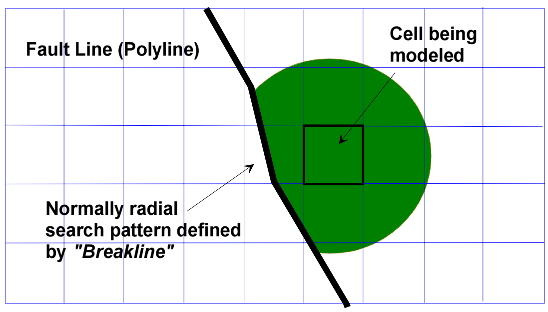
Technote:

Contouring with Fault Lines (Breaklines)

November, 2003
Intermediate

Modeling of topographic contours, or of subsurface features using TECHBASE, assumes that your control points represent a uniform surface with predictable transitions between randomly scattered data points. Following this assumption, it is a straightforward process to model this surface using one of the many modeling programs distributed by TECHBASE. However, if your values are from a surface interrupted by definable breaks, such as faults, stream valleys, ridge lines or pit crests, modeling becomes more complex.

Recognizing the need to more allow more flexibility in modeling surfaces, TECHBASE introduced their *Breaklines* option to allow the user to independently model different sets of data within a single project area. *Breaklines* is available in both the *KRIGE* and *INVERSE* modeling programs. *Breaklines* allows you to uniquely identify groups of data by using polylines that separate this data. When using breaklines to model cells, a cell on one side of the line can't "see" datapoints on the other side of the line. The *Breaklines* option works well for faults that displace units in either a strike-slip or dip-slip motion. Thrust faults, where rocks are thrust up and over each other, require different modeling techniques not addressed here.



Step 1: Create a Cell, Layer or Block Table

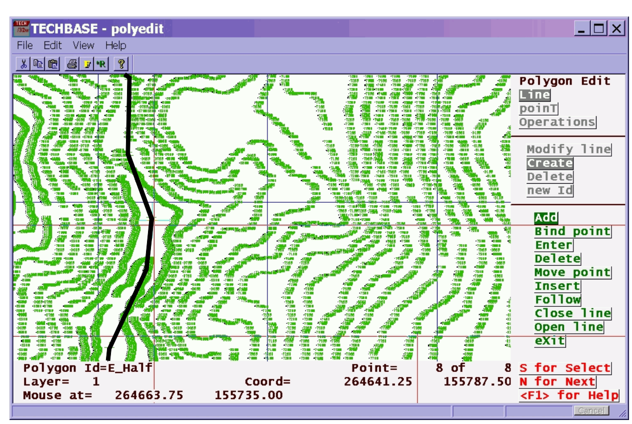
The first step in modeling your data using *Breaklines* is to create a *CELL* or *LAYER* table to store the results. Make sure the cells are small enough so that detailed features of your surfaces are preserved, yet large enough to be modeled in a reasonable time. You can also model your data into a block table using the *Breaklines* option, as described below.

Step 2: Determine the Best Modeling Method

Next you want to determine if Kriging or Inverse Distance is the best method of interpolating between your data points. As a general rule, the inverse distance algorithm (*Inverse)* works best with evenly spaced data that vary with distance. It assumes that data that are close together are more similar, and data that are far apart are less similar. The Kriging algorithm (*Krige)* assumes that data varies with distance and direction, according to a variogram calculated by the user using the *Vario* program.

Step 3: Define Polygons

Evaluate your data and draw polylines in *Polyedit* to identify the trace of faults or other features which interrupt data. For example, you might want to model data sets to the east and west of a river valley separately. A polyline, separating the two data sets, can be drawn in polyedit as shown to the right. Save the polygons you draw to a polygon file.



Step 4: Model Your Data Using the Breakline Polygons

The next step is to model your data into a cell table, using the polygons you defined in step 3. Enter the polygon file where prompted in the Inverse/Breaklines/Filename menu. This polygon file must be in TECHBASE polygon file format, and must contain at least one line or polygon. You may include as many polygons as you wish in this file.You can also point to a TECHBASE *POLYGON TABLE* in this menu.

You can also use the *Breaklines* feature when estimating into a block table. *Breaklines* will apply your polygon to your entire Block Table in a cookie cutter fashion. If you wish to work with only one level at a time, set filters on your block table so that you are only working with one level of blocks. For instance, if your block table (named Block) is set up such that blocks are 100 feet thick, and start at 5000 feet elevation, set a filter such that the block center (Block_zc) is equal to 5000. Model into these blocks using the breaklines feature, pointing to the polygons that apply to that level. Then change the filter so you are working with the next level up (i.e. Block_zc = 5100), model this level and so on.

Step 5: Contour your Results

Use *Gridcont* to contour the results. Note that contoured maps of data modeled using the Breaklines feature commonly show awkward transitions as you cross the breaklines. To minimize these transition zones on your final maps, use a small cell size for your model. Another method to make transitions more graphically pleasing is to draw the breaklines over the contours on your final map, using wide lines to hide these transition zones.

The example graphic below shows a modeled surface in an oil field transected by northeast trending faults. The use of *Breaklines* was instrumental in modeling this data as units were spatially offset along the faults making traditional modeling inaccurate. Note how faults were drawn in wide red lines on this map to minimize abrupt transitions across these structures.