

# A Short Note on the Integration of GSLIB and GOCAD

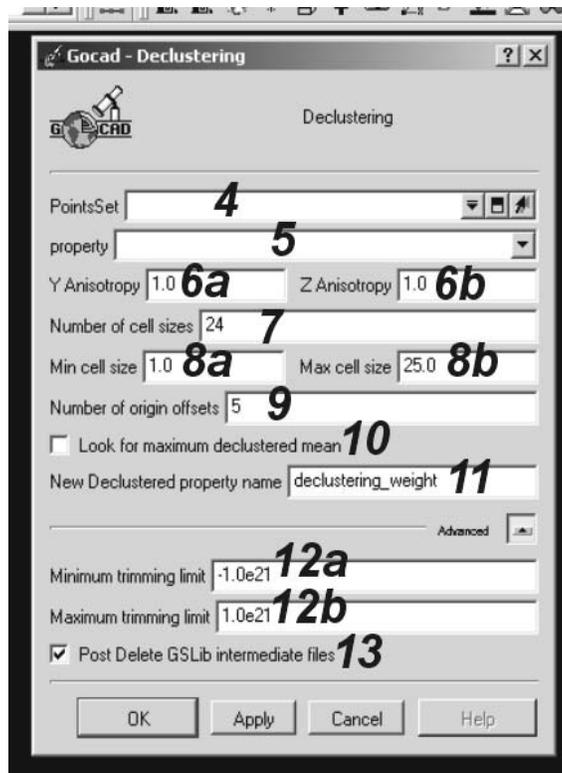
W. Glenn ([wglenn@ualberta.ca](mailto:wglenn@ualberta.ca))  
University of Alberta, Edmonton, Alberta, CANADA

*This short note describes some of the work undertaken by Will Glenn as a summer intern supported by T-Surf to incorporate some of the latest GSLIB/CCG codes in gOcad. Please contact C. V. Deutsch or T-Surf to discuss implementation at your site.*

## Users guide to implement GSLib dll in Gocad

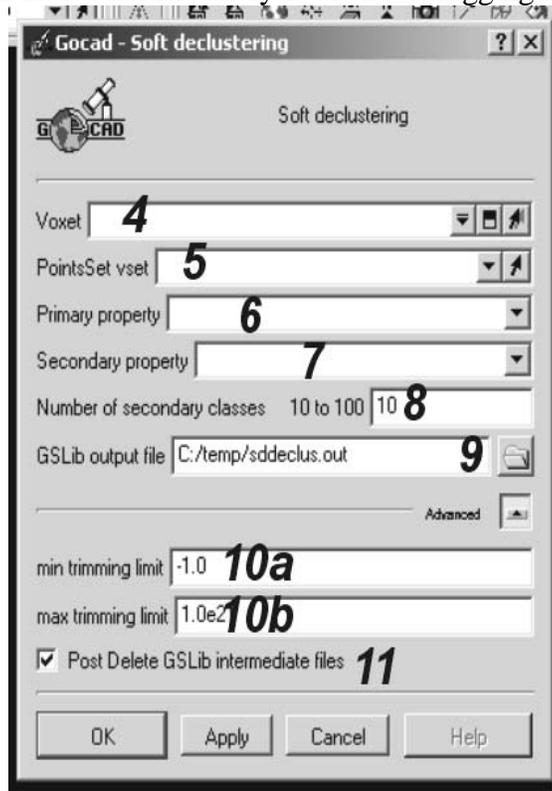
Here are some general guidelines related to using the GSLib DLL in gocad. Note that this approach requires a C:/temp folder to hold GSLib intermediate files.

Cell declustering (declus) - see below - requires a loaded Pointset with a property and assigns the declustering weights to each point in Vset. The steps are summarized by (1) display the Vset and property in the camera, (2) select declus from the GSLib algorithms menu, (3) the declustering dialog window will pop up, (4) enter the name of the Pointset, (5) select the Pointset property, (6) enter the Y and Z anisotropy variables, (7) number of cell sizes to consider in the operation, (8) minimum and maximum cell size, (9) number of origin offsets (noff), (10) select the check box for maximum declustered mean, (11) choose a suitable name for the declustered weight, (12) enter the minimum and maximum trimming limits, and (13) select the check box for Post deleting intermediate files (this is recommended if no further file analysis or file debugging is needed).



Soft data declustering (sddeclus) requires a loaded Pointset with a property and a loaded Voxet with cell properties. The result is a file with a representative histogram. The steps are

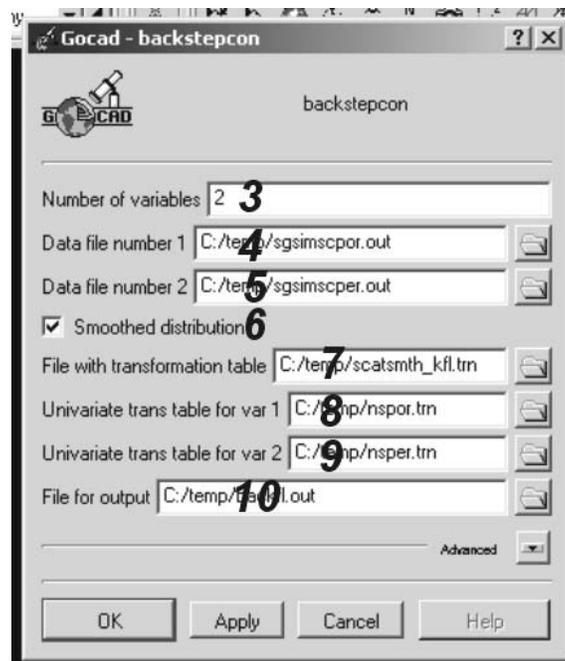
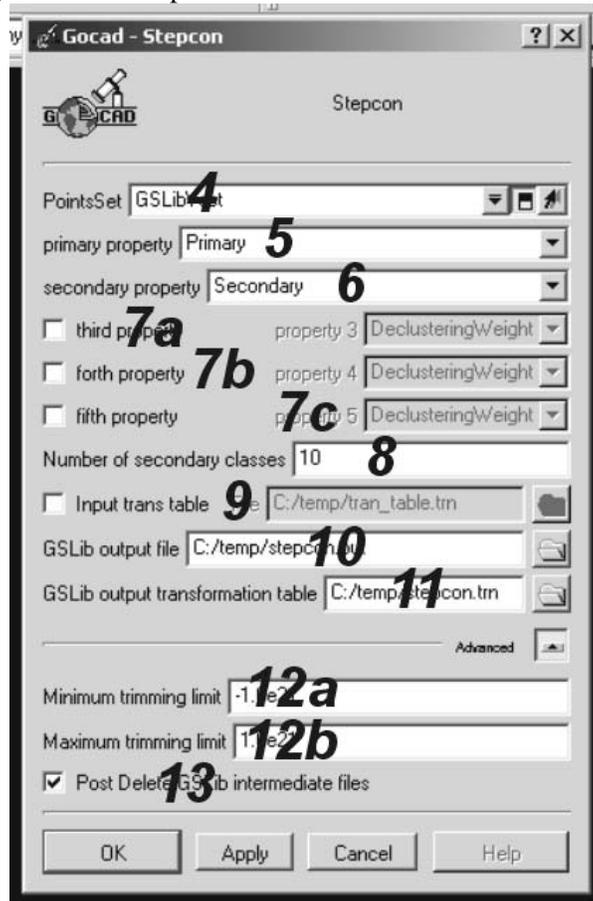
summarized by (1) display the Pointset with property and Voxet with cell property in the camera, (2) select sdeclus from the GSLib algorithms menu, (3) the soft declustering dialog window will pop up, (4) enter the name of the Voxet, (5) enter the name of the Vset, (6) enter the name of the Vset property, (7) enter the name of the Voxet property, (8) enter the number of secondary classes to consider, (9) GSLib sddeclus output file, (10) minimum and maximum trimming limits of data, and (11) post deleting intermediate files. This is recommended if no further file analysis or file debugging is needed.



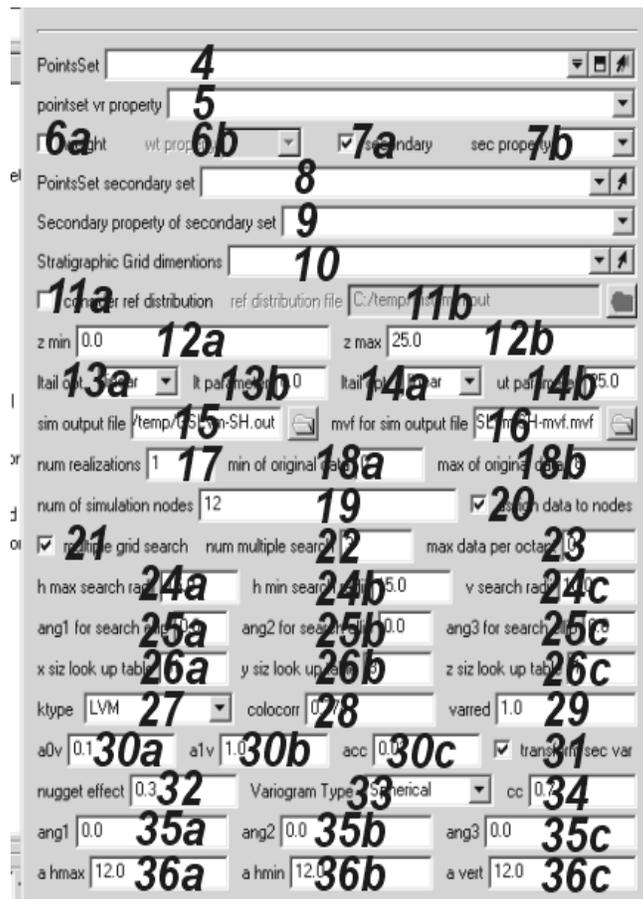
Stepwise conditional transformation (stepcon) requires a loaded Pointset with two or more properties. The result is a file with new columns for the transformed values and another file with the output transformation table. The steps are summarized by (1) display the Pointset with property and Voxet with cell property in the camera, (2) select stepcon from the GSLib..Algorithms menu, (3) the stepcon dialog window will pop up, (4) enter the name of the pointset, (5) enter the primary property, (6) enter the secondary property, (7) if there are other properties of the pointset that must be used, select the properties accordingly, (8) enter the number of secondary classes, (9) an optional transformation table file can be considered in the algorithm by indicating its location, (10) the GSLib output file containing the original data set with additional columns with the new conditioned data, (11) the GSLib output file containing the transformation table, (12) minimum and maximum data trimming limits, and (13) post deleting intermediate files. This is recommended if no further file analysis or file debugging is needed.

The reverse of the stepwise conditional transformation (backstep) requires two data files and the transformation data. The result is the back transformed values. The steps are summarized by (1) select stepcon from the GSLib algorithms menu, (2) the backstepcon dialog window will popup, (3) enter the number of variables, (4) enter data file number 1, (5) enter data file number 2, (6) select check box for smoothed distribution, (7) enter the input transformation table, (8) enter the

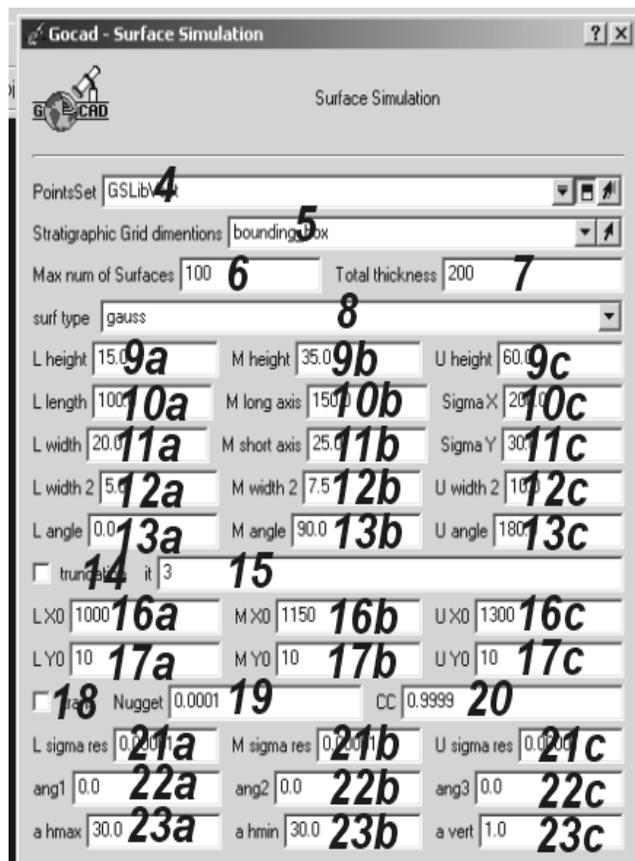
univariate transformation table file for variable 1, (9) enter the univariate transformation table file for variable 2, and (10) select the output file.



The sequential Gaussian simulation program with self healing (sgsim\_sh) requires two pointsets and generates a new SGrid properties for each cell, a simulation output file, an MVF for simulation output file. The steps are summarized by (1) display the two Pointsets with property and SGrid in the camera, (2) select sgsim\_sh from the GSLib algorithms menu, (3) the sgsim\_sh dialog window will pop up, (4) enter the name of the primary Pointset, (5) enter the primary property, (6) a and b: select the weight checkbox if there is a weight property in the primary Pointset to be used in sgsim\_sh and enter the property name, (7) a and b: select the secondary checkbox if there is a secondary property in the primary Pointset to be used in sgsim\_sh and enter the property name, (8) enter the name of the secondary Pointset, (9) enter the secondary property to be used from the secondary pointset, (10) enter the Sgrid, (11) a and b: select the checkbox to consider the reference distribution file, (12) a and b: enter the minimum and maximum Z values, (13) a and b: enter the lower tail option and parameter, (14) a and b: Enter the Upper tail option and parameter, (15) enter the simulation output file, (16) enter the mvf file for simulation output, (17) enter the number of realizations, (18) a and b: Enter the minimum and maximum original data for simulation, (19) enter the number of simulation nodes, (20) select the checkbox to assign data to nodes, (21) select the checkbox to perform a multiple grid search, (22) enter the number of multiple searches, (23) enter the maximum data per octant (0=not used), (24) a, b, and c: - maximum search radii (hmax,hmin,vert), (25) a, b, and c: -angles for search ellipsoid, (26) a, b, and c: -size of covariance lookup table, (27) kriging type, (28) colocorr, (29) varred, (30) a, b, and c: a0v, a1v, acc, (31) select checkbox to transform secondary variable, (32) enter the nugget effect, (33) select the Variogram type, (34) enter the cc parameter, (35) a, b, and c: enter the angle parameters, and (36) a, b, and c: enter the a\_hmax, a\_hmin, and a\_vert parameters. Note 1A: the SGrid must be an translation of the original XYZ grid.



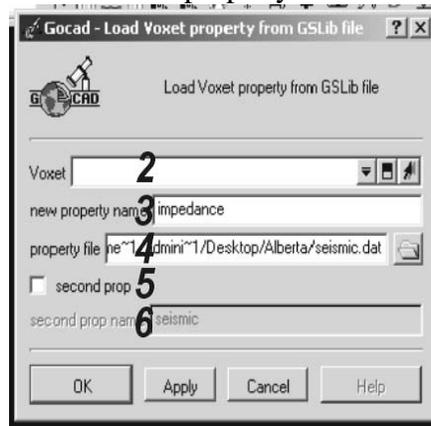
Surface simulation (*surfsim*) requires a loaded Pointset generates four output files with the surface parameters. The steps are summarized by (1) display the poinset and Sgrid in the camera, (2) select surfsim from the GSLib ..Algorithms menu, (3) the surfsim dialog window will pop up, (4) enter the name of the poinset, (5) select the Sgrid to use for the algorithm, (6) enter the Maximum number of surfaces, (7) enter the total thickness, (8) select the surface type –gauss-helmet-ellipse, (9) a, b, and c: Enter the L, M, and U height of surface, (10) a, b, and c: (simple) Enter the L length,(ellipse) M long axis, and (gauss) Sigma X, (12) a, b, and c: (simple) Enter the L width, (ellipse) short axis, and (gauss) Sigma Y, (13) a, b, and c: (simple)(not for ellipse & gauss) L, M, U width 2, (14) select box for truncation, (15) select box for it variable, (16) a, b, and c: if trunction(14), Enter the L, M, and U of X0, (17) a, b, and c: if trunction(14), Enter the L, M, and U of Y0, (18) select box for itrans variable, (19) enter the nugget effect, (20) enter the cc variable, (21) a, b, and c: enter the L, M, and U Sigma of residual, (22) a, b, and c: enter angle\_1, angle\_2, and angle\_3 parameter, (23) a, b, and c: enter the a\_hmax, a\_hmin, and a\_vert variables. Note 2A: the SGrid must be a translation of the original XYZ grid.



Create a Voxet from GSLIB takes a GSLIB data file and generates a file for GOCAD. The steps are summarized by (1) select Create\_Vset\_from\_GSLib from the GSLib-tools menu, (2) enter an appropriate name for the Pointset, (3) select a GSLib data file from the file selector, (4) check checkbox if GSLib data file contains a well prefix (ie. Is the first column in the GSLib data file a well number?), and (5) select checkbox if the GSLib data file contains a column for Z values (ie. Is the file in 3-D or 2-D format?).



Load a Voxet. The steps are summarized by (1) select Load Voxet Property from the GSLib-tools menu, (2) choose the Voxet that will have the additional property, (3) give the new property a name, (4) choose the GSLib file<sub>A</sub> that contains the property, (5) select the checkbox if the GSLib file has a second property that is to be included in the Voxet, and (6) depending on step 5, Enter the second property.



Load a SGrid. The steps are summarized by (1) select the load\_sgrid\_property from the GSLib..tools menu, (2) choose the SGrid that will have the additional property, (3) enter a property name, (4) choose the GSLib file that contains the property, (5) select the checkbox if the GSLib file has a second property that is to be included in the SGrid, and (6). depending on step 5, Enter the second property name



## How to integrate the GSLIB DLL in MSDEV C++:

This is an example implementation of the IDECLUS function using the GSLIB.dll and GSLIB.lib.

- 1 download and install the VFRUN65AI.exe from:  
<http://www.compaq.com/fortran/kb/q1023.html>  
this exe will install the necessary dlls to run the DECLUS.dll or any other fortran dll created by visual fortran in C++ code.
2. Place the GSLIB.dll in the system path
3. In the MSVisual Studio environment, update include the GSLIB.lib file in the Project->settings->Link(tab)->Object/library modules: line
4. Use the sample C++ code to run the IDECLUS function inside the implementation files  
the IDECLUS function requires a .dat and .par file inside a sample .cpp file:

```
#include<string.h>
...
char instring[50];
char outstring[50];
strcpy(instring,"C:/temp/declus.par");
strcpy(outstring,"C:/temp/data.out");
    /* Call Fortran routine -pass length of outstring explicitly */
int warnint = IDECLUS(instring,strlen(instring),outstring,50);
    // if warnint = 0      -Normal operation
    // if warnint = 1      -Missing parameter or input file
    // if warnint = 2      -Error in parameter file
    // if warnint = 9      -Extraneous Unknown error
...

```

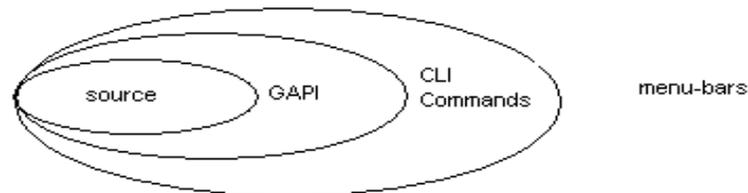
inside a sample .h file:

```
...
extern "C"    // must declare extern "C" to avoid C++ name mangling
              // IDECLUS being the name of the function
{
    __declspec(dllimport)int __stdcall IDECLUS
        (char *STR_IN,
         int STR_IN_LEN,
         char *STR_OUT,
         int STR_OUT_LEN);
}

```

## General Gocad Plugin information

For future development of CCG Gocad plug-ins. The CCG Gocad plug-in was developed using the three stage implementation. Each plug-in element begins with the source stage. The source files are interfaced with the API stage. A single CLI command encapsulates the entire process of calling the API and source objects.



Specific Libraries were tailored for the CCG Plug-in. the libraries are InputOutput, and Algoritihms. The InputOutput library contained most of the file reading and writing. The Algoritihms library contained all of the CCG\_DLL.dll functionalities and GSLib parameter file creations. The source contains the GSLib function calls and extern”c” declarations. It also contains the file writing procedures for the GSLib intermediate files.

The GAPI contains the calls to the source classes and performs error checks. It also prepares Gocad to receive properties and contains calls to update properties. The GAPI classes that the CCG has developed in summer 2001 have contained only public static functions ( similar to Gocad api convention ) that mainly return Boolean values as output. Inside the GAPI static functions the main source classes are called. In the CCG Gocad plug-in each library that was created, there was a corresponding GAPI file used to call the library’s code.

The CLI and XCLI simplified the creation of the user interface windows. Inside the CLI commands the CCG used file selectors, Object selectors, property selectors, parameter entry fields, and Boolean variables to create a dialog to users. The CLI commands call the GAPI static functions.

The menu-bars section defines the menu item names, CLI calls, and ordering of menu layout. The Gocad plugin takes care of the implementation of the menu-bar, so developers need only worry about the ordering of the menu items.

## Instructions on how to use each GSLib dll function in Gocad Using specific parameters and files.

### declus:

Using this parameter file:

```
START OF PARAMETERS:
./data/cluster.dat      \file with data
1  2  0  3              \ columns for X, Y, Z, and variable
-1.0e21  1.0e21        \ trimming limits
declus.sum              \file for summary output
declus.out              \file for output with data & weights
1.0  1.0                \Y and Z cell anisotropy (Ysize=size*Yanis)
0                       \0=look for minimum declustered mean
(1=max)
24  1.0  25.0          \number of cell sizes, min size, max size
5                       \number of origin offsets
```

Steps:

1. In GSLib..tools.. Create VSet from GSLib,
2. Enter the file cluster.dat as the GSLib file.
3. make sure the check boxes are unchecked because cluster.dat does not have columns for well id or Z
4. In GSLib..Algorithms..declus
5. choose the Pointset previously entered and enter "Primary" property in the property box
6. Use the default parameters(they are the same as the above)

### stepcon:

Using this parameter file:

```
START OF PARAMETERS:
data.dat                - file with data
2                       - number of variables to transform
3  5                    - columns for variable transformation-
primary and Secon
-1.0e21  1.0e21        - trimming limits
10                     - number of classes
1                       - input transformation table,
yes=1,no=0
scatsmth_k.trn          - file with transformation table
stepcon.out             - file for output
stepcon.trn             - file for output transformation table
```

Steps:

1. In GSLib..tools..Create\_Vset\_from\_GSLib,
2. Enter the file cluster.dat as the GSLib file.
3. Make sure the check boxes are unchecked because cluster.dat does not have columns for well id or Z
4. In GSLib..Algorithms..stepcon
5. choose the Pointset previously entered and enter "Primary" in the primary box and "Secondary" in the secondary box.
6. Use the default parameters(they are similar to the ones above).
7. Out put files should appear in the C:/temp folder.

### ssdeclus:

Using this parameter file:

```
START OF PARAMETERS:
pairs.dat               -file with paired calibration data
1  2                    - columns for secondary and primary
-1.0  1.0e21           - trimming limits
```

```

seismic.dat          -file with secondary data
1                    - column for secondary variable
sddeclus.out        -file for representative histogram
50                   -number of sec. classes to use (10-100)

```

Steps:

1. In GSLib..tools..Create\_Vset\_from\_GSLib,
2. Enter the file welldata.dat as the GSLib file.
3. make sure the check boxes are checked because welldata.dat does have columns for well id and

Z

4. In Objects right click on Voxet ..New..From\_Objects\_cage.
5. give the Voxet a name, set Scale W dimation by 1, set nu= 81, nv= 81, nw= 78
7. Populate the Voxet nodes with properties from the file "seismic.dat". This file contains only one column with the property "impedance". The file data will expire before filling all Voxet nodes.
8. go to Voxet..Attributes.. select the Object and make the "volume" atoms visible.(this will slow things down!)
6. In GSLib..Algorithms..sddeclus
7. Put in the above Voxet, Pointset, and "Porosity" property from Pointset in primary property and "impedance" in the secondary property.
8. use the rest of the default parameters (similar to the above parameter file) uncheck Post\_delete\_par\_file to view the intermediate GSLib files (ie. the pairs.dat file).

#### backstep:

Using this parameter file:

START OF PARAMETERS:

```

2                    - number of variables
sgsimscpor.out      - data file number 1
sgsimscper.out      - data file number 2
1                    - smoothed distribution, 1=yes, 0=no
scatsmth_kfl.trn    - file with input transformation table
nspor.trn           - univariate transformation table for variable 1
nsper.trn           - univariate transformation table for variable 2
backfl.out          - file for output

```

Steps:

- Not much to this Program,  
Just put in the files and Select where the output files go!

#### sgsim\_sh:

Using this parameter file:

START OF PARAMETERS:

```

./data/cluster.dat  -file with data
1 2 0 3 0 4         - columns for X,Y,Z,vr,wt,sec.var.
-1.0e21             - trimming limits
1.0e21              -transform the data (0=no, 1=yes)
1                    - file for output trans table
sgsim.trn           - consider ref. dist (0=no, 1=yes)
0                    - file with ref. dist distribution
histsmth.out        - columns for vr and wt
1 2                 - zmin,zmax(tail extrapolation)
0.0 25.0            - lower tail option, parameter
1 0.0               - upper tail option, parameter
1 25.0              -debugging level: 0,1,2,3
1                    -file for debugging output
GSLvm-SH.dbg        -file for simulation output
GSLvm-SH.out        -mvf file for simulation output mvffl
GSLvm-SH-mvf.mvf    -number of realizations to generate
1                    -nx,xmn,xsiz
25 0.5 2.0         -ny,ymn,ysiz
25 0.5 2.0         -nz,zmn,zsiz
1 0.5 1.0

```

```

123065          -random number seed
0          8    -min and max original data for sim
12          -number of simulated nodes to use
1          -assign data to nodes (0=no, 1=yes)
1          3    -multiple grid search (0=no, 1=yes),num
0          -maximum data per octant (0=not used)
15.0 15.0 10.0 -maximum search radii (hmax,hmin,vert)
0.0 0.0 0.0   -angles for search ellipsoid
31 31 1       -size of covariance lookup table
2 0.774 1.0   -ktype: 0=SK,1=OK,2=LVM,3=EXDR,4=COLC
1 0.1 1.0 0.02 -self healing (0=no, 1=yes)
../data/ydata.dat - file with secondary variable
(LVM,EXDR,COLC variable
4              - column for secondary variable
1              - transform secondary variable (0=no,
1=yes)
1 0.3         -nst, nugget effect
1 0.7 0.0 0.0 0.0 -it,cc,ang1,ang2,ang3
12.0 12.0 12.0 -a_hmax, a_hmin, a_verta

```

Steps:

1. In GSLib..tools..Create\_Vset\_from\_GSLib,
2. Enter the file cluster.dat as the GSLib file.
3. make sure the check boxes are unchecked because cluster.dat does not have columns for well id

or Z

4. In GSLib..tools..Create\_Vset\_from\_GSLib,
5. Enter the file ydata.dat as the GSLib file. MAKE sure the name is not the same as above!!
6. un-check check boxes because ydata.dat does not have columns for well id or Z

7. create an SGrid .. New ..From\_Step\_Vectors . put origin at 0, 0, -0.5 to encapsulate the data, also set nu=51, nv=51, nw= 2 (these are the number of nodes) and leave the property\_cell\_centered
8. Run the GSLib..Algorithms..sgsim\_sh picking "Primary" as the pointset\_vr\_property, and "Secondary" as the sec\_property. Also, pick the secondary pointset and its "Secondary" property and proper SGrid.

9. DELETE the OLD properties if running again to avoid the Gocad database from rejecting the additional properties with identical names

### surfsim:

Using this parameter file:

START OF PARAMETERS:

```

1          -number of realizations
cond1.dat   -input file of conditioning data
cond1_.dat  -output scaled conditioning data
w2000C1idx.out -output file of the index
w2000C1thk.out -output file of the thickness
w2000C1par.out -output of the parameters
1500 0.5 1.0 -nx,xmn,xsiz
20 0.5 1.0   -ny,ymn,ysiz
100 0.5 1.0  -nz,zmn,zsiz
100          -Maximum no. of surface
200          -total thickness
69069       -seed of random number
3          -surface type (1-Helmet 2-Ellipse 3-gauss)
15.0, 35.0, 60.0 -L,M,U height of a surface
100.0,150.,200.0 -L,M,U length (Simple), long axis (Ellipse),
SigmaX (Gauss)
20.0, 25., 30.0 -L,M,U width (Simple), short axis (Ellipse),
SigmaY (Gauss)

```

```

5.0, 7.5,10.0          -L,M,U width 2 (Simple) (not for Ellipse &
Gauss)
0.0,90.0,180.0        -L,M,U angle
0                      -truncation flag
1000,1150,1300        -L,M,U of X0 if trunFlag = 0
10, 10, 10            -L,M,U of Y0 if trunFlag = 0
-----
0                      -itrans
0.00001, 0.00001, 0.00001  -L,M,U sigma of residual
1    0.0001              -nst, nugget effect
3    0.9999 0.0    0.0    0.0  -it,cc,ang1,ang2,ang3
          30.0  30.0    1.0  -a_hmax, a_hmin, a_vert

```

steps:

1. In GSLib..tools..Create\_Vset\_from\_GSLib,
2. Enter the file cond.dat as the GSLib file.
3. make sure the check box for well id is unchecked because cond1.dat does not have a column for well id
4. create an SGrid from right clicking on SGrid ..New..From\_Step\_Vectors. Name the SGrid and set the origin to x=0.5, y=0.5, z=0.5, and set nu = 1500, nv= 20, and nw = 100
5. Run GSLib..Algorithms..surfsim using the default parameters