



Chapter 1

**So You've Got
Questions ...**

What's in this Manual?

Welcome

Welcome to the Poly3D Tutorial-Based Users' Manual. If 3D modeling and numerical modeling is completely new to you, fear not. This manual was written so even the beginner can understand and use Poly3D. For those of you who are numerical modeling gurus, this manual will be an easy jump-start into using Poly3D.

This manual includes:

- Comparison of different numerical models
- Basics of Poly3D (with tutorials)
- Viewing Poly3D input and output files with IRIS Explorer*

For More Information

This is not the first users' manual written for this program. In June 1993, Andrew Lyle Thomas wrote his Master of Science thesis as a users' manual and explanation of his program Poly3D. In his thesis, he describes all the mathematical assumptions, equations, and calculations that he used to write the code. The manual you are reading now is a simplified version of his users' manual with tutorials added throughout for ease in learning the program. It is not a substitute for his thesis, but only a starting point for learning Poly3D. Detailed mathematical explanations are left to Thomas. Once a general understanding is gained, the more complicated concepts can be understood by reading his thesis:

Thomas, A.L., 1993, *Poly3D: A Three-Dimensional, Polygonal Element, Displacement Discontinuity Boundary Element Computer Program with Applications to Fractures, Faults, and Cavities in the Earth's Crust*: M.S. Thesis, Stanford University, CA.

Citing Your Work

In his thesis, Thomas asks that you cite his thesis as shown above if you use Poly3D to produce results for a report or publication. He also asks that you send one reprint of each report or publication to Prof. David D. Pollard at:

Rock Fracture Project
ATTN: Poly3D
Department of Geological & Environmental Sciences
Stanford University
Stanford, CA 94305-2115 USA

* While this introductory manual was being written, Juliet Crider (Rock Fracture Project, Stanford University) completed a manual for viewing Poly3D input and output files with IRIS Explorer. For the convenience of the user, her manual has been included as Chapter 4 of this manual.

Whom Should I Call?

This manual contains a lot of information. If you should have a problem, make sure you are contacting the correct person to prevent getting the “run-around”:

Problems with this Manual:

This manual was written by Kate Griffin, an undergraduate at the University of Wisconsin, as an internship project for her technical communications certificate. The computer copy is now maintained and distributed by the *Rock Fracture Project* at Stanford University. For problems with this manual, contact:

Rock Fracture Project
ATTN: Poly3D Tutorial-Based Users' Manual
Department of Geological & Environmental Sciences
Stanford University
Stanford, CA 94305-2115 USA

or contact Prof. David D. Pollard at:

`dpollard@pangea.stanford.edu`

Problems with Poly3D:

Poly3D is a C language computer program written by Andrew L. Thomas as part of an M.S. thesis in the Department of Geology at Stanford University. The program will be updated and maintained by the *Rock Fracture Project* at Stanford University.

Rock Fracture Project
ATTN: Poly3D
Department of Geological & Environmental Sciences
Stanford University
Stanford, CA 94305-2115 USA

or contact Prof. David D. Pollard at:

`dpollard@pangea.stanford.edu`

Problems with IRIS Explorer:

IRIS Explorer was originally developed by Silicon Graphics, Inc. and was licensed to Numerical Algorithms Group (NAG) in 1992. For questions or concerns regarding this program, contact:

IRIS Explorer Center (North America)
1400 Opus Place, Suite 200
Downers Grove, IL 60515-5702 USA
Tel: (630) 971-2367
Fax: (630) 971-2346
E-mail: `explorer@nag.com`
Internet: `http://www.nag.com/IEC`

What Are Numerical Modeling Tools?

Generally, analytical solutions to engineering or geology problems are limited to simple geometries. For example, analytical solutions are available to solve for stresses around a hole in a plate. For 3-D non-planar faults, however, there is no analytical solution available. To solve these more complex problems, mathematicians and engineers have developed computer-aided numerical methods that are more powerful. These methods allow us to solve problems with complicated initial conditions, such as the 3-Dimensional deformation that occurs between two offset faults. Numerical methods can take these initial conditions and use them to predict distortions and derive the stress and strain experienced by a body or its parts. It can also be used to determine natural frequency, mode shapes of components, and buckling loads of structures.

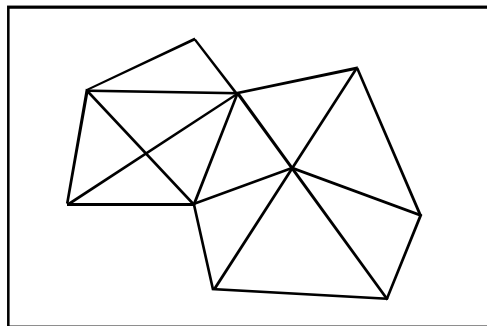
There are two main types of numerical methods: differential methods and integral methods. The Finite Element Method (FEM) is a differential method that was developed in the 1950's and is one of the most common numerical methods in use today. Since its development, other differential methods, including the Boundary Element Method (BEM), have been developed to solve different types of problems.

The most important concept to understand when considering numerical methods is that of discretization. Discretization refers to the breaking-up of a field into a finite number of elements. Drawing a free body diagram to solve a problem allows us to determine the stress and strain along the outside edges of the body by summing the forces in each **direction** ($\sum F_x = 0, \sum F_y = 0, \sum F_z = 0$) and the moment about any point ($\sum M_o = 0$). To find the forces inside the body, you have to break-up that body into smaller free body diagrams. When you discretize a body for use in a modeling program, you create many smaller free body diagrams that are all adjacent. By breaking the body into these smaller parts, the numerical code can determine stresses and strains throughout the entire body.

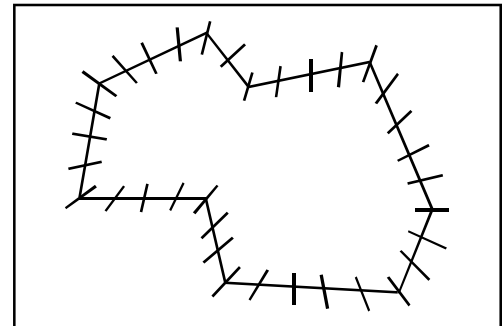
Why Should I Use Poly3D?

The main difference between FEM and BEM is that FEM requires the entire body be discretized into elements, while BEM, as suggested by its name, uses only elements on the boundary of the body. This means that if you have a three-dimensional object, FEM will use 3-D elements and BEM will use 2-D elements. If your object is a surface, you will need to create 2-D elements for FEM and 1-D elements for BEM. Why is this important? Less complex elements means a smaller system of simultaneous equations to solve, and ultimately, less computation time. Figure 1 shows the way FEM and BEM discretizes the same body.

Figure 1:
The basic difference between the Finite Element Method (FEM) and the Boundary Element Method (BEM) is the way it discretizes its elements.



Finite Element Method (FEM) uses nodal points and elements to approximate differential equations. The solution requires continuity and equilibrium across the elements.



Boundary Element Method (BEM) uses numerical integration over the boundary (surface) of the object which is subdivided into small segments.

Poly3D is a BEM, so let's look at some of the other characteristics of BEMs:

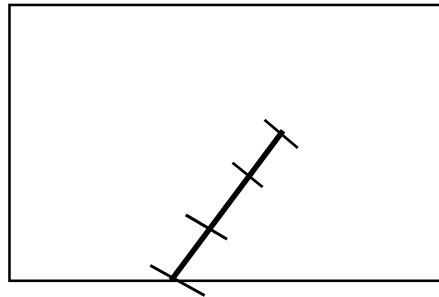
- Because BEM reduces every problem by one dimension, it is not appropriate for one-dimensional problems such as 1-D water flow.
- The BEM can assume boundaries at infinity, but the FEM cannot. In FEM programs, you must create an artificial boundary. This may require special efforts in order to minimize errors due to approximations.
- Because it only discretizes the boundaries, errors due to discretization and numerical approximations arise only on the boundaries.
- Because no approximations were made on interior areas, you can obtain values of variables at any specific interior point with very high resolution.
- It is necessary to specify only the bounding conditions, so it takes less time to prepare the data for input.

With all these advantages, BEM is well suited for modeling fractures (joints, faults, bedding planes, etc.), which, by their nature, are two-dimensional in a three-dimensional body.

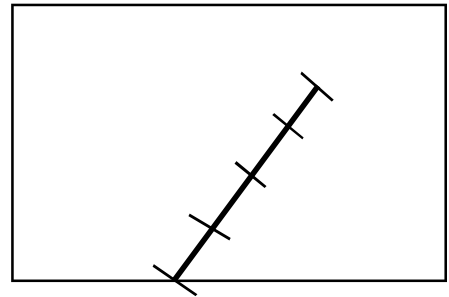
Numerical Modeling of Fractures

Previously, I said that the BEM is well-suited for modeling fractures. Let's look at an example to see why. Suppose you are looking at a vertical cross-section of the earth that crosses a fault as shown in Figure 2. With a BEM program, you can use an infinite boundary, so you only have to discretize the fault. If you are modeling fault propagation, you need only to add an element to the tip of the fault.

Figure 2: Modeling a fault with BEM, you can specify infinite boundaries and discretize only the fault. As the fault grows, you only need to add an element to the end of the fault.

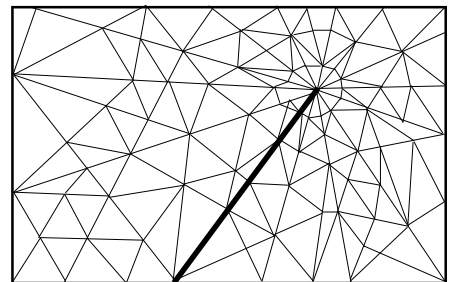
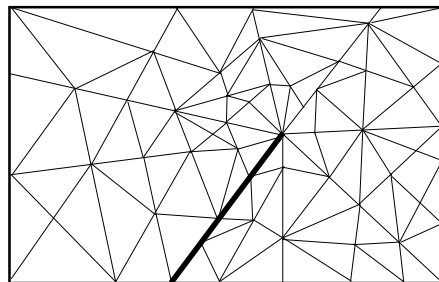


Initial Fault



Growing Fault

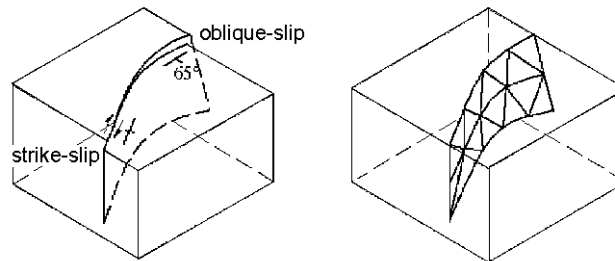
With FEM, you have to define an artificial boundary and discretize the entire area inside this boundary. As the fault grows, the area will have to be re-discretized.



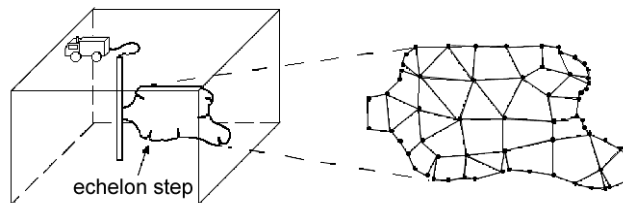
If you are working with the FEM, you have to define an artificial boundary which could reduce the accuracy of your results. You also need to discretize the entire area inside your artificial boundary, which might be time consuming for you or the computer. If you are modeling fault propagation, you need to re-discretize the entire area as the fault grows.

The polygonal elements in Poly3D are also well-suited for modeling complex surfaces with curving boundaries because:

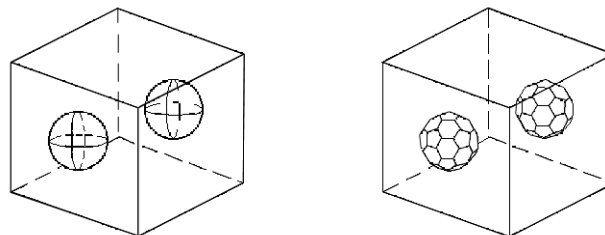
- ♦ Fault surfaces which change in both strike and dip can be meshed without creating gaps.



- ♦ Polygonal elements easily replicate the irregular boundary of a hydraulic fracture.



- ♦ A spherical void can be modeled by assembling hexagonal and pentagonal elements in the manner of a soccer ball.



These kinds of problems would be difficult or impossible to attempt using rectangular elements alone. For this reason, Poly3D represents a significant improvement over exiting rectangular element boundary element codes such as Dis3D.

What BEM Programs are Available?

So if BEM is so great, why wouldn't you use it all the time? Finite Element Modeling has been around for much longer than Boundary Element Modeling, so current FEM codes can incorporate material differences and non-linear rheologies. It is possible to incorporate these characteristics into BEM, but no BEM programs have been written with this capability yet.

The BEM codes that have been written are variable in their capabilities. Some of the 2-D and 3-D codes allow you to incorporate infinite boundaries and frictional elements, some can model fault propagation, and others vary in the shape of the faults. A few of the BEM codes that have been written are listed in Table 1 with their current capabilities.

Capabilities of BEM Codes

	Dimensions	Infinite Boundaries	Fracture Propagation	Frictional Elements	Fracture Shape
¹ Frac2D	2-D	X	X	—	—
² Fric2D	2-D	—	X	X	—
³ Dis3D	3-D	X	—	—	Rectangular
Poly3D	3-D	X	—	—	Polygonal

Table 2: BEM codes have different capabilities including incorporating infinite boundaries, fault propagation, friction, and fault shape.

¹Thomas, A.L. & Pollard, D.D. (1993). The geometry of echelon fractures in rock: implications from laboratory and numerical experiments. *Journal of Structural Geology* **15**, 323-334.

²Cooke, M. (1996). Frictional slip and fractures associated with faults and folds. Unpublished PhD thesis, Stanford University. (Includes opening-mode fracture propagation and opening-mode fracture initiation along frictional interfaces away from the tips).

³Erickson, L.L. (1986). A three-dimensional dislocation program with application to faulting in the earth. Unpublished M.S. thesis, Stanford University.

What Can't Poly3D Do?

Currently, Poly3D cannot model:

- fracture propagation
- frictional slip along interfaces
- elements with more than 40 sides
- material differences or layered materials
- non-linear rheologies
- variable remote boundary conditions

Perhaps future generations of BEM will incorporate these capabilities.