Engineering with Computers (2004) 20: 210–221
DOI 10.1007/s00366-004-0289-z

ORIGINAL ARTICLE

Mark W. Beall · Joe Walsh · Mark S. Shephard

A comparison of techniques for geometry access related to mesh generation

Received: 28 July 2003 / Accepted: 12 April 2004 / Published online: 20 August 2004

Abstract One of the major issues of mesh generation today is access to geometry in an accurate and efficient manner. This paper will review several of the issues associated with accessing geometry for mesh generation. This paper will also evaluate alternative techniques for accessing geometry and review how these techniques address, or do not address, the issues related to geometry access for mesh generation. The techniques for geometry access to be reviewed include: translation and healing, discrete representations, direct geometry access, and unified topology accessing geometry directly. The intent of this paper is to provide an overview to the alternative approaches and how they address the specific issues related to accessing geometry for mesh generation. It is not the intent of this paper to provide detailed algorithms related to accessing or repairing geometry data.

Keywords CAD · Geometry · Topology · Tolerances · Design integration · Adaptivity · Mesh generation · Geometry-based · Geometry access · Unified topology model · Defeaturing

1 Introduction

Automatic and semi-automatic mesh generation has seen dramatic improvements over the last ten years. One of the most important, and often overlooked, aspects to mesh generation is accessing geometry. The emphasis on analysis in recent years has moved from failure analysis and validation to becoming an active part of the design process. There is a growing demand from manufacturing companies to include a performance evaluation based on simulation results earlier in the design process, making simulation an integral part of their design process. To do this in a cost-effective manner requires automation of all of the steps involved in performing such simulations from the product design data. Accessing geometry for mesh generation is still one of the major technical issues related to moving simulation forward as an essential ingredient of the design process. This desired ability to move simulation forward in the design process requires a review of current techniques for accessing geometry.

This paper will review potential sources of geometry, along with several of the issues related to geometry access, and will evaluate four techniques for geometry access as follows:

1. Translation and healing
2. Discrete representations
3. Direct geometry access
4. Unified topology accessing CAD geometry directly

M. W. Beall (✉)
Simmetrix Inc., Clifton Park, NY, USA
E-mail: mbeall@simmetrix.com

J. Walsh
Simmetrix Inc., Clarkesville, GA, USA
E-mail: jwalsh@simmetrix.com

M. S. Shephard
Rensselaer Polytechnic Institute, Troy, NY, USA
E-mail: shephard@scorec.rpi.edu

2 Potential sources of geometry

The most common source of design data is CAD geometry. Almost all CAD systems have evolved into similar representations for their models. This representation often includes feature-based data and a resulting B-rep instance or a B-rep model.

The B-rep model consists of much more than just geometry. B-rep models contain geometry (shape), topology (how things are connected), and tolerances (how closely do they actually fit together). This combination of model data is then accessed by the CAD systems' methods to define a valid B-rep model. Therefore, a valid B-rep model should be considered to consist of geometry, topology, tolerances, and methods used by the CAD system it was defined within [1].

CAD systems often use relatively large tolerances on an entity-by-entity basis to provide robustness to model operations. This approach is referred to as variable tolerances or tolerant modeling by different CAD systems. The use of these large variable tolerances produces gaps and overlaps in the geometry and topology of the CAD system B-rep model, as illustrated in the simple (and extreme) example in Fig. 1.

The algorithms used in the CAD system modeling engines are written to deal with these tolerances in a consistent manner, and they do not see the gaps or overlaps.

Geometric modeling kernels such as ACIS, Granite, and Parasolid are often used to supply the methods and model representations used by CAD system modeling engines. CAD systems that use a common geometric modeling kernel also share common methods for evaluating tolerances and the validity of a B-rep model. These methods can be accounted for directly in the mesh generation process in a consistent manner using information easily provided by the CAD system API [1, 2].

Another potential source of geometry for mesh generation is an internal representation of geometry in the system doing the mesh generation. This is a common approach for simulation applications that contain their own native representation. This internal geometry representation may come in many forms, including; Voxel- or facet-based representations, simplified geometry representations, or geometric modeling kernels such as ACIS, Parasolid, or Granite.

The third potential source of geometry for mesh generation is existing mesh data. A large number of legacy meshes are stored without the original CAD data. Use of legacy surface meshes as a geometry definition usually requires some form of coarsening or refining of the surface mesh. An example of a legacy mesh model is illustrated in Fig. 2.

The final potential source of geometry for mesh generation is scan data. There is an increasing desire to use scan data directly for mesh generation. This introduces a unique set of issues that include; large number of facets (millions), robust algorithms to ensure valid surface triangulations, and the need to coarsen data. Figure 3 illustrates an example of facet data received from scan data.

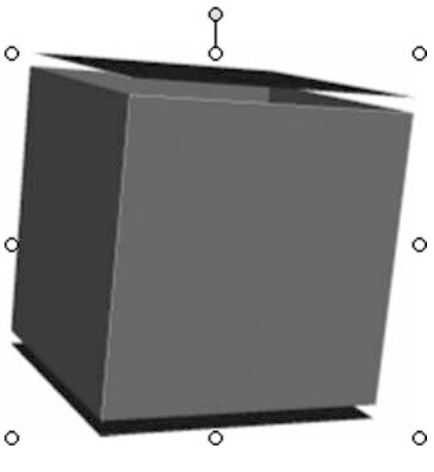


Fig. 1 Large/variable tolerances result in gaps and overlaps

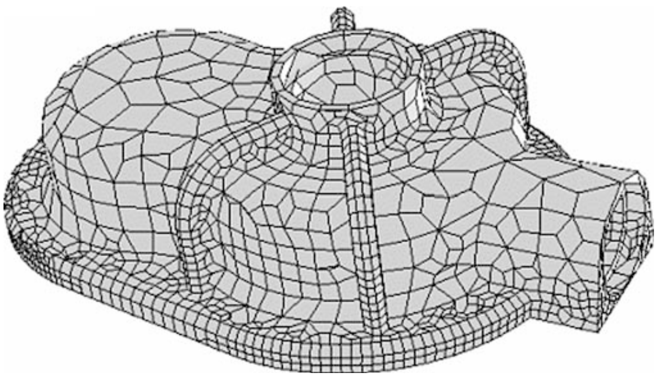


Fig. 2 Legacy mesh data

3 Geometry-related issues for mesh generation

There are several issues associated with effective and efficient access of geometry for mesh generation. This section will provide a quick overview of several of the major issues and the ramifications that these issues have on mesh generation. A detailed review of these effects is beyond the scope of this paper. Specifically excluded from this paper are model abstraction or idealization for analysis and domain decomposition.

3.1 Understanding the analysis requirements

The first major issue with geometry access for mesh generation is the need to understand the analysis

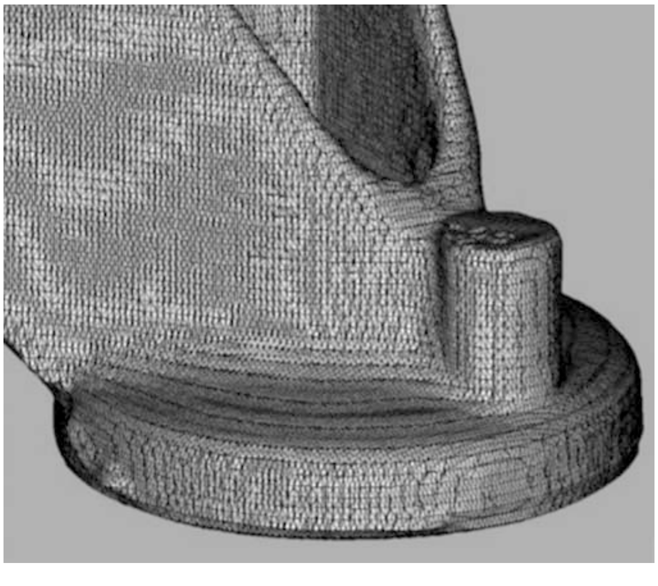


Fig. 3 Facet data from scan data

requirements. The appropriate mesh and geometry to be used for meshing is a function of the analysis to be performed and the desired accuracy [3]. There does not exist an optimal mesh independent of the analysis to be performed. A priori element shape quality tests have often been used as a misleading indicator of a good mesh, independent of the analysis to be performed or the accuracy desired. An appropriate mesh is one that produces the desired accuracy for the problem to be solved. In practice, the only means to reliably achieve such control is through the application of adaptive mesh control.

Different types of analyses require different instances of the geometry to capture the physics. For example, we can perform a dynamic structural response analysis and a computational fluid dynamics (CFD) analysis on the same part. The dynamic structural response analysis requires the solid geometry of the part while the CFD analysis requires the geometry of the cavities through which the fluid will flow. This simple illustration of the different uses of geometry representations is illustrated in Fig. 4.

Physics simulations, such as external flow, electromagnetics, and radiation, are actually concerned with the volume not occupied by the part.

Different types of analysis also require different resolutions of mesh to achieve the desired accuracy on a particular design.

3.2 Defeaturing

Defeaturing is one of the most complex issues associated with geometry access for mesh generation. We will classify features into two main groups.

The first group of features will be called ‘‘explicit features.’’ Explicit features are features that were explicitly defined as features in the model that drive the resulting geometry. In this case, a feature-based modeling system was used to create a model which contains explicit features. Explicit features can only be created by feature-based CAD modeling systems and can be suppressed by the original CAD modeling system.

The second group of features will be called ‘‘implicit features.’’ Implicit features are features that are created indirectly by the modeling process. One example of implicit features is the creation of engineering features, such as holes, by a modeling system that is not feature-based. The second example of implicit features is the creation of recognizable patterns of geometry/topology data that create a valid design model, but, at the level of mesh resolution desired, makes it difficult to create a well-conditioned mesh. Artifact features can be created from any modeling system and cannot be suppressed in the original modeling system. The only features that exist in geometry sources other than feature-based CAD systems are implicit features.

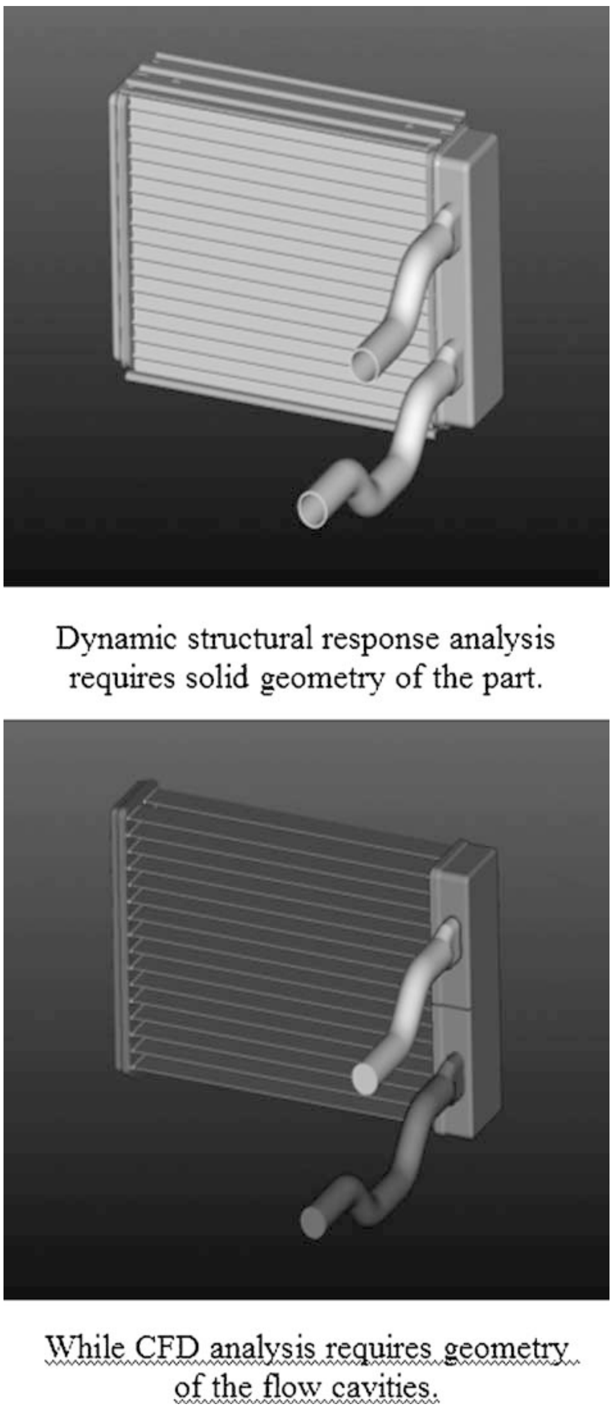

Dynamic structural response analysis requires solid geometry of the part.
While CFD analysis requires geometry of the flow cavities.

Fig. 4 Different analyses require different geometric representations

Part of the complexity associated with geometry access for mesh generation is due to the fact that, historically, analyses are performed too late in the design process and the design model contains more details than are appropriate for analysis. Moving the analysis earlier in the design process will help to reduce, but will not remove, the need for defeaturing. Since multiple analysis types may be required for any design state, there remains a need for defeaturing to various levels to support the range of analyses to be performed.

One of the most common unwanted implicit features encountered are "slivers." Slivers can be described as very small implicit features that are larger than the geometric tolerances, but extremely small with respect to the model size. These very small implicit features can provide problems for mesh generation algorithms and are meaningless to the analysis [4]. Slivers are introduced into models to maintain validity and integrity of the model. Native models contain far fewer slivers than translated models. A very common method of healing or repair algorithms used in translation is to introduce slivers to resolve gaps, overlaps, and tangency conditions.

Another common unwanted implicit feature type is "small" model features. Small model features can be described as implicit features that are very large with respect to the geometric tolerances, but small with respect to the local target mesh size. This definition of small features indicates that the classification of a small feature is a function of the target element size and accuracy desired. The actual definition of the small sizing with respect to target mesh size can vary with each analysis to be performed. Some typical values for small features are less than 25–30% of the target mesh size. This definition also allows for the support of an adaptive representation of geometry used for meshing as part of the mesh adaptivity process that we will discuss further later in this section. The issue of dealing with small geometric features in the mesh generation process has been discussed in various references [5, 6]. An example of a small feature and its potential impact on mesh generation is illustrated in Fig. 5.

Slivers may be re-classified as a special case of small features that will remain small through all possible target mesh sizes.

CAD models may include geometric features that are important for design, but are irrelevant for the simulation to be performed. These unwanted features can be classified as "simple" features and "complex" features. These features can be suppressed by the CAD system if and only if they were explicit features.

Simple features can be described as features which, when suppressed or removed, refer back to a single parent face on the B-rep model. Simple features may be explicit features or implicit features, but are most likely explicit features from a feature-based modeling system. Simple features are defined in terms of the topology of their base features rather than their size. Examples of simple features are illustrated in Fig. 6.

Complex features can be described as features that are not simple. Complex features may be explicit features or implicit features, but are most likely explicit features from a feature-based modeling system. These features include a variety of features as follows:

- Features whose base feature spans across multiple faces
- Features whose base features need to be extended for feature removal or suppression, such as fillets and chamfers
- Features that interfere with other features

Complex features are the largest challenge to deal with in defeaturing. If these features are not small with respect to target mesh size, careful consideration should be given regarding why these are being defeatured and the impact on accuracy. If these features are small, then they can be treated as small features independent of their complexity. For complex features that need to be removed or suppressed and that are not small, a thorough understanding of the feature data is required and it is usually best to suppress these in the CAD system prior to geometry access.

3.3 Tolerances and methods for evaluating tolerances

Understanding tolerances and methods for evaluating tolerances plays an important role in accessing CAD geometry for mesh generation. One of the key areas influenced by tolerances and their associated methods is that of tangencies and near tangencies. The methods

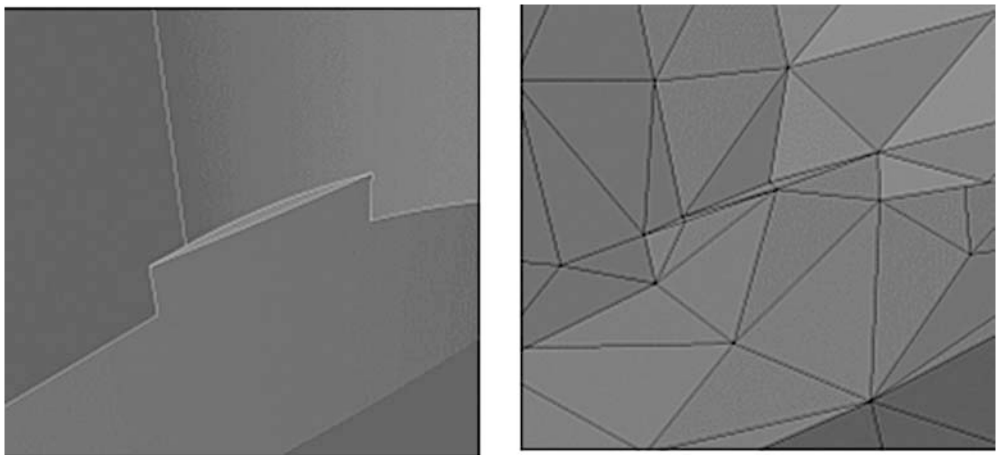


Fig. 5 Small feature

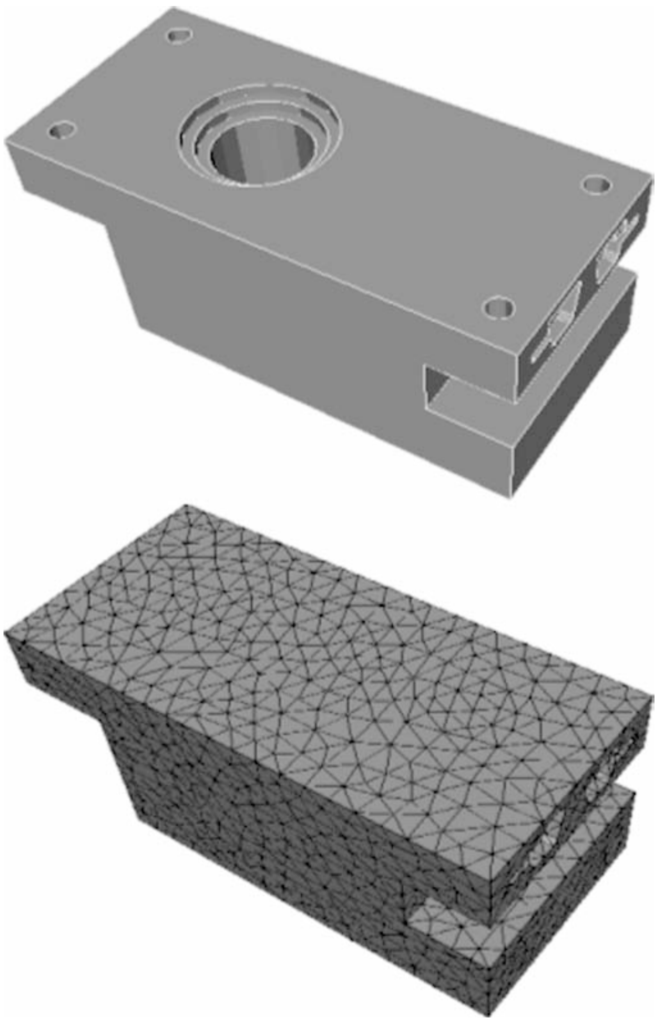


Fig. 6 Simple features on top face

used in CAD system modeling engines are written to deal with tolerances in a consistent manner. These methods are not available outside of the CAD system modeling engines; therefore, translated data often introduces "dirty" geometry.

3.4 "Dirty" geometry

Dirty geometry has been one of the most nagging issues related to geometry access. Dirty geometry consists of gaps, overlaps, and other incompatibilities in the model, preventing it from being valid. These incompatibilities do not exist in the native CAD system and are introduced from translating the native CAD geometry to another format. Differences in representations, methods, and tolerances between modeling engines create dirty geometry. Translators must then heal or repair the geometry to represent it as a valid model in the non-native system [4, 7, 8]. Note that, without knowledge of the modeling system tolerances and methods, there is no a priori means to ensure a healing process will successfully recover the correct model representation.

3.5 Support for curved meshing

The previous issues associated with geometry access have focused on ensuring the correct geometry representation and level of detail in the geometry be used for mesh generation. The next three issues deal with ensuring that the geometry access can support key mesh generation functionality. The first mesh generation functionality to be considered is curved meshing. Curved meshing involves the ability to create curved mesh edges and faces that have the level of geometric approximation needed to ensure that, as the simulation results are improved by the introduction of higher-order equation approximations (e.g., high-order finite elements), the geometric approximation errors do not control the solution accuracy. The need for the ability to properly curve the mesh entities arises as soon as higher-than-linear basis functions are used and, as demonstrated by a simple example in [9], the order of geometric approximation needs to be increased as the basis order increases.

In the simplest cases, the appropriate curved meshes can be created by moving mesh on the boundary of the model to the "closest" location on the model geometry. However, even in the simplest, and common, case of quadratic h-type finite elements (see the upper example of Fig. 7), a more complex algorithm is required to ensure that the elements can be properly curved [10]. The complexity of the curved mesh generation process increases further in the case of p-version methods where coarse meshes, such as the example at the bottom of Fig. 7, must have higher-order geometric approximations.

3.6 Support for curvature-based mesh refinement

The next meshing functionality to be considered as desirable is curvature-based mesh refinement. This meshing functionality provides automatic refinement of the mesh based on the underlying geometry curvature. The benefits of this functionality are: (1) the ability to capture the geometry with a considerably smaller number of elements and/or grid points, and (2) a resulting improvement in mesh quality in areas of rapid geometric changes. Figure 8 illustrates the benefits of curvature-based mesh refinement.

3.7 Support for geometry-based mesh adaptivity

The final mesh generation functionality to be considered as an issue for geometry access is the support for geometry- based mesh adaptivity. This functionality involves the ability of the adapted mesh to adhere to the original geometry, as illustrated in Fig. 9, and requires access to

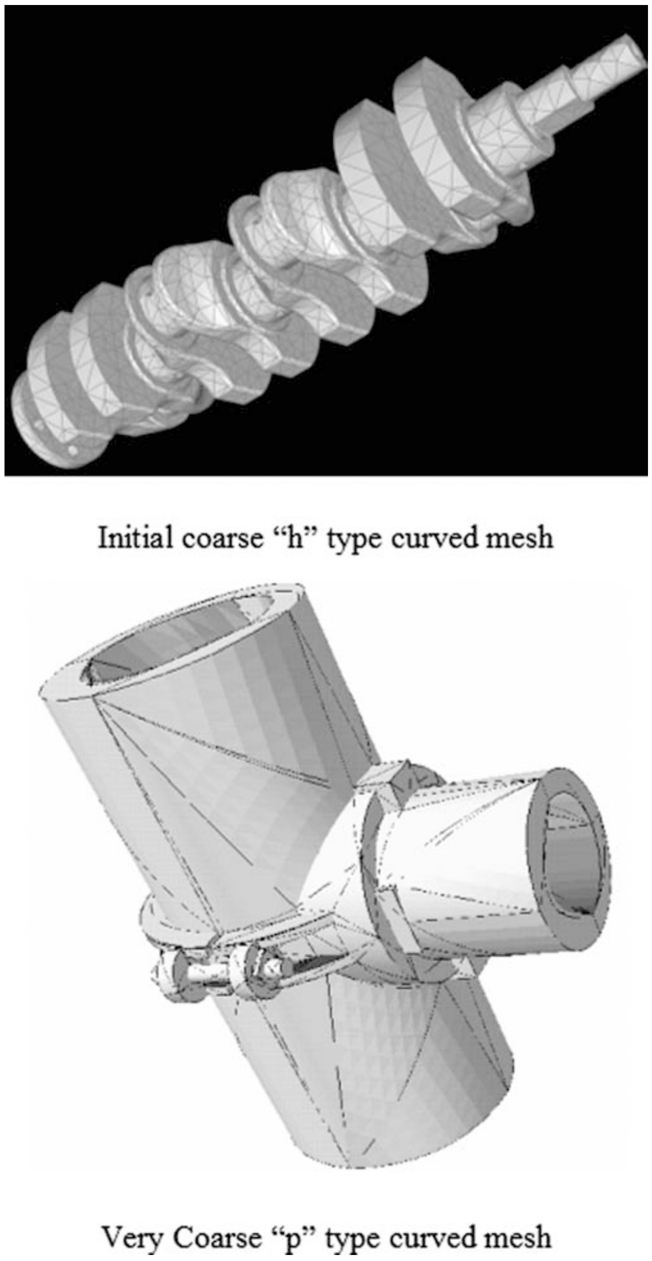

Initial coarse "h" type curved mesh
Very Coarse "p" type curved mesh

Fig. 7 Curved meshing

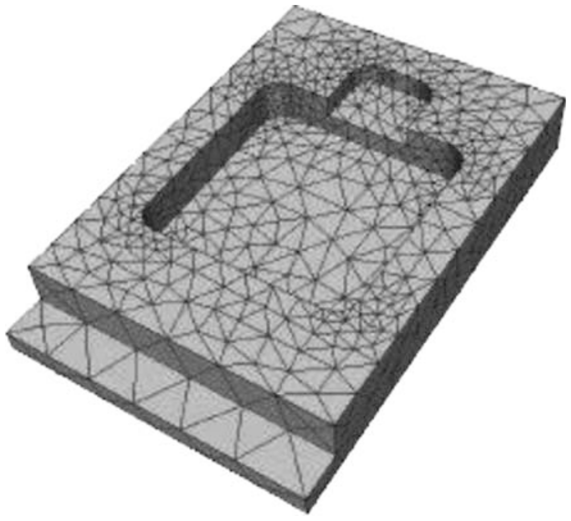


Fig. 8 Curvature-based mesh refinement

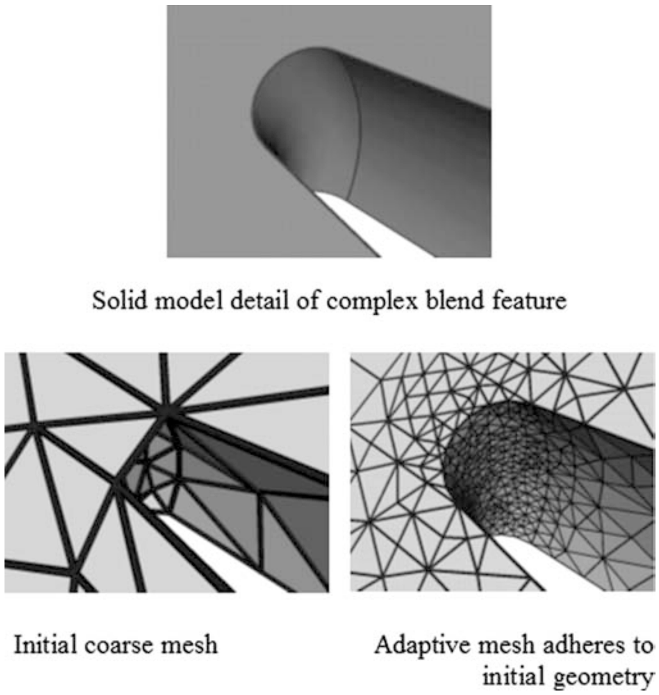

Solid model detail of complex blend feature
Initial coarse mesh
Adaptive mesh adheres to initial geometry

Fig. 9 Geometry-based mesh adaptivity

the original geometry to be present. Mesh adaptivity that does not adhere to the geometry is limited by the initial mesh geometric approximations, and can provide results that are meaningless. For example, Fig. 9 is a close-up of a geometric feature in an accelerator cavity geometry where the simulation procedures must provide highly accurate estimates of the electrical and magnetic losses. The sensitivity of the results to the local geometric shape is so high that, if the mesh geometric approximation did not improve as the adaptive simulation process continued, the results obtained would have been not just a poor approximation, but meaningless.

In many problems of interest, the mesh edges and faces are of the same size as the small geometric features that are often critical to the analysis, such as the accelerator cavity. In these cases, the simple movement of new nodes introduced during refinement to the curved model surfaces can yield invalid elements. The algorithms needed to effectively deal with these situations must include general mesh modification operations and a control algorithm that ensures that the procedure is progressing in a positive manner [11].

The advantages of geometry-based mesh adaptivity include: (1) the ability to start with coarser initial meshes, and (2) the ability to ensure that the resulting model adheres at an appropriate level of accuracy to the design geometry. An additional benefit that may not be apparent is the combination of geometry-based mesh refinement with the small feature defeaturing as a function of target mesh size. This can result in an adaptive geometry representation for mesh adaptivity where small features are ignored in the initial mesh and accounted for as a function of target mesh size in each stage of the mesh adaptivity process. This combined approach dramatically reduces the defeaturing requirements associated with geometry access for mesh generation and allows for initial coarse meshes of detailed geometric models. Figure 10 illustrates an example of this combined approach to adaptive geometry representation.

3.8 Evolving geometry problems

There are a number of situations where the model shape and topology can evolve during the simulation. When the simulation is performed using a Lagrangian-type analysis and there are large deformations and/or model fracturing, it is often necessary to update the domain and mesh several times during the simulation (e.g., in fragmentation simulations [12] or metal forming [13]). In these situations, the model topology and shape must be updated based on the simulation results. Even in the case where the original geometric model was defined in a CAD system, it is most likely not desirable to continue to use the original CAD system to update the CAD model. This is because the new geometric information available from the simulation is limited to node point coordinates on the mesh facets, and most CAD systems do not effectively support such geometry updates.

An important aspect of properly updating the geometric model for these cases is to update the model topology based on the simulation information, and to associate the appropriate collections of mesh edges and mesh faces with the resulting model edges and faces to use in the subsequent definition of shape information. Algorithms to do this based on mesh-based geometry parameters and/or simulation contact or fracture information have been developed [13–15]. Once the model topology has been defined, the geometric shape information can be defined directly in terms of the mesh facets, or can be made higher-order using subdivision surfaces [15, 16] or higher-order triangular patches [17, 18]. Reference [13] provides a description of an automated adaptive medal forming procedure where the updated geometric model is defined based on the simulation information and higher-order updated shapes of the edges and surfaces are defined by subdivision patches applied on a model entity level.

3.9 Integration of simulation in the design process

Integration of simulation in the design process is a driving factor for improved geometry access for mesh generation and support of this integration should be

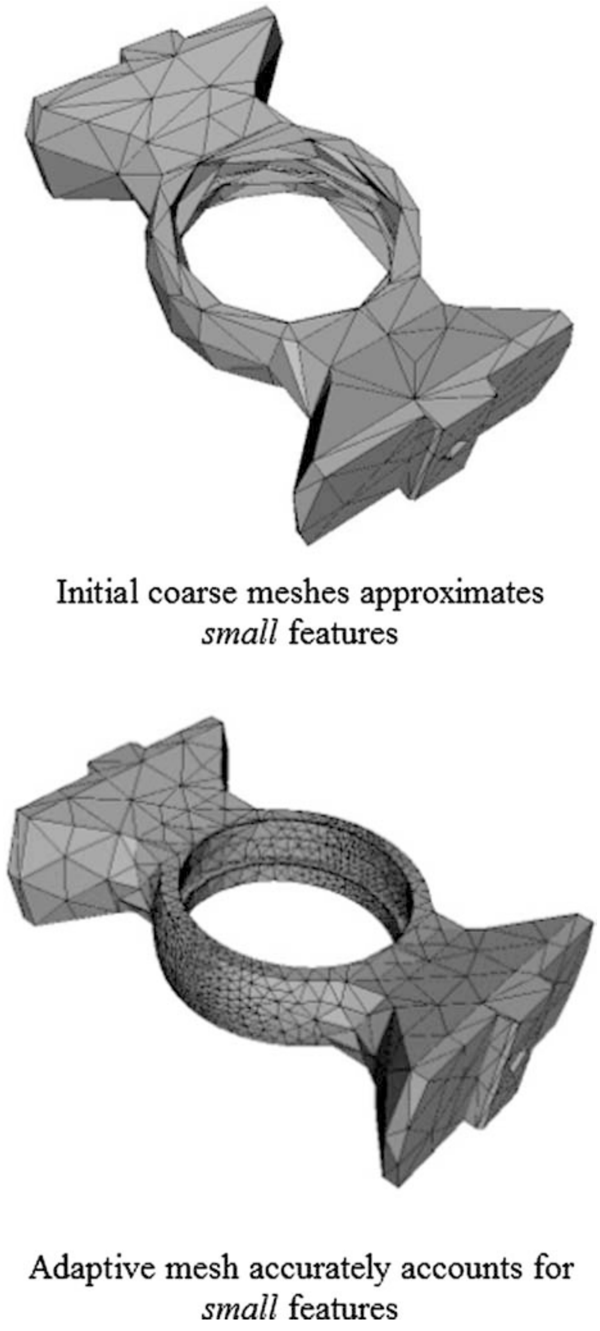

Initial coarse meshes approximates *small* features
Adaptive mesh accurately accounts for *small* features

Fig. 10 Adaptive geometry representation

considered as a major issue when considering geometry access. This integration allows for simulation to be an integral part of the design process and requires the use of the native CAD system geometry as the geometry source to allow for effective reuse through multiple design iterations. Mesh generation needs to access the current design state and evolve with the design [19]. Automatic meshing and geometry-based mesh refinement are fundamental requirements to ensure efficiency and accuracy. Integration of simulation in the design process also requires sophisticated management of simulation attributes to support design change in a manner that avoids the constant redefinition of the attributes.

3.10 Multiple geometry sources

The geometry access issues discussed so far are limited to a single geometry source. These issues are further complicated by the need to support multiple geometry sources. Each geometry source may use different representations for geometry and topology, and different tolerances and methods for evaluating tolerances. Direct interface utilities to multiple CAD systems is both complex and expensive to develop and support. Modeling kernels such as ACIS, Granite, and Parasolid help to reduce the scope of this problem by providing generic API's that provide roughly equivalent capabilities. This makes it possible to develop a uniform procedural approach to integrate geometry-based applications, like mesh generation, with them.

4 Techniques for accessing geometry for mesh generation

There are several techniques currently used and being developed to address the geometry access issues. The techniques used for geometry access can be classified into four major approaches as follows:

– Translation and healing
– Discrete representations
– Direct geometry access
– Unified topology accessing geometry directly

A basic comparison of these alternative approaches is provided by considering the ability of each of the approaches to address:

– Operating with the alternative forms of input geometry
– Application of defeaturing operations
– Propensity to introduce dirty geometry
– Meshing operations needed to properly deal with curved geometry domains
– Evolving geometry simulations
– Integration into the design process
– Integration with multiple CAD geometry model sources

4.1 Translation and healing

Translation and healing has historically been the most commonly used technique for geometry access. The translation may involve the use of standard file formats or direct translators.

IGES does not address issues with representations, global tolerances, features, tolerancing or tolerance methods, and typically results in dirty geometry [4]. Standards such as VDAFS and STEP do address issues with representations and global tolerances, but do not address features, tolerancing, or tolerance methods, and often results in dirty geometry (typically cleaner than IGES).

Many companies have invested millions to resolve the translation-related issues (ITI, Elysium, Spatial, TransMagic, CAD-CAMe, TTI, TTF, etc.). An entire interoperability industry has evolved to attempt to address the issues of translation and healing. Progress has been made but the translation and healing process is still not reliable or robust. Without the addition of a means to obtain the modeling system tolerance and methods, healing processes must employ heuristic algorithms to identify and resolve ambiguities as simple as, "is there a gap here?" and "should a face be introduced to close it?"

Evaluation of the translation and healing technique as related to the geometry access issues presented is as follows:

– Does not address legacy mesh data or scan data as potential sources of geometry
– Defeaturing is difficult since explicit feature information is lost in translation and unwanted implicit features may be created
– Feature-based translators attempt to reproduce models from feature representations, but do not address tolerance methods and may fail to rebuild models, or may introduce slivers and small features
– Healing typically introduces slivers and small features to resolve dirty geometry
– Non-feature-based translators require explicit feature removal
– Feature suppression with non-feature-based translators requires feature recognition algorithms
– Translation and healing introduces dirty geometry due to differences in CAD systems' modeling engines' representations, tolerances, and methods
– The resulting geometry representation, typically, can support curved meshing, curvature-based refinement, and geometry-based mesh adaptivity on modified representation
– It is possible to support adaptive geometry representations on modified a representation with small feature recognition
– The ability to support evolving geometry is limited by the geometry representation available
– The integration of simulation in the design process is not effectively addressed unless the translation process maintains appropriate links to the simulation attributes
– Differences in algorithms and tolerances between modeling engines make it impossible to exactly exchange data between them. Results and robustness vary dramatically with different CAD systems

4.2 Discrete representations

The discrete representations technique is based on the generation of a faceted model and accessing the resulting faceted model for mesh generation. This is most commonly done based on simple facets generated by the CAD system faceter. Recently, a few investigators have introduced some ability to account for the fact that the facets are approximating curved boundaries by the introduction of facet-based subdivision surfaces [15, 16] or higher-order triangular patches [17]. Legacy mesh data and scan data may also be potential sources of facet data.

This technique is taken in an attempt to eliminate dirty geometry and to resolve differences between different CAD systems. A key assumption in doing this is that the CAD system facet generator will create a properly closed set of surface facets. Since many of these procedures were developed to be as fast as possible to create display data, they do not always include all the procedures and checks needed to ensure a properly closed set of facets. Figure 11 shows such an example in which there are both small holes and dangling faces, both of which can cause many mesh generation algorithms to fail.

The successful use of the simple facets in the discrete representations technique is highly dependent on the original facet representation. All discrete representation techniques result in an approximation of the geometry, and do not retain the explicit feature data and geometry of the CAD model.

Evaluation of the discrete representations technique as related to the geometry access issues presented is as follows:

- Does address all potential sources of geometry
- Defeaturing of any type is difficult since all explicit feature information is lost
- Simple facet representations are designed for visualization and may still have some problems with dirty geometry
- Simple facet representations cannot support curved meshing and geometry-based mesh adaptivity

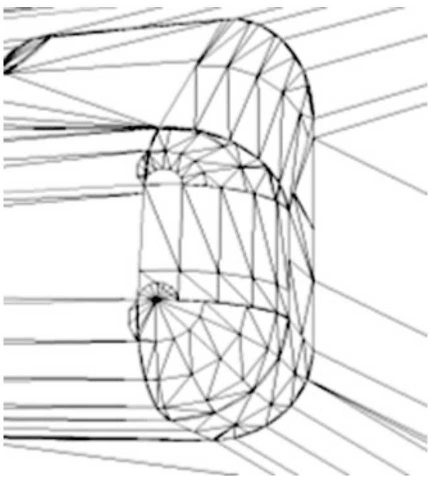


Fig. 11 Facet representations from major CAD system modeling engines may not close

- More sophisticated discrete representations, such as subdivision surfaces and higher-order triangular patches, can support an approximate version curved meshing, curvature-based refinement, and geometry-based mesh adaptivity
- It is difficult to support adaptive geometry representation on modified representations with small feature recognition
- The definition of evolving geometry can be supported.
- The integration of simulation in the design process is not effectively addressed since the linkage to the model entities that the simulation attributes is typically not properly maintained
- Handles data from different systems in a consistent manner but results may vary dramatically due to differences in facet representations

4.3 Direct geometry access

Direct geometry access is a technique based on accessing CAD geometry directly through CAD system toolkits such as CATIA CAA and Pro/Toolkit [20]. Use of the CAD system toolkits requires that a seat of the CAD system is available for geometry access.

Since many CAD systems use geometric modeling kernels, this approach can also be achieved by licensing the same geometric modeling kernel as the CAD system and accessing the geometry through the modeling kernel APIs [2, 21, 22].

The main theme of this approach is to leave the data in the native modeling engine and to use that native modeling engine to access geometry so that the native tolerances and methods are used for geometry access and, wherever possible, the explicit feature data is retained.

Evaluation of the direct geometry access technique as related to the geometry access issues presented is as follows:

- Does not address legacy mesh data or scan data as potential sources of geometry
- Defeaturing is an issue for implicit features that cannot be suppressed
- Small features, slivers, and multiple faces cannot be suppressed
- Native geometry is not dirty
- Can support curved meshing, curvature-based refinement, and geometry based mesh adaptivity
- Adaptive geometry representation with small feature recognition is extremely difficult (if not impossible) to support
- The ability to support evolving geometry is limited by the geometry representation available
- The integration of simulation in the design process can be effectively addressed with unique solutions for each CAD modeling source
- Requires multiple direct interfaces for a broad range of geometry support

- Each CAD system has a different geometry and topology representation to interrogate for meshing
- Each CAD system has different tolerances and methods to understand
- Each CAD system has a different toolkit for accessing geometry and topology data

4.4 Unified topology accessing geometry directly

The final geometry access technique to be considered is the unified topology accessing geometry. This is an extension of the direct geometry access technique, with enhancements to overcome the shortfalls of that technique. This approach is based on an abstraction of the geometry that allows multiple sources of geometry to be treated the same by the mesh generator [2, 21, 22]. This abstraction of the geometry will be referred to as the unified topology model.

The unified topology model is a representation of the model for simulation purposes that retains its connection to the original system geometry and topology. This provides a separate topology data structure that allows for multiple forms of defeaturing, whilst retaining the original geometry and topology.

The geometry is directly accessed from the native modeling system, as per the direct geometry access technique, and a unified topology model is created. This unified topology model accounts for the topology of the original modeling system and enhances this representation to make it more suitable for analysis. These enhancements may include; support for multi-dimensional models, non-manifold model (extremely useful for assemblies), defeaturing of unwanted features, and support for models from multiple geometry sources for a single analysis.

One important aspect of the unified topology model is to maintain a relationship between the unified topology model and the topology of the original model. This may be a one-to-one relationship or a one-to-many relationship. Maintaining these relationships allows the unified topology model to be modified for analysis without affecting the underlying CAD model, whilst still maintaining the direct geometry access for all geometric queries.

One example of a unified topology model is the Simulation Modeling Suite provided by Simmetrix. In the Simulation Modeling Suite, the unified topology model builds on top of the geometry source data to present a standard representation for all modeling sources (non-manifold topology similar to the radial edge data structure [23]). Geometric queries are passed through to the CAD system via direct access to APIs or modeling kernels, or evoked directly on facet data. The resulting unified topology model used is illustrated in Fig. 12.

Evaluation of the unified topology accessing geometry technique as related to the geometry access issues presented is as follows:

- Does address all potential sources of geometry
- Allows for various forms of defeaturing
- Slivers and small features can be addressed as a function of global and local target mesh sizes; Fig. 13 illustrates the effect on meshing results related to defeaturing of the small features illustrated in Fig. 5
- Simple features can be suppressed in the unified topology model for meshing purposes
- Complex features may be addressed either by suppression of explicit features in the CAD system, or as small features in the unified topology model
- Uses native system tolerances and methods

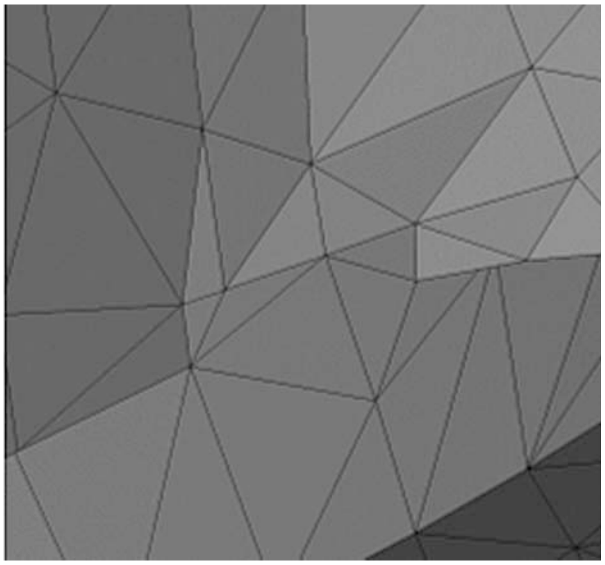
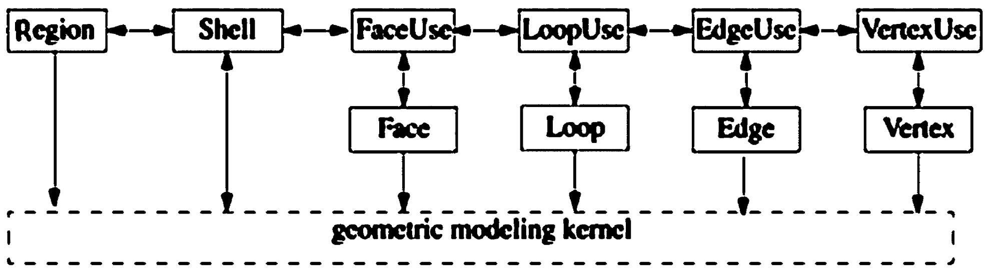


Fig. 13 Small feature removed

Fig. 12 Simulation Modeling Suite unified topology model



– Native geometry is not dirty
– Curved meshing, curvature-based refinement, and geometry-based mesh adaptivity can be supported
– Can support adaptive geometry representation with small feature recognition
– Creation of new topology in the unified topology model based on a discrete geometry basis provides support for evolving geometry problems
– Proven effective to address issues related to integration of simulation in the design process
– Used in large simulation-based design initiatives and commercial CAE Software (Visteon, John Deere, Blue Ridge Numerics, CFD Research Corporation, ESRD, Coventor, PVM Corporation, and many others)
– Provides a single interface for a broad range of geometry support
– Geometry abstraction layer handles all geometry-source-specific issues
– Mesh generation algorithms access a consistent unified topology model

5 Summary

The desire to use simulation as an integral part of the design process has necessitated an evaluation of the issues and techniques associated with geometry access for mesh generation. A broad range of issues were highlighted and four techniques for geometry access were reviewed with respect to these issues.

Translation and healing was the initial technique reviewed and was found to lack the reliability and robustness necessary to support design/analysis integration. The translation and healing technique does not address several of the geometry access issues outlined.

The second technique reviewed was discrete geometry representations. This technique does address some of the geometry access robustness issues but does not address well those issues related to feature representations, curvature-based meshing, and design integration.

The third technique reviewed was direct geometry access. This technique does address many of the geometry access issues but does not address well those issues related to defeaturing of implicit features and multiple CAD systems or access to facet-based data.

The final technique reviewed was unified topology accessing geometry directly. This technique provides an effective means to address to the geometry access issues outlined. The unified topology model accessing geometry directly technique is the most flexible technique for addressing issues related to accessing geometry for mesh generation.

The unified topology accessing geometry directly technique can provide a single environment to effectively deal with integration to geometry from multiple sources and defeaturing of implicit features, providing a firm foundation for design/analysis integration.

References

1. Braid I (1991) A history of geometric modeling. Spatial Tech-Ex, pp 1.1–1.17
2. Shephard MS, Georges MK (1992) Reliability of automatic 3-D mesh generation. Comput Methods Appl Mech Eng 101:443–462
3. Walsh JL (1993) Exposing the myths of design to analysis data exchange. In: Proceedings of the ABAQUS user's conference, Aachen, Germany, June 1993, pp 659–672
4. Butlin G, Stops C (1996) CAD data repair. In: Proceedings of the 5th international meshing roundtable, Pittsburgh, Pennsylvania, October 1996, pp 7–12
5. Shephard MS, Beall MW, O'Bara RM (1998) Revisiting the elimination of the adverse effects of small model features in automatically generated meshes. In: Proceedings of the 7th international meshing roundtable, Dearborn, Michigan, October 1998. Sandia National Laboratories report SAND 98-2250, pp 119–131
6. Dey S, Shephard MS, Georges MK (1997) Elimination of the adverse effects of small model features by local modifications of automatically generated meshes. Eng Comput 13(3):134–152
7. Mezentsev AA, Woehler T (1999) Methods and algorithms of automated CAD repair for incremental surface meshing. In: Proceedings of the 8th international meshing roundtable, South Lake Tahoe, California, October 1999. Sandia National Laboratories report SAND 99-2288, pp 299–309
8. Ribo R, Bugeda G, Onate E (2002) Some algorithms to correct a geometry in order to create a finite element mesh. Comput Structures 80:1399–1408
9. Luo X, Shephard MS, Remacle J-F, O'Bara RM, Beall MW, Szabó BA, Actis R (2002) p-version mesh generation issues. In: Proceeding of the 11th international meshing roundtable, Ithaca, New York, September 2002. Sandia National Laboratories, pp 343–354
10. Dey S, O'Bara RM, Shephard MS (2001) Curvilinear mesh generation in 3D. Comput Aided Des 33:199–209
11. Li X, Shephard MS, Beall MW (2003) Accounting for curved domains in mesh adaptation. Int J Numerical Methods Eng 58(2):247–276
12. Pandofi A, Ortiz M (2002) An efficient procedure for fragmentation simulations. Eng Comput 18(2):148–159
13. Wan J, Kocak S, Shephard MS, Mika D (2003) Automated adaptive forming simulations. In: Proceedings of the 12th international meshing roundtable, Santa Fe, New Mexico, September 2003
14. Krysl P, Ortiz M (2001) Extraction of boundary representation from surface triangulations. Int J Numerical Methods Eng 50:1737–1758
15. Lee CK (2003) Automatic metric 3-D surface mesh generation using subdivision surface geometry model. Part 1: construction of underlying geometric model. Int J Numerical Methods Eng 56:1593–1614
16. Cirak F, Ortiz M, Schroder (2000) Subdivision surfaces: a new paradigm for thin shell finite-element analysis. Int J Numerical Methods Eng 47:2039–2072
17. Owen SJ, White DR (2001) Mesh-based geometry: a systematic approach to constructing geometry from a finite element mesh. In: Proceedings of the 10th international meshing roundtable, Newport Beach, California, October 2001. Sandia National Laboratories report SAND 2001-2967C, pp 83–96
18. Owen SJ, White DR, Tautges TJ (2002) Facet-based surfaces for 3-D mesh generation. In: Proceedings of the 11th international meshing roundtable, Ithaca, New York, September 2002, pp 297–311
19. Walsh JL (1991) Geometrically associative analysis modeling. Spat Tech-Ex, pp 9.1–9.16
20. Merazzi S, Gerteisen EA, Mezentsev A (2000) A generic CAD–mesh Interface. In: Proceedings of the 9th international meshing roundtable, New Orleans, Louisiana, October 2000. Sandia National Laboratories report SAND 2000-2207, pp 361–369

21. Tautges TJ (2000) The common geometry module (CGM): a generic, extensible geometry interface. In: Proceedings of the 9th international meshing roundtable, New Orleans, Louisiana, October 2000. Sandia National Laboratories report SAND 2000-2207, pp 337–359
22. Shephard MS (2000) Meshing environment for geometry-based analysis. Int J Numerical Methods Eng 47(1–3):169–190
23. Weiler KJ (1988) The radial edge structure: a topological representation for non-manifold geometric boundary representations. In: Wozny MJ, McLaughlin HW, Encarnacao JL (eds) Geometric modeling for CAD applications. North Holland, Amsterdam, The Netherlands, pp 3–36