PROCEEDINGS, Twenty-Third Workshop on Geothermal Reservoir Engineering
Stanford University, Stanford, California, January 26-28, 1998
SGP-TR-158

A GEOMETRIC MODELING FRAMEWORK FOR THE NUMERICAL ANALYSIS OF GEOTHERMAL RESERVOIRS

Brian Hardeman
Daniel Swenson

Kansas State University
302 Rathbone Hall
Manhattan, KS 66506
http://www.mne.ksu.edu/~geocrack

ABSTRACT

In the past decade, advancements in automatic mesh generation and topological data structures have made possible the use of a more general and abstract geometric model for the description of an analysis problem. Using a geometric model for describing a reservoir results in a conceptual model of the geothermal system, rather than a simple numerical simulation. The analyst builds the geometric model using true features of a reservoir, such as well bores and known fracture locations. Material properties and boundary conditions are then assigned to these features, not to the underlying finite element mesh or finite difference grid. This independent storage of the problem description makes it simple to run multiple analyses, potentially using different solution schemes. Definition of both 2-d and 3-d models can be greatly simplified through this more interactive, intuitive model creation process. State-of-the-art visualization and manipulation methods assist in conveying the conceptual model of the reservoir and the assumptions made for performing a simulation.

MOTIVATION

Current reservoir analysis software (GEOCRACK, TOUGH2, TETRAD) requires the user to have in-depth knowledge of the numerical solution scheme used. The problem description, in the form of a finite element mesh or finite difference grid, serves as the fundamental carrier of information into the simulation. This mesh must satisfy not only the restraints dictated by the problem at hand, such as geometric boundaries and features (reservoir extent, fractures, wellbores), but also must comply with numeric restrictions of the solution method, such as maximum or minimum element sizes, grid density, and any special input parameters. This approach requires the analyst to think ahead when creating the mesh, because if any of the desired characteristics are not met, the mesh or grid must be regenerated, incurring a substantial cost in time and effort.

Relying on the input grid or mesh for problem information can also cause difficulty when attributes (material properties, boundary conditions, analysis assumptions) or geometry change during the course of an analysis. Placing all of the problem description in the mesh requires a change to the mesh definition. For example, to change attribute data, one must only change a numeric value stored with the mesh elements, but this change must occur for every element in the region of the change. While interactive tools for querying and manipulating these values can help, the task is still tedious, particularly for meshes with a great level of detail and a fine discretization. Even if the interface allows the user to specify the new value and lets the software determine which elements require updating, the program must still perform repeated searches to find the affected elements.

A change in geometry is even more serious and user intensive. Because of the close link between the location and size of elements and the physical boundaries of the problem, a change in geometry requires regeneration of the mesh. Such changes cause problems for finite element meshes, which must conform to boundaries, and can also affect finite difference grids, where the grid density may be closely related to high solution gradients caused by boundary conditions or boundaries of different materials. For many analysis systems based on input files, this would involve recreating the entire grid and reassigning all attributes. Even in systems which allow interactive modification of the mesh, these changes can be very labor intensive, particularly for large or detailed grids.

While modifying a two-dimensional solution grid can be tedious, it is relatively straightforward. However, direct manipulation **of** a three-dimensional grid **or** mesh becomes nearly impossible. Spatial visualization is difficult, even when using interactive and graphical modification tools. In addition, the number of elements or cells increases greatly in three dimensions, further complicating modification tasks. In order to avoid such low-level manipulations, the **user** needs a more intuitive and automated way of incorporating geometric changes.

The above arguments also apply to situations where the solution mesh needs modification independently **of** the defining geometry. Such situations occur as a result of the numerical techniques used in the solution. **For** instance, as the analysis of a particular reservoir evolves through time, areas of high temperature **or** pressure gradients may move to different parts **of** the reservoir, requiring smaller elements in a new area of the mesh. Again, it can be tedious for the analyst to manually modify the mesh to introduce new areas of refinement and assign proper attribute information.

Adaptive meshing, where automatic mesh generation is used in conjunction with an estimation of the error in the calculated solution, is a powerful approach to addressing appropriate mesh refinement (Lewis **1991**). But if the mesh provides the only source of problem information, data such as geometry and material properties, must be inferred from the current mesh before a new one may be generated and assigned the proper information. Storing the geometry and attributes independent of the mesh greatly simplifies the implementation of **an** adaptive meshing scheme.

Another reason for the use of geometric modeling arises when the analyst desires to change the numerical method **of** the solution, for instance from a finite difference to a finite element or control-volume finite element solution. This requires the creation of another mesh with all of the proper data attached. **A** higher-level description of the problem, independent of the actual solution method, could allow the analyst to describe the problem's geometry and input parameters only once and provide this data in an automated way to the solution mesh **or** grid actually used for the solution. The user would only have to provide additional parameters specific to the chosen method, as opposed to redefining the entire problem to use the new method.

GEOMETRI DELIN

A etric model is a level, geometry representation of an ɔj This representation pro a natural and convenient location to store all of the necessary information associated with a numerical analysis problem, such **as** material properties and boundary conditions. Combining the geometric model with the parameters of the analysis creates a problem description unique to the problem at hand, but independent of any given analysis method.

The geometric model serves as a means of communication between an analyst and the analysis software. This model is independent of the solution method, yet contains all of the necessary parameters, making it a natural way to convey information about a given problem to the simulation software. The geometric model contains information about the nature of the problem and can store the results obtained from simulations of the problem. In this way, the analyst need only interact with the abstract description of a reservoir, not with the underlying solution mesh or grid. This simplifies and streamlines the job of the analyst and provides an organizational tool for defining, performing, and interpreting reservoir simulations. The chore of developing input files and translation file formats can be greatly simplified and automated by the modeling software because all of the problem information is centralized in the geometric model.

Several different ways exist to implement a geometric model. Each method has different levels of representational capability and introduces a variety of benefits and drawbacks. The three primary types of geometric models are decomposition models, constructive solid geometry, and boundary representations (Mantayla **1988).**

Topology

Boundary representation, a form of geometric modeling where the explicit storage of boundary information describes an object, provides the underlying support for the geometric modeling framework discussed herein. The boundary of a model consists of adjacency information between point sets known as ***regions, faces, edges,*** and ***vertices.*** This adjacency information is commonly referred to as *topology.*

A ***region*** defines a separate portion of three-dimensional space that may be either a bounded (finite) or unbounded (infinite) subset of $\mathbf{R}^3$. A ***face*** is a bounded, two-dimensional subset of $\mathbf{R}^3$ that corresponds to a surface, and an ***edge*** is a one-dimensional subset corresponding to a curve. A ***vertex*** is a zero-dimensional entity that represents a unique point in space.

The boundary of a region is comprised of a set of elements of dimension less than three. **A** typical region' boundary would be simply a set of faces. Similarly, faces are bounded by a collection of edges. Every edge is then bound at either end by a vertex. Together, all of the regions, faces, edges, and vertices of the model make up the entire model space of $\mathbf{R}^3$. This hierarchical representation of an object is depicted in Figure 1.

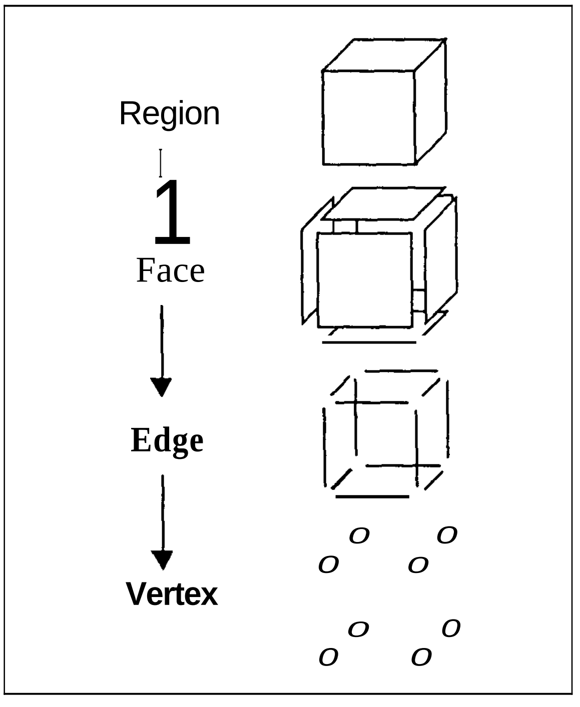

Figure 1 Topological elements in boundary representation modeling.

Storage of the boundaries of all entities in the model explicitly maintains the adjacency information of the model. Because the adjacencies are stored directly, a boundary representation supports very efficient query and traversal methods. This simplifies many tasks, such as display and local modifications of the model.

Because the adjacency information simplifies local modification of a boundary representation, construction and modification operators are designed around this property. These operators, known as the Euler operators, were first introduced by Baumgart (1974). The Euler operators guarantee that modifications keep the model in a consistent topological state. While simple and robust, the Euler operators are very low-level tools and are not suitable as the sole means of creating a large and complex model. However, the use of Boolean set operations and feature-based modeling in conjunction with the Euler operators provides a convenient and powerful geometric modeling interface.

Non-Manifold Topology

Topology may be divided into two major categories – manifold and non-manifold. Manifold refers to the boundaries of the point set elements. A manifold boundary is one that is of dimension one less than the entity that it bounds at every location on the boundary. For instance, a traditional solid model has a manifold boundary if, at every spot on the boundary, the boundary is two-dimensional. This is equivalent to stating that every point on the boundary is topologically identical to an open disk in $\mathbf{R}^2$. Because the boundary must be two-dimensional everywhere, each edge must have two and only two faces incident upon it. These properties have been used to optimize data structures for manifold models, such as the winged-edge data structure (Baumgart, 1974). Every physical object has a manifold boundary, so a manifold assumption does not limit the real objects that a boundary representation can model.

Manifold boundaries simplify some data structure and manipulation tasks, while not limiting the class of objects that can be represented, yet are still too restrictive for a truly general geometric modeling system. For example, it is very natural to model a truss structure as a set of wireframe edges in space, or a car fender as an open surface (face). Under manifold boundary assumptions, these situations would have to be modeled exactly as they are in real life. The truss elements would have to be modeled explicitly, each with their own cylindrical face boundary, and the fender would have to be modeled as a complete solid with faces for both the front and back and for the thin area around the edges. A non-manifold boundary allows the bounding elements to be a mixture of elements of dimension one less that the bounded region. This extends the representational capability of the boundary representation from real objects to also handle useful abstractions that arise in numerical analysis and simulation.

The geometric modeling framework discussed here utilizes a fully non-manifold topological database. The data structure, called the Multi-Link data structure, is a combination of ideas and techniques from previous work on non-manifold boundary representation (Weiler 1986, Choi 1989, Rossignac 1990).

Feature-Based Modeling

A geometric model is based upon a mathematical, geometric description of an object. This high degree of abstraction makes developing an interface to the model based upon the features of the problem domain natural. In models with a low degree of abstraction, such as finite element meshes, the mesh must be created with the geometry in mind. Nodes and elements must be placed so that the final mesh conforms to the material boundaries and allows boundary conditions to be appropriately imposed. For example, to represent a wellbore as a one-dimensional feature, nodes would have to be placed along the well and elements formed so that edges lie exactly along the well.

Using a geometric model makes a higher level of abstraction possible. A wellbore may be represented by simply inserting an edge at the location of the well in the model. Intersections with other edges and faces create vertices, and all information can be propagated to the solution mesh or grid.

To increase abstraction even further, a well may be inserted simply by specifying the geometric description of the wellbore and allowing the geometric model to take care of the details of representation. Similarly, the user would add a fracture or fault to a reservoir model by locating it within the model and supplying its strike/dip specification. A geometric description of a planar surface patch would be constructed from the input, and a face would be intersected and added to the geometric model using that geometry. Just as the abstraction of a geometric model insulates a user from solution method details, using features to describe the problem insulates the analyst from the implementation details of the geometric model.

EXAMPLE

A simple example demonstrates the expressive power and intuitive interface provided by a geometric modeling system. A basic representation of a geothermal system requires only a few operations. The hot dry rock reservoir in Hijiori, Japan provides a useful example.

The first step **is** to provide the dimensions of the overall boundaries **of** the reservoir. At Hijiori, the active region of circulation covers approximately **500m** by 1000m in plan view and extends 1000m in depth. Figure 2 shows the bounding volume of the reservoir as represented in the geometric modeling framework.

Next, the wells are added to the model. At Hijiori, three wells of interest extend into the fractured region, HDR-1, **HDR-2a**, and HDR-3. For a simple approximation, the wells are represented as straight lines. The wells may be input as a sequence of line segments or as a spline curve if desired for a more accurate representation. A wireframe view of the model is shown in Figure 3.

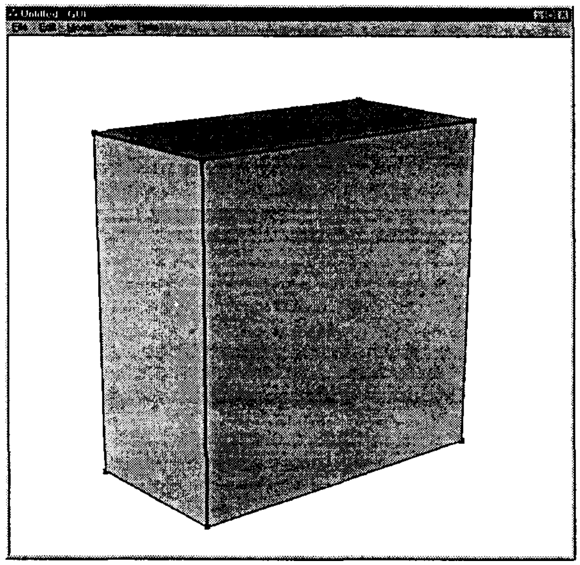

Figure 2 Boundary of a reservoir.

Fractures are added to the model by inserting planar polygons or infinite planes that are truncated by the reservoir boundary. Two zones of high permeability are observed at the Hijiori reservoir. A simple model of this system that can be used for analysis of basic behavior of the reservoir contains two infinite fractures. Figure 4 shows the basic reservoir system with two flowing fractures. The fractures are located and oriented by the input of a point on the plane and the known strike and dip information.

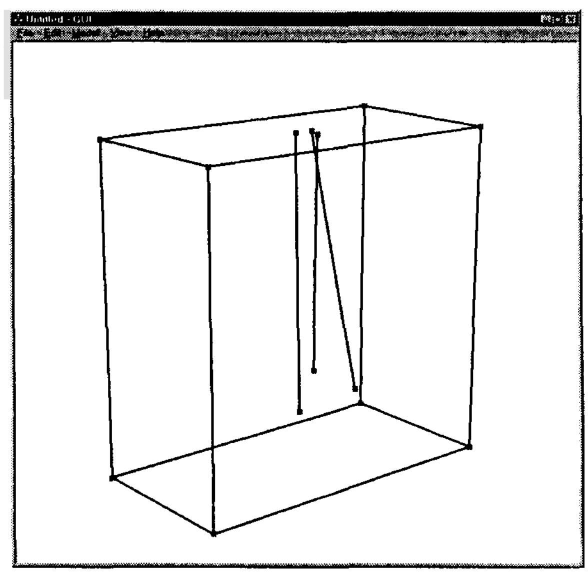

Figure 3 Geometric model including boundary and wells.

To assist in visualization and conceptualization of the reservoir geometry, the software supports real-time manipulation and rotation of the model. The reservoir can be viewed as either a wireframe or solid model, and cutting planes can be used to strip away portions and look inside, as shown in Figure *5.*

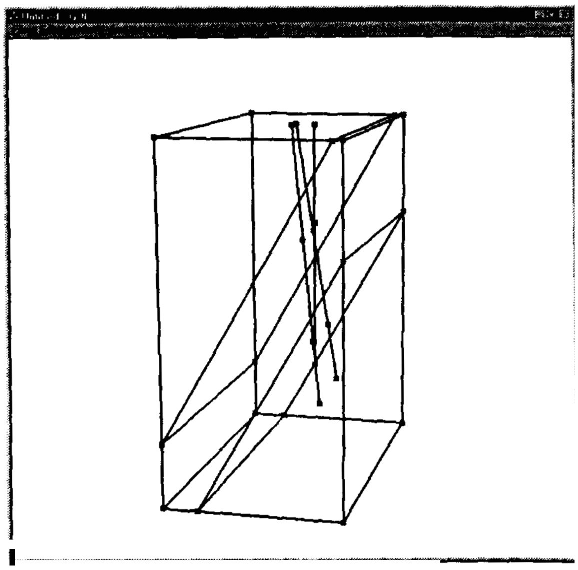

Figure 4 Two fracture model of Hijiori.

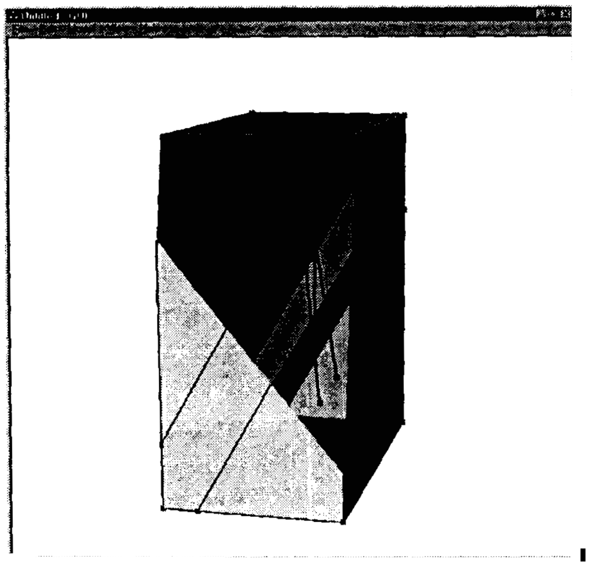

*Figure 5 Use of **a** cutting plane to view the interior of a reservoir.*

After creation of the geometry, attribute information representing rock material properties, fracture opening and permeability, and operating conditions will be assigned to the elements of the model. Ultimately, this information will be used to generate an appropriate solution grid or mesh. All data is then available to perform an analysis of the reservoir or create an input file for an external analysis package.

CONCLUSIONS

Geometric modeling provides a flexible and efficient analysis interface to serve as a high-level description of a geothermal reservoir. Mesh-based approaches are simply too detailed and dependent on the associated numerical method to be generally useful. The specification of a geometric model and associated problem data are logical first steps in an analysis and can serve as the medium of communication between an analyst and the analysis software.

The use of geometric modeling, combined with advanced visualization and manipulation tools, can greatly simplify the task of preparing complex problems for analysis. These advances have the potential to allow reservoir engineers perform calculations previously limited to scientists and computer programmers.

ACKNOWLEDGEMENTS

W thank Mark m for long and insightful s us on non d geometric m his work was supported by the c of l Te U.S. Department of Energy, and performed in the Applied Computation Laboratory at Kansas State University

REFERENCES

Baumgart, B. "Geometric Modeling for Computer Vision." Ph.D. Thesis, Stanford University, 1974.

Choi, Young. "Vertex-Based Representation of Non-Manifold Geometric Models." Ph.D. Thesis, Carnegie-Mellon University, August 1989.

Lewis, R. W., et. al. "Finite Element Analysis of Heat Transfer and Flow Problems Using Adaptive Remeshing Including Application to Solidification Problems." *International Journal for Numerical Methods in Engineering,* Vol. **32,** pp. 767-781, 1991.

Mantyla, Martti. ***An** Introduction to Solid Modeling.* Computer Science Press, 1988.

Rossignac, Jaroslaw R. and Michael **A.** O'Connor. "SGC: A dimension-independent model for pointsets with internal structures and incomplete boundaries." *Geometric Modeling for Product Engineering.* Eds. Wozny, M. J., J. U. Turner, and K. Preiss. Amsterdam: North-Holland, 1990. 145-180

Swenson, Daniel, et. al. "GEOCRACK: A Coupled Fluid Flow / Heat Transfer / Rock Deformation Program for Analysis of Fluid Flow in Jointed Rock." Mechanical Engineering Department, Kansas State University, Manhattan, KS.

Weiler, Kevin. "Topological Structures for Geometric Modeling." Ph.D. Thesis, Rensselaer Polytechnic Institute, August **1986.**