Proceedings World Geothermal Congress 2000
Kyushu - Tohoku, Japan, May 28 - June 10, 2000

A 3-D FINITE ELEMENT MODEL OF FLOW IN FRACTURED RESERVOIRS

B. Hardeman, S. Kulkarni, and D. Swenson

Mechanical and Nuclear Engineering Department, Kansas State University, Manhattan, KS, 66506, USA, swenson@ksu.edu

Key Words: Finite element, geothermal, fracture, geometry, analysis, object oriented, topology

ABSTRACT

We present the design and implementation of a 3-D finite element model of flow in a fractured reservoir. Three key features of our approach are discussed: (1) the need for discrete modeling of flow on fractures, (2) the use of a topological data base to describe the model independent of an underlying finite element mesh or finite difference grid, and, (3) the design and implementation of the finite element analysis module using object-oriented programming techniques. A simple example analysis illustrates the approach.

The goal of our approach is to allow the analyst to work at an intuitive level relevant to the physical nature of the problem, rather than being concerned with the details of the solution procedure. Material properties and boundary conditions are assigned to features in the model, not to a finite element mesh or finite difference grid. This high level, independent storage of the problem description makes it much easier for an analysis to be performed. 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.

1. MOTIVATION

In geothermal reservoirs, flow in fractures can be dominant when cold water flow is re-injected into the circulation loop. If a fracture provides a preferential flow path (channeling), cooling can occur at the production wells earlier than desired. Such situations can arise in hydrothermal reservoirs during re-injection or in Hot Dry Rock (HDR) or Enhanced Geothermal Systems (EGS) where flow is on fractures.

In a recent review of HDR/HWR technology (Structured Academic Review of HDR/HWR, 1997) the following statements are made:

> "The nature of the flow field within the reservoir, and most importantly, within the planes of the fractures that link to form through-going conduits, represents one of the most crucial yet least understood aspects affecting the performance and longevity of HDR Systems. Data from rock mechanics experiments and theoretical analyses provide firm grounds to expect that channelling of flow within fractures will occur to some extent, and that this will reduce the surface area swept by the primary flow. … The available data from tracers and thermal drawdown are insufficient to define the extent to which the channeling phenomenon is a problem."

> "Cooling of the reservoir during operation will result in the generation of thermo-elastic stress. Both positive and negative consequences can be anticipated."

> "It may be difficult to model rates of chemical processes, but it is important to model the processes by which chemical interference may take place, and the hydraulic consequences which would be observed. If a remedy was found for such difficulties, it could be applied when they were observed, rather than waiting until the damage was done."

These statements indicate both the need for modeling of flow on fractures and some of the challenges facing the developer, including coupled physics (hydro/thermal/structure), coupled chemistry, and non-uniform flow on fractures.

Data from the Hijori reservoir in Japan (GERD, 1996; GERD 1997) illustrates some of these aspects. The Hijiori reservoir is located on the southern boundary of the Hijiori caldera. The reservoir wells intersect fractures that are part of the ring structure around the caldera and strike approximately east-west and dip steeply to the north, at an angle of about 70 degrees from the horizontal. The intersections of these fractures with the wells are shown in Figure 1. HDR-2a and HDR-3 are production wells that are open (not cased) below about 1500 m. HDR-1 is the injection well and is cased to a depth of about 2150 m.

Pressure-temperature-spinner (PTS) data show the distribution of flows in the fractures, Figure 2 (GERD, 1997). It is believed that at least four of the producing points in each well correlate to common fractures that intersect both wells. Two of the pairs are indicated: F2a-9/F3-7 and F2a-3/F3-2. This means that significant connections between the wells occur on the common fractures. Although many more fractures have been identified in televiewer data, only a limited number of the fractures are actively flowing. One of the reasons only certain fractures are active may be their orientation with respect to the in-situ stresses (Barton et al., 1997). In any case, the distinct entry points illustrate the importance of fracture flow in the Hijiori reservoir.

Using the borehole data and known fracture orientations, it is possible to begin to develop a model of the reservoir. Figure 6 illustrates four of the connecting fractures between HDR-2a and HDR-3 (this is discussed further in the example model section).

2. GEOMETRIC MODELING

2.1 The Need for a High-Level Geometric Description

Current reservoir analysis software, such as TOUGH2 (Pruess, 1991), TETRAD (Au, 1995) requires the user to have in-depth knowledge of the numerical solution scheme. 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 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. Changing either the geometry or problem attributes (material properties, boundary conditions, analysis assumptions) can also become a labor intensive and error-prone task.

It should be noted that aid for model development and input file preparation is available using codes such as GEOCAD (Burnell and White, 1996), that helps the user prepare TOUGH input files. However, these codes typically focus on the mesh and do not include a high-level geometric description of the reservoir as described in this paper.

In addition to simplifying the analysis task, use of a higher level geometric model makes it possible to develop a model independent of a particular solution scheme (finite difference, finite element, or control-volume). A higher-level description of the problem can 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 only has to provide additional parameters specific to the chosen method, as opposed to redefining the entire problem to use the new method.

Clearly, using a geometric description that directly corresponds to features in the reservoir has many advantages. Such a representation provides a natural and convenient location to store all of the necessary information associated with a numerical analysis problem, but independent of any given analysis method.

2.2 Topology and Boundary Representation

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* (Mantayla, 1988). 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 and every edge is bound by two vertices. 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 is depicted in Figure 3.

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.

2.3 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 the surface of a solid. Manifold boundaries are 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 fracture as an open surface (face). 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).

2.4 Feature-Based Modeling

A geometric model is based upon a mathematical, geometric description of an object. 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.

3. OBJECT-ORIENTED FINITE ELEMENT MODEL

3.1 Background

The goal is to design a finite element program framework that will include our current approach to the coupled hydro-thermal-structure problem, while allowing extension and application to other problems.

The primary variables for solution are displacements (ux, uy, and uz), pressure (P), and temperature (T). The equilibrium variables (physical variables used in writing the differential equation) will be forces (fx, fy, and fz), mass flow of fluids (q), and heat flux (h). We may also want to solve for concentrations (c_0, c_1, …, c_n). In general, we want the design to accommodate the addition of new degrees of freedom, such as rotations.

The design must accommodate different element types. Each element type must provide the tangent matrix for all the degrees of freedom. For instance, each element must provide the standard structure stiffness matrix

$$K_{struct} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \tag{1}$$

in addition to the heat transfer conductivity matrix

$$K_{conduct} = \frac{\partial \mathbf{h}}{\partial \mathbf{T}} \tag{2}$$

and the fluid permeability matrix

$$K_{fluid} = \frac{\partial \mathbf{q}}{\partial \mathbf{P}} \tag{3}$$

To allow a full Newton-Raphson iteration to be applied, the cross-coupling terms should be supplied if requested.

$$\frac{\partial \mathbf{q}}{\partial \mathbf{u}}, \frac{\partial \mathbf{h}}{\partial \mathbf{u}}, \frac{\partial \mathbf{f}}{\partial \mathbf{P}}, \frac{\partial \mathbf{h}}{\partial \mathbf{P}}, \frac{\partial \mathbf{f}}{\partial \mathbf{T}}, \frac{\partial \mathbf{q}}{\partial \mathbf{T}} \tag{4}$$

The element must also evaluate equilibrium variables (f, q, and h) and residuals. For transient analyses, the appropriate mass (storage) matrices must be evaluated. It is desired to accommodate the addition of new elements, such as plate or beam elements.

3.2 Object Design and Classes

The design of an object-oriented program attempts to represent the problem domain using objects and their interactions. The most significant object classes are briefly described below.

Degree of Freedom

A degree of freedom, Dof, stores the values of a solution variable. Dof's can either be active (to be solved) or inactive (value specified). The design must be general enough to accommodate the addition of new degrees of freedom.

Node

A node holds the geometry information (coordinates) and acts as a container for the various degrees of freedom. The number of DOF's in a node can vary.

Element

An element discretizes the geometry and interpolates the solution. It also acts as a container for its nodes, material, and Gauss point data. It provides an iterator to traverse and access its nodes.

In traditional finite element programs an element such as a T6 (Triangular 6 noded quadratic element) may have multiple implementations, each differing from the other only in the way it treats the physical behavior of the system. This introduces multiplicity of code to do the same purpose. The current approach emphasizes that the main purpose of an element is discretization and interpolation. The underlying element physics is independent of the element functionality and is separated from geometry by the use of behaviors. Thus a T6 can model fluid flow or elasticity by simply changing the behavior that is attached to it.

The Element classes have a typical hierarchy as shown in Figure 4. The abstract Element base class defines the interface using pure virtual functions. The implementation is provided by the specific derived classes. Additional type-specific pure virtual functions are added at the Line/Joint/Surface/Volume level as needed.

Behavior

The Behavior classes handle the physics of the problem being solved, for example, fluid flow in a fracture. The behavior requests geometry, interpolated data, and Gauss point information from the element to which it is attached. The actual integration point looping is performed in the behavior, as it knows the required order of integration. The behavior class hierarchy is similar to the element hierarchy.

The typical methods include tangentStiff, which calculates an element tangent stiffness matrix, and internalForce which calculates the internal element forces. Another important functionality provided is that of reserving the degrees of freedom. An element requires particular type of degrees of freedom allocated to all of its nodes depending upon the kind of behavior that its solving for. Thus, behavior is the class to which such a functionality is delegated by the element.

Material

Every element that models the physical world has some physical properties associated with it. The Material class is a generic container of material properties. A material can contain another material; thus allowing generality of implementation. For example, a material "fracture" will have "aperture" as a material value and another embedded material "water". The material "water" in turn will have material values (properties) like "density" and "viscosity".

Dof Manager

During a finite element solution process, the current solution is stored in the solution vector. When converged, the data will be stored in the appropriate Dof. The Dof Manager controls this interaction.

Integrator

An integrator is basically an iterator over the Gaussian integration points. It provides the coordinate value at an integration point and the associated weight. An element returns an integrator over itself when supplied with the order of integration that is required

Domain

This one of the most important container classes in the system. It has all the problem data, including the nodes, elements, materials, behaviors, boundary conditions and tracer data. The Domain class does not know about the solution methods that are going to be used to solve the problem nor anything about the solvers. It contains data in the form of dynamically growing arrays for providing constant time accessing of data as it is going to be referred multiple number of times during the complete initialization and solution process. The domain class also has various iterator classes (for example, iterator over nodes, elements etc.) as its friend classes.

Driver

Driver controls the overall solution of the problem, choosing the appropriate solution method. The driver creates the domain and adds all the nodes, elements, boundary conditions, tracer data, etc. to it. After the finite element solution is obtained, output files for post-processing are written out. This driver acts as the main "action" class.

Solution Strategy

The main purpose of Solution Strategy class is to bring the problem from one state into a future state. It uses the current solution (in the domain) and saves the new solution in the domain. If a transient solution is requested, it controls the time stepping until the requested time is reached. From the pure abstract base class, SolutionStrategy, the LinearSolutionStategy is derived. Similarly, any of the non-linear solution schemes such as Newton-Raphson can be derived from it. It provides methods for initializing and solving the solution.

4. EXAMPLE

A simple example demonstrates the expressive power and intuitive interface provided by a geometric modeling system as implemented in the program Geocrack3D (Swenson, et al., 1999). 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. Next, wells can be added to the model (features can be added in any order, fractures could be added first rather than wells). At Hijiori, three wells of interest extend into the fractured region, HDR-1, HDR-2a, and HDR-3. Figure 5 shows the model after the wells have been added.

Fractures are added to the model by inserting planar polygons or infinite planes that are truncated by the reservoir boundary. Figure 6 shows the model with four fractures. These are oriented to match the known intersections with HDR-2a and HDR-3. Note that the intersections of the fractures with the wells was automatically detected by the topological model.

After creation of the geometry, attribute information representing rock material properties, fracture opening and permeability, and operating conditions are assigned to the model features. Up to this point, all information has been stored independent of the solution scheme. Now a finite element mesh is created, Figure 7. Instead of a finite element model, a finite difference grid could be created.

The mesh and attribute data is sent to the finite element program for solution. After the solution is obtained, results can be plotted. Figure 8 shows pressure contours for the problem.

5. 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.

The object-oriented approach used to implement the finite element solution helps in maintaining and adding new features to the model.

At the time of paper preparation, the model could solve for fluid flow on fractures. Active development is continuing of heat transfer and elasticity.

6. ACKNOWLEDGEMENTS

The authors thank Mark James for extensive discussions during the design of Geocrack3D. We thank the New Energy and Industrial Technology Development Organization (NEDO) for providing the Hijiori data used in this paper.

7. REFERENCES

Au, A. (1995) *TETRAD User Manual*. ADA International Consulting Inc., 705 Hawkwood Blvd., N.W., Calgary, Alberta, Canada, T3G 2V7, 403-630-3872.

Barton, C. A., Hickman, S., Morin, R., Zoback, M., Finkbeiner, T., Sass, J., and Benoit, D. (1997). Fracture Permeability and Its Relationship to In-Situ Stress in the Dixie Valley, Nevada, Geothermal Reservoir. PROCEEDINGS, Twenty-Second Workshop on Geothermal Reservoir Engineering, Stanford University, Stanford, California.

Baumgart, B. (1974). *Geometric Modeling for Computer Vision*. Ph.D. Thesis, Stanford University, Stanford, CA.

Burnell, J. G. and White, S. P. (1996). *GEOCAD V3.0 For Windows User Manual*, Applied Mathematics, Industrial Research Limited, Lower Hutt, New Zealand, ISSN 1171-9281.

Choi, Y. (1989). *Vertex-Based Representation of Non-Manifold Geometric Models*. Ph.D. Thesis, Carnegie-Mellon

University, Pittsburgh, PA.

GERD (1996). *FY 1995 Summary of Hot Dry Rock Geothermal Power Project in Japan*. New Energy and Industrial Technology Development Organization, Tokyo, Japan.

GERD (1997). *FY 1996 Summary of Hot Dry Rock Geothermal Power Project in Japan*. New Energy and Industrial Technology Development Organization, Tokyo, Japan.

Mantyla, M. (1988). *An Introduction to Solid Modeling*. Computer Science Press.

Pruess, K. (1991). *TOUGH2 - A General Purpose Numerical Simulator for Multiphase Fluid and Heat Flow*. Lawrence Berkeley Laboratory Report LBL-29400, Lawrence Berkeley Laboratory, Berkeley, CA, May 1991.

Rossignac, J. R. and O'Connor, M. (1990). 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, pp. 145-180.

Structured Academic Review of HDR/HWR. (1997) Engineered Geothermal Systems: How We Think Reservoirs Work and What To Do Next. Tohoku University, Sendai, Japan.

Swenson, D., Hardeman, B., Kulkarni, S. (1999) Geocrack3D Examples and Manual, www.mne.ksu.edu/~geocrack, Mechanical and Nuclear Engineering Department, Kansas State University, Manhattan, KS, 66506.

Weiler, K. (1986). *Topological Structures for Geometric Modeling*. Ph.D. Thesis, Rensselaer Polytechnic Institute.

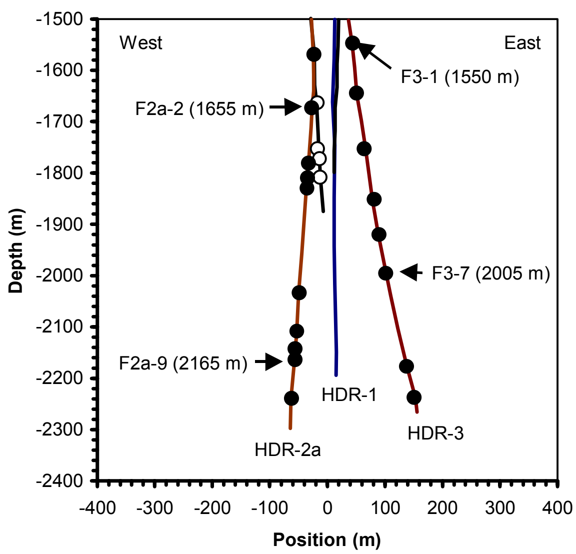

Figure 1: Vertical section of Hijiori reservoir showing fracture intersections with wells

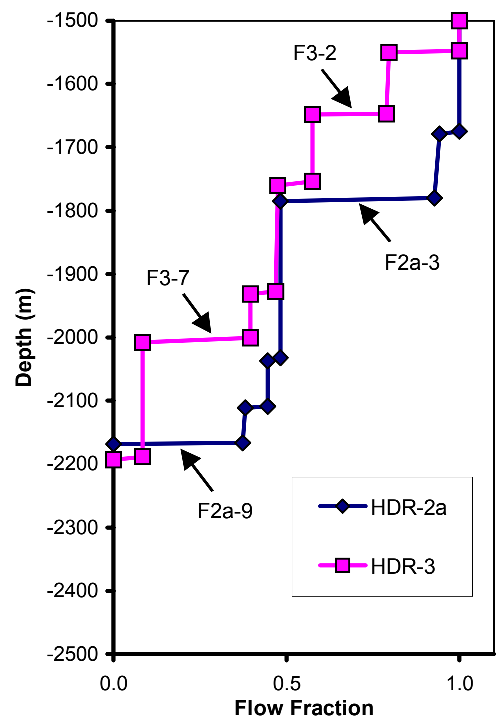

Figure 2: Measured flow fractions in producing wells

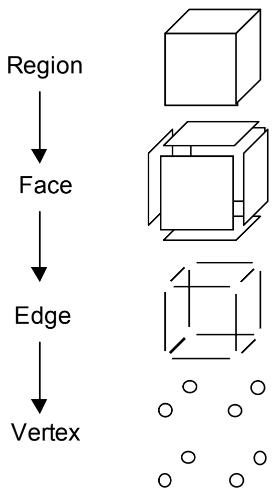

Figure 3: Topological elements in boundary representation modeling

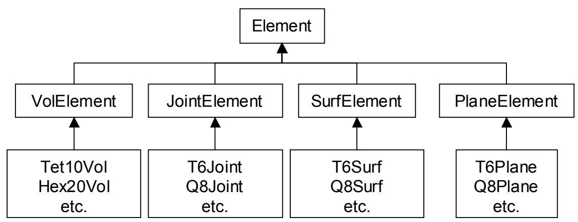

Figure 4: Element classes

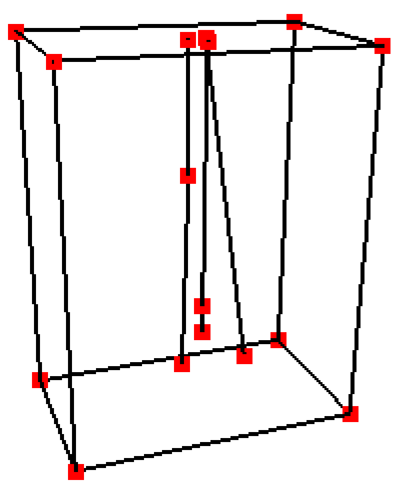

Figure 5: Boundary and three wells defined

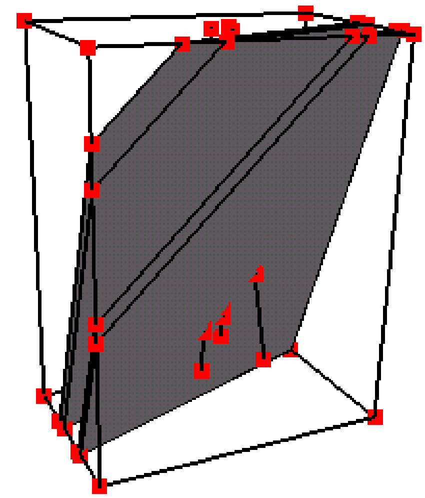

Figure 6: After adding four fractures

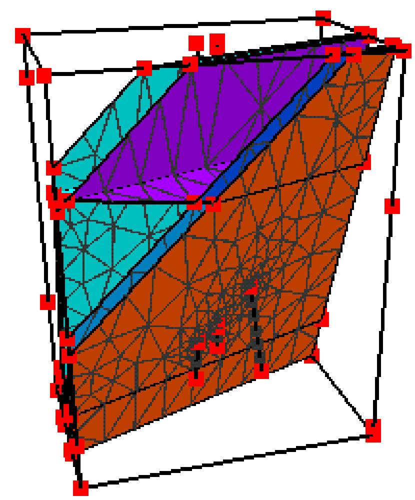

Figure 7: After meshing the fractures

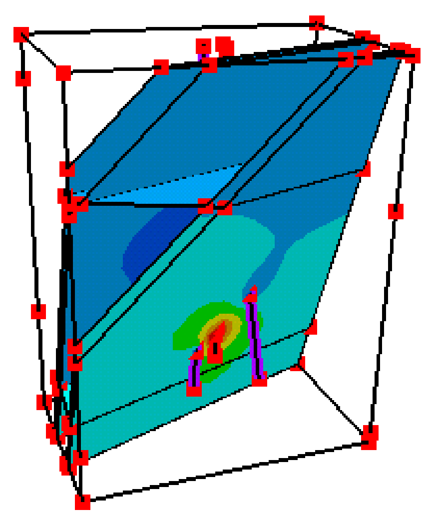

Figure 8: Pressure contours