LA-UR-00-2956
June 2000

Department of Energy
Environmental Cleanup
Program

Interfacing Mesh Generators and Geometric Modelers in Geological Modeling - Progress Report for LANL/ER Program

Los Alamos

National Laboratory
Los Alamos, NM 87545

Los Alamos National Laboratory, an affirmative action equal opportunity employer is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

Produced by Porous Flow Team, EES-5

Authors: Rao V. Garimella, Carl W. Gable

Contact: rao@lanl.gov, gable@lanl.gov

Contents

1 Executive Summary

The "Geological Modeling with CAD" project was started earlier this year with the help of a grant from the Environmental Restoration project. The goal of of this project is to demonstrate the possibility of integrating Geographic Information Systems (GIS), geometric modeling (or CAD) and automatic mesh generation procedures in porous flow simulations for the ER effort of LANL and also general geological modeling and simulations.

The work-flow of the project is to obtain discrete data from GIS teams, automatically construct a geometric model from this data and mesh the geology using an automatic mesh generator.

It was decided that the most convenient and flexible way of obtaining data from the GIS work would be in the form of "triangulated" or faceted surfaces. In this form, surfaces of the geology are described in the form of an irregular network of triangles, rectangles or a combination of the two. The nodes of the triangles or rectangles correspond to actual points on the surface measured in the field. In forming this faceted representation of the surface, there may be work-flowmultiple choices for connections between nodes, some of which provide a better approximation to the true surface than the others. The decisions regarding the best choice of connections to represent the surface as accurately as possible has been left to the GIS team.

The choice of a geometric modeler is crucial to the success of this project. The geometric modeler used must have the following capabilities:

1. Capability to handle non-manifold models (general combinations of solids, surfaces and curves).
2. Capability to handle faceted/triangulated surfaces.
3. Robust programming interface to the geometric modeling kernel
4. Reliability of software

A number of geometric modelers were reviewed and finally, the Shapes(TM) kernel from XOX Corp. was chosen as it met all the requirements of the project. This software has been obtained and preliminary work has begun in testing its capabilities and reliability. In the next few months the software capabilities to take information in a few forms from GIS teams and automatically convert it to a geometric model in Shapes will be developed.

In the next month, work is expected to start simultaneously on incorporating in-house and external mesh generation components into the . This will provide the framework for taking simple geological models through all the steps of the proposed process. When a certain degree of robustness has been established in the different components with simple models, more complex geological models will be modeled and meshed.

2 Introduction

Mesh generation is the process of deriving an approximation of a geometric domain by the union of a number of smaller geometric domains that assume a fixed number of simple topological configurations but may have flexible geometrical definitions.

The topology of these simpler domains (called *elements* of the mesh) are typically simply connected, 2-manifold polygons and polyhedra, the common ones being triangles, quadrilaterals, tetrahedra, triangular prisms and hexahedra, although it is possible to consider others (Voronoi polyhedra). Of these, the simplices, namely triangles and tetrahedra are the most popular, since they are more easily tackled by the theorems of computational geometry.

The geometry of these elements is flexible with the most common type being linear or bilinear, i.e., the edges of the elements are straight lines. Some uses of meshes such as finite element methods for structural mechanics, fluid mechanics and acoustics do require curved element geometry. The curving of elements may be according to a polynomial or mapped to the true geometry of the domain being discretized.

The two most important uses of meshes are for graphics visualization and for numerical analysis of the physics of complex domains. Of these, the former application is more relaxed in its requirements of the mesh than the latter. Many numerical methods for instance require the mesh elements to not intersect each other except at their boundaries whereas graphics methods often do not care.

The number of elements of the mesh, or the *refinement* of a mesh as it is called, depends primarily on how well the original geometric domain and any fields associated with the domain must be approximated. These two criteria can be met by making the elements smaller and by improving the approximation of of the true boundary by making element edges and faces curved - often a combination of the two may be used (referred to in numerical methods as h- and p-refinement respectively). If the geometry of the elements is constrained to be of a certain type (say linear or quadratic), then the only recourse for improving the approximation to the true geometric domain is to refine the elements making them smaller and increasing the number of elements in the mesh.

In this article, the representation of meshes, models and the interaction between mesh generators and geometric modelers is discussed with special emphasis on geological modeling.

Geological models raise a unique set of issues in geometric modeling and mesh generation. Typically geological models have to be constructed from data provided by Geographical Information Systems (GIS). This data is naturally discrete data which must be used to approximate the shape of the geological surfaces. Therefore, the most common method of representing such surfaces is by a triangulation of the discrete points sampled on the surfaces. This is also the most valid representation, since any interpolation between the data points, say by fitting a B-spline surface to the data points, makes unjustified assumptions about the behavior of the surface. In fact, in highly constrained portions of the model, the B-spline surfaces derived from two non-intersecting triangulated surfaces might be intersecting.

Since the data from GIS systems is often just an ordered set of surfaces and some additional data, it has to be processed further to incorporate it into a geometric modeler. The surfaces have to be trimmed in the lateral direction by a bounding box or some other shape. Also artificial surfaces, such as wells, faults, fractures and tunnels, may have to be introduced into the model, This involves interaction (intersection) of triangulated and analytical surfaces. These individual surfaces must then be sewn together to form the shells of material regions. These operations are not commonly available for faceted surfaces in geometric modelers which have traditionally focussed on modeling structural components.

Geological models are almost always non-manifold[1] due to the presence of multiple material regions. Sometimes other types of non-manifold topology may also be present in the model, e.g. a fault surface. Therefore, the geometric modeler must be capable of representing such models and the mesh generator capable of meshing them.

Therefore, the task is to identify a suitable geometric modeler that has the capability of supporting the construction of geological geometric models and the use of advanced mesh generation techniques for them. To this end, the rest of this article discusses the

- various representations for geometric models and meshes,
- the relationship between meshes and models,
- the proposed formats for the mesh and model representation,
- the interaction required between a mesh generator and a geometric modeler, and
- a review of available geometric modelers and their suitability for geological modeling.

3 Representations for geometric models

Two common forms of geometric modeling are Constructive Solid Geometry (CSG) and Boundary-representation (B-rep). In CSG, a library of primitive solids, such as cubes, spheres, cones, cylinders, tori, prisms, etc. form the basis for constructing more complex shapes. This construction is performed through point set operations, namely, union, subtraction and intersection. For example, a plate with a hole may be constructed by subtracting a cylinder from a rectangular prism. CSG modelers are quite sophisticated and construction of

[1] i.e. having general combinations of solids, surfaces and curves

CSG models is intuitive. However, since CSG modelers do not explicitly represent the boundary, geometric inquiries about boundary information is computationally expensive. Another disadvantage of CSG modeling is that it can create non-manifold situations, but it cannot detect them easily. Finally, the capability of the system is restricted to boolean operations with a predetermined set of primitives.

B-Rep modeling involves the description of a solid model by explicit description of its boundary (refer Figure 1). In any model there may be one or more solids called *bodies*. Each body is made up of a number of regions which are continuous portions of the 3D space (one can go from any point in a region to any other point without crossing the boundary of the region. Each region is made up of one or more *shells*. Shells (and therefore, boundaries of bodies and regions) are made up of one or more model *faces* which are portions of surfaces. Each model face has one or more *loops* which have the same relationship to faces as shells to regions. Loops (and therefore, the boundary of model faces) are made up of *edges* which are parts of curves. Boundaries of edges are in turn described by *vertices* which are associated with a point in space.

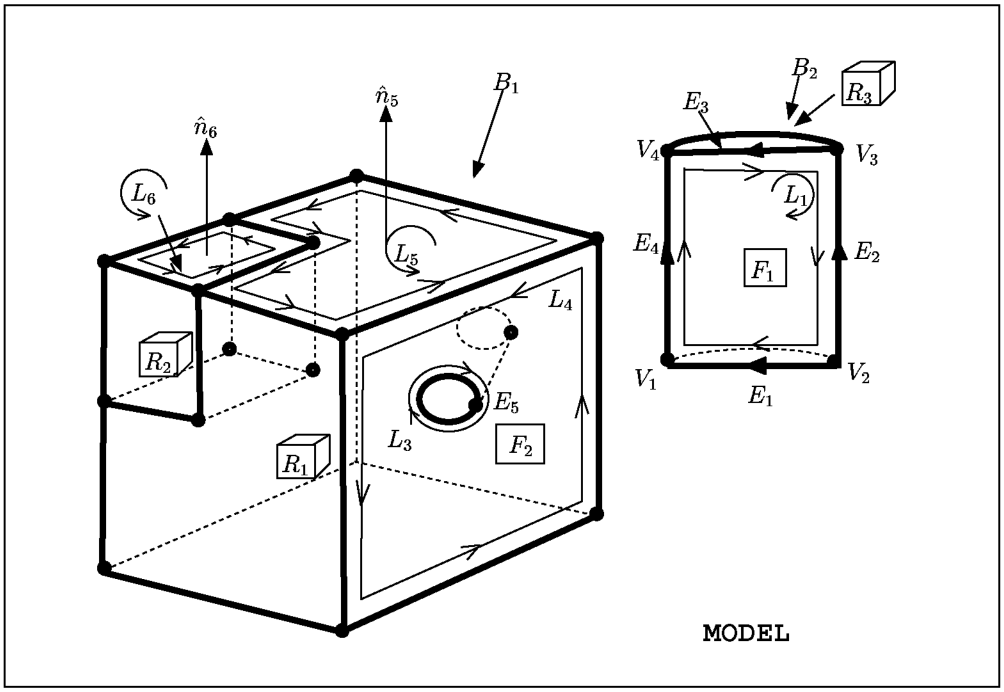

Figure 1: Topological entities in a B-rep model - shells not shown for clarity

Bodies, regions, shells, faces, loops, edges and vertices are called the topology of the model since they represent a general framework for a class of models without any reference to shape information. The various models in the class differ in the geometry underlying the faces, edges and vertices, i.e., the geometric surfaces, curves and points. Thus the topology of a cylinder and a section of a torus are the same but the geometry of the curved surface is different (See Figure 3). B-rep models posses the advantage that the boundary is explicitly represented which does not require repeated evaluation. Also, with the right data structure, B-rep models can unambiguously represent all types of non-manifold models.

The geometry of the models that can be constructed by a B-rep modeler does not depend on the set of parameterized primitives available; rather, it is determined by the various types of geometric surfaces and curves that can be created in the modeler and whether they can be combined to form a valid solid. Thus, one can envision a model in which some surfaces of the model are tessellated surfaces, others planes and still others B-surfaces. Almost all good B-rep modelers (and possibly CSG and surface modelers) can map curves

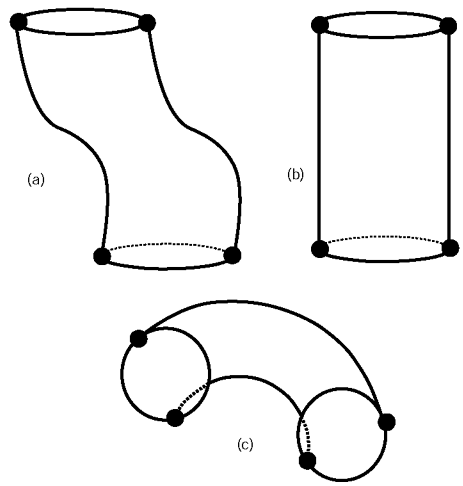

Figure 2: Equivalent topology of warped cylinder, regular cylinder and sector of a torus

and surfaces to one- and two-dimensional *parametric spaces.* While the mapping from parametric to real spaces is always unique, the reverse may not be true. In closed curves and surfaces, like circles, cylinders, spheres, tori and cones a single point in real space can map to two points in parametric space. Such spaces are called periodic. In the case of a the tip of a cone or poles of a sphere, a single point in real space maps to an infinite set of points along a line segment in parametric spaces. Such parameterizations are said to have degeneracies. In Figure 3, a 3D curve and a 3D surface are shown along with their 1D and 2D parametric spaces respectively. Also shown in each case is a point in the parametric space along with its corresponding location in the real space. Note that the parametric space is not constrained to go from 0 to 1 but can take on any range.

Finally, operations not possible with CSG modelers can be performed in B-rep modelers, e.g., sweeping a cross-section along an arbitrary curve to create a solid and uniting two regions while maintaining the common interface. The distinction between topology and geometry is also useful in simplifying mesh generation procedures that can perform many checks more efficiently than in CSG modelers. However, since construction of complex models by boolean operations on simpler parts is intuitive, many B-rep modelers allow CSG-like operations. The output of a CSG-like operation in a B-rep modeler is directly computed as a new boundary representation.

For the purpose of geological modeling, B-rep modelers are particularly useful since geological surfaces are seldom available in analytical form. They are necessarily constructed from discrete data and therefore, the common method of representing them is through a set of concatenated planar facets (either triangles or quadrilaterals). Also, such models have to be created by sewing together various surfaces, a natural functionality of B-rep modelers. This along with boolean operations to introduce artificial structures (for example, a well) or other natural formations such as faults permits the constructions of arbitrarily complex non-manifold representations of geological data.

4 Mesh Representations:

A mesh is inherently a non-manifold model since elements of a mesh share common faces, edges and vertices. Each element of a mesh, however, is much simpler than a full geometric model since the element is simply connected, 2-manifold and the geometry of its boundary entities are typically described by a finite class of geometries.

In cases where the mesh is described only by linear elements (i.e. all edges are straight lines and all

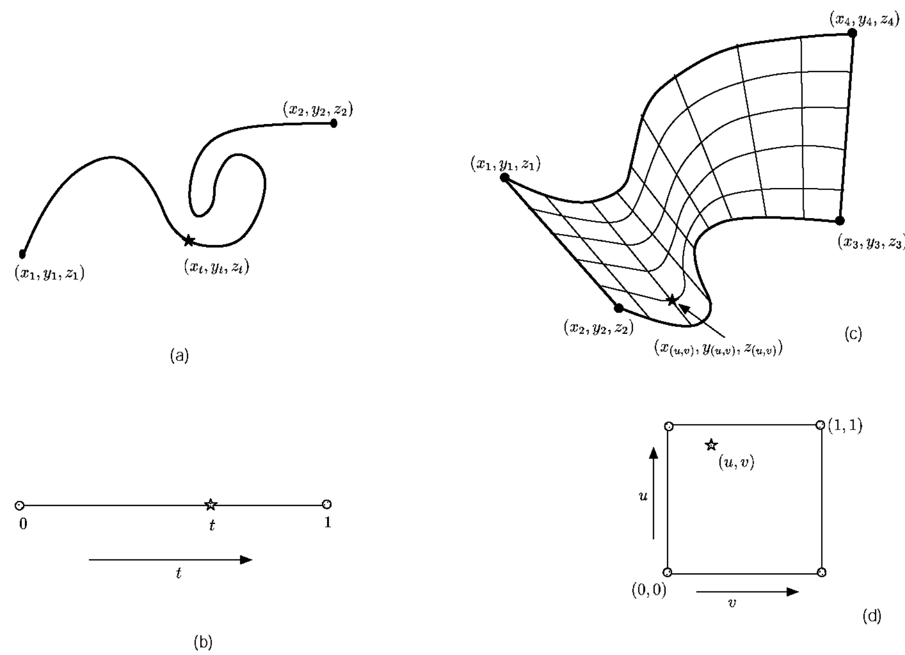

Figure 3: (a) 3D curve (b) Parametric space of curve (c) 3D surface (d) Parametric space of surface

faces are linear or bi-linear) a common approach has been to represent mesh elements by their nodes in a specific order. This suffices for meshes of a large class of simple 2-manifold models. When the models are more complex, non-manifold domains, additional information, is often convenient and even required. For example, it is often necessary to store information regarding the elements adjacent to the implicitly defined faces of an element. This information is important in identify boundary facets in a mesh. Also, the standard element-node data structure is inadequate for providing unique answers to some queries. As an illustration, consider the connection between two nodes (i.e. an edge) known to be on the boundary of a model. The information about the nodes is insufficient to indicate if the connection between them is a boundary connection or an interior connection. To determine this, one has to identify the mesh elements connected to the edge (by identifying the common elements in the sets of elements connected to each node), find facets of the elements connected to the edge, and then check if any of those facets lie on a model boundary. Clearly, this is a cumbersome operation and is impossible if the additional information cited above (elements connected to mesh faces) is not furnished.

Therefore, it is advantageous to have a more descriptive representation for the mesh similar to the Boundary-representation used for geometric models, i.e., the mesh may be formed of mesh regions, mesh faces, mesh edges and mesh vertices. Mesh regions are bounded by mesh faces, mesh faces by mesh edges and mesh edges by mesh vertices. The geometry underlying mesh faces and mesh edges may be linear or curvilinear (described either by additional points or by more general means). The representation does not have to be as complex as that required for geometric modeling though. For example, it is known that each element by itself is 2-manifold, all mesh entities are simply connected, mesh entities cannot have special cases as in geometric modeling (for example, the vertices of an edge, the edges of a face and the faces of a region have to be distinct), the downward connections of the element graph are of fixed complexity for a given element type. This limits the complexity of a boundary representation of a mesh while giving it the requisite richness of information.

The other, more involved issue of mesh representations is that of connectivity between mesh entities.

Recall that in the simple element-node mesh representation discussed above, the only connectivity (also called *adjacency*) information present is from elements to nodes, namely the downward adjacency. If the list of elements connected to a specific node, namely upward adjacency information, is needed then cumbersome and expensive searches must be conducted or additional information must be stored. Downward adjacency information has small fixed costs associated with it for structured and unstructured meshes but the cost of storing upward adjacency information is large and variable for unstructured meshes. If memory usage is not an issue, the simplest solution would be to store a full mesh representation with complete upward and downward adjacencies. The issue, however, is to devise a structure that balances searching costs with storage costs.

For now, let us assume that we use a full data structure with full downward adjacency information and partial upward adjacency information (e.g. mesh vertex to mesh region information).

5 Classification:

Definition 5.1 *A mesh entity is said to be classified on a model entity if it forms all or part of the discretization of the model entity but not its boundary.*

This unique relationship between a mesh entity and a model entity is called classification. A mesh entity of a particular order (3 - regions, 2 - faces, 1 - edges and 0 - vertices) can be classified only on a model entity of an equal or higher order.

Classification is an important concept since it allow procedures to distinguish between various boundary mesh entities and between boundary entities and interior ones. This allows for simplified and robust checks for validity of meshes and operations on them. For example, since the interior of each model face is 2-manifold, any mesh edge classified on the model face has to be connected to 2 and only 2 mesh faces classified on that model face. Also, if a mesh edge is being collapsed in a mesh, classification and mesh topology alone can be used to unambiguously determine if the operation will cause dimensional reduction in the mesh, i.e., the dimensionality of the local mesh is less than the corresponding dimensionality in the model. Say the mesh edge is classified on a model region and its two vertices are classified on different model faces - the collapse operation will definitely cause a zero thickness region in the mesh while this condition does not exist in the model (illustrated in 2D in Figure 5). While this is only a necessary but not sufficient condition for the collapse to be invalid, it is clear that a simple topological check can replace an otherwise expensive and unreliable geometric calculation.

Also, since mesh generation, mesh modification and numerical analysis pre-processing requires different courses of action for various boundary mesh entities and interior mesh entities, classification is a significant aid in such procedures.

6 Need for abstracting geometric models and the interface to geometric modelers:

Of the various B-rep data structures in use the radial edge data structure [1] is one of the most general and flexible data structures capable of representing all non-manifold models. The radial edge data structure introduces the concept of 'uses' in a model to deal with the complexities of a non-manifold model. In this representation, each model face has two face uses. Each of these two face uses represents one side of the face. Similarly, each model edge has edge uses associated with it. The number of edge uses associated with an edge is equal to the number of face uses connected to it or in other words twice the number of faces connected to the edge. Finally, each model vertex has vertex uses connected to it. The number of uses of a vertex depends on the number of edges connected to the vertex and the number of edge uses of each edge. With the help of use data structures, it is possible to unambiguously characterize all non-manifold situations.

The Radial Edge Data structure is unnecessarily verbose. Therefore a reduced data structure called the minimal use data structure [2] has been devised to effectively represent non-manifold models with lesser

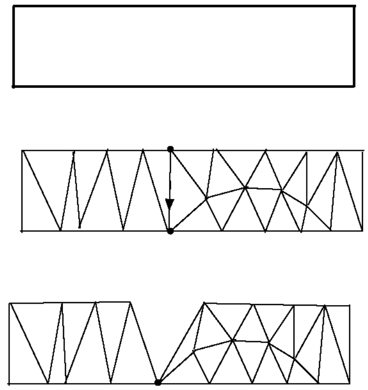

Figure 4: (a) Geometric model (b) Mesh of geometric model (c) Dimensional reduction in mesh caused by improper edge collapse.

information. In this data structure, redundant edge uses and vertex uses are merged to form a more concise representation.

Different geometric modelers available in the industry, however, use a wide variety of data structures to represent their models. Some of them are not even non-manifold although they contain all the functionality necessary to build a non-manifold representation given a set of 2-manifold components and some additional information about their interaction[3]. Finally, each geometric modeler has a unique programming interface in the absence of an established or de facto standard. When developing an application making use of a geometric modeler's capability, it is therefore very important to insulate the application from the specific features of a particular geometric modeler. If this is not done, the application would have to be adapted to every geometric modeler used. Not only that, any time a change is effected in the interaction of the application with the geometric modeler, every version of the code would have to be modified. Rather it is much more efficient to build a generic geometric model and modeler interface. Any application needing the modeler functionality only interacts with this generic geometric modeler and never sees the specifics of any geometric modeling software. The functionality of the generic geometric modeler is then implemented using the specifics of the particular geometric modeling software used. When the application is modified, the geometric modeler interface is not affected. When the geometric modeler interface is modified, the application has to be changed only once. If a new geometric modeler is to be used, only the generic modeler must be adapted to the specifics of the new software and none of the applications built atop the generic interface are affected. Finally, one can use a 2-manifold modeler to build a full non-manifold representation inside of this generic geometric modeler and the application never knows that it is dealing with a 2-manifold modeler [3].

7 Required interaction of mesh generators and geometric modelers:

In order to generate an approximate representation of a geometric domain, an application such as a mesh generator must be able to query a geometric modeler for details of both the topology and geometry of a model. Queries about the topology of the model ensure that the mesh along with the classification information of the entities faithfully represents the topology the model. For example, if a model edge has 3 model faces connected to it, then a mesh edge classified on the model edge must have three mesh faces classified on these 3 model faces in addition to ones classified on the model region. Thus, topological inquiries to the model serve to provide information that allows the mesh generator to make a mesh compatible with the model.

Geometric inquiries are necessary to determine the local shape of the model entities and make decisions about the placement of mesh vertices based on this information. These decisions which rely on intersection queries, distance queries, etc, influence factors such as the quality of elements, the approximation of the mesh to the true shape of the model entities, the necessity for refinement of local regions, the shape of curved mesh entities, etc. These queries also serve as the basis for certain heuristics to make meshes with particular characteristics. For example, in generating meshes for viscous flow simulations, it is necessary to generate elements elongated along model faces and very short perpendicular to the model faces. Therefore, it is necessary in this case to query the modeler for normal vectors at many points on model faces.

The queries to a geometric modeler required for a modeling and analysis application that does not modify the geometric model (such as a mesh generator or analysis pre-processor) are listed below.

- Topological Queries
  1. List of Bodies, Regions, Shells, Faces, Loops, Edges or Vertices in the Model (Variations of this and other similar queries might be to ask for the number of bodies in the model and to ask the bodies one by one)
  2. List of Regions, Shells, Faces, Loops, Edges or Vertices in a Body
  3. List of Shells, Faces, Loops, Edges or Vertices in a Region
  4. List of Faces, Loops, Edges or Vertices in a Shell
  5. List of Loops, Edges or Vertices in a Face
  6. List of Edges or Vertices in a Loop
  7. List of Vertices of an Edge
  8. List of Edges, Loops, Faces, Shells, Regions or Bodies connected to a Vertex
  9. List of Loops, Faces, Shells, Regions or Bodies connected to an Edge
  10. Face of a Loop
  11. List of Shells, Regions or Bodies connected to a Face
  12. Region of a Shell
  13. Body containing a region
  14. Region on the same/opposite side of a face as its normal
  15. Direction of use of a face by a shell or a region
  16. Direction of use of an edge by a loop or an edge
  17. Check for topological containment of lower order entity on the boundary of higher order entity
  18. List of higher order entities common to a set of entities
  19. List of lower order entities common to a set of entities
- Geometric Queries
  1. Geometric tolerance of entity or model
  2. Coordinates of Vertex
  3. Curve of Edge
  4. Sense in which Edge uses Curve
  5. Surface of Face
  6. Sense in which Face uses Surface
  7. Parametric bounds of Curve or Edge (One pair of double precision numbers)
  8. Parametric bounds of Surface or Face (Two pairs of double precision numbers)
  9. Nature of parametric space of geometry (Continuous, Differentiable, Periodic, Degenerate)

10. Nature of parametric space within bounds of Edge or Face (Continuous, Differentiable, Periodic, Degenerate)
11. Point on curve corresponding to given parameter
12. Parametric value of given point on curve
13. Point on surface corresponding to given parameters
14. Parametric values corresponding to given point on surface
15. Curve parameter corresponding to vertex point
16. Surface parameters corresponding to vertex point
17. Surface parameters corresponding to curve parameter and vice versa
18. Normal to a surface at given parametric location
19. Curvature of surface at given parametric location
20. Tangent to curve at given parametric location
21. Closest point to Face from given point
22. Closest point to Edge from given point
23. Intersection of a line segment and a surface
24. Intersection of a plane and curve
25. Containment of point in Edge, Face, Region or Body
26. Bounding Box of Edge, Face, Region, Body or Model

8 Proposed representation of geometric model:

The proposed representation of the geometric modeler is a non-manifold representation **without** explicitly representing use information. This will allow for all classes of non-manifold models to be represented but if use information is required it has to be constructed on the fly or if this cannot be done some geometric computations may have to be used to answer a query. The representation described here closely matches the general description of a non-manifold B-rep model earlier (also see Figure 5)

The geometric model is made up of model bodies, regions, shells, faces, loops, edges, vertices. A model is a collection of bodies (B_1 and B_2 in Figure 5 which in turn may be made up of one or more regions or faces (B_1 is made up of regions R_1 and R_2, and B_2 is made up of regions R_3). Each model region is made up of one or more shells. Each region has one and only one outer shell but it can have multiple inner shells which enclose holes. In Figure 5, R_1 is made up of one outer shell, R_2 is made up of one outer shell and one inner shell (enclosing the cylindrical hole), and R_3 is made up of one shell. Shells are made up of face uses or in other words faces used in a particular orientation. A shell may use one or more of its faces in both directions if necessary. This permits the existence of faces partly protruding into a model region or fin faces outside a model region. It also allows for a completely disconnected face inside a model region or outside all model regions. Thus a shell may not enclose a region at all, although it is made up of faces. These features contribute to the non-manifold capabilities of this representation.

Model faces are made up of one or more loops (e.g., in Figure 5 F_1 is made up of only one loop L_1 but F_2 is made up of loops L_3 and L_4). Each model face must have an outer loop, like L_4, and can have multiple inner loops enclosing holes, like L_3. Loops are made up of edge uses or directed edges. Like shells, loops may use edges in both directions but this is a possibility even in 2-manifold models and is not unique to non-manifold representations. In a fully non-manifold representation, the analogy between shells and loops may be carried to completion, and loops may be allow to be made up of completely disconnected edges on a model face, in a model region or outside a model region. Only the first of these, i.e., the existence of an edge or edges forming a loop by themselves inside a model face but not enclosing a hole will be allowed since it is the most likely to occur. Of course, the case of loops using edges both ways in the outer loop or inner loops enclosing holes is still permitted. Also, note that a loop may use some edges of a face in one way and

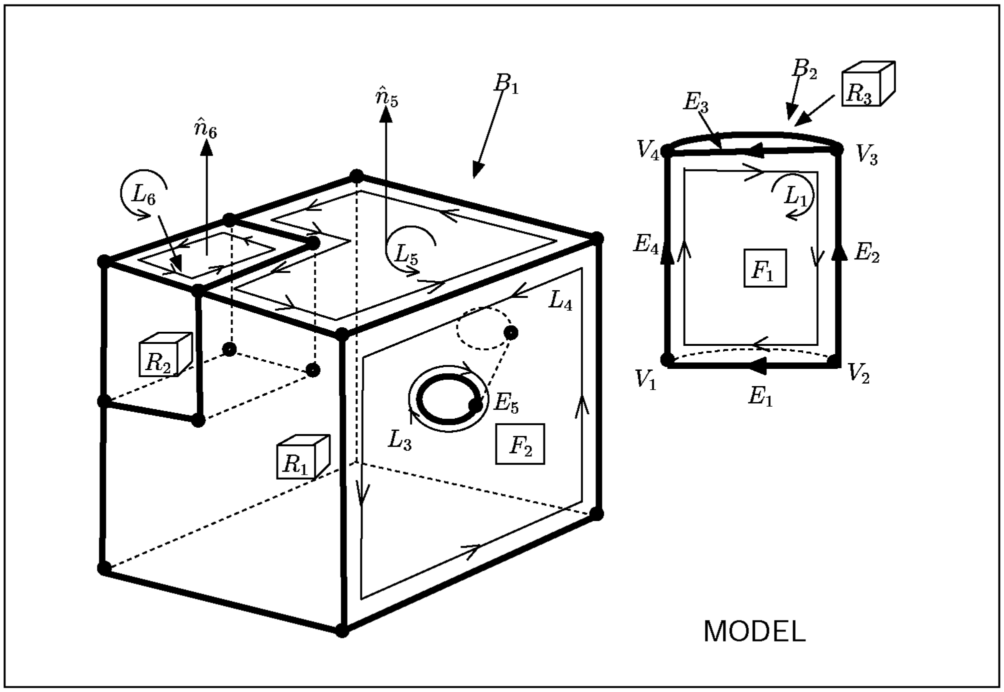

Figure 5: Topological entities in a B-rep model - shells not shown for clarity

other in another, as long the direction of the loop is consistent. Typically, the direction of a loop is chosen such that the inside of the model face is always to the left as one travels along the loop.

Finally edges are made up of vertices. In the geometric model it is possible for an edge to have two, one or no vertices. Edges with one or no vertices are called closed edges and require special consideration (e.g., Edge E_5 in Figure 5 has only one vertex). Model faces containing closed edges may be closed, e.g., cylinder, torus, cone, sphere. Again these faces require special consideration by applications.

The connectivity or adjacency matrix for the geometric modeler representations is shown in Figure 6. The following assumptions are made about the model in deriving this matrix:

1. Degenerate points on surfaces have to be represented by a vertex. For example, a sphere will always have a vertex at each pole and a cone will have a vertex at the tip.

2. Fin faces (not connected to any region), wire edges (not connected to any face) and hanging vertices (not connected to any edges) are not permitted. Fin faces may be permitted in the future, not because they will find use in geological modeling but because they find ample use in many other engineering fields. This will require maintaining direct connectivity between bodies and at least some shells.

3. Closed entities have at least one lower order entity on the boundary. In particular, periodic edges have at least one vertex and periodic faces have at least one edge along each direction of periodicity. This precludes the existence of edges without vertices and faces without any edges.

The first items ensures that faces always have at least one loop and the second item every model body has at least one shell (even if it is made up of only faces). This allows the model representation to maintain only one level adjacencies. The last assumption makes it simpler to deal with parametric spaces of periodic faces and edges.

Entity	Body	Region	Shell	Face	Loop	Edge	Vertex
Body	–	n	n	n	n	n	n
Region	1	–	n	n	n	n	n
Shell	1	0,1	–	n	n	n	n
Face	1	0,1,2	1,2	–	n	n	n
Loop	1	0,1,2	1,2	1	–	n	n
Edge	1	n	n	n	n	–	n
Vertex	1	n	n	n	n	n	–

Figure 6: Adjacency (Connectivity) matrix for topological entities in geometric model ($n > 0$).

The geometry of faces is defined by trimmed surfaces which may be analytical or faceted (triangulated/tessellated). Similarly the geometry of edges is defined by segments of curves which may be analytical or faceted. The geometry of vertices is merely a point in space. Shells and loops do not have geometry associated with them since they are not physical entities but topological abstractions.

9 Survey of geometric modeling software, and their suitability for geological modeling, their meshing and analysis:

Given below is a description of the geometric modelers reviewed for the purpose of building geological models. The factors considered in the software survey were:

1. Capability to handle non-manifold geometric models.
2. Capability to handle faceted/tessellated surfaces.
3. Robust programming interface to the kernel so that the software can be tied into existing mesh generation software.
4. Demonstration of reliability and capabilities by application in independent applications.
5. Support of the essential capabilities by group or company offering the software.
6. Cost to purchase and maintain software.

9.1 Geometric Modelers:

1. *Parasolid (Unigraphics Solutions Inc, USA):*

   Parasolid(TM) is a robust, powerful geometric modeler that has been widely used in the industry for many years now. Its geometry definition, geometry manipulation, import capabilities together with tolerant modeling and the recently introduced non-manifold modeling capability make it an industry leader in geometric modeling. It is used as a kernel of many CAD packages and other applications including mesh generators. Its main shortcoming for use in geological modeling is that the underlying geometry of model faces must be analytical and cannot be defined by a faceted (triangulated) surfaces.

2. *ACIS (Spatial Technologies Inc, USA):*

   ACIS (TM) is the main contender for the position of industry leader to Parasolid. ACIS is also a robust and powerful geometric modeler and is used widely as the kernel for many CAD packages and applications. ACIS also has strong geometry definition and manipulation capabilities. The modeler, however, lacks the full functionality of tolerant modeling available in Parasolid. ACIS does have a Mesh Surface Module which is designed to handle faceted geometry with limited capability for operations that include analytical geometry as well. The biggest obstacle to using ACIS, is that it no longer supports this module and there appear to be no customers using this module. ACIS does provide full access to the source code for this module and unrestricted rights to its use. Therefore, this may be an option to look at if one were to build on the existing source code.

3. *Shapes (XOX Corp, USA):*

   Shapes (TM) is a relatively new entry into the geometric modeling kernel arena and therefore has some advantages and disadvantages compared to the older kernels. Since it is new, it has been designed from the beginning as a full fledged non-manifold geometric modeler capable of mixing topological entities of any dimension (up to 3). This is contrast to other modelers which have added on the non-manifold modeling capability. On the other hand, it is not as mature as Parasolid and ACIS, it suffers from reliability problems and lack of some essential functionality.

   The main feature of Shapes of use for geological modeling is it's ability to handle faceted surfaces called web geoms (in its terminology). It has an impressive list of capabilities designed to handle such models:

   - Create web geoms from a list of points and connectivity information
   - Handle combinations of triangles and quadrilaterals
   - Sew multiple web geoms together to form a shell
   - Orient web geoms with respect to a shell so that a model region can be defined from a closed shell
   - Make web geom triangulations match at the common edges
   - Fit a parametric space to a web geom (although very often the mapping between the real and parametric spaces is very distorted)
   - Perform boolean operations on web geoms
   - Allow access to the web geom as a whole or to the individual facets that form the web geom
   - Allow the full set of geometric queries (except second derivatives) on web geoms

   While the reliability of these operations particularly the modification operations is not perfect, this is by far the most comprehensive set of functions available for building, meshing and analyzing geological models.

4. *Pro/ENGINEER (Parametric Technology Corp, USA):*

   Pro/ENGINEER is another stalwart in the field of solid modeling and is used widely in the mechanical design industry. It is based on the philosophy of parametric design where one can design an entire part based on parametric relationships and then decide what values of parameters are suitable for the model. Also, the modeler kernel is reported to be primarily a CSG type kernel although Parametric has recently been offering a programming interface to its kernel first as Pro/DEVELOP and then as Pro/TOOLKIT. These interfaces can provide B-rep type information to the calling application.

   This modeler is also incapable of representing faceted surfaces in its modeler and therefore, it is not of use to the geological modeling effort.

5. *BRL-CAD (Army Research Laboratory, USA):*

   This is modeler from the Army Research Laboratory is a CSG modeler with no apparent programming interface callable from FORTRAN, C or C++ codes. The primary emphasis seems to be that of a CAD package for building and visualizing models. Therefore, this package is not of any use for applications such as mesh generators.

6. *Open CAS.CADE (open source from Matra Datavision, France):*

   Open CAS.CADE is a full featured geometric modeler from Matra Datavision in France. Matra Datavision has decided to adopt an open source business model and therefore, the source code for this modeler is available freely. The modeler itself is a full fledged non-manifold geometric modeler with a

C++ API. Most important modeling and querying operations appear to be available in the modeler. Also, a significant advantage this modeler has over all except Shapes is that it can represent faceted surfaces. However, according to its developers most of its modeling operations, having been designed to work with analytical surfaces, will not work for faceted surfaces.

7. *Irit (open source from Technion, Israel):*

   Experimental modeler for educational and research purposes. From the documentation there is no indication that it is B-rep modeler or even if it is it does not subscribe to the concept of hierarchical topological structure.

8. *Varkon (Microform AB, Sweden):*

   Varkon is a geometric modeling kernel being developed by Microform AB in Sweden. According to information on their web site, Varkon is not a true solid modeler although it has many geometric modeling capabilities. From this it must be supposed that it also does not understand non-manifold topology.

9. Miscellaneous

   A number of other CAD packages were studied as part of the survey. Some of them could not handle non-manifold topology or faceted surfaces. Others had an impressive suit of capabilities integrated into a graphical user interface thus precluding any interface with existing meshing software at LANL.

9.2 Conclusions:

Of the various modelers reviewed, it was determined that Shapes best suited the needs of the geological modeling effort. A license for Shapes has been obtained from XOX Corp. Work has started on studying Shapes capabilities in detail.

References

[1] K. J. Weiler. The radial-edge structure: A topological representation for non-manifold geometric boundary representations. In M. J. Wozny, H. W. McLaughlin, and J. L. Encarnacao, editors, *Geometric Modeling for CAD Applications*, pages 3–36. North Holland, 1988.

[2] M. W. Beall and M. S. Shephard. A general topology-based mesh data structure. *International Journal for Numerical Methods in Engineering*, 40(9):1573–1596, May 1997.

[3] R. V. Garimella and M. S. Shephard. Generation of non-manifold representations using the capabilities of 2-manifold modelers. Scientific Computation Research Center, Rensselaer Polytechnic, Troy, NY 12180, 1993.