

Current Affairs

A Window on Software Engineering

MS2 Engines as Stand-alone Programs

DTB, SGR, UNION, CLIP, and PCLIP - you've used them inside MS2, but these powerful engines are also available as separate, stand-alone programs. This article summarizes how each program works using simple examples.

Inside MS2, the various program switches and options are set automatically for you. To execute these programs individually from the command line, you must manually specify the various program switches and options for the task you want to accomplish. The advantage to using the programs separately outside of MS2 is that they can be used in batch mode and you don't have to fire up MS2. This suggests they could be run on a less powerful machine, with less powerful graphics capabilities than is necessary for running MS2. Nevertheless, after execution you will need to import the resulting data or shell file into MS2 in order to verify it for correctness.

If you execute each program from the command line without any arguments, the usage syntax will be displayed. Use the help, "-h", option for more detailed, specific information about how to run each of these programs, as shown in Figure 1.

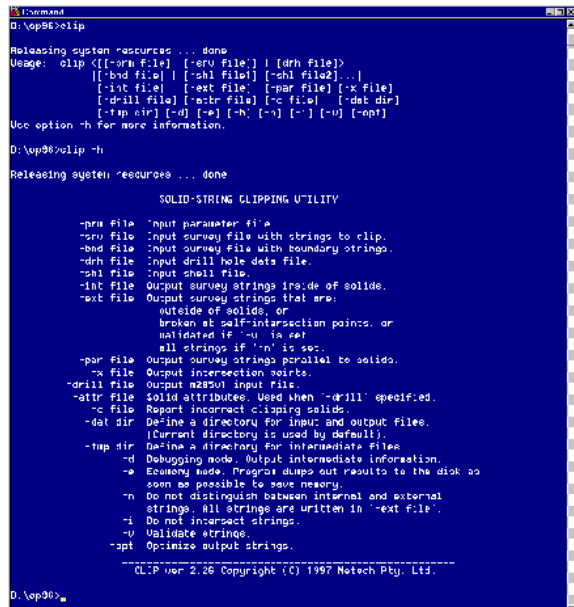


Fig. 1

To use these programs outside of MS2, your MS2 geometry objects must be exported to shell files. As stand-alone programs, these functions use shell files, not MS2 objects. Shell file format is a special ASCII file containing data structure information about your geometry object. Export data from MS2 to shell format via **Surface**→**Export**→**Shell file**. The file chooser will be displayed allowing you to name the new file.

DTB is the engine behind triangulation of survey data. Survey data can consist of points and strings which are used to create a mesh of triangles. There are four different setup options for this program: interactive mode, piping, command line, and run file.

In the example shown below, DTB is used to triangulate MS2 survey data to create a shell file. To use this program, you must already have a survey codes parameter file (param.cod) and an input survey file (current.srv). The program is executed using the following syntax to output a shell file (current.shl):

```
dtb +B -I -p param.cod -s current.srv
-o current.shl -F S -a
```

The "+B" switch tells DTB to not cross breaklines, "-I" to output the surface inside the boundary, "-F" for type of output format (S = shell format) and "-a" for ASCII format.

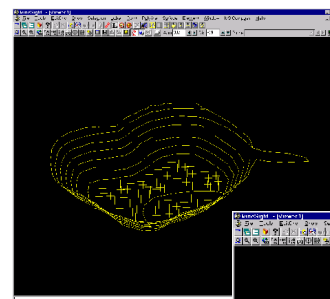
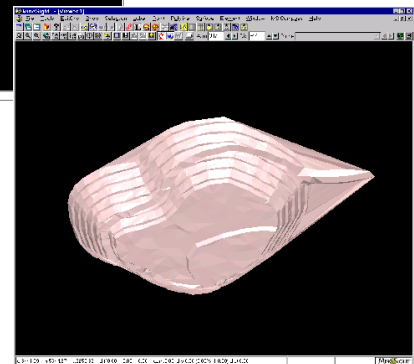


Fig. 2 Survey data as displayed in MS2.

Fig. 3 Resultant triangulated survey data shell file imported into MS2.



(continued on page 9)

(MS2 Engines as Stand-alone Programs continued from page 6)

SGR is used for surface gridding and volume calculation. This program calculates 2-D grids of triangulated surfaces and calculates volumes between two surfaces using gridding logic. There are three main working modes to this program; gridding a triangulated surface, simple cut and fill volumes between two surfaces, and cut and fill partials and level volumes between two surfaces.

In this example, the fill volume will be calculated for a pit. First, export both topography and the pit to shell files. In this case topography is the upper surface (topo.shl) and the pit is the lower surface (pit.shl). To calculate the volume between these two surfaces, the following syntax is used on the command line to create the ASCII text file, volume.txt: `sgr -topshl topo.shl -botshl pit.shl -bench bench.txt -xmin 1000 -ymin 4000 -nx 150 -ny 125 -dx 20 -dy 20 -sbx 1 -sby 1 -vol volume.txt`

The -xmin/-ymin, -nx/-ny, -dx/-dy, -sbx/-sby key word options are used in creating the grid used in the calculations. The values used, along with these key words, are the project's extents and block size.

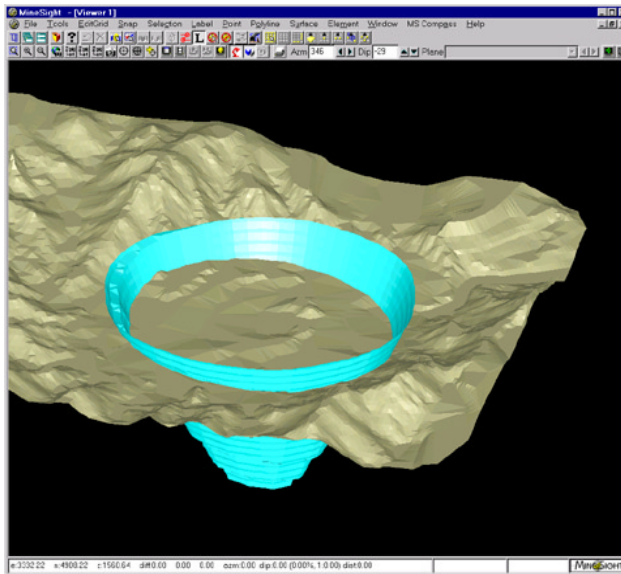


Fig. 4 Calculate the cut and fill volume between a pit and topography.

UNION is the engine used in solid-solid intersection. It is also used for finding holes and edges in solids as well as self-intersections, disassembling a shell consisting of many solids or surfaces into pieces, and for volume calculations.

The example below shows how to calculate the intersecting pieces of geometry between two 3-D solids. First, the two intersecting 3-D geometry objects must be exported to shell files (object1.shl and object2.shl). Using the syntax below, Union will output the piece of the object surface that occurs inside the other object to a shell file (out1.shl), and vice versa (out2.shl):

```
union -I1 object1.shl -I2 object2.shl
      -O1 mi out1.shl -O2 mi out2.shl
```

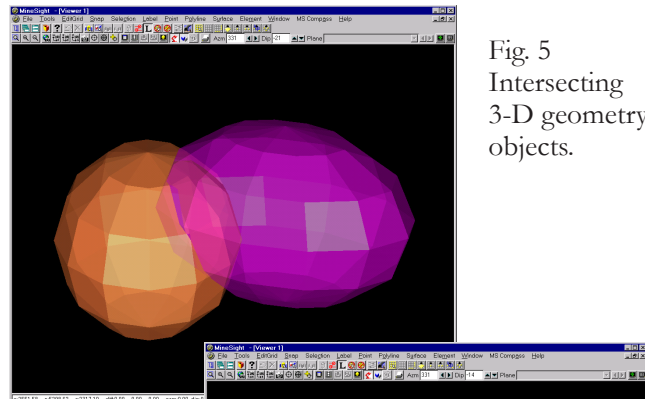


Fig. 5 Intersecting 3-D geometry objects.

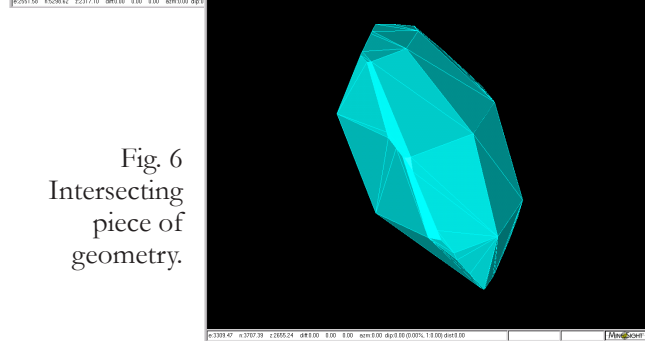


Fig. 6 Intersecting piece of geometry.

Import the resulting shell files (out1.shl and out2.shl) into MS2 to merge them, creating one resulting 3-D geometry object.

CLIP is used to clip strings against 3-D triangulated solids or surfaces or 2-D polygons/polylines (solid-string clipping utility). In particular, this program is used to spear 3-D geometry objects with drillholes and code the intervals. This complex program has eight different optional modes: 3-D solid/string clipping (and determination of inside/outside), 2-D polygon/string clipping (also includes determination of inside/outside), self-intersection of strings within the same file, string validation, solid validation, string reconstruction, and string optimization.

(continued on page 10)

(MS2 Engines as Stand-alone Programs continued from page 9)

The example discussed below shows how to code drillhole intervals that pierce a 3-D solid. First, the drillhole data must be dumped in m219v1 format (dat219.oa). You can use procedure p21901.dat in MS2Compass™. Next, export your 3-D solid to shell format (gradshell.shl). Then use the following syntax to run program clip to create a data file in m205v1-input format (dat205.x):

```
clip -drh dat219.oa -shl
gradshell.shl -drill dat205.ia
```

After running CLIP, use procedure p20501.dat to load the data in file dat205.ia to a specific code item in your drillhole file.

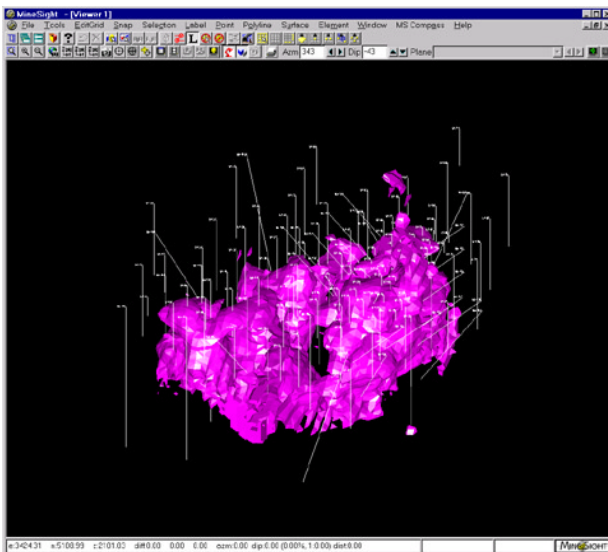


Fig. 7 3-D solid pierced by drillholes.

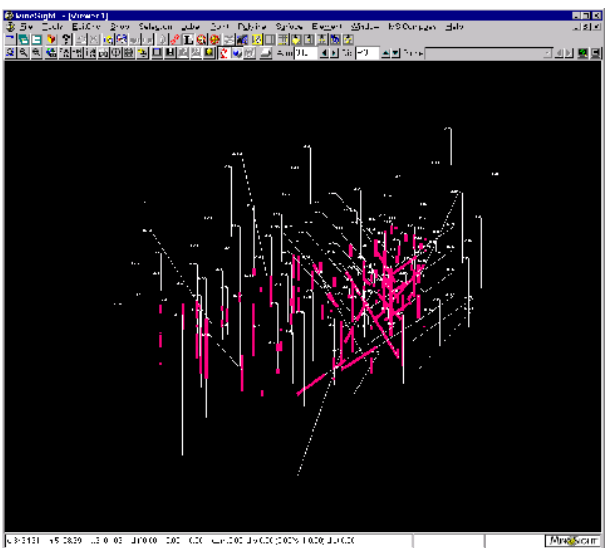


Fig. 8 Resulting drillhole intervals, coded where they intersect the 3-D solid.

PCLIP is the polygon-polygon intersection program that allows polygons in 3-D space to be intersected with other polygons to produce the difference, intersection, or union of the polygons in the X-Y (2-D) plane.

The following example shows how three closed polylines can be combined into one object. Since two of the polylines overlap one another, PCLIP combines the intersecting portions. However, after execution there are now two polylines which are assigned the same geometry code, hence the three original objects get combined into one. First, export all three features to one VBM (ASCII) file (geomall.vbm). Then use the following syntax to have pclip create a VBM (ASCII) file of the combined polylines (combine.vbm):

```
pclip -vbm c geomall.vbm -union
combine.vbm
```

The “c” switch stands for Combine and the “-union” option tells PCLIP to calculate the union of polygons.

Fig. 9 Three closed polylines, each with unique material and VBM feature codes.

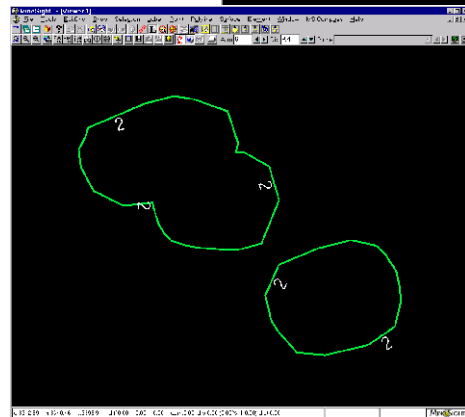
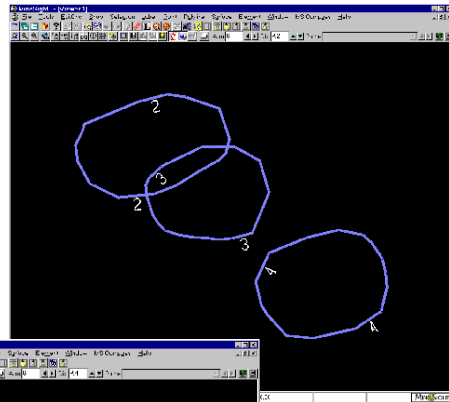


Fig. 10 Resulting combined polylines.

These powerful engines come to us from Metech, our Australian affiliate. There are procedures available for each of these engines. Please contact Mintec Technical Support to obtain a copy of each of these programs as well as even more complete details regarding the various program options and command line instructions and syntax.