology is shown if Fig. 12.

The described procedure is fully automatic, however manual control can be taken when needed to prevent undesirable connects or to specify entities to connect even if not inside the tolerance prescribed.

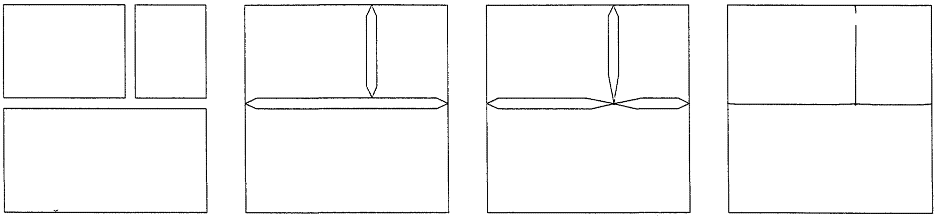


Fig. 13. Connect stages of three disconnected patches, including a "T"-intersection. (a) Original topology & geometry. (b) Structure after vertex connect. (c) Structure after a "T"-connect - where the edge in the "T"-intersection was split and then a common vertex for the "T" was created. (d) The fully connected structure after edge connect.

5.1.4. "T"-Intersections

The term "T"-intersection is commonly used to describe a situation where a corner (a vertex) of one face lies (geometrically) on an edge of another face, but is not topologically a part of it (as shown for example in Fig. 13 (a)). Such situations occur quite often when dealing with CAD models of complex surfaces described as a set of Bézier or other NURB patches. "T"-intersections can be resolved using a high level "T"-connect operator, described above, which performs a combination of virtual split and connect operators, topologically "projecting" the vertex to the edge to allow correct connection of the patches, as shown in Fig. 13.

5.2. Geometry Simplification

Once a complete and correct model is built, the next step in preparing it for meshing is usually the simplification of over-detailed geometries. Insignificant details may have a minor effect on the analysis result but severely complicate automatic mesh generation. The major constraint such details impose is placement of mesh nodes on the insignificant entity boundaries, which is especially restrictive when the entity is relatively small with respect to the mesh element size. Commonly occurring minor details include sliver faces and edges as well as minor features like fillets, blends, small holes, etc.

Another type of overspecified model complicating the mesh generation process is faceted geometries. Because of the CAD system representation used, surfaces/curves are often broken into multiple small faces/edges. Faceted models impose similar mesh nodes placement constraints as minor details. Hence in order to create good quality mesh sets of small facets need to be replaced by a single face.

Removing such details by changing the geometry, as suggested in Refs. [11,4], would require finding new surfaces and curves approximating the existing geometry over the constraining topological boundaries. Computation of such geometries might often be very costly, and even not always possible. Such change of geometry also leads to inevitable lose of accuracy.

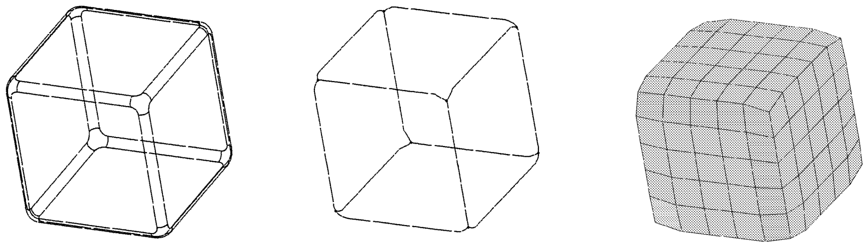


Fig. 14. A fully automated topology simplification of a blended cube. (a) the original topology; (b) the simplified - basic cube topology; (c) mesh of the simplified topology.

Fig. 15. (a) An apple constructed of multiple facets. (b) The simplified topology after face merge.

Using virtual topology provides the means of suppressing the insignificant details by use of the merge and collapse operators. Applying those operators removes the node placement constraint with no actual geometric changes, and therefore with no accuracy loss (Since the geometry is left unchanged, the accuracy of the mesh is dependent only on the element size.). Examples of the use of those operators for simplification are in Figures 14 and 15. In Fig. 14 a cube with blended edges and corners is edited by means of the collapse operator, to have a topology of a simple cube, enabling easy generation of a hexahedral mesh. In Fig. 15 an example of faceted geometry, merged to enable coarse meshing is shown.

Large design models often require simplification of multiple features before the model is suitable for meshing. Performing such simplification manually is time consuming and tedious. Hence the need for automated and semi-automated tools for such detail detection and subsequent suppression. Careful selection of which details to suppress should ensure sufficient FE analysis accuracy, while simplifying the mesh generation process. Several methods for such detail recognition and suppression has been suggested in the past.

The application of medial axis, for detail suppression was introduced by Armstrong.[1] While the medial axis does provide all the information required for detail suppression, it's computation (as described in Ref. [17]) is based on global analysis of the object and therefore is highly complex, expensive, and not always possible. The construction of medial axis requires the analyzed object to be well defined and doesn't handle non-manifold topologies or models containing sets of faces not contained by a volume.

(a) (b)

Fig. 16. (a) A nut with complex blends and bulges; (b) Mesh after topology simplification.

Another approach that could be used for detail suppression is that of feature recognition as described by Dabke.[5] This approach requires defining a set of insignificant features to be detected and the removal from the model of the detected features. It requires heavy computations and demands a correspondence of details and features, which doesn't always exist (e.g. sliver faces).

An automated scheme for detail suppression using virtual topology was introduced by Sheffer.[19] This clustering algorithm suggests an alternative topology for the model, based on the model geometry. The suggested topology is based on subdivision of the model faces envelope into regions of restricted curvature and distance deviation.

The clustering algorithm provides automated determination of entities to be merged or collapsed, based on a user defined approximation tolerance. It is based on the analysis of a faceted approximation of the model faces, enabling very fast computation, since it involves manipulation of linear computations only. It avoids the complexity of both global analysis or explicit detection of the suppressed details. Some examples of automated geometry simplification are in Figures 16 and 19, where the original objects and the mesh after simplification are shown.

While the provided scheme does not detect *all* topological changes resulting in simplification it enables fast simplification of a vast amount of geometries, and significantly shortens the geometry simplification part of the model pre-processing for meshing.

5.3. Geometry Decomposition

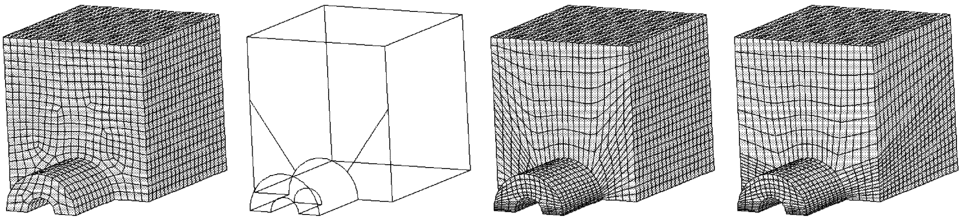
Decomposition of the model into several parts is often the final step of model pre-processing for meshing. It is used both to simplify the meshing and to achieve the desired mesh properties.

Various boolean operators have been historically used to split off portions of the real geometry. The main drawback to this approach is that often the cutting tool (surface or volume) is itself hard to define. For example, the volume shown in Fig. 3 contains the oblique intersection of a cylinder with a cube. Finding a fitting surface for such an intersection containing all the edges is non-trivial and might often result in surfaces with very high curvature deviating far from the surrounding edge loop.

The virtual topology provides the tools to split the volume using a given edge loop, avoiding the heavy computations required for the boolean operation, by constructing a *parasite* virtual face on the volume using as its lower topology the dissection edges. The face is then used as the splitting tool for a virtual split operator decomposing the volume into two parts.

5.4. Mesh Adjustment

Often after the model is finally prepared for meshing and meshed, the resulting mesh does not satisfy the user requirements and the model has to be re-edited and remeshed. One of the more common adjustments required is correcting the model decomposition. These adjustments can be categorized into those which require a change of topology and those which require a change in "position only" (e.g. the location of a vertex). The required changes in topology are straight forward using the virtual topology operators.



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

Fig. 17. Adjusting a model mesh by use of decomposition. (a) Original volume mesh. (b) Virtual split of the top face using *parasite* edges. (c) The volume mesh after the split. (d) The mesh adjusted based on sliding the edges along the face.

A change in 'position only" is especially well suited to virtual topology. In many cases, decompositions result in splits with the splitting entity being a *parasite* vertex or edge. This implies that the only constraint on the entity is that it remain on its host and hence can be dynamically slid on the host to reposition the mesh interactively as shown in Fig. 17 (d). Also the split can be made with the mesh in place, requiring no remeshing at all. The virtual topology then becomes a mechanism to perform macro mesh edits.

6. Examples and Results

The examples displayed below demonstrate two of the major pre-processing steps: geometry correction and model simplification. In Fig. 18, a model of a sedan loaded from an IGES format file is displayed. The model contains 87 NURB patches, and has a tolerance of 0.6 (the solid modeler in our system has a tolerance of 10^{-6}), there is no topological connectivity between patches and multiple "T"-intersections exist. Automatic correction of the model using the input tolerance fixes 99% of the inconsistencies. The automatic correction takes about 5 minutes on HP900, with the manual correction of the remaining problems taking about the same time. After a correct model was established it was simplified to contain only 15 faces by merging groups of faces prior to the mesh generation.

In Fig. 19, a bracket-like shape is described, with multiple blends, fillets and

Fig. 18. (a) A sedan model as loaded from an input IGES file. (b) Surface mesh of the sedan after model correction and simplification.

other minor details. The model was simplified using the fully automated clustering algorithms based on user prescribed angle and distance tolerance. This resulted in a simple topology meshable by the "Cooper Tool" as described by Blacker.[3]

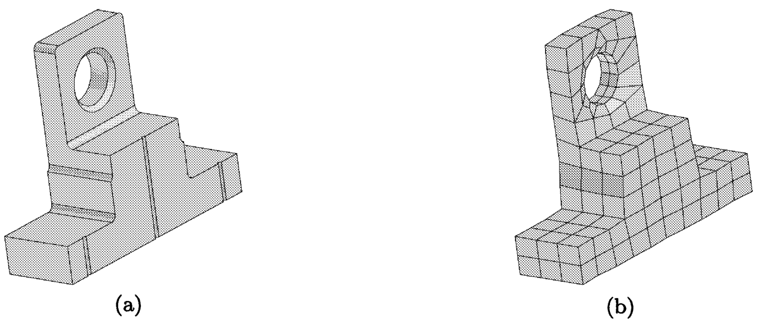


Fig. 19. (a) A bracket-like object with multiple blends, fillets and notches; (b) Volume mesh after topology simplification.

The virtual topology is implemented as part of the preprocessing package called GAMBIT, being developed at Fluent Inc. As seen in the examples it provides the tools to transform a CAD model into a meshable model, allowing flexible topology editing while maintaining the object geometry. The automated geometry correction and simplification tools serve to reduce significantly the amount of required user interaction to speed up the model preprocessing. The system is written in C++ (using inheritance so as to make the access to real/virtual entities completely transparent for the interface) and it runs on multiple platforms.

7. Conclusions

In this paper an overview of the problems arising when integrating CAD data into an analysis system was presented, together with an inclusive system for resolving such problems using the virtual topology concept. Both the virtual B-Rep enhancement and a set of virtual operators sufficient to perform the model preprocessing for meshing were described. The use of virtual geometry at numerous phases of the meshing process was described.

This technique provides the tools to easily perform:

- correction of common problems found in legacy geometries;
- editing of the geometry to be more suitable for meshing;
- decomposition of the geometry to make it more suitable for existing meshing tools;

- dynamic adjustment of the mesh by relocating the virtual topology.

As has been shown, the virtual geometry is a broadly applicable approach to solving the challenges associated with meshing real world geometry.

The tools presented provide significant automation of model pre-processing for meshing. However further automation can be achieved. Hence further research is required both in extension and refinement of the existing algorithms and in the development of other automation techniques and tools.

References

1. C. G. Armstrong, D. J. Robinson, R. M. McKeag, T. S. Li, S. J. Bridgett, R. J. Donaghy, C.A. McGleenan, "Medials for Meshing And More". *Proc. 4th International Meshing Roundtable*, SAND95-2130, Sandia National Laboratories, Albuquerque, NM, Oct. 1995, pp. 277–288.
2. T. D. Blacker, R. J.Meyers, "Seams and wedges in plastering: a 3D hexahedral mesh generation algorithm", *Engineering with computers* **9** (1993) 83–93.
3. T. D. Blacker, "The Cooper Tool", *Proc. 5th International Meshing Roundtable*, SAND 96-2301, Sandia National Laboratories, Pittsburgh, USA, Oct. 1996, pp. 13–29.
4. G. Butlin, C. Stops, "CAD Data Repair", *Proc. 5th International Meshing Roundtable*, SAND 96-2301, Sandia National Laboratories. Pittsburgh, USA, Oct. 1996. pp. 7-12.
5. P. Dabke, P. Vallury, S. Sheppard, "Using Features to Support Finite Element Idealizations", *ASME Computers in Engineering Conf.*, Minneapolis, MN, Sept. 1994.
6. D. A. Field, "The legacy of automatic mesh generation from solid modeling", *Computer Aided Geometric Design* **12-7** (1995) 651–673.
7. P. L. George, "Improvements on Delaunay-based three-dimensional automatic mesh generator", *Finite Elements in Analysis and Design* **25** (1997) 297–317.
8. D. I. Holmes, "Generalized Method of Decomposing Solid Geometry into Hexahedron Finite Elements", *Proc. 4th International Meshing Roundtable*, SAND95-2130, Sandia National Laboratories, Albuquerque, NM, Oct. 1995, pp. 141–152.
9. *International Standard ISO 10303-42:1994(E)*, (International Organization for Standardization), First edition, 1994.
10. *Initial Graphics Exchange Specification (IGES), Version 5.1*, (U.S. Department of Commerce, National Bureau of Standards, Sept. 1991).
11. M. R. Jones, M. A. Price, G. Butlin, "Auto-Meshing", *Proc. 4th International Meshing Roundtable*, SAND95-2130, Sandia National Laboratories, Albuquerque, NM, Oct. 1995. pp. 153-164.
12. A. D. Kalvin, R. H. Taylor, "Superfaces: Polygonal Mesh Simplification with Bounder Error" *IEEE Computer Graphics and Applications* **16-3** (1996) 64–77.
13. "*How to* - Integrate CAD and Analysis", CAD/FE Integration Working Group, (NAFEMS Finite Element Methods & Standards, 1996)
14. M. Mäntilä, *An Introduction to Solid Modeling* (Computer Science Press, Maryland, 1988)

15. M. A. Price, M. A. Sabin, C. G. Armstong, "Hexahedral Mesh Generation by Medial Axis Subdivision: I. Solids with Convex Edges", *Int. J. of Numerical Methods in Eng.* **38** (1995) 3335–3359.
16. R. Schneiders, "Automatic Generation of Hexahedral Finite Element Meshes", *Proc. 4th International Meshing Roundtable*, SAND95-2130, Sandia National Laboratories, Albuquerque, NM, Oct. 1995. pp. 103–114.
17. D. J. Sheehy, C. G. Armstrong, D. J. Robinson, "Computing the medial surface of a solid from a domain Delauney triangulation", *ACM Symposium on Solid Modeling Foundations and Applications*, Utah, 1995, 201–212.
18. A. Sheffer, T. D. Blacker, J. Clements, M. Bercovier, "Virtual Topology Construction and Applications", *Geometric Modeling Theory and Practice*, (Springer Verlag, 1996) 247–259.
19. A. Sheffer, T. D. Blacker, M. Bercovier, "Clustering: Automated Detail Suppression Using Virtual Topology", *Trens in Unstructured Mesh Generation, ASME*, AMD-Vol. 220 (1997) 57–64.
20. R. J. Smith, M. A. Leschziner, "A Novel Cartesian Grid Method for Complex Aerodynamic CFD Applications", *5th International Conference on Numerical Grid Generation in Computational Field Simulations*, eds. B. K. Soni, J. F. Thompson, J. Hauser, P. Eiseman, Mississippi State University, 1996.
21. T. J. Tautges, T. D. Blacker, S. A. Mitchell, "The Whisker Weaving Algorithm: A Connectivity-Based Method for Constructing All-Hexahedral Finite Element Meshes", *Int. J. of Numerical Methods in Eng.* **39-19** (1996) 3327–3349.