UNIVERSITY OF ALBERTA
Department of Civil and Environmental Engineering
School of Mining and Petroleum Engineering
Centre for Computational Geostatistics

CCG Software Catalogue

Chad Neufeld and Clayton V. Deutsch

September 2007

CCG Software Catalogue

CCG Software Catalogue

Chad Neufeld and Clayton V. Deutsch

Preface

Geostatistics provides a philosophical approach and specific tools for the spatial modeling of heterogeneity and the quantification of uncertainty. Teaching, learning and applying geostatistics is unavoidably linked to software. This catalogue contains a variety of programs that have been developed by different people at different times for different purposes; it is a real hodgepodge.

In all cases, the programs were developed in an academic non-commercial setting and the authors are distributing the software with no restrictions. This does not mean that the source code is given to everyone. The Centre for Computational Geostatistics (CCG) is supported by industry sponsors who expect some benefits for their membership fees. Preferential access to the source code of programs developed at CCG is one of those benefits.

The 39 programs of GSLIB (Deutsch and Journel, 1992/1998) brought together most of the important geostatistical tools as of 1991. Programs were left out. New programs have been developed in the last decade. A number of the programs documented here are derivatives of various vintages of GSLIB code. The aim of GSLIB was to unify geostatistical code in a consistent style; the aim here is to collect relevant software regardless of style or vintage. This is a catalogue of different programs and not consistent programs.

This compilation is neither a textbook nor a comprehensive user's guide. Most of the programs will do what you ask, but not necessarily what you want. Bugs will be fixed from time to time and known deficiencies will be documented. We appreciate feedback and bug reports, but no support is provided. The disclaimer for GSLIB remains relevant: *these programs are distributed in the hope that they will be useful, but without any warranty. We accept no responsibility to anyone for the consequences of using them or for whether they serve any particular purpose or work at all.*

This software was written by a number of people over the years and the authors are acknowledged conspicuously in the source code and in the documentation.

Contents

Preliminaries

Most of these programs were fashioned after the GSLIB programs. The format of the data files and some general description is contained here so that it does not have to be repeated with each program. The GSLIB book is a valuable resource and should be reviewed for a more complete description of many programs and the underlying algorithms (Deutsch, C.V. and Journel, A.G., *GSLIB: Geostatistical Software Library and Users Guide*, Oxford University Press, New York, second edition, 1997, 369 pages) [10]. It can be purchased from any online bookseller such as amazon.com.

Grid Specification

Regular grids of data points or block values are often considered as input or output. The conventions used throughout GSLIB are:

- The X axis is associated to the east direction. Grid node indices ix increase from 1 to nx in the positive X direction,
- The Y axis is associated to the north direction. Grid node indices iy increase from 1 to ny in the positive Y direction,
- The Z axis is associated to elevation. Grid node indices iz increase from 1 to nz in the positive Z direction,

The user can associate these three axes to any coordinates system that is appropriate for the problem at hand. For example, if the phenomenon being studied is a stratigraphic unit, then some type of stratigraphic coordinates relative to a marker horizon could make the most sense.

As shown in Figure 1, the coordinate system is established by specifying the coordinates at the center of the first block (*xmn, ymn, zmn*), the number of blocks/grid nodes (*nx, ny, nz*), and the size/spacing of the blocks/nodes (*xsiz, ysiz, zsiz*).

Data Files

The data file format is sometimes refered to as a simplified Geo-EAS format. The data are flat ASCII files with a header and data in free format. The header consists of (1) a title line, (2) the number of variables (an integer on the second line), (3) a one-line description of each variable, then (4) all the variables per line. This format is very simple, see Figure 2.

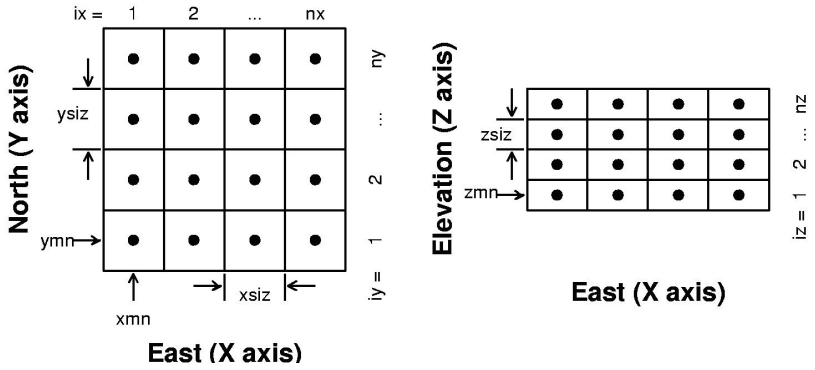

Figure 1: GSLIB grid definition.

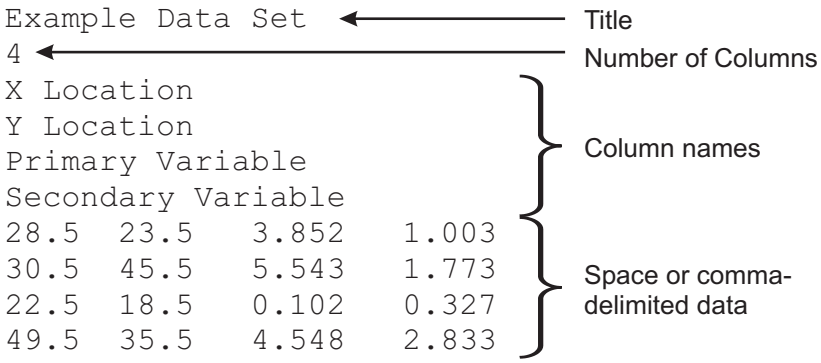

Figure 2: Example data file.

There are some common pitfalls. The data values should be separated by spaces or commas; tab characters are not always recognized. Missing values should be specified as large negative or large positive values — they should **not** be left as blanks. No alphanumeric characters are allowed in the data values: all data IDs should be replaced with integer or real numbers. There should not be multiple blank lines at the bottom of the data file — a read error will be reported.

A special ordering is used to store a regular grid. This avoids the requirement of storing node coordinates or grid indices. The ordering is point by point to the east, then row by row to the north, and finally level by level upward, i.e., X cycles fastest, then Y, and finally Z. The index location of any particular node ix, iy, iz can be located by:

$$index = ix + (iy - 1) \cdot nx + (iz - 1) \cdot nx \cdot ny$$

Given the above one-dimensional index location of a node the three coordinate indices can be calculated as:

$$iz = 1 + int((index - 1)/nx \cdot ny)$$

$$iy = 1 + int((index - (iz - 1) \cdot nx \cdot ny)/nx)$$

$$ix = loc - (iz - 1) \cdot nx \cdot ny - (iy - 1) \cdot nx$$

When multiple realizations are in a data file, they are stored one after another. The `addcoord` program from GSLIB will extract a particular realization and add the X, Y, Z coordinates to the gridded values.

Parameter Files

The parameters are also stored and read from a flat ASCII file. Following is an example. All lines before "START" are skipped; they can be used for notes. The little description to the right side (often starting with a "-" or a "́) are not required — they make it a little easier to set the parameters. The ordering of the parameters is important.

```
                  Parameters for RELOCATEDATA
                  ***************************

START OF PARAMETERS:
Database.dat             -file with data
2  3  0                  -  columns for X, Y, Z
300  490050.  100.       -nx,xmn,xsiz
350 6200050.  100.       -ny,ymn,ysiz
  1       0.    1.       -nz,zmn,zsiz
relocatedata.out         -file for output
```

Data files are specified by name and variables are specified by column numbers (the integer position in the file); note the first two lines (after START) in the parameter file above. The nine parameters required for a 3-D grid are normally specified by three lines; see the three lines after the data specification. The ordering of the parameters is program dependent, but there is often some structure.

Output Files

The output files are almost always ASCII data files or PostScript graphics files. The PostScript files can be previewed with gsview, copied to the clipboard, or reformatted to any of the common image formats.

Memory Allocation in Fortran

Prior to 1993 fortran did not support dynamic memory allocation. All fortran programs prior to 1993 required arrays to be dimensioned in the code, compiled, and then used for the problem at hand. This meant that all of the GSLIB programs written for the first edition did not use dynamic memory allocation and the user was responsible for changing the array dimensions in the program for each case.

Ideally all of the programs contained within this catalogue use dynamic memory allocation. This is not the case, and *a common problem will be the static memory allocation used in some programs*. Many of the programs in this compilation use static memory allocation. We will make every effort to document the programs that use static memory allocation, but cannot ensure all of the programs will be adequately mentioned in the text. Please check that the program will be capable of handling the problem you are trying to solve.

Fortran Compilers

All of the programs in the compilation can be compiled with the Compaq Visual Fotran Compiler. With minor modifications these programs can be compiled with any other compiler. There is an open source Fortran 95 compatible compiler available: g95[1]. It is an open source compiler that can compile the gslib programs with little to no changes.

To use the g95 Fortran compiler minor modifications to the source code may be required. The most common change will be an error message caused by the `use msflib` command. This command has become obsolete with the new fortran standard, but is still commonly used in the GSLIB programs. Removing, or commenting out, the line where `msflib` is called will not change the functionality of the programs.

The other common change relates to the format statements used for writing out information. Compaq Fortran allows some flexibility that is not in the fortran standard. Because of this, some format statments that work with the Compaq compiler will not work with the g95 compiler. The following format statement is not allowed according to the fortran standard, but does work with the Compaq Compiler:

```
100 format('number of data = ',i)
```

When the line is changed as below it will work with g95.

```
100 format('number of data = ',i4)
```

g95 does not allow a variable in the format statement. This is used for writing out arrays where the number of elements changes for different applications. The following format statement is correct, but will not work with g95:

[1] g95 is available for different platforms www.g95.org

```
110 format(<nvar>(1x,g14.6))
```

It has to be changed to the following for it to work.

```
110 format(200(1x,g14.6))
```

There is one other way to get this to work. This is to write the format statement to a string. Then use the string when writting out the information instead of the format statement.

Chapter 1

Data Sets

Data drives the practical application of geostatistics. However, many companies are hesitant to make their data publicly available. This is understandable. Some companies have made a few data sets public, while other data sets have been made from scratch. This chapter presents some data sets that are publicly available, real or synthetic, that can be used for testing and training purposes.

1.1 Lead smelter data (`dallas.dat`)

This is a collection of 180 lead samples collected in an area centered at an old lead smelter in Dallas. The data file contains the x location, y location, and assayed lead level.

1.2 Integer coded data (`ipcdat.dat`)

`ipcdat.dat` contains 140 integer coded data for GSLIB problems. These data were used to test and develop the indicator principal component software that was in the first edition of GSLIB. Those programs never made it to the second edition.

1.3 Small multi-part data set (`part.zip`)

`part.zip` contains three small data sets: (1) `parta.dat`, (2) `partb.dat`, and (3) `partc.dat`. These data are for small kriging exercises in the GSLIB book.

1.4 Vein type data (`red.dat`)

`red.dat` contains 67 data points sampled from a vein type gold deposit. The samples are 2-D and also contain thickness, silver, copper, and zinc.

1.5 GSLIB reference data set (GSLIB.zip)

The `true.dat` data file serves as a reference data set for the GSLIB book [10]. The data sets used throught the GSLIB book were created from this reference data set.

A clustered sample of 140 values, `cluster.dat`, was drawn from the reference data set. The first 97 samples were taken on a pseudo-regular grid and the last 43 samples were clustered in the high valued areas. The first 97 samples are in `97data.dat`.

A subset of 29 samples was retained in `data.dat`. This smaller number is more typical of many applications and also better illustrates the use of secondary information.

An exhaustive sampling of the reference data was created. `ydata.dat` contains the 29 primary samples and the exhaustive sampling of the secondary variable. This can be used to illustrate techniques for incorporating different attributes.

A complete 2D gridded variable was created by simulated annealing where the first lag of a "low nugget" isotropic variogram was matched. The gridded data, defined below, provide reference data for all subsequent problem sets. The reference 2D data file of 2500 values is characterized by the following geometric parameters:

xmn = 0.5 (origin of x axis)

nx = 50 (number of nodes in x direction)

xsiz = 1.0 (spacing of nodes in x direction)

and similarly, **ymn**=0.5, **ny**=50, and **ysiz**=1.0.

Four sample data sets were derived from this full valued 2D grid of 2500 values.

1. The complete reference data set contained in `true.dat`.

2. A clustered sample of 140 values was drawn from the reference data set. The first 97 samples were taken on a pseudo-regular grid and the last 43 samples were clustered in the high-valued regions (as identified from the first 97 samples). This data set, found in `cluster.dat` is used for the first two problem sets on exploratory data analysis and variogram analysis.

3. A subset of 29 samples was retained in `data.dat` for selected kriging and simulation exercises. This smaller number of samples is more typical of many applications and also better illustrates the use of secondary variables.

4. An exhaustively sampled secondary variable (2500 values) was created to illustrate techniques that allow the incorporation of different attributes. These 2500 secondary data *and* the 29 primary sample data are contained in `ydata.dat`.

1.6 Amoco reservoir data (Amoco.dat)

There are three data files with this set: (1) 2-D summaries of 62 wells over an area that is about 2 miles by 2 miles, (2) the detailed 3-D well data over the same aream and (3) a seismic attribute over the same area. These data were provided by Amoco many years ago for testing and development. This is a mixed siltstone-carbonate reservoir in West Texas.

1.7 Well log data (`twowell.dat`)

This file contains interpreted well log informatin for 2 wells. The logs contain: (1) facies, (2) VCLAY, (3) Porosity, (4) Permeability, and (5) Log10 Permeability.

1.8 Jura data (`jura.zip`)

The Jura data set was presented in Goovaerts' book [20]. The data set consists of 359 samples that include: (1) X location, (2) Y location, (3) rock type, (4) land use, (5) Cadmium, (6) Copper, (7) Lead, (8) Cobalt, (9) Chromium, (10) Nickel, and (11) Zinc. The metal concentrations are in parts per million (ppm). There are 5 rock types: (1) Argovian, (2) Kimmeridgian, (3) Sequanian, (4) Portlandian, and (5) Quaternary. The Land use is divided into 4 categories: (1) Forest, (2) Pasture, (3) Meadow, and (4) Tillage. There are prediction and validation data sets.

1.9 Spatial interpolation rainfall data (`sic97data.zip`)

The spatial interpolation contest in 1997 used 467 daily rainfall measurements made in Switzerland on the 8th of May 1986. A subset of 100 observed data were used to estimate the rainfall at the remaining 367 locations. The data can be freely downloaded from www.ai-geostats.org/resources/sic97_data.html.

There are 5 data files contained within `sic97data.zip`:

- Full data set (`sic_full.dat`): This file contains the 467 measured rainfall values. There are four columns in the data file. They are: (1) location code , (2) X coordinate in meters, (3) Y coordinate in meters, and (4) rainfall in 1/10th of mm.

- Subset of 100 locations (`sic_obs.dat`): This file contains a subset of 100 locations from the 467 measured rainfall values. They have the same format as the full data set.

- Country borders (`borders.dxf`): The country borders are available as an Autocad DXF file. Can be used as a base map.

- Elevation map: A digital elevation model is available for the area where the sample were taken. The grid is available in 2 different formats: (1) ASCII/ARC format (`demstd.grd`), and (2) surfer format (`surfdem.grd`). The grid definition is included in the files.

This data can be freely used as long as the following source is acknowledged:

> EUR 20667 EN. (2003) Mapping radioactivity in the environment. Spatial Interpolation Comparison 1997. Dubois G., Malczewski J. & De Cort M. (Eds). 268 pp.

1.10 Walker lake data (`WalkerLake.zip`)

The Walker Lake data set from Isaak & Srivastava's book [27]. The exhaustive data set that was used as a reference is in `walker_lake_exhaustive.dat` and the subset used as a test set is in `walker_lake_samples.dat`.

The exhaustive data set is a grid of 2 variables, U and V, that cover the entire area. There are 4 variables in the grid file: (1) X location, (2) Y location, (3) the U variable, and (4) the V variable.

The smaller data set is a set of samples taken from the exhaustive data set. It contains 6 variables: (1) sample number, (2) X location, (3) Y location, (4) V variable in ppm, (5) U variable in ppm, and (6) an indicator variable T.

1.11 Scanned rock image (`AGJ rock.dat`)

These gridded values were scanned from a small hand sample acquired by AGJ in 1991. The values were used to demonstrate the challenge of two- point statistics to reproduce complex patterns. This is the original eight category scanned image.

1.12 Yellowstone image data (`Yellowstone.dat`)

This data was scanned from a small rock sample that was purchased in West Yellowstone. There are spatial features that make the data difficult for many applications. The data is in GSLIB format and is 300 by 200.

Chapter 2

Data Manipulation

This Section describes some programs to prepare and analyse data. Every geostatistician will have their own set of programs, scripts and procedures for preparing data. Many settings are unique.

2.1 Calibrate data for Markov-Bayes (`bicalib`)

Whenever soft data are considered in an indicator formalism, it is necessary to calculate the soft indicator values (the prior distributions). Also, when considering the Markov-Bayes option in the indicator simulation program `sisim`, the B(z) calibration parameters are called for. The program `bicalib` computes the prior distributions and the $B(z)$ calibration parameters. These prior distributions are written to one output file and the $B(z)$ parameters are written to a reporting file from which they must be transferred to the `sisim` parameter file.

Often in practice the sparse sample size does not provide enough resolution for the scattergram of primary vs. secondary variables. Program `scatsmth` could be used to smooth and add resolution to the sample scattergram. Instead of the original data pairs (size n) the input data file datafl to program `bicalib` could contain the smoothed sample scattergram.

Author: Hua Zhu
Date: July 1990
Modified: C.V. Deutsch
Date Modified: 2003
Current Version: 3.000

2.2 Bivariate distribution modeling (`bimodel`)

Debiasing is a procedure whereby a representative distribution of a petrophysical variable is derived based on a spatially exhaustive secondary variable and a calibration relationship. We often deal with continuous variables, but categorical variables could be used. The exhaustive secondary variable could be seismic or a geological map. The calibration comes from the available well data or forward geophysical modeling. This algorithm is important to

create the required representative input to resource calculations and geostatistical modeling. [19]

Author: C.V. Deutsch
Date: 2005
Current Version: 1.000

2.3 Write out selected columns (`colsel`)

This program extracts a select set of columns from a GSLIB file. The columns do not have to be in ascending order or continuous; i.e. you could write out column 6 then column 4.

Author: Chad Neufeld
Date: 2002
Current Version: 1.000

2.4 Combine two columns from seperate files (`combine`)

This program extract columns from 2 different files and writes them out to a single file. The number of lines in each file must be the same.

Author: Chad Neufeld
Date: January 2002
Current Version: 1.000

2.5 Calculate stratigraphic coordinates (`corr_crd`)

Calculate erosional, onlap, and proportional coordinates from regular elevation well data.

Author: Jason M^{c}Lennan
Date: 2005

2.6 Extract samples within an area of interest (`extaoi`)

This program to extract all of the samples that fall within a 3-D rectangle specified in the parameter file. The program requires the x, y, and z coordinates to trim the data. Any data that falls outside the box is trimmed.

Author: Chad Neufeld
Date: January 2002
Current Version: 1.000

2.7 Extract samples based on trimming limits (`exthl`)

This program to extract samples based on an upper and lower trimming limit. Any samples that fall outside of the trimming limits are trimmed, and the samples that are within the trimming limits are written to the output file. This effect can be achieved by using the trimming limits included with the GSLIB programs and with some of the non-standard

GSLIB like programs. However, not all of the programs have that functionality built in. This is where `exthl` is useful.

Author: Chad Neufeld
Date: January 2002
Current Version: 1.000

2.8 Pairing spatial data (`getpairs`)

This program reads two data files with coordinates and writes an output file of pairs. Multiple data types can be compared when the pairs fall within some maximum distance. In the earlier version, the maximum number of data were hard coded. This had been changed to dynamic allocation. An additional option has been added to the parameter file. It allows the user to keep all of the pairs that fall within the specified radius, or only the closest pair.

Author: C.V. Deutsch
Date: 2004
Author: C. Neufeld
Date: 2006
Current Version: 2.000

2.9 Manipulate columns (`manip`)

This is a simple program to perform an arithmetic operation on a column in a data file or between two columns in two seperate files. If two files are used, the number of lines in each file must be the same.

The type of operation is specified with an integer in the paramter file (see below): specifying 1 performs addition, 2 is subtraction, 3 is multiplication and 4 is division .

```
1       -operator 1=+ 2=- 3=* 4=/
0.0     -factor to add/sub/mult/div (not used with 2nd file)
```

The operations are performed using the following formula:

$$\textit{result} = (\textit{value from file 1})\ \textbf{operator}\ (\textit{value from file 2 or factor})$$

Author: Chad Neufeld
Date: August 2002
Current Version: 1.000

2.10 Merge multiple files (`merge_multi`)

Merge multiple columns from multiple files into one file. Any number of columns can be read from any number of files.

Author: Jason M^{c}Lennan
Date: 2005

2.11 Merge multiple files (`merge_ult`)

This program will merge multiple columns/models from multiple files and do any combination of simple arithmetic manipulations on these columns. The merging works the same as `merge_multi` program. You can manipulate a single column, type 1 manipulation, or a set of two columns, type 2 manipulation. A type 1 manipulation for variable X is:

$$Z = X \quad [op] \quad c$$

where *op* can be $+(1)$, $-(2)$, $\times(3)$, or $\div(4)$ and c is a constant. A type 2 manipulation for two variables X and Y is:

$$Z = (X \quad [op1] \quad c1) \quad [op2] \quad (Y \quad [op3] \quad c2)$$

where *op1*, *op2*, and *op3* are operations and *c1* and *c2* are constants.

Author: Jason M^{c}Lennan
Date: 2006

2.12 Relocate data to grid node locations (`relocatedata`)

This program relocates scattered data according to a user-specified grid definition. The X/Y/Z values are modified to exactly coincide with the grid spacing and the data are written back out. A grid is not written — only the X/Y/Z coordinates are modified.

The parameters are self explanatory. The maximum number of data are hard coded (50000 at the time of this documentation). Multiple data points may be assigned to the same grid node.

Author: C.V. Deutsch
Date: 2002
Current Version: 1.000

2.13 Remove data at duplicate locations (`remdups`)

This is a hardcoded little program to read/write a set of data. Data that are within some arbitrarily small distance of previous data are not written. Only the first data point is kept.

Author: C.V. Deutsch
Date: 1995

2.14 Change coordinate system (`rotcoord`)

The `rotcoord` program performs a 2-D translation and rotation of coordinates (or the reversal of a previous 2-D translation and rotation). A 3-D rotation can be accomplished by two successive steps. The main purpose of a global translation and rotation is to make the grid system more efficient by aligning the coordinate system parallel and perpendicular to an elongated region of interest.

Author: C.V. Deutsch

Date: 1989
Date Modified: 2003
Current Version: 3.000

2.15 Split data from different categories (`splitcat`)

This program reads through a data file and splits the data according to a chosen set of categories. It is useful to split data according to rock type, well/drillhole ID and so on. Data that have an unrecognized code are skipped. The parameters are self explanatory. The maximum number of categories is set to 512. There are versions of this program out there that are called `split`, but that is an unfortunate choice of name since it is a valid UNIX command.

Author: C.V. Deutsch
Current Version: 1.000

2.16 Extract columns from a GSLIB file (`splitGSLIB`)

This program reads in a data file and seperates out n columns into one file per column. There is no parameter file, the user has to specify the parameters on a command line.

Author: C.V. Deutsch
Date: 2004

Chapter 3

Data Transformation

Data transformation is essential to geostatistics. Most of the programs presented here are used for transforming the data to be Gaussian.

3.1 Normal scores back-transformation (`backtr`)

The normal score back-transformation (`backtr`) program is rarely needed since Gaussian simulation programs, like `sgsim`, automatically back transform the simulated values. At times, however, it is useful to simulate in "normal-space" and keep the results so that they can be checked. Then, this program could be used for back transformation.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 3.000

3.2 Despike tied data (`despike`)

The normal score transformation is important for geostatistics. There is usually no concern when transforming an arbitrary distribution to normal scores. However, problems can arise when there is a large proportion of the data, ¿10%, that have the same value. Despiking is a method for "breaking the ties" between constant values for the normal score transformation.

All of the data need to be ordered, or ranked, as part of the normal score transformation. Despiking uses a local neighborhood for ranking data that have the same value. An average is calculated at each data value within the local neighborhood. The data values are ranked according to the average value from the local neighborhood centered at that data value.

Consider a region where the values are all zero. The standard approach is to transform the values using a random despiking procedure. This could lead to the zero values that are surrounded by other zeros having higher transformed values than the zeros that are close to samples with non-zero values. This problem can be avoided by using proper despiking with a local window for ranking the data

Author: C.V. Deutsch
Date: January 2003

Current Version: 3.000

3.3 Convert normal deviates to lognormal (`lognorm`)

This program converts a normal distribution to a lognormal distribution. There is no parameter file. The user specifies the mean and variance of the lognormal distribution and the input and output file names from the command line. This is particularly useful to create synthetic data sets for testing.

Author: C.V. Deutsch
Current Version: 1.000

3.4 Normal scores transformation (`nscore`)

This program transforms an arbitrary distribution to be Gaussian. Almost all simulation algorithms require normal score data. Despiking is needed when there is a significant number of data that have the same value.

The sorting routine inside of nscore was modified slightly. An additional array that is identical to the data array is created. Then a small random number is added to each value in the new array. This new array is used for sorting the data. It increases the sorting speed substantially when there are a large proportion of constant values in the data.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2006
Current Version: 3.000

3.5 Calculate the mean and variance for proportional effect (`propeff`)

It is very common for the variance of a variable to be non-stationary. Typically, the variance in high valued area is higher than the variance in low valued areas. This is called the proportional effect. `propeff` calculates the mean and variance of the data using a user specified grid.

Author: C.V. Deutsch
Date: January 2003
Current Version: 2.905

3.6 Calculate quantiles/CDF from non-parametric distribution (`quantile`)

Reads in a distribution of data, that has an optional weight, and will report any specified number of percentiles and quantiles.

Author: C.V. Deutsch
Date: September 1990

Current Version: 1.000

3.7 Stepwise conditional transformation (`sctrans`)

The stepwise conditional transformation algorithm has proven useful in recent times to transform complex multivariate relations to independent Gaussian variables. The need for many data hinders reliable implementation of stepwise conditional transformation for multivariate geostatistical simulation. Enforcing order relations consistency in the multivariate distribution and permitting flexible tail extrapolation for the conditional distributions improves the reliability of the transformation and back transformation. [14]

Author: Oy Leuangthong
Date: 2001
Modified: C.V. Deutsch
Date: 2005
Current Version: 2.100

3.8 Reverse stepwise conditional transformation (`scback`)

The variables from stepwise conditional transformation are Gaussian and independent. They are often simulated with SGS and must be back transformed in the reverse order to reintroduce the relationships transformed away during the stepwise transformation. This program performs that back transformation

Author: Oy Leuangthong
Date: 2001
Current Version: 1.000

3.9 General distribution transformation program (`trans`)

`trans` is a generalization of the quantile transformation used for normal scores, the p-quantile of the original distribution is transformed to the p-quantile of the target distribution. This transform preserves the p-quantile indicator variograms of the original values. The variogram (standardized by the variance) will also be stable provided that the target distribution is not too different from the initial distribution.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2003
Current Version: 3.000

3.10 General transformation with a trend (`Trans_Trend`)

This extended version of the `trans` program allows reproduction of vertical and horizontal trends. An iterative approach is required to ensure that local data, the target histogram, and target trends are reproduced. The program will also consider an externally specified mean value coming from, say, the spatial bootstrap. The ability to transform to any target

mean value is useful. The uncertainty in the mean can be established with the spatial bootstrap, and then realizations can be transformed to selected quantiles, (e.g., the 0.05, 0.5, and 0.95 quantiles) for sensitivity analysis.

Author: C.V. Deutsch
Date: 2005
Current Version: 1.000

3.11 Geometric unfolding (`Unfolding`)

A number of programs are contained in the unfolding directory. These programs will transform the coordinates of data so that modeling can be performed in straightened coordinates. The results can be back transformed.

Author: C.V. Deutsch
Date: 2004
Current Version: 1.000

Chapter 4

Distributions and Statistics

4.1 Bootstrap resampling for the average (`boot_avg`)

Bootstrapping is a statistical tool that is used to estimate the uncertainty in the data. It amounts to randomly drawing n samples from the distribution to assemble a new ditribution. Then the statisic of interest is calculated using the new distribution that was drawn from the original one. This procedure is repeated many times, m, and the results tabulated.

The maximum number of data is fixed at 2500 and the maximum number of simulation is fixed at 1000.

Author: C.V. Deutsch
Date: 1996
Current Version: 1.000

4.2 Bootstrap resampling for the correlation (`bootcorr`)

This program is an extension of the average bootstrap program. It is used to calculate the uncertainty in the correlation between 2 variables. The parameters are similar to `boot_avg`.

The maximum number of data is fixed at 2500 and the maximum number of simulation is fixed at 1000.

Author: C.V. Deutsch
Date: 1996
Current Version: 1.000

4.3 Bootstrap resampling from well data (`bootwell`)

This program resamples well or drillhole data from an input file. This is particularly useful with the data are highly correlated along wells and it would be inappropriate to sample the original data.

The maximum number of data is fixed at 5000.

Author: C.V. Deutsch
Current Version: 1.000

4.4 Bootstrap from three distributions (`boot03`)

This program draws from three distributions simultaneously. This is useful for simple Monte Carlo programs that relate three input variables specified by different distributions in different files.

The maximum number of data is fixed at 5000.

Author: C.V. Deutsch

Current Version: 1.000

4.5 Compare multiple point statistics (`compare`)

When selecting a training image (TI) for multiple point simulation it is often difficult to assess the appropriateness of potential TIs. Once the list of potential TIs is reduced based on the available geological knowledge `compare` can be used to compare the multiple point histogram or the distribution of runs for all potential TIs and the exploration data. Moreover, `compare` can be used to compare geostatistical realizations to the parent TI that generated the realizations to quantify how well the realizations reproduced the statistics of the TI. Use `mphist` or `runs_dist` to generate the required distributions for input to COMPARE. The output of compare includes a summary file giving the difference between all distributions as well as a rank file that ranks potential TIs or realizations [3].

Author: J. Boisvert

Date: 2006

4.6 Conditional statistics (`condstat`)

Old program to calculate conditional statistics of a variable within classes. This program is not aimed at multiple variables - one variable is considered.

Author: C.V. Deutsch

Date: May 1991

Current Version: 1.000

4.7 Cell declustering (`declus`)

Data are often spatially clustered; yet, we may need to have a histogram that is representative of the entire area of interest. To obtain a representative distribution, one approach is to assign declustering weights whereby values in areas/cells with more data receive less weight than those in sparsely sampled areas. The program declus provides an algorithm for determining 3-D declustering weights in cases where the clusters are known to be clustered preferentially in either high or low valued-areas. In other cases, polygon-type declustering weights might be considered whereby the weight is made proportional to the sample area (polygon) of influence.

The declustering weights output by declus are standardized so that they sum to the number of data. A weight greater than 1.0 implies that the sample is being overweighted

and a weight lesser than 1.0 implies that the sample is being downweighted (it is clustered with other samples).

Author: C.V. Deutsch
Date: 1989
Date Modified: 2003
Current Version: 3.000

4.8 Monte Carlo drawing (`draw`)

The Monte Carlo Drawing program is a very basic simulation program that can be used for the bootstrap or to understand the elements of stochastic simulation.

The `draw` program reads as input a K-variate probability density function, computes the corresponding K-variate cdf and draws a user specified number L samples from that distribution. The input probability density function could be the equal weighted data distribution, the declustered distribution, a smoothed version of the data distribution (e.g., the output of histsmth or scatsmth) or any user-specified distribution

Author: C.V. Deutsch
Date: 1989
Date Modified: 2003
Current Version: 3.000

4.9 Simulated annealing histogram smoothing (`histsmth`)

Creates a smooth univariate distribution model constrained to a mean, variance, quantiles, and smoothmess. The smoothing is done with simulated annealing.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 3.000

4.10 Gaussian kernel smoothing of histogram (`kernel`)

Kernel smoothing of an arbitrary distribution. The distribution can be a declustered or weighted distribution. The maximum number of data is hardcoded at 250, and the maximum number of discretization points is fixed at 500.

Author: C.V. Deutsch
Date: January 1994
Current Version: 1.000

4.11 Fit least squares line (`lsline`)

This program calculates the best fit line using the least squares regression criteria. The input file, column numbers, and output file are read from the command line, not a parameter file. The output is written to the screen.

All arrays are hardcoded.
Author: C.V. Deutsch
Date: 1996

4.12 Quick statistics from a data file (`minmax`)

This is a hardcoded little program to read a set of data and report some statistics for a particular column. Negative values below -900.0 are trimmed. The results are written to standard output (screen).
Author: C.V. Deutsch
Date: February 1992
Modified: Chad Neufeld
Date Modified: November 2003

4.13 Multiple-point histogram calculation (`mphist`)

Characterization of relations beyond two-point moments can be difficult. The multiple-point histogram is one way of describing such complex structure mathematically. The MPH is expressed as the frequency of arrangements of facies, with a one-dimensional index describing each arrangement. With K different facies, there are K^N classes in an N-point histogram. This program calculates the frequencies of each class.
Author: S. Lyster
Date: 2004
Current Version: 1.000

4.14 Multiple point scaling (`mps-scaling-random3D`)

Multiple-point statistics are used in geostatistical simulation to improve forecasting of responses that are highly dependent on the reproduction of complex features of the phenomenon that cannot be captured by conventional two-point simulation methods. Inference of multiple-point statistics is often based on a training image that depicts the features that provide the character to the geological event being modelled. One limitation of this approach is that the univariate distribution of categories (facies or rock types) in the training image may not match the target statistics of the final model. The question of scaling multiple-point statistics arises, the idea being to take the statistics from the training image and scale them in a repeatable manner and honouring the target univariate proportions of categories. [56]
Author: Julian Ortiz C.
Date: 2005
Current Version: 2.000

4.15 Polygonal declustering (`polydec`)

Another method for estimating a representative histogram from clustered data is polygonal declustering. The maximum number of points is hardcoded at 500. The program works in 2D only.

Author: C.V. Deutsch
Date: June 1988
Current Version: 1.000

4.16 Calculate the distribution of runs (`runs_dist`)

Calculates the distribution of runs on a set of drill holes/wells or on a model. If calculating on a model the direction (X,Y, or Z) is required. The cumulative distribution of runs as well as the distribution of runs is included in the output [3].

Author: J. Boisvert
Date: 2006

4.17 Simulated annealing scatterplot smoothing (`scatsmth`)

Creates a smooth bivariate distribution model constrained to marginal distributions, correlation coefficient, bivariate quantiles, and smoothmess. The smoothing is done with simulated annealing.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 3.000

4.18 Declustering with seismic data (`sddeclus`)

Data are rarely collected for their statististical representativity. This is particularly true in petroleum exploration and production where wells are often drilled in locations to maximize future production or to delineate the most productive portions of the reservoir. Nevertheless, geostatistical simulation methods require distributions of facies proportions and petrophysical properties that are representative of the *entire* reservoir being modelled. Therefore, there is a need of declustering procedures, which adjust the histogram and summary statistics to be representative of the entire volume of interest.

In presence of severely limited well data, conventional declustering algorithms such as the polygonal and cell-declustering methods are not effective; there are too few data to assign relative weights. Regions of poorer reservoir quality may be indirectly observed by seismic data or geological interpretation, but may not be sampled. In this case, there are no hard data to "weight." In such cases, we must obtain a representative distribution with the aid of soft secondary data and a calibration relationship between the soft data and the primary variable under consideration [9].

Author: C.V. Deutsch

Date: 1997
Date Modified: 2002
Current Version: 3.000

4.19 Spatial bootstrap resampling (`spatial_bootstrap`)

The usual assumption made in statistics is that the data are *iid* — independent and identically distributed. This is not the case when working with spatially correlated data.

Assuming independence:

$$Var\{\bar{X}\} = \frac{Var\{X\}}{n^2}$$

When the data are correlated, the uncertainty in the mean is always higher.

The spatial bootstrap program accounts for the correlation between the data values through the modelled variogram. It uses the LU simulation method for generating the realizations. This is very fast, but is limited to 3000–10000 data points [12].

Author: C.V. Deutsch
Date: 2003
Current Version: 1.000

4.20 Wide array declustering (`wa_declus`)

The program performs four different declustering methods and outputs a report detailing the results of each method and appends four weight columns to the data file [63].

Author: B. Wilde
Date: 2007
Current Version: 3.000

Chapter 5

Variograms

5.1 Apparent range in non-principal direction (app_range)

Experimental variograms are not usually isotropic and aligned with the coordinate system. The rotation angles and anisotropic distances can make interpreting the variogram model difficult. A very simple program app_range was written to calculate the maximum range of continuity for a specified direction using the modelled variogram. The paramter file requires the variogram model, from any GSLIB program, and the azimuth and dip directions for checking. Any number of directions can be checked at once [49].

Author: Chad Neufeld
Date: August 2004
Current Version: 1.000

5.2 Azimuth and dip of the principal directions (azm_dip)

It is useful to know how the three principal directions are oriented relative the coordinate system. However, this is not easy when you only have a set of GSLIB rotation angles. This program reads in the three rotation angles and caculates the azimuth and dip of the principal directions [49].

Author: Chad Neufeld
Date: August 2004
Current Version: 1.000

5.3 Indicator variograms of a theoretical bi-Gaussian distribution (bigaus)

Most geostatistical simulation is based on the Gaussian model. At times this may be inappropriate and one could check the theoretical indicator variograms (that correspond to the Gaussian model) to the actual indicator variograms. The indicator variogram computation (bigaus) calculates the theoretical indicator variograms, which can be plotted with the variogram plotting program.

This program allows checking for bivariate normality; a model of the normal scores variogram is input to `bigaus`. The corresponding indicator semivariograms for specified cutoffs, lags, and directions are output from the program. The idea is to see how far the experimental indicator variograms deviate from those implied by the bivariate Gaussian distribution. A numerical integration algorithm is used to compute the Gaussian model-derived indicator variograms.

Author: C.V. Deutsch
Date: 1997
Current Version: 3.000

5.4 Theoretical cross indicator variograms (`bigausfull`)

The old program for generating the indicator variograms calculates only the indicator direct variograms, this is, for single cut-off's. This new program calculates the full matrix of indicator direct and cross variograms. It uses a Monte Carlo approach for sampling the bigaussian distribution defined by the correlation model and calculating the volumes delimited by the cut-off under the distribution surface.

Two versions were built: version 1.000 gives results compatible with the old variogram format. Version 2.000 variogram results have the same format as `gamv2004` program. For both versions the parameter file is the same as for the `bigaus` program. [45]

Author: D.F. Machuca and C.V. Deutsch
Date: 2006
Current Version: 1.000 and 2.000

5.5 Calculate Gaussian variogram from indicators (`bigaus2`)

This program returns is an extension of the original GSLIB `bigaus` program. The program returns the values of the theoretical indicator semivariograms for various thresholds given a bivariate Gaussian distribution with a specified input normal scores semivariogram, or the values of the theoretical normal scores semivariogram of a bivariate Gaussian distribution that, after truncation at the 1-p quantile, will yield a binary image having the desired (input) standardized indicator semivariogram (1-p being the input cutoff/proportion)

Author: Phaedon C. Kyriakidis
Date: May 1997
Current Version: 2.000

5.6 Variogram from gridded data (`gam`)

The program for Regularly Spaced Data (`gam`) is for gridded secondary data (from seismic or remote sensing) or to check the result of simulation. The directions are specified by unit increments in the grid directions. The output from this program is not in GSLIB data format; it is in a format ready for the variogram plotting program.

Author: C.V. Deutsch
Date: 1992

Current Version: 3.000

5.7 Variogram from irregularly spaced data (gamv)

In general, the primary variable of interest will not fall on a regular grid. The program for irregularly spaced data (gamv) is commonly used for such data. The program will simultaneously calculate an arbitrary number of variograms in an arbitrary number of directions. The lag distance specification, however, is the same for all directions; therefore, multiple executions of the program are necessary in practice. The output from this program is not in GSLIB data format; it is in a format ready for the variogram plotting program.

Author: C.V. Deutsch
Date: 1997
Current Version: 3.000

5.8 Variogram calculation in cylindrical coordinates (gamv_cyl)

Calculates experimental variogram points. In the horizontal direction the (theta) axis wraps to allow for the calculation of distance between data pairs in cylindrical coordinates. Input data have (z-theta) coordinates with a given variable [7].

Author: Jeff Boisvert
Date: 2007
Current Version: 1.000

5.9 Locally varying variogram calculation (gamv_lva)

Calculates experimental variogram points. For the distance between points the anisotropic distance (considering a locally varying anisotropy field) is used. Program parameters are similar to GAMV with the addition of a gridded file for the locally varying anisotropy [5, 6].

Author: Jeff Boisvert
Date: 2007
Current Version: 1.000

5.10 Calculation of non stationary experimental variograms (gamvlocal)

Non stationary, location dependent variograms can be calculated by weighting the sample pairs inversely proportional to their distance to pre-defined control points. The gamvlocal program, which is based in the gamv2004 program is aimed for such task. The locations of control points are either provided in a Geoeas format file or defined by a grid. The remaining parameters are similar to Gamv2004. The program generates two files, the output file contains the local experimental variograms for each control point location in Gamv2004 format, and the debugging file contains the local means and variances for the same locations [39].

At the present model `gamvlocal` can be used for calculating the local semivariogram, covariance and correlogram.

Author: David F. Machuca
Date: 2007
Current Version: 1.000

5.11 Calculate Z variogram from Y (`gamvyz`)

In general, the normal score variogram is easier to calculate and model compared to the original unit variogram. The Z variogram can be calculated from the calculated experimental Y or modelled Y variogram. The program `gamvyz` reads in a gamv style output file with the Y variogram, the data transformation table and calculates the Z variogram [60].

Author: Brandon Wilde
Date: 2007
Current Version: 1.000

5.12 Variogram of irregularly spaced data (`gamv2004`)

The limitation of the gam/gamv formats in GSLIB is that little information is passed regarding the parameters used to calculate the variogram. This requires the user to keep those parameters separately and pass them into other programs such as `varfit`. This is inefficient and error-prone. The same basic format will be retained, but now there will be the possibility of additional lines of input parameters.

Note that the number of lags, the lag spacing, and the lag tolerance must now be specified for each direction. This permits the simultaneous calculation of horizontal and vertical directional variograms. It also permits the more reliable specification of directional variograms when the domain is anisotropic. The basic calculation scheme within the program remains the same.

The other major change is to the output format. Input data parameters are saved with the calculated variogram points. They are saved on four input lines after each title line. Each line begins with a four character index.

```
-HDIR   -30   22.5    25.0    -azm, azm tol, azm bandw
-VDIR   -90   22.5    25.0    -dip, dip tol, dip bandw
-LAGS    10   15.0     7.5    -# lags, lag dis, lag tol
-VARI     9   1   1   0.73    -var type, tail var, head var, ind cat/cut
```

The vargplt2004 program recognizes this format and removes any lines with a recognized four character index. The new `varfit` programs require this format to get information about the experimental variograms.

`Gamv2004` has been updated to allow the calculation of a standardized semi-madogram [37].

Author: C.V. Deutsch
Date: 2004
Modified: D.F. Machuca-Mory
Date Modified: 2007
Current Version: 3.100

5.13 Inference of variogram parameters from the composite samples variance (`nuggcalc`))

In some cases it is difficult to model the correct nugget effect, even from the experimental variogram parallel to the down-the-hole direction. Provided a drillhole assays file, `nuggcalc` program derives the correct nugget effect from the composite variances at different lengths. As a by product, the appropriate range in the down-the-hole direction is also calculated [38].

Author: David F. Machuca
Date: 2007
Current Version: 1.000

5.14 Correct apparent anisotropy (`trueanis`)

The user will be prompted for the apparent geometric anisotropy ratio and the angular tolerance used to compute this anisotropy. The program will report the "true" anisotropy associated with this apparent anisotropy.

Author: C.V. Deutsch
Date: September 1990

5.15 Semiautomatic variogram fitting (`varfit`)

The variogram is an important input in geostatistical techniques. Calculated experimental variograms must be modelled with a legitimate analytical model. This program obtains a legitimate model of spatial variability that closely fits directional experimental variograms. The program mimics the iterative procedure performed by the experienced geostatistician. In addition this program can fit linear models of coregionalization that are licit and positive semi-definite. The program varfit is GSLIB compatible, but it can be run with any variogram calculation output that has been properly reformatted.

The nugget effect, sill contributions, structure types and ranges, are fit to experimental variogram points in up to three directions simultaneously. The user can fix components of the variogram as needed. The fitting can be weighted by lag distance and/or number of pairs at each point, as an option for a better fit at shorter distances or at lags with more pairs, respectively [29].

The first `varfit` program developed by CCG has received widespread attention because it greatly simplifies variogram fitting, particularly in 3D and with multiple variograms making up a linear model of coregionalization. A number of enhancements were made in 2004: (1) a new gamv output format is considered as input, which simplifies the input parameters and makes it straightforward to input directional, indicator, and cross variograms, (2) any number of directions can be optimized simultaneously, not just the three principal directions, (3) the angles can be optimized independently, optimized and fixed between structures, or frozen at user input values, and (4) indicator direct and cross variograms can be fit with constraints on reasonable variations between thresholds [50].

Author: C.V. Deutsch and Paula Larrondo
Date: 2002

Modified: Chad Neufeld
Date Modified: July 2006
Current Version: 2.600

5.16 LMC variogram fitting (`varfit_lmc`)

This is a version of the `varfit` program written for fitting linear models of coregionalization. Fitting an LMC involves fitting many direct and cross varigrams. For example, with 4 variables there are 4 direct variograms and 6 cross variograms to fit. This program will fit a licit variogram model for up to n variables.

Author: C.V. Deutsch and Paula Larrondo
Date: 2002
Modified: Chad Neufeld
Date Modified: April 2004
Current Version: 2.500

5.17 Inidicator variogram fitting (`varfit_ind`)

Indicator variograms must be fit consistenly between thresholds. If not, order relations problems will arise. `varfit_ind` will fit indicator variograms for multiple thresholds while ensuring consistency between the thresholds. An output file contains the variogram parameters in a format that can be easily imported into EXCEL to plot how the different variogram parameters change between thresholds [50].

A new idea at CCG was to consider indicator cokriging to alleviate the patchy prevalent in simulated indicator maps. This introduced another problem for variogram fitting. In addition to the regular constraints for fitting an LMC variogram model, the bivariate transition probabilities must be non-negative. Two programs for fitting indicator LMC's were written: (1) `varfit_int_cat` for fitting categorical indicator LMC's and (2) `varfit_int_cont` for fitting continuous indicator LMC's. The paper in the 2005 CCG Report contains all of the details for fitting indicator LMC's and the kriging algorithm [55].

Author: Chad Neufeld
Date: June 2005
Current Version: 2.500

5.18 Local variogram fitting (`varfit-loc`)

This program is a modification of the `varfit` program aimed to sequentially fit the local experimental variograms generated by `gamv-local` for each of multiple control points. The summary file contains the local parameters of the fitted variogram models [39].

Author: David F. Machuca
Date: 2007
Current Version: 1.000

5.19 Calculate variogram map or volume (varmap)

Variograms are traditionally presented as 1-D curves as a function of the distance **h** along a particular direction. It is often useful to have a global view of the variogram values in all directions. The variogram map is a 2-D plot of the sample semivariogram for all experimentally available separation vectors.

A variogram map (varmap) is useful for determining directions of continuity. After computing and displaying a variogram map it is common to go back and use the gam or gamv program to calculate the variogram in the directions of greatest and least continuity. The output from this program is in GSLIB data format and is intended to be used in pixelplt.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2003
Current Version: 3.000

5.20 Variogram model (vmodel)

This program will take the semivariogram model and write out a file with the same format as the gam program so that it can be plotted with vargplt. The primary uses of vmodel are to overlay a model on experimental points and also to provide a utility to check the definition of the semivariogram model.

The variogram file from model (vmodel) program calculates the variogram values for specific distances and directions to be plotted with experimental values for fitting and checking a variogram model. The user must specify the variogram model — this program just calculates the values for plotting. It is common to set up this program and the vargplt program in a batch file so that the plot can be displayed right away after changing a variogram model parameter.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2003
Current Version: 3.000

5.21 Y-Z Variogram Mapping (vmodelyz)

An important problem in Mining Geostatistics and dealing with multiscale data is inference of a reliable small-scale variogram of the original variable (Z). Robust alternatives such as the pairwise relative variogram, correlogram and normal scores variograms converge to incorrect values. Our goal is to establish a robust calculation scheme where the values converges to the correct result, that is, there are no known biases in the variogram that will be used for subsequent volume variance and kriging steps. [59]

Author: C.V. Deutsch
Date: 1988
Modified: B. Wilde

Date Modified: 2005
Current Version: 1.002

Chapter 6

Volume Variance Calculations

6.1 Scale histogram with affine shape change (`affine`)

This program uses a reduction factor and the mean to complete an affine change of variance The affine correction assumes that the shape does not change - simply leave the mean and reduce the variance. Some comments on this correction: (1) introduces artificial minimum and maximum, (2) no shape change is unrealistic, and (3) experience shows that it works as long as f = 0.7.

Author: C.V. Deutsch

Date: 1996

Modified: C. Neufeld

Date: 2005

Current Version: 1.100

6.2 Scale histogram with discrete Gaussian model (`dgm`)

This program allows the calculation of Hermite polynomials and coefficients, change of support coefficient, and global distribution for block support.

The updated version of `dgm` allows the change of support calculations for different styles of mineralization, ranging from the pure mosaic model (measure of dissemination = 0), to the pure diffusive model (measure of dissemination = 1). In the last case it will perform as the old `dgm` program [37].

Author: Julián Ortiz C.

Date: 2000

Modified: Olena Babak

Date Modified: 2007

Current Version: 2.000

6.3 Average variogram for a block (gammabar)

This program calculates the average variogram or gammabar value from a semivariogram model. These average variogram values are required to calculate how the variance reduces as the volume increases.

The programs `gammabar` and `gammabar2` perform the exact same calculation with one exception; `gammabar` corrects for the zero effect and `gammabar2` does not.

This program has been updated version allows the calculation of the average measure of dissemination for a given volume. If the Madogram model is not provided, the program will behave as the previous version [37].

Author: C.V. Deutsch
Date: 1996
Modified: D.F. Machuca-Mory
Date Modified: 2007
Current Version: 3.000

6.4 Average variogram for arbitrary volume (gammabar-irr)

This program calculates the gammabar value from a semivariogram model. This program requires the user to enter the discretization points, which allows any irregular volume to be discretized. It has been used for actual production periods from mines where the locations are not contiguous.

Author: C.V. Deutsch
Date: 2002
Current Version: 2.000

6.5 Scale histogram with indirect lognormal shape change (indlog)

The transformation that would be used if both the block and point distribution were lognormal. It is more realistic than the affine correction as it does not impose artificial minimum and maximum. Some comments on this correction: (1) arbitrary shape change, and (2) works for a larger range of f, say f = 0.5, compared to the affine correction. This program was derived from the `postik` program from GSLIB. The indirect lognormal correction algorithm was taken from the book of Isaaks and Srivastava.

An error was found in `indlog` version 1.000. The bug was fixed in version 1.100.

Author: C.V. Deutsch
Date: 1996
Modified: C. Neufeld
Date: 2005
Current Version: 1.100

Chapter 7

Model Construction

7.1 Annealing categorical variable simulation with annealing (`afs`)

Conventional geostatistical simulation techniques such as sequential indicator simulation (SIS) and truncated Gaussian only account for two point correlations through the variogram. Moreover, realizations from such categorical variable simulation techniques often have systematic biases in the proportions of the categories. The restriction to two point statistics and bias in proportions significantly affect response variables calculated from simulated realizations. Simulated annealing can be used to change a specified starting image to match a set of desired statistics. Using simulated annealing with multiple point statistics to post process simulated realizations can capture realistic geologic features while removing bias in the results.

`afs` is an annealing post-processing program and demonstrate how it works. The required multiple point statistics are taken from a training image, which must be deemed representative. The results are shown to visually match the initial features on SIS realizations while simultaneously reproducing the high-order character taken from the training image. The results of a transfer function are shown to be unbiased.

Author: C.V. Deutsch
Date: October 1991
Modified: Julián Ortiz C. and Steve Lyster
Date Modified: 2004
Current Version: 1.401

7.2 Image cleaning with annealing (`anneal`)

The `anneal` program will extract statistics from a training image and post- process input realizations to more closely honor the input statistics. This program requires some tradecraft for difficult problems to set realistic annealing schedules.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005

Current Version: 3.000

7.3 Calculate 3-D trend from 2-D and 1-D trends (bbn)

Calculate $P(A|B,C)$ given $P(A|B)$, the aerial trend, and $P(A|C)$, the vertical trend, assuming full independence or incremental independance (permanence of ratios).

Author: Jason M^{c}Lennan
Date: 2005

7.4 Block sequential indicator simulation (BlockSIS)

Stochastic simulation of facies or geologic units is important before the assignment of continuous rock properties. Sequential indicator simulation (SIS) remains a reasonable approach when there are no clear genetic shapes that could be put into object-based modeling. Constraining SIS to soft secondary data coming from geological interpretation or geophysical measurements is important. There are a number of techniques including IK with a local mean, collocated cokriging, Bayesian updating, permanence of ratios, block kriging and block cokriging. BlockSIS implements all of these and more (nine all together). The images may also be cleaned using maximum a-posteriori selection. [16]

Author: C.V. Deutsch
Date: 2005
Current Version: 1.100

7.5 Block LU simulation of a 3-D Grid (blusim)

The blusim program was assembled from the kt3d and lusim codes in GSLIB with significant modifications to post process the local distributions. The program runs as a GSLIB program [54].

Author: Julián Ortiz C.
Date: December 2006
Current Version: 1.000

7.6 Multivariate block LU simulation (blusim_mv)

This program is the same as the blusim program, but can be used to simulate multiple variables. A linear model of coregionlization is required [4].

Author: Jeff Boisvert
Date: 2007
Current Version: 1.000

7.7 P-Field simulation and Bayesian updating (bupfsim)

Bayesian updating technique generates local uncertainty distributions. P-field simulation can be used to generate simulation realizations.

The implementation of P-field simulation after Bayesian updating is to draw a set of standard normal deviates that are spatially correlated within the field A (using sGsim). Then, conditional these standard normal values with the Bayesian updated mean and standard deviation to get simulated values.

Author: W. Ren
Date: 2005
Current Version: 1.000

7.8 Cloud transform (`cltrans`)

The Cloud Transform is a related technique. The goal is to construct a 3-D model of a petrophysical variable using a previously modeled variable and a bivariate calibration relationship. The probabilities to draw can be independent from each other or correlated in space. The technique is commonly referred to as P-field simulation if the probability values are correlated. [19]

Author: C.V. Deutsch
Date: 2003
Current Version: 1.000

7.9 Multiple coal seam modelling (`CoalSeams`)

This is a set of four programs for modelling multiple coals seams. The source code for the programs is contained in a zip file called `CoalSeams.zip`.

The program `faci_dist` creates and appends a new variable called faci-dist. It is calculated using the volume function defined as the shortest distance to another facies type. The program `coal_picks` is used to model multiple coal seams using the stratigraphic faci-dist variable. The elevations of the top and bottom of the seams are located using multiple grids. The output is in interval (from-to) format. The program `grid_picks` takes the interval data from `coal_picks` and creates a 3-D proportion model. The proportion being the proportion of the cell that is coal. The program `pickselplt` takes the interval data output from `coal_picks` and creates a plot of a given slice. The drillholes that fall within that slice can also be shown [62].

Author: B. Wilde
Date: 2007
Current Version: 1.000

7.10 Cokriging (`cokb3d`)

A 3-D Cokriging Program (`cokb3d`) is available in GSLIB to illustrate the potential of cokriging and provide a simple implementation for cokriging one variable using one or more secondary covariates.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005

Current Version: 3.000

7.11 Conditioning turning bands simulations (cbk)

Turning bands (TB) method is one of first practical 3-D simulation methods and it is used. TB generates unconditional simulation realizations based on isotropic covariance or variogram models. These realizations can be conditioned by kriging. Although the TB method has several limitations, it is faster than the common sequential Gaussian simulation (SGS) when multiple realizations are required. [57]

Author: W. Ren
Date: 2005
Current Version: 3.000

7.12 Volume function calculation with data conditioning factors (dceid)

Calculates a volume function for a specified 3d grid. The program uses the conditioning data and an inverse weighting for calculating the volume function.

Author: Jason A. McLennan
Date: 2006
Current Version: 1.000

7.13 Direct Multiscale sequential simulation (dssim_ms)

Direct sequential simulation with multiscale data. This first version is quite inefficient because of the CPU requirements to calculate all of the volume averaged variogram values. It is seed code for the development of a more complete DSS program

Author: C.V. Deutsch
Date: May 1999
Current Version: 2.901

7.14 Exact downscaling (edmdss)

Issues of scale pervade geostatistics. We deal with data of different scale and are often required to upscale and downscale models. This research is aimed at constructing scale-consistent models at different scales. Coarse scale models are often constructed over large areas consistent with larger scale seismic data and geologic trends. Then, finer scale models are constructed to represent the heterogeneity at a scale that affects recovery calculations. A number of models at different scale may be constructed. Fine scale models must be constructed so that they average exactly to the large scale models. This is required to avoid conflicts between different models and to capture the trends and large scale seismic data contained in the large scale models. [58]

Author: W. Ren

Date: 2005
Current Version: 2.000

7.15 Object based boolean simulation (ellipsim)

The ellipsim program is to illustrate the concept of randomly placing objects to create a categorical-variable 3-D realization. This program does not honor local data. This program provides a very simple example of Boolean simulation: ellipsoids of various sizes and anisotropies are dropped at random until a target proportion of points within-ellipse is met. The ellipsoids are allowed to overlap each other.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2003
Current Version: 3.000

7.16 Back transform quantiles from multi-Gaussian kriging (getquantiles)

The output from multi-Gaussian kriging is a normal score estimate and estimation variance. These 2 parameters completely define the distribution at a specific location. The program getquantiles is used to backtransform any number of quantiles from multi-Gaussian kriging.

Author: C.V. Deutsch
Date: 1989
Modified: C. Neufeld
Date Modified: 2004
Current Version: 2.907

7.17 Truncated Gaussian simulation (gtsim)

The Gaussian Truncated program may be used in conjunction with sgsim to accomplish simulation of categorical variables. This program performs no simulation, rather simulated Gaussian values (from sgsim) are truncated according to locally-variable thresholds.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 3.000

7.18 Indicator kriging in 3D (ik3d)

The indicator kriging program (ik3d) provides for indicator kriging with hard and soft indicator data. The use of soft indicator data requires some attention. In all rigor, a form

of cokriging should be used for soft indicator data. The output of this program can be post-processed by postik or the probabilities can be mapped directly.

The program `ik3d` considers ordinary or simple indicator kriging of either categorical and cdf-type indicators, it performs all order relation corrections, and, optionally, allows the direct input of already transformed indicator data, including possibly soft data and constraint intervals.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 3.000

7.19 Indicator kriging of a 3-D rectangular grid (`ik3d_fc`)

Indicator based geostatistical techniques provide great flexibility to integrate soft data and account for more complex spatial features than conventional Gaussian techniques. A long-standing problem, however, has been order relation deviations partly caused by inconsistently modeled indicator variograms. The bivariate distributions summarized by indicator variograms must be valid distributions, that is, there can be no negative probabilities, the probabilities must sum to one and the corresponding marginal distributions must be reasonable.

Typical implementations of indicator kriging and simulation consider independent kriging of each category/threshold. Cokriging with cross covariances is avoided primarily because of awkward inference and modeling. The semiautomatic fitting permits all direct and cross variograms to be fit simultaneously. We implement full indicator cokriging as an option in indicator kriging and simulation, which brings potential benefits [55].

Author: C.V. Deutsch
Date: 1989-2005
Modified: Chad Neufeld
Date Modified: 2005
Current Version: 3.000

7.20 Kriging in 2D (`kb2d`)

The 2D Kriging (`kb2d`) program is a small program designed for people learning about kriging and needing a simple program to see how it works. `kb2d` does 2-D point or block kriging with simple or ordinary kriging. This program is not designed for practical kriging applications; the program `kt3d` should be used.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 3.000

7.21 Kriging near boundaries (kt3d_bound)

Gradational or soft boundaries are common in several types of geological settings due to the transitional nature of geological mineralisation processes. Contacts were grades change transitionally across the boundary are usually characterized by a non-stationary behavior of the variable of interest, that is, the mean, variance or covariance are no longer constant within a zone of influence of one rock type into the other, and their values depends on the location relative to the boundary. The M.Sc. thesis of Larrondo developed the framework for non-stationary (co)kriging in the presence of a non-stationary soft boundary. This approach reduces the misclassification of ore and waste within the transition zone. [28]

Author: P. Larrondo
Date: 2005
Current Version: 3.000

7.22 Kriging in 3D (kt3d)

The 3-D kriging program (kt3d) is a general purpose kriging program for different types of kriging. The program will perform kriging on a grid, cross validation, and jackknife. This program could be tweaked to perform virtually any type of kriging (of course, there is the cokriging and indicator kriging program).

The program kt3d provides a fairly advanced 3-D kriging program for points or blocks by simple kriging (SK), ordinary kriging (OK), or kriging with a polynomial trend model (KT) with up to nine monomial terms. The program w orks in 2-D and is faster than kb2d if there are many data. One of the features that makes this program fairly fast is the super block search.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 3.000

7.23 Kriging in 3D – non-Gaussian collocated (kt3d_colo)

The program kt3d_com implemented collocated cokriging with a Gaussian assumption. This program implements collocated cokriging without the Gaussian assumption.

Author: C.V. Deutsch
Date: 1989
Modified: C.V. Deutsch
Date Modified: 2006
Current Version: 3.000

7.24 Kriging in 3D – complete (kt3d_com)

The 3-D kriging program (kt3d) is a general purpose kriging program for different types of kriging. The program will perform kriging on a grid, cross validation, and jackknife. This

program could be tweaked to perform virtually any type of kriging (of course, there is the cokriging and indicator kriging program).

The program `kt3d` lacks some commonly used kriging implementations. The first is collocated cokriging, and the second is an alterante iplementation of the locally varying mean. The program `kt3d_com` has all of the kriging methods contained in the original `kt3d` and the 2 methods mentioned earlier.

Author: C.V. Deutsch
Date: 1989
Modified: C.Neufeld
Date Modified: 2005
Current Version: 3.000

7.25 Global kriging (`kt3d_gl`)

In some cases, it is a good idea to use all of the available sample data. An example of this is building trend models or elevation maps. The program `kt3d_gl` does not perform any type of search. It simply reads in all of the sample data, builds a large n by n covariance matrix, inverts the matrix, and then uses all of the samples to estimate all of the blocks in the model. This produces smooth maps with no artifacts. The number of data is limited to approximately 1000 because of the matrix inversion. Note that not all of the kriging types are availble. The notable exception is collocated cokriging. [52]

Author: C.V. Deutsch
Date: 1989
Modified: C.Neufeld
Date Modified: 2005
Current Version: 3.000

7.26 Factorial kriging (`kt3d_fact`)

This is a modified version of te `kt3d` program to perform factorial kriging. Factorial kriging is commonly used to filter, or remove, components in a model. This decomposition can be useful for feature identification [18].

Author: C.V. Deutsch
Date: 2007
Current Version: 3.100

7.27 Locally varying kriging (`kt3d_lva`)

Performs Kriging with a locally varying anisotropy field. The distance between all necessary data and estimation locations is obtained from a Newton Method optimization. There is a possibility for indefinite matrices, which are flagged and indicated in a second file. This program requires a gridded file for the locally varying anisotropy [5, 6].

Author: Jeff Boisvert
Date: 2007

Current Version: 1.000

7.28 Finite domain kriging (`kt3d_upd`)

This program is extended version of the `kt3d` program which allows you to perform typical and Finite Domain Kriging; a robust technique with respect to 'string effect', at the same time. The parameter file for this program is identical to the one for `kt3d`. The difference is in the results: when you choose either option 0 (for Simple) or 1 (for Ordinary) in Kriging type, the results produced by running the program will contain in the first columns results produced by the typical Kriging, and in subsequent columns produced by the Finite Domain Kriging [2].

Author: O. Babak
Date: 2006
Current Version: 1.000

7.29 Estimation with non-stationary moments (`kt3d_lp`)

If non stationary variogram parameters are provided as grid files at the same resolution of the estimation grid, this program uses t his information to perform Quasi-non stationary simple and ordinary Kriging. If some variogram parameter is not provided in a grid format, the program will use the default stationary parameter. The parameter file is similar to the original Kt3d parameter file, with the exception that extra lines have been added at the end for the names of files containing the gridded variogram parameters [36].

Author: David F. Machuca
Date: 2007
Current Version: 1.000

7.30 Combine multiple secondary variables (`likelihood`)

Co-located cokriging and simulation are commonly used methods for incorporating secondary information into a geostatistical model. However, most implementations of co-located cokriging and simulation can only use one secondary variable. When more than one secondary variable is present, something must be done. The code could be modified, or the secondary variables could be combined into 1 "super secondary" variable. The `likelihood` program ccombines multiple secondary variables into a combined secondary variable using the correlations between the secondary variables and the variable that will be predicted. `Corrmat` can be used to generate the input correlation matrix. The combination is done using the normal score transformed secondary variables under a multi-variate Gaussian assumption. [65]

Author: C.V. Deutsch
Date: 2002
Modified: C. Nufeld
Date Modified: 2004
Current Version: 1.000

7.31 LU simulation (`lusim`)

The LU simulation program is a small matrix-method program that is suitable for small grids (less than 2500 or so grid nodes). It creates Gaussian-based realizations with the LU matrix method

Author: F. Alabert
Date: 1986
Modified: C.V. Deutsch
Date Modified: 2003
Current Version: 3.000

7.32 Annealing based multiple-point simulation (`mpasim`)

Facies simulation honouring multiple-point statistics can be quite difficult to implement. Simulated annealing is an algorithm used for optimizing very complex problems that may be very difficult to solve analytically. This program uses the square difference between target and actual MP histogram frequencies as an objective function to reproduce complex features [35].

Author: S. Lyster
Date: 2005
Current Version: 1.100

7.33 MPS simulation using a Gibbs Sampler and multiple-point events (`mpesim`)

This program implements a proposed Gibbs sampler algorithm for reproducing high- order structure. The Gibbs sampler is a Markov chain Monte Carlo algorithm which uses conditional distributions to approximate samples from marginal or joint distributions that may be unattainable analytically. Using several MPS templates simultaneously in a multiple normal equations framework, conditional distributions may be calculated which go beyond the variogram[34].

Author: S. Lyster
Date: 2006
Current Version: 0.600

7.34 Probability field simulation (`pfsim`)

The probability field program is another post-processing program that takes one file of conditional distributions (Gaussian or indicator) and simultaneously draws from them with correlated probabilities that are input as either Gaussian or uniform scores.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2003
Current Version: 3.000

7.35 Corrected probability field simulation (pfsim_ucc)

Probability field simulation has 2 well known artifacts: (1) the covariance between the simulated values near data is too high and (2) the data appear as local minima and maxima. The covariance bias in p-field simulation was derived and a correction scheme developed so that the simulated model will have the correct covariance after p-field. The program pfsim_ucc generates an unconditional simulation using a non-stationary covariance matrix that will produce the correct covariance in the simulated model. [51]

Author: C. Neufeld

Date: 2005

Current Version: 3.000

7.36 Probability kriging (pk3d)

Probability kriging is indicator cokriging with the indicator transform and the rank order transform. There was a belief that the rank transform would allow improved resolution within the classes of indicator kriging.

Author: C.V. Deutsch

Date: 1989-2005

Modified: Leonid Shmaryan

Date Modified: 2005

Current Version: 3.000

7.37 Facies modeling using probabilistic data integration (probint)

Building facies or geologic unit model is important before continuous rock properties modeling. Sequential indicator simulation has been widely used, however, several secondary data gets more available and useful. Incorporating secondary data can provide more realistic modeling results. Traditional co-kriging is unfortunately inadequate when mixing discrete variable and continuous variable such as facies indicator and seismic attributes. This program is for integrating primary and secondary data to construct facies model in 3-D based on probabilistic approach. One advantage of this approach is that different types of data can be honored as long as each data is calibrated to produce probability cubes (3-D). For example, qualitative geologic map, training images, and seismic attributes are used and separately calibrated to estimate probability of facies. Then, this program will integrate probability cubes to generate a posteriori probability of facies. Permanence of ratios, Tau-model, and Lamda-model(new method in this program) are implemented in this program. Final results are probabilities of facies over the entire modeling girds [26, 24, 25].

Author: S. Hong

Date: 2007

Current Version: 1.000

7.38 3-D simulation by annealing (sasim)

sasim allows conditional simulations of a continuous variable honoring any combination of the following input statistics: histogram, variogram, indicator variograms, correlation coefficient with a secondary variable, or conditional distributions with a secondary variable. This program requires some tradecraft for difficult problems to set realistic annealing schedules.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 3.000

7.39 Sequential Gaussian simulation (sGs) (sgsim)

The sgsim program is widely used for simulation and cosimulation. It is possible to simulate very large grids sequentially. Care should be taken to check histogram reproduction when collocated cosimulation or simulation witha locally varying mean is applied.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 3.001

7.40 sGs with block cokriging (sgsim_bc)

Modified version of sgsim to perform block cokriging in the presence of large scale secondary data.

Author: C.V. Deutsch
Date: October 1997
Current Version: 2.000

7.41 sGs with Bayesian updating (sgsim_bu)

sgsim was modified to use a locally varying Bayesian updating secondary variable. Use build_lh to build the gridded likelihood for input the sgsim_bu.

Author: C.V. Deutsch
Date: May 1999
Modified: Chad Neufeld
Date: 2004
Current Version: 3.000

7.42 sGs with cokriging (sgsim_fc)

Modified version of sgsim to perform cokriging in the presence of secondary data.

Author: C.V. Deutsch

Date: November 2000
Current Version: 2.001

7.43 sGs in a finite domain (`sgsimfd`)

Customized to work with "Kriging in a Finite Domain." This program avoids the string effect, but the weights and variance do not correspond to the theoretically required simple kriging weights and variance; therefore, the simulated values do not reproduce the input histogram or variogram.

Author: C.V. Deutsch
Date: August 1990
Current Version: 1.201

7.44 sGs with Locally Varying Angles (`sgsim_lva`)

Modified version of `sgsim` to perform sequential Gaussian simulation with locally varying directions of continuity. The locally varying directions are specified with an input grid where each cell has its own azimuth and dip for the direction of continuity. [31]

Author: O. Leuangthong
Date: September 2006
Current Version: 2.907

7.45 Variogram calculation in cylindrical coordinates (`sgs_cyl`)

Performs sequential Gaussian simulation in cylindrical coordinates. In the horizontal direction the (theta) axis wraps to allow for simulation in cylindrical coordinates (used in calculation of the distance between points). If the "radius" is used as the variable the realizations are of the pipe geometry and can be used to calculate the uncertainty in the pipe volume [7].

Author: Jeff Boisvert
Date: 2007
Current Version: 1.000

7.46 Locally varying variogram calculation (`sgs_lva`)

Performs sequential Gaussian simulation with a locally varying anisotropy field. The distance between all necessary data and estimation locations is obtained from a Newton Method optimization. There is a possibility for indefinite matrices, which are flagged and indicated in a second file. This program requires a gridded file for the locally varying anisotropy [5, 6].

Author: Jeff Boisvert
Date: 2007
Current Version: 1.000

7.47 Sequential lndicator simulation (`sisim`)

The `sisim` programs are for the simulation of either integer-coded categorical variables or continous variables with indicator data defined from a cdf. There are three sequential indicator simulation programs. The `sisim` program is a general purpose indicator simulation program. There is a special version for local means (`sisim_lm`) and for a gridded secondary variable (`sisim_gs`).

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 3.000

7.48 Direct lognormal simulation (`slogsim`)

The idea of direct simulation is becoming more established with the incorporation of unstructured grids in ore body and reservoir modeling. Direct kriging and simulation permits reliable integration of multiscale data. Simply performing kriging on data in original units, however, leads to a variance that is incorrect as real data exhibit a proportional effect. This paper introduces a direct simulation algorithm for data that appears lognormally distributed. Two types of direct simulation can be executed: (1) a naíve type that performs kriging and simulation on the data directly and (2) a form that corrects the kriging variance according to the proportional effect inherent in lognormal data. In fact, the proposed algorithm is not direct simulation — it is not based on the simple kriging (SK) principle that underlies the theory and publications of direct simulation. The proposed algorithm considers links the kriging estimate and the kriging variance according to the lognormal model; the SK principle requires independence of the estimate and variance. [42]

Author: Clayton V. Deutsch
Date: 1989
Modified: John Manchuk
Date Modified: 2005
Current Version: 2.000

7.49 Facies proportion from a surface model (`SurFac`)

Given a third-order surface model, which can be built by `Turbsim`, `LE_model`, or other resources/softwares, this program will calculate the facies proportion according to its 3D relative position of each cell using a given facies template. At present, only a linear facies template is used. The source code can be easily modified to use any kind of facies template. After calculating the facies proportion, a conventional SIS-based program can be used to build the facies model. The result is geologically realistic [66].

Author: Xingquan (Kevin) Zhang
Date: June 2006
Current Version: 1.000

7.50 Stratform Surface Simulation (`surfsim`)

General surface based simulation framework. It works for any geometries by using different flattening methods. Piecewise linear trend model can be well honored, and common bounding surfaces correlation can also be well honored. The result can be output for history visualization purpose or general surface modeling purpose. And there are also other small improvements [66].

Author: Michael J. Pyrcz
Date: March 2003
Modified: Xingquan (Kevin) Zhang
Date Modified: April 2006
Current Version: 2.995

7.51 Unconditional turning bands simulation (`tb3d`)

Generates and unconditional Gaussian simulations using turning bands. It was included with the first edition of the GSLIB book, but not the second. The results could be post processed to create conditional simulations. This program has largely been replaced by SGS except for historical interest and certain fast simulation exercises.

Author: A.G. Journel
Date: 1978
Modified: C.V. Deutsch
Date Modified: 1990
Current Version: 1.201

7.52 Indicator cosimulation using a training image (`tisis`)

The standard method used to build a rock type model is to code the facies as indicators, characterize the spatial structure of each indicator with variograms and/or covariance functions, and then perform sequential simulation. In the typical implementation, the indicator variograms of all facies are calculated separately; there is an implicit assumption that the different rock types are independent of one another. In most cases this is not true, and there are in fact spatial relations between the facies. The difficulty in producing a licit linear model of coregionalization including all direct and cross variograms (or covariances) leads to the simplification. If a suitable training image is available, direct calculation of covariances and cross-covariances for all facies is straightforward with no fitting of functions and the resulting model of spatial structure is guaranteed to be positive semidefinite as there is at least one realization which honours all of the statistics (the training image) [33].

Author: S. Lyster
Date: 2006
Current Version: 0.610

7.53 Ultimate sGs (ultimatesgsim)

The sequential Gaussian simulation (SGS) algorithm is popular because of its remarkable robustness; however, the flexibility offered by SGS is rarely used as much as it could be. The `UltimateSGSIM` program was generalized from the latest FORTRAN 90 `SGSIM` program to provide practitioners with access to some additional flexibility. Some of the special features of `UltimateSGSIM` include: (1) simulation by rock type accounting for hard and soft boundaries between rock types; one integer coded rock type can be set to "missing" and that rock type "keyed out" and not simulated, (2) multiple variables are simulated simultaneously by either (i) full cokriging, (ii) collocated cokriging with a Markov model, or (iii) after stepwise conditional transformation, (3) the mean values of any or all of the different variables can be made locally variable; the self-healing algorithm can be applied with this locally variable mean option (or with the collocated cokriging option), (4) the three angles that define anisotropy can be made locally variable to improve reproduction of large-scale curvilinear features, and (5) the correlation coefficients in collocated cokriging can be made locally variable [11].

Author: C.V. Deutsch
Date: 2004
Current Version: 1.000

7.54 Uniform conditioning (uc)

Uniform Conditioning is used in the mining industry, yet the literature available on this topic remains limited. This set of programs was the result of a guidebook aimed to explain the details of UC in a concise and complete manner [47]. The `pre_uc` program does the discrete Gaussian change of support and calculates the change of support coefficients the the UC program needs. The UC program, `uc`, reads in kriged panel estimates and calculates the recoverable reserves inside the panel given the estimate and the fitted discrete Gaussian model. The panel reserves can be upscaled to global reserves using `post_uc`.

Author: C. Neufeld
Date: 2005
Current Version: 1.100

7.55 Combine kriged estimate and likelihood (update)

The output from `likelihood` can be used as a co-located secondary variable for kriging, or it can be combined with kriged estimates once the kriging has been completed. The `update` program combines the likelihood with the kriged estimate to get a final estimated model. [65]

Author: C.V. Deutsch
Date: 2002
Modified: C. Neufeld
Date Modified: 2004
Current Version: 1.000

7.56 Non-parametric Bayesian Updating (`update_np`)

The `update_np` program combines the non-parametric likelihood and prior distributions [48]. This initial program was written to perform bayesian updating for a continuous variable using an indicator framework. The program `build_lh_np` can be used to calculate the gridded likelihood using a bivariate calibration.

Author: C. Neufeld
Date: 2006
Current Version: 1.000

Chapter 8

Model Checking and Validation

Model checking is very important. There are a number of programs that perform cross validation and ensure statistics reproduction.

8.1 Accuracy plot (`accplt`)

Probabilistic models built using geostatistical tools are built with some degree of uncertainty. One basic check is to verify that the probability intervals of the local distributions are consistent with the underlying model of uncertainty. For a specific probability interval (PI), p, we should expect to find that over multiple realizations, the proportion of times the true value falls within the PI is approximately equal to p for all p in [0,1] (Deutsch, 1996, [8]). For instance, a symmetric PI of 80% ($p = 0.80$) means that the lower and upper probability values in the interval is 0.10 and 0.90, respectively. Ideally, the proportion of times the true value falls within the 80% PI should be close to 0.80. If this fraction is much greater than 0.80, then the probability interval is too wide, and the local uncertainty may be too high. Conversely, if the fraction is much smaller than 0.80, then the probability interval is too narrow and the distribution has too low a variance.

The file containing the true values and the file containing the distributions must be in the same order – the program has no way of checking. The recent update changed the hardcoded data arrays to dynamic data arrays. There is no longer a maximum number of data for the program.

Author: C.V. Deutsch
Date: November 1995
Modified: Chad Neufeld
Date Modified: February 2007
Current Version: 2.000

8.2 Accuracy plot (`accplt-sim`)

This is a version of the accuracy plot program modified to have the realizations and data values in the same file.

Author: C.V. Deutsch

Date: November 1995
Current Version: 1.000

8.3 Check local data in a 3-D model (`checkmod`)

This program is used to see how well data are honored in a 3-D model. All of the data are checked – often the closest is assigned to the grid node location and the rest within the cell are not used.

The program `checkmod` writes out the data values from the input file and the value from the model grid for checking. `checkmod_mod` performs the same check as `checkmod`, but in addition it writes out an updated model where the data values have been assigned to the nearest cell in the model.

Author: C.V. Deutsch
Date: 1998
Modified: Chad Neufeld
Date Modified: August 2005
Current Version: 2.000

8.4 Check variogram reproduction (`gamsim_ave`)

This program is used to calculate the variogram for each realization in a simulated model. The original `gam` program was modified to loop through the simulation output and calculate variograms for each realization. It greatly simplifies checking variogram reproduction.

Author: C.V. Deutsch
Date: 1998
Modified: Julián Ortiz. C.
Date Modified: 2003
Current Version: 3.000

8.5 Check histogram reproduction (`histpltsim`)

This program is used to calculate the histogram for each realization in a simulated model. The original `histplt` program was modified to loop through the simulation output and calculate the multiple histograms. It greatly simplifies checking for histogram reproduction.

The recent modifications include: (1) specifying a reference disribution that will be plotted as a solid red line, and (2) performing random despiking prior to the data sorting. Plotting the reference distribution is a quick visual check that can be done to ensure the simulated histogram reproduces the input histogram. The mean and variance of the simulated realizations is ouput as well as the mean and variance for the reference distribution. Random despiking is done to improve the efficiency of the sorting routine used for calculating the CDF. When there are large proportions of constant values, zero's, one's, or categories, the sorting takes a very long time to complete. Despiking reduces the amount of time required to sort the data. The despiked values are only used for the sorting, the original values are used for calculating the distribution.

Author: C.V. Deutsch
Date: 1998
Modified: Chad Neufeld
Date Modified: 2005
Current Version: 3.200

Chapter 9

Model Manipulations and Transformation

9.1 Add spatial coordinates to grid (`addcoord`)

The GSLIB programs write gridded values in a specific order (X fastest, then Y, then Z, then by realization). It is often convenient to add the spatial coordinates so that the results can be imported to commercial software. This program adds 3-D coordinates to gridded points in an ordered GSLIB output file (from kriging or simulation).

Author: C.V. Deutsch
Date: 1989
Date Modified: 2003
Current Version: 3.000

9.2 Average values from one block size to another (`blkavg`)

This program averages a realization to a coarser scale. This program is not particularly flexible - the grids must nest perfectly. A more flexible program with dynamic memory allocation, rotated grids, and non- evenly fitting grids is requried.

Author: C.V. Deutsch
Date: 2004
Current Version: 1.000

9.3 Clip grid to a polygon (`clipgrid`)

Clips an input 3D grid by a bounding ploygon and optionally to a top surface. The polygon is specified inside the parameter file and the top surface in a second input file.

The maximum number of nodes in the polygon is fixed at 450. The top surface has a maximum size of 500 cells by 500 cells. There is no limit on the size of the grid to be trimmed.

Author: C.V. Deutsch
Date: 2002

Current Version: 1.100

9.4 Sample a 3-D grid (`clwell`)

Samples a 3D grid on multiple nested grids for psuedo wells. Small randomized offsets are implemented.

Author: Jason M^{c}Lennan
Date: 2005

9.5 Reformat Eclipse property values for GSLIB (`ecl2gsl`)

Converts a gridded property value from Eclipse format to GSLIB format. The output grid is in standard Geo-EAS format and can be read with any GSLIB program. The reformatting will most likely be specific to each case, resulting in the need to edit and recompile the program.

Author: Chad Neufeld
Date: March 2005
Current Version: 1.000

9.6 Reformat GSLIB grids for Eclipse (`eclgeom`)

This program takes structure specified in GSLIB format and writes the results in a format for Eclipse. The inputs are an upper and lower grid file that define the boudary of the model. The output is an eclipse structure file. The model properties can be formatted for Eclipse using the `gsl2ecl` program below.

Author: C.V. Deutsch
Date: 1991
Modified: Chad Neufeld
Date Modified: November 2003
Current Version: 1.100

9.7 Reformat GSLIB property values for Eclipse (`gsl2ecl`)

This program converts a GSLIB gridded file for Eclipse. Porosity and permeability can be read in and written out. Additional variables that are a function of porosity and permeability can be calculated in the code and written out as well. These custom variables need to be coded into the program.

Author: C.V. Deutsch
Date: 1991
Date Modified: November 2003
Current Version: 1.100

9.8 Scale up by steady state flow simulation (`flowsim`)

This program computes the effective permeability in the three principal directions. The arithmetic, geometric and harmonic averages are also calculated and saved.

Author: C.V. Deutsch
Date: 1987
Modified: C.V. Deutsch
Date Modified: K.M. Malick
Current Version: 2.000

9.9 Fuzzy probabilities from categorical model (`fuzzlvm`)

A deterministic categorical variable model is filtered to calculate fuzzy probabilities near the boundaries between categories. These probabilities can be used in SIS to assess uncertainty in categories.

Author: C.V. Deutsch
Date: February 2004
Current Version: 1.000

9.10 Randomly sample a gridded model (`gridsamp`)

This program gets pseudo-gridded values within a polygonal limit. This program has been used to create synthetic data sets for research purposes.

Author: C.V. Deutsch
Date: 1996
Current Version: 1.000

9.11 Categorical variable model cleaning (`maps`)

Annealing based cleaning of categorical realizations. The probability of each category is calculated on the basis of surrounding categories, local conditioning data, and the departure from the input proportions. Different window sizes can be used to generate results of varying smoothness.

Author: C.V. Deutsch
Date: 1997
Date Modified: 2004
Current Version: 2.000

9.12 Post processing object based models (`MAPSpp`)

Stochastic simulation of facies is important before the assignment of porosity and permeability. Object based models are particularly suited to situations where the original sedimentary structures are preserved and have a large affect on fluid flow. Object based models reproduce well data with varying degrees of success, with more difficulty in situations with large

numbers of wells. There is a need to post process object based models that do not exactly reproduce all of the available small scale well data. The facies intersections at well locations must be reproduced without unrealistic short scale variations away from the well locations. A post processing algorithm is implemented to accomplish this post processing. [15]

Author: C.V. Deutsch
Date: 2005
Current Version: 1.000

9.13 Merge 3-D models by category (`mergemod`)

This program merges multiple GSLIB output files based on an input facies model. One facies realizations is needed for each simulated realization. A default value may be written out for each facies type instead of a simulated value. The maximum number of lithologies is set at 10 in the array definitions.

Author: C.V. Deutsch
Date: 2004
Current Version: 2.000

9.14 Soft boundary model merging (`mix`)

Petrophysical properties often do not behave as if the boundaries that represent the prior decision of stationarity are hard. Both the original conditioning data and subsequently predicted petrophysical property data on either side of a boundary influence both sides to some degree. This program weights models from both sides of a boundary according to a user-defined distance and slope.

Author: Jason M^{c}Lennan
Date: 2006
Current Version: 1.000

9.15 Extract a single block (`picknode`)

This is a very basic program to read in a simulated model, and write out the simulated values at a single location.

Author: C. Neufeld
Date: 2002
Current Version: 1.000

9.16 Extract a single realization (`pickreal`)

This is a very basic program to read in a simulated model, and write out a selected realization.

Author: C. Neufeld
Date: 2002
Current Version: 1.000

9.17 Post process indicator kriged distributions (postik)

Indicator kriging reports the probabilities associated to specified cutoffs. The indicator kriging post processing program will calculate a number of summary statistics of the local indicator kriging-based distributions, e.g., the fraction and mean value above a threshold

The output from ik3d requires additional processing before being used. The program postik performs order relations corrections, change of support calculations, and computes various statistics according to the following options:

- Compute the "E-type" estimate, i.e., the mean value of the conditional distribution
- Compute the variance the conditional distribution.
- Compute the probability of exceeding a fixed threshold, the average value above that threshold, and the average value below that threshold.
- Compute the value where a fixed conditional cumulative distribution function (cdf) value p is reached, i.e., the conditional p-quantile value.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2003
Current Version: 3.000

9.18 Post process kriged Gaussian distributions (postmg)

Post process multi-Gaussian kriging results. Simple kriging in Gaussian space calculates a mean and variance that completely define conditional distributions in Gaussian units. These distributions, however, must be back transformed to calculate the average, variance, quantiles and such in original units. The results of simple kriging and the transformation table are input to this program for back transformation.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2004
Current Version: 1.000

9.19 Post process simulated realizations (postsim)

Often, it is convenient to summarize multiple realizations to display the local average (which will be close to kriging), the probability to exceed a critical threshold and so on. The post processing program will read multiple realizations from any of the GSLIB programs and calculate the requested summary statistics.

postsim allows a number of summaries to be extracted from a set of simulated realizations:

- the "E-type" estimates, i.e., the point-by-point average of the realizations.

- Compute the variance of the conditional distribution.
- the probability of exceeding a fixed threshold, the average value above that threshold, and the average value below that threshold.
- the value where a fixed conditional cumulative distribution function (cdf) value p is reached, i.e., the conditional p-quantile value.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2003
Current Version: 2.908

9.20 Post process simulated realizations (`postsim_new`)

This program is based off of the original GSLIB `postsim` program. It has been modified to shorten the run time drastically. By doing this the number of options has been reduced. Only the e-type estimate, variance, and probability to be above a threshold can be calculated. The original program must be used to calculate the probility intervals or specific p-values.

Author: C.V. Deutsch
Date: 1989
Modified: Chad Neufeld
Date Modified: 2005
Current Version: 2.908

9.21 Scale up by power law averaging (`powavg`)

This program will calculate specified power law averages within specified blocks. This is a another upscaling technique. The results of `flowsim` are often used to calibrate the power law exponents for this program.

Author: C.V. Deutsch
Date: January 1994
Current Version: 1.000

9.22 Full permeability tensor upscaling on unstructured grids (`ptensor`)

Heterogeneity in a hydrocarbon reservoir is captured by the rock permeability. This heterogeneity exists in reservoir both in fine scale and large scale. Upscaling techniques are applied to average the fine scale permeability up into the final flow simulation grids. In cases where unstructured grids are used, full permeability tensors arise instead of a diagonal tensor. The program called `ptensor` is developed, based on the `flowsim` program, to calculate full tensor permeability on an unstructured grid model [21, 22].

Author: R.M. Hassanpour

Date: 2007
Current Version: 2.000

9.23 Extract a 3-D Volume at random (`rndvol`)

Randomly extract a 3D volume from a 3D grid system.
Author: Jason M^{c}Lennan
Date: 2005

9.24 Upscale catersian grid (`upscale_cart`)

Upscale geology from a fine scale to a coarse scale appropriate for flow simulation.
Author: Jason M^{c}Lennan
Date: 2005

Chapter 10

Model Ranking and Scale Up

10.1 Compare realization ranking measures (`compare_rank`)

Compare realization rankings based on different criteria. The ranking measures from two different files are extracted and put together for cross plotting with, say, `scatplt`.

Author: C.V. Deutsch

Date: January 1996

Current Version: 1.000

10.2 Feasibility grade control (`fgc`)

The program performs FGC by optimizing the way blocks are accumulated into mining units. A report classifying each block as ore, waste, lost ore, or dilution is output [61].

Author: B. Wilde

Date: 2007

Current Version: 1.000

10.3 Calculate connected 3-D groups of cells (`geoobj`)

This program will scan through multiple 3-D realizations of rock type, porosity, and permeability and calculate connected regions. Net is defined based on specified rock types and the continuous variables exceeding user specified thresholds. Then, connected regions are defined by face-connected blocks. There are options to consider corner- connections and edge-connections, which are useful for sensitivity analysis. Two output files are created: (1) a 3-D grid of the geoobject number sorted from largest to smallest, and (2) a summary file of the geoobjects in each realization.

Author: C.V. Deutsch

Date: 1998

Current Version: 1.000

10.4 Rank by connectivity to well locations (rank_loc)

Rank realizations based on the volume connected to well locations. This program uses the results of the geoobj program to establish what values are connected.

The following parameters are hard coded in the program: (1) maximum number of realizations is 1000, (2) the maximum number of well locations is 50, and (3) the dimensions of the model are 200 cells for x, 100 cells for y, and 50 cells for z.

Author: C.V. Deutsch
Date: 1998
Current Version: 1.000

10.5 Rank by volume between well pairs (rank2loc)

Establish the number of cells that are shared between pairs of wells (arbitrary well paths) and then rank realizations in descending order of the connected volume. This program uses the results of the geoobj program to establish what values are connected.

Author: C.V. Deutsch
Date: 1998
Current Version: 1.000

10.6 Rank realizations by connected groups (rankobj)

Given the output statistics file from geoobj, establish a ranking of the realizations. This program uses the summary file created by the geoobj program. The number of geoobjects to use in ranking must be specified.

Author: C.V. Deutsch
Date: 1998
Current Version: 1.000

10.7 Rank realizations by simple statistics (rank_stat)

Realization ranking program based on simple statistics such as the average above a threshold.

Author: C.V. Deutsch
Date: May 1995
Current Version: 1.000

10.8 Local ranking (r_local)

Calculate the locally connected pore volume of multiple geological realizations.

Author: Clayton V. Deutsch
Modified: Jason M^{c}Lennan
Date Modified: 2005
Current Version: 1.002

10.9 Shale thickness ranking (`r_shthk`)

Calculate a shale thickness index within a SAGD drainage volume realizations with an associated facies model.

Author: Jason M^{c}Lennan
Date: 2005

10.10 Uncertainty matrix (`calres, calcrtu, calcsources`)

Assessing and presenting uncertainty from multiple realizations is a challenge in modern mining geostatistics. Most practitioners have developed their own procedures and tricks to present uncertainty. The aim of this short note is to collect some of that practice together for newer practitioners. One important goal of risk assessment is to understand where the uncertainty is coming from: the geologic rock types or the grades within the rock types. An uncertainty matrix is used to assess the sources of uncertainty. [17]

Three programs are required to generate the uncertainty matrix. The `calcres` program calculates resources/reserves within zones. The `calcsources` program merges the $nsim^2$ realizations and writes the results to a file for calculating the variances. The `calctru` program calculates the uncertainty due to the rock type model. The remaining uncertainty is due to the grade model.

Author: C.V. Deutsch
Date: 2005
Current Version: 1.000

Chapter 11

Production Data Integration for Petroleum Applications

Programs for production data integration have been collected in the first CCG Monograph. That document is highly recommended to get a description of the programs and the underlying assumptions. The following programs may be useful.

11.1 Sequential self calibration (SSC)

The sequential self calibration (SSC) algorithm was developed by Jaime Gomez-Hernandez and coworkers in Spain to account for pressure data in a hydrogeologic setting. The program mentioned here was adapted to a petroleum context. Initial permeability realizations are iteratively perturbed until specified pressure data are matched.

Author: X.H. Wen
Date: January 1996
Current Version: 1.000

11.2 Single well single phase (SW\SP)

These programs have been assembled for the integration of well test data. There are programs to calibrate the drainage radius and the type of averaging (the `calib` subdirectory). The `spsim` program will simulate the well test. The `wtperm` program will calculate the well test permeabilities from well test pressure responses. The `swsasim` program is an annealing based 3-D geostatistical simulation program that will create realizations that reproduce a histogram, variogram and well test derived permeabilities.

Date: January 2000
Current Version: 1.000

11.3 Multiple well single phase (MW\SP)

These programs have been assembled for the integration of multiple well single phase data, which is appropriate in primary production. Another version of the SSC code is contained

in this directory and a version of `sasim` that will use the results of SSC in the construction of fine scaled realizations is also provided.

Date: January 2000

Current Version: 1.000

11.4 Multiple well multiple phase (`MW\MP`)

These programs have been assembled for the integration of complex production data, that is, multiple well and multiple phase data. The programs are quite limited, but the basic approach to complex production data is established. A customized version of SSC that uses streamlines is available. The program is setup for 2-D 2 phase flow simulation.

Date: January 2000

Current Version: 1.000

Chapter 12

PostScript Plotting

12.1 2D plot of arrows from gridded angles (`arrowplt`)

This program is for plotting 2D map of grided angles by arrows. Angles are specified clockwise from the north direction (Azimuth) [23].

Author: R.M. Hassanpour
Date: 2007
Current Version: 1.000

12.2 View a bivariate histogram (`bivhist`)

This program creates histograms of two variables on the axes of a crossplot of these variables. This program is intended to plot the results of the bivariate smoothing program that is part of GSLIB. The input data formats are particular to that program.

Author: C.V. Deutsch
Date: December 1993
Current Version: 1.002

12.3 Bivariate PDF plot (`bivplt`)

The `histsmth` and `scatsmth` programs smooth univariate and bivariate distributions. The `bivplt` program displays the results. The plot also shows the original data pairs and the original unsmoothed data histogram.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 3.000

12.4 View a CDF (`cdfview`)

This program is used to create and view a cumulative distribution function and histogram. It also outputs a file containing a quantile function.

Author: C.V. Deutsch
Date: November 1993
Current Version: 1.000

12.5 Plot a correlation matrix (corrmat_plot)

This program generates a plot of a correlation matrix. A GSLIB gridded file with the correlations is required as input. The output is a postscript plot of the correlations. The grid cells are colored according to the correlation and the correlation is written in the center of each grid.

Author: C.V. Deutsch
Date: November 1993
Modified: C. Neufeld
Date Modified: 2005
Current Version: 3.000

12.6 View a grid (gridview)

This program uses three input files to create and view a grid. This is intended to show slices through stratigraphic layers.

Author: C.V. Deutsch
Date: November 1992
Current Version: 1.000

12.7 Plot grade tonnage curves (gtcurve)

This program creates a grade tonnage curve. The input may be data, declustered data, variance correct data or a kriged or simulated model. The output is written to a postscrtipt plot and a GSLIB output file. The output file may be used with the gtcure_plot program to plot the output of different grade tonnage curves on a single plot.

There is a version of the grade tonnage curve program written for processing simulated models, gtcurve_sim. It will loop through the simulated model and calculate the grade tonnage curve for each realization as well as the average grade tonnage curve. Note that this will be different than calculating the average of the simulations and then the grade tonnage curve. The output is written to a postscript plot and to an output file.

As mentioned earlier, there is a grade tonnage curve plotting program, gtcurve_plot. The input files must be in GSLIB format but can come from any program. The different lines can be given different colors, thicknesses, and dashing.

Author: C.V. Deutsch
Date: 1998
Modified: Chad Neufeld
Date Modified: 2005
Current Version: 2.000

12.8 Plot histogram (`histplt`)

This program has been custom written to generate some univariate statistical summaries and a visual output that is compatible with a PostScript display device. The input data is a variable from a simplified Geo-EAS input file where the variable in another column can act as a weight (declustering weight, specific gravity, thickness f or 2-D data,...). These weights do not need to add up to one. Minimum and maximum trimming limits can be set to remove missing values and outliers. The program will automatically scale the histogram. The user can choose to set the minimum and maximum histogram limits, number of classes, and whether or not to use a logarithmic scale.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 3.000

12.9 Create a 3-D isometric grid view (`isogrid`)

This program creates an isometric view through a 3-D grid. The output file is quite large since each cell face is represented by its own polygon.

Author: C.V. Deutsch
Date: December 1992
Current Version: 1.000

12.10 Plot isometric view of 3-D model (`isoview`)

This program creates an isometric view. It is executed with no command line arguments. The user will be prompted for the name of a parameter file. As in `isogrid`, the program creates large output files.

Author: C.V. Deutsch
Date: December 1992
Current Version: 1.000

12.11 Plot a histogram of the pairs in different lags (`laghist`)

This program creates a histogram of lags. The azimuth, it's tolerance and bandwidth, the dip and it's tolerance and bandwidth are specified by the user. The program scans through a set of data and assembles the data pairs (just like the `gamv` program) and displays the results.

Author: C.V. Deutsch
Date: 1997
Current Version: 1.000

12.12 Plot lines (`lineplt`)

This program plots lines onto an $x - y$ graph. Each line must be in its own file and it can be given its own color, thickness, and dash style. The program is based off of the GSLIB program `scatplt`.

The program support multiple polygons contained in the same file. Each polygon must be seperated by a / and have the x and y coordinates in the same column. The line plotting options cannot be changed for different polygons contained in the same file.

Author: C. Neufeld

Date: 2004

Current Version: 3.000

12.13 Location map (`locmap`)

An X/Y (or any two variables) plot for scattered sample values indicated by gray- or color-scale coded circles. The data values may also be labeled. The axis have the same PostScript distance units as the 2-D map (`pixelplt`) so the experienced PostScript user can directly edit the points into the pixelplt output for data checking.

Author: C.V. Deutsch

Date: 1989

Date Modified: 2005

Current Version: 2.906

12.14 Location map with Polygons (`locmappoly`)

This is exactly the same as `locmap`, but with the ability to plot polygons on top of the location map. Each polygon must be in its own file. The color, thickness, and line dashing can be different for each polygon.

Author: C.V. Deutsch

Date: 1989

Modified: C. Neufeld

Date Modified: 2005

Current Version: 2.906

12.15 Create a well log plot (`logplot`)

This program is a user friendly well log plotting program. The program is a little awkward to use, but the resulting graphics are quite flexible.

Author: Xingzhou (Frank) Liu

Date: May 1994

12.16 Plot a sampling nomograph (nomograph)

This program plots a sampling protrocal input in the parameter file. The user can specify a maximum sampling error line that will be plotted on the nomograph. The program also calculates the fundamental sampling error for the protocal that is being plotted.

Author: Chad Neufeld
Date: 2002
Current Version: 1.000

12.17 2-D plot from gridded model (pixelplt)

This program is for plotting 2-D slices through gridded 3-D data sets, e.g., as generated by kriging or simulation algorithms, one can use gray or color scale maps.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 2.905

12.18 A flexible 2-D plotting program (pixelplt_mp)

This program is for plotting 2-D slices through gridded 3-D data sets, e.g., as generated by kriging or simulation algorithms, one can use gray or color scale maps. In addition, wells can be plotted on top of the model and the grid can be clipped by a polygon.

Author: J. Manchuk
Date: 2006
Current Version: 3.100

12.19 2-D plot from grid with polygons (pixelpltpoly)

This is exactly the same as `pixelplt`, but with the ability to plot polygons on top of the gridded section. Each polygon must be in its own file. The color, thickness, and line dashing can be different for each polygon.

Author: C.V. Deutsch
Date: 1989
Modified: C. Neufeld
Date Modified: 2005
Current Version: 2.906

12.20 Combine PostScript plots onto a single page (plotem)

This program combines multiple GSLIB PostScript plots onto a single page. The number of figures in the x and y directions are specified in the parameter file. The plot names are given one to a line after the parameters. If a line is left blank, or a file does not exist, that

position is simply left blank. A title for for the page can be specified after the last file name and a footer for the page can be given after the title.

Author: C.V. Deutsch
Date: January 1995
Date Modified: 2005
Current Version: 3.000

12.21 Combine `plotem` output into multiple page PostScript (`plotem_mp`)

This program combines the output from multiple `plotem` runs into a single multiple page PostScript document.

Author: C. Neufeld
Date: 2005
Current Version: 2.000

12.22 Plot multiple probability plots (`plotprob`)

A small program written to plot multiple probability plots on the same PostScript output file.

Author: C.V. Deutsch
Date: January 1995
Current Version: 1.000

12.23 Plot variogram vectors (`plotvecs`)

This program plots vectors that could be used to construct a locally varying direction model for geostatistical modeling.

Author: C.V. Deutsch
Date: November 1992

12.24 Permeability versus porosity plotting (`poroperm`)

General plotting of permeability versus porosity. The program also fits a linear regression to the log of permeability versus porosity.

Author: C.V. Deutsch
Date: March 2007
Current Version: 3.000

12.25 Create a Postscript plot: lines, markers, text, fill areas (postplot)

General plotting of X-Y graphs and such. This program can be used for very general plots. The simple XY plotting capability is useful to plot lines and symbols on the same plot. The code predates the GSLIB plotting programs.

Author: C.V. Deutsch
Date: January 1988
Current Version: 1.000

12.26 Plot probability paper plot (probplt)

This program generates either a normal or a lognormal probability plot. This plot displays all the data values on a chart that illustrates the general distribution shape and the behaviour of the extreme values. Checking for normality or lognormality is a secondary issue.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 3.000

12.27 Creates a PostScript listing of a file (psexamp)

This program creates a postscript listing of a file, ie. the header information, the top 10 data lines, and the bottom 10 data lines

Author: C.V. Deutsch
Date: November 1990

12.28 Create a Postscript file that lists another file (pslist)

This program creates a postscript file that lists a parameter file, include file, or source code file of a GSLIB version 2.0 program.

Author: C.V. Deutsch
Date: March 1995
Current Version: 2.000

12.29 Generate a pretty listing of source code (psnotes)

Creates a Postscript listing of a source code file. The lines are numbered and a border is drawn.

Author: C.V. Deutsch
Date: 1991

12.30 Create a PostScript section plot (pssect)

This program creates a postscript section plot. The program is setup to plot a section through multiple stratigraphic slices.

Author: C.V. Deutsch
Date: February 1994
Current Version: 1.000

12.31 Q-Q or P-P plot (qpplt)

This program takes univariate data from two data files and generates either a Q-Q or a P-P plot. Q-Q plots compare the q uantiles of two data distributions, and P-P plots compare the cumulative probabilities of two data distributions. These graphical displays are used to compare two different data distributions, e.g., compare an original data distribution to a distribution of simulated points.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 3.000

12.32 Plot scatter plot (scatplt)

This program displays bivariate scatterplots and some statistical summaries for PostScript display. The summary statistics are weighted, except for the rank order (Spearman) correlation coefficient.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 3.000

12.33 Plot section through stratigraphic surfaces (secline)

Multiple surfaces are extracted along a section line and plotted.

Author: C.V. Deutsch
Date: 1997
Current Version: 1.000

12.34 Create a section view (secview)

This program creates a section view from data files which define the restored and existing top and bottom surfaces.

Author: C.V. Deutsch
Date: Summer 1996
Current Version: 2.000

12.35 Plot calculated variogram points (vargplt)

The program vargplt takes the special output format used by the variogram programs and creates graphical displays for PostScript display devices. This program is a straightforward display program and does not provide any facility for variogram calculation or model fitting.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 3.000

12.36 Plot calculated variogram points (vargplt2004)

The program vargplt takes the special output format used by the variogram programs and creates graphical displays for PostScript display devices. This program is a straightforward display program and does not provide any facility for variogram calculation or model fitting. The vargplt2004 program recognizes the new gamv2004 format and removes any lines with a recognized four character index.

Author: C.V. Deutsch
Date: 1989
Date Modified: 2005
Current Version: 3.000

Chapter 13

Miscellaneous Programs and Subroutines

13.1 Conditioning Event-based Facies Models to Well Data (`alluvsim`)

The `alluvsim` program introduces an alternative approach in conditioning event-based fluvial models. The geological complexity generated by event-based (also known as process-based or pseudo-genetic or advanced object based models) is appealing to many geologists. The complexity is believed to have a significant affect on fluid flow and recovery predictions. Event-based models are difficult to construct such that well and seismic data are reproduced. One approach to use the structure of such models is to use them as training images for multiple point statistics based algorithms. Those algorithms have their own challenges; it would be useful to create event-based facies model reproducing one to five wells. The `alluvsim` program was adapted to reproduce channel fill and non-channel intersections from multiple wells together with areal and vertical trend information that comes from seismic and well data. The results of this new approach are promising; however, challenges remain in presence of many well data or the requirement to reproduce non-channel facies such as levees and crevasse splays. [64]

Author: Michael Pyrcz

Date: 2005

Current Version: 1.000

13.2 Refine Boundary Volume Function (`bref`)

This program refines the volume function around a potential boundary location. The outputs are the point-set of refined nodes as well as a refined boundary indicator model and refined volume function around the boundary [43].

Author: Jason McLennan

Date: 2006

13.3 Code Drillhole Data for Boundary Modelling (dcode)

This program codes 3D drillhole data to volume function data appropriate for conditioning volume function interpolation for boundary modeling [43].

Author: Jason McLennan
Date: 2006

13.4 Optimal Design Matrix (dmatrix)

The use of experimental design for optimal sensitivity analysis was proposed in the 2003 CCG report. A methodology was developed to determine a design matrix of realizations that should be processed through to a transfer function such as flow simulation or engineering design. This presents an efficient means of performing sensitivity analysis with a minimum of realizations. The methodology consisted of minimization of an objective function that characterizes the difference between reference and observed first and second order sensitivity terms. Flexibility in the algorithm permitted consideration of any number of input variables, response variables and case values. [30]

Author: Oy Leuangthong
Date: 2005
Current Version: 2.006

13.5 Multivariate Gaussian Sensitivity Analysis (krig_tests)

Integration of multiple variables in a Gaussian context is remarkably useful: the required parameters can be inferred with relatively few data, the approach is simple and robust to implement, and data transformation schemes like the stepwise transformation can handle problematic relationships in a preprocessing step. This note discusses a methodology and small program to gain insight into the importance of different variables[41]. The importance of multiple variables considered simultaneously is investigated.

Author: John Manchuck
Date: 2005
Current Version: 1.0

13.6 Calculate L-optimal estimates (lossopt)

Making decisions in the prescence of uncertainty can be difficult. Choosing which realization to use for decision making is not an easy task, nor is it recommened. By using the full space of uncertainty and a loss function optimal decisions can be made.

A loss function is a quantification that represents the economic cost of making a wrong decision. The cost of over, or under, estimation is used to determine the optimal estimate for reducing the potential loss. In other words, by minimizing the loss, the profit should be maximized.

Author: Clayton Deutsch
Date: 1997

Current Version: 1.200

13.7 Loss function plot (`lossplot`)

This program creates a loss function plot.

Author: Clayton Deutsch
Date: 1997
Current Version: 1.000

13.8 Optimize dig limits for surface mining (`diglim`)

Dig limit selection is a difficult problem in mining. There is imprecise knowledge of the mineral grade and the dig limits must account for the mining equipment. This paper illustrates a technique for semi-automatic selection of dig limits. The technique is called semi-automatic because the user is required to supply initial dig limits. The technique consists of three steps: (1) transformation of a model of grade uncertainty to a map of expected profit, (2) selection of the initial dig limits, and (3) iterative perturbation of the initial dig limits to account for uncertainty and the limitations of mining equipment.

The optimization algorithm, called simulated annealing, is used to iteratively perturb the initial dig limits until they converge to dig limits that maximise profit and account for the mining equipment. A number of implementation details are addressed including: selecting initial dig limits, nested dig limits, and the protocol for dealing with multiple ore types.

Step-by-step examples using real mining data illustrate the technique and its features, and also a comparison between hand drawn dig limits and automatically generated dig limits is provided.

Author: K. Norrena and C. Deutsch
Date: 2000-2002
Modified: Chad Neufeld
Date Modified: 2003
Current Version: 2.000

13.9 Hierarchical Fluvial Reservoir Simulation (`fluvsim`)

The program is executed with no command line arguments. The user will be prompted for the name of a parameter file. The parameter file is described in the documentation (see the example fluvsim.par)

The output file will be a GEOEAS file containing the simulated facies codes. The file is ordered by x,y,z, and then simulation (i.e., x cycles fastest, then y, then z, then realization number).

Author: C.V. Deutsch
Date: 2000
Current Version: 2.900

13.10 Fit the Percolation Threshold (ptsa)

Fit the percolation threshold using simulated annealing with a best-fit power-law model above and below the percolation threshold on a plot of effective k vesus Vshale.

Author: Jason M^{c}Lennan
Date: 2005
Current Version: 1.000

13.11 Shale Thickness Measure (shthk_z)

Calculate a number of shale thickness measures from well data.

Author: Jason M^{c}Lennan
Date: 2005

13.12 Optimal Well Placement (QS_optimize)

The placement and timing of production and injection wells is a significant decision in reservoir development planning. Well placement is difficult with a single deterministic model of reservoir structure and petrophysical properties. The duration of plateau production must be maximized, water handling must be minimized, recovery should be maximized and key economic indicators should also be maximized. These response variables depend on the placement/timing of the wells, their operating conditions and the subsurface reservoir description. Many of the complex nonlinear interactions are resolved by simulating the reservoir behavior with a number of cases. A combination of sound engineering judgment and flow simulation are used in practice. The problems associated with well placement become more complex in presence of multiple geostatistical realizations. The number of flow simulation runs becomes intractable and it becomes impossible to visualize all possibilities. There is a need for numerical measures to assist in the optimization of well locations to minimize risk and maximize reservoir performance. Well placement is improved by establishing a quickly-calculated proxy for the performance of a particular well plan and reservoir description. The static proxy must be calibrated with some flow simulation results. Then, optimization of the well locations can be performed over multiple realizations simultaneously. The results can be used to supplement good judgment and flow simulation of base case realizations. [13]

Author: C.V. Deutsch
Date: 2005
Current Version: 1.6

13.13 Classification drillhole spacing and density (Spacing3D)

Geometric criteria for resources classification require of sample and drillhole spacing and density measures. This program calculates the local drillhole density and nominal spacing in two or three directions, either elliptic or rectangular search windows can be used, and the smoothing is controlled by changing the minimum and maximum number of drillholes

considered for calculations. Results can be plotted for creating maps of local spacing and density [46].

Author: D.F. Machuca and C.V. Deutsch
Date: 2006
Current Version: 1.000

13.14 Stope Geometry Optimization (`stopeoptimize`)

Economic improvement of underground mining operations by various optimization techniques is a growing area of research. Sectors of an operation such as designing stopes and scheduling equipment can benefit from computational optimization techniques. For many operations, stopes are designed by hand using ore grade cutoff shells, geology maps and geotechnical constraints; however, these stopes may not be offering their full potential to the economics of the operation. This research focuses on developing techniques for flexible optimization of stope geometry to maximize economic return. Practically any designed stope ranging from drift-like for cut and fill mining to very expanse for sublevel stoping are considered for optimization [40].

Author: John Manchuk
Date: 2006

13.15 Training Image Modification (`timod`)

Simulation with MPS may be improved by nesting the facies or rock types, and simulating hierarchically. To do this training images with several facies combined are required. This program reads in a given TI then outputs a specified indicator code for each input code; if several of the output values are the same then this effectively combines the facies.

Author: S. Lyster
Date: 2005
Current Version: 1.000

13.16 Training Image Resampling (`tisamp`)

Training images are not always at the same scale as that desired for simulation. If larger blocks are to be used than in the TI, a coarser grid is needed. This program resamples a TI at a specified resolution, which may be anisotropic.

Author: S. Lyster
Date: 2005
Current Version: 1.000

13.17 MPS Template Generation Using Entropy (`titemp`)

The use of multiple-point statistics (MPS) shows great promise for characterizing data which display high-order structure such as curvilinearity or complex relations between facies. The use of MPS for facies modeling requires a template of points to be defined, within which the

relevant statistics will be calculated and stored. This approach is used to minimize both the CPU time and memory requirements. Determining which arrangement of points to use in a MPS template is not trivial. Some points contain more relevant information; others contain significantly less, and at a certain distance from the central point some locations may not add any information at all. The methodology in this program uses two-point entropy to quantify the goodness of points in a possible template [32].

Author: S. Lyster
Date: 2006
Current Version: 1.000

13.18 Trimming the Borders of an Image (`titrim`)

Iterative simulation methods often leave edge effects around the borders of realizations. There are several ways to fix this problem: wrap the grid, which is geologically unrealistic; modify the algorithm near the boundaries, which is computationally inefficient; or simulate a larger field than the one desired and clip the excess cells.

Author: S. Lyster
Date: 2006
Current Version: 1.000

13.19 Optimal SAGD Well Placement (`wellopt`)

The application of steam-assisted gravity drainage (SAGD) to recover heavy oilsands is becoming increasingly important in the northern Alberta McMurray Formation because of the vast resources/reserves accessible with this production mechanism. Choosing the vertical locations of SAGD well pairs is a vital decision to be made for reservoir evaluation and planning. The inherent uncertainty in the distribution of geological variables should be an integral part of this decision. Geostatistical simulation is used to capture geological uncertainty. This geological uncertainty is used to determine a distribution of the best possible well pair locations. [44]

Author: J. McLennan
Date: 2004

Chapter 14

Subroutines

Common tasks that are repeated many times in a program, or multiple programs are best placed in subroutines. This allows a common interface between the geostatistical programs and the tasks that are common to the different programs. GSLIB contains approximately 30 subroutines that can be called by any fortran program. Other subroutines have been written and will be included. These subroutines are integral to the functioning of the programs, so they are documented here.

14.1 Random number generator (`acorni`)

All simulation algorithms require random numbers. There are no true random number generators. However, there are very good numerical methods for generating strings of psuedo-random numbers. The `acorni` subroutine uses the additive congruential method for generating random numbers. It has been shown that this is one of the best generators in use today [53].

Author: R.S.Wikramaratna

Date: October 1990

14.2 Backtransform (`backtr`)

This subroutine backtransforms a standard normal deviate from a specified back transform table and option for the tails of the distribution.

14.3 Interpolate and extrapolate CDF points (`beyond`)

This subroutine is a general purpose subroutine to interpolate within and extrapolate beyond discrete points on a conditional CDF. If the Z value is specified then the corresponding CDF value will be computed, if the CDF value is specified the corresponding Z value will be computed.

14.4 Blue color level (blue)

Provided with a real value this subroutine returns the blue portion of the color specification in integer and hexidecimal numbers.

14.5 Check file name (chknam)

This subroutine takes a character string and removes all leading blanks and blanks out all characters after the first blank found in the string (leading blanks are removed first). If "fi" or " fi" is found in the file name, all of the characters from the dash or backspace are blanked out.

14.6 Check title (chktitle)

This subroutine takes a character string and blanks out all characters a specific character string is found. The character string it looks for is two spaces, a back slash, "-Titl", "-titl"', "-TITL", "-X la", or "-Y la".

14.7 Covariance between two points (cova3)

This subroutine calculates the covariance between 2 points associated with a variogram model specified by a nugget effect and nested varigoram structures. The anisotropy definition can be different for each nested structure.

14.8 Gaussian inverse (gauinv)

Computes the inverse of the standard normal cumulative distribution function with a numerical approximation from : Statistical Computing, by W.J. Kennedy, Jr. and James E. Gentle, 1980, p. 95.

14.9 Cumulative Gaussian distribution (gcum)

Evaluate the standard normal cdf given a normal deviate x. gcum is the area under a unit normal curve to the left of x. The results are accurate only to about 5 decimal places.

14.10 Grid index for a point (getindx)

Calculates the grid index of a point in space using the location of the point and the grid definition in the x, y, or z direction.

14.11 Back transform univariate data from normal scores (getz)

This subroutine backtransforms a standard normal deviate from a specified back transform table and option for the tails of the distribution.

14.12 Green color level (green)

Provided with a real value this subroutine returns the green portion of the color specification in integer and hexidecimal numbers.

14.13 Hexadecimal representation of a number (hexa)

Returns the hexadecimal representation of a number.

14.14 Solve kriging system (no pivoting) (ksol)

Solution of a system of linear equations. This subroutine does not use pivoting, and will not be able to solve all matrices possible. A more stable subroutine is `ktsol`.

14.15 Solve kriging system (with pivoting) (ktsol)

Solution of a system of linear equations by gaussian elimination with partial pivoting. Several right hand side matrices and several variables are allowed.

14.16 Locate point in an array (locate)

Given an array and a value x, this subroutine returns a value j such that x is between $xx(j)$ and $xx(j+1)$. The array must be monotonic, either increasing or decreasing. $j = 0$ or $j = n$ is returned to indicate that x is out of range. `locate` is the single precision version and `dlocate` is the double-precision version of the subrutine.

Author: Bisection Concept From "Numerical Recipes", Press et. al. 1986 pp 90.

14.17 Transform univariate data to normal scores (nscore)

This subroutibe takes nd data $vr(i), i = 1, \ldots, nd$ possibly weighted by $wt(i), i =, \ldots, nd$ and returns the normal scores transform N(0,1) as $vrg(i), i = 1, \ldots, nd$. The extra storage array is required so that the data can be returned in the same order (just in case there are associated arrays like the coordinate location).

14.18 Write numbers to a string inside a program (`numtext`)

This subroutine will write a value into the string so that the label of the value can be written on the postscript file with the gramma of string.

14.19 Correct order relation problems (`ordrel`)

This subroutine identifies and corrects order relation problems in a conditional distribution known at a specified number of cutoffs.

14.20 Establish which super blocks to search (`picksupr`)

This subroutine establishes which super blocks must be searched given that a point being estimated/simulated falls within a super block centered at $(0, 0, 0)$.

14.21 Power interpolation (`powint`)

Power interpolate the value of y between $(xlow, ylow)$ and $(xhigh, yhigh)$ for a value of x and a power pow. `powint` is the single precision subroutine and `dpowint` is the double precision subroutine.

14.22 Plot polygon filled with a grey scale color (`psfill`)

Write a polygon and fill it with a grey scale color in a Postscript file.

14.23 Write a line to a postscript file (`psline`)

Postscript line commands to a file.

14.24 Write text to a postscript file (`pstext`)

Write Postscript text commands to a file.

14.25 Linear congruential (`rand`)

This random number generator generates random numbers in]0,1[Note that if the seed value is zero on the first call, a default value of 7931 will be used in a linear congruential generator to generate 55 odd integers for the array "itab". These values are preserved by a common statement, so that they may be used in subsequent calls by setting the seed to zero. If the value of "seed" is greater than zero in a call to the subroutine, then the array "itab" will be initialized and a new seed value will be returned by the subroutine. Best results are obtained by making the initial call with a seed of your choice and then setting the seed to "0" for all subsequent calls.

14.26 Red color level (red)

Provided with a real value this subroutine returns the red portion of the color specification in integer and hexidecimal numbers.

14.27 Linear rescaling function (resc)

Simple linear rescaling (get a value in coordinate system "2" given a value in "1").

14.28 Draw postscript plot axis (scal)

Draws a reasonable graph axes for a PostScript plot. The appropriate labelling and tic mark interval are established.

14.29 Setup anisotropic search matrices (setrot)

Sets up the matrix to transform cartesian coordinates to coordinates accounting for angles and anisotropy (see manual for a detailed definition).

14.30 Setup superblocks for searching (setsupr)

This subroutine sets up a 3-D "super block" model and orders the data by super block number. The limits of the super block is set to the minimum and maximum limits of the grid; data outside are assigned to the nearest edge block.

The idea is to establish a 3-D block network that contains all the relevant data. The data are then sorted by their index location in the search network, i.e., the index location is given after knowing the block index in each coordinate direction (ix,iy,iz):

$$ii = (iz-1) * nxsup * nysup + (iy-1) * nxsup + ix$$

An array, the same size as the number of super blocks, is constructed that contains the cumulative number of data in the model. With this array it is easy to quickly check what data are located near any given location.

14.31 Sort nested arrays (sortem)

This is a subroutine for sorting a real array in ascending order. This is a Fortran translation of algorithm 271, quickersort, by R.S. Scowen in collected algorithms of the ACM.

The method used is that of continually splitting the array into parts such that all elements of one part are less than all elements of the other, with a third part in the middle consisting of one element. An element with value t is chosen arbitrarily (here we choose the middle element). i and j give the lower and upper limits of the segment being split. After the split a value q will have been found such that $a(q) = t$ and $a(l) <= t <= a(m)$ for all $i <= l < q < m <= j$. The program then performs operations on the two segments

$(i, q-1)$ and $(q+1, j)$ as follows The smaller segment is split and the position of the larger segment is stored in the lt and ut arrays. If the segment to be split contains two or fewer elements, it is sorted and another segment is obtained from the lt and ut arrays. When no more segments remain, the array is completely sorted.

`sortem` is the single precision subroutine and `dsortem` is the double precision subroutine.

14.32 Squared anisotropic distance (`sqdist`)

This routine calculates the anisotropic distance between two points given the coordinates of each point and a definition of the anisotropy.

14.33 Super block search (`srchsupr`)

This subroutine searches through all the data that have been tagged in the super block subroutine. The close data are passed back in the index array "close". An octant search is allowed.

14.34 Determine the length of a string (`strlen`)

Determine the length of the string minus trailing blanks. A much faster alternative is to use the intrisic Fortran 90 function `trim()`.

Author: C.V. Deutsch
Date: July 1992
Modified: Chad T. Neufeld
Date Modified: February 2002

Chapter 15

Scripts

Many mundane tasks required for geostatistics can be done with simple scripts. Scripts do not need to be compiled, they can be called just like programs, they can be simpler to write, and they can even be faster than fortran programs.

The scripts presented here are called in the following manner:

```
script_name argument_1 argument_2 ...
```

If there are no arguments, or the incorrect number of arguments, after the script name, it will write a description of the required arguments to the screen:

```
Usage: gocad_format.sh inputFile colNb varName outputFile
```

For example; to use the `gocad_format` script, it needs to know the input file, the column number, the variable name, and the output file.

The downside of writing scripts is that the user must have access to cygin, or UNIX, and be familar with a UNIX like environment.

15.1 Compile Fortran from Cygwin (`cvf`)

It is not possible to compile fortran programs with the Compaq Visual Compiler from a cygwin command line. This is because cygin and the Compaq Compiler use the same name for the linker (although there is one for cywin and one for Compaq). A simple sript `cvf` was written to facilitate compiling programs from a cygwin command line. Instead of typing `fl32 program_name`, type `cvf program_name`.

Author: Chad Neufeld
Date: 2004

15.2 Filter based on an integer code (`filter_integer_code`)

Seperating files based on facies type, or any integer code, is a common task. This simple script will extract lines from a file based on an integer code. It is usually alot faster than Fortran programs that do the same task and there is no limit to the line length.

Author: Chad Neufeld
Date: 2004

15.3 Format GSLIB grid for GOCAD (gocad_format)

Before importing a GSLIB gridded file into GOCAD, it needs to be put into a simplified format. The format consists of a 1 column file with the title of the property on the first line, followed by the model, the same order as GSLIB, and then a forward slash at the end of the file. The script `gocad_format` is run just like a program. You type the name of the script followed by the parameters that the scipt needs. If no parameters are given, or an incorrect number of parameters, the cript will write the correct usage to the screen.

An important point to note: the way that the grid was defined in GOCAD is important. You will have to ensure that the indexing in the GOCAD is the same as GSLIB before importing any model with this script. If the indexing is not the same, a more complicated reformatting program will be needed.

Author: Chad Neufeld

Date: 2004

Chapter 16

Non-CCG Programs

These programs were not written at CCG. They are used by most of the students and researchers. Text editors, such as TextPad, are needed for writing programs and scripts. Each person will have their own preference for a text editor, we have included the 2 most popular at CCG. A Fortran compiler will be needed to use any of the software included with this compilation. There are numerous commercial compilers available. An open source Fortran 95 compiler is available free of charge and has been included. Most of the programs in this compilation are run from the command line. This can take time, especially when performing repetitive tasks. Scripting helps to automate and speed up these tasks. CYGWIN is a UNIX like environment for windows that allows the use of scripts. A postscript interpreter will be needed to view any of the plots made by the GSLIB programs. The postscript interpreter is free.

16.1 Windows text editor (`TextPad`)

Editing multiple parameter files, source code, and postscript images requires a powerful text editor. `TextPad` is an easy to use editor for windows that is available for a nominal licensing fee.

16.2 Unix text editor (`gvim`)

People who have work on UNIX should be familiar with `vi`. It is an extremely powerful text editor that is now available for any operating system. `gvim` contains all of the functionality from `vi` with more. It has a steep learning curve, but it pays off after a while.

16.3 Open source Fortran compiler (`g95`)

The `g95` project's goal is to provide a Fortran 95 standard compiler to the public. Limited testing has been done with `g95` and GSLIB, and there is no problem compiling the programs. The only quirk with `g95` is that the executables it produces run much slower than a commercial compiler.

16.4 Unix emulator for Windows (`cygwin`)

The standard command line included with windows, DOS, is not very powerful. An alternative is the `cygwin` shell. It is a UNIX emulator that runs on windows. This allows you to run almost all of the native UNIX text editing commands and in addition to use scripting for automating tasks. `Cygwin` is available free from cygwin.com.

16.5 PostScript interpretation (`ghostscript and gsview`)

Postscript is the common printer language. All of the images produced by GSLIB programs are in PostScript. You will need a PostScript interpreter to view any of the plots.

Bibliography

[1] and Clayton V. Deutsch. — Paper xxx. In *Centre For Computational Geostatistics, Report 9*, 2007.

[2] O. Babak and C. V. Deutsch. Estimation in a finite domain: Fixing the string effect — Paper 111. In *Centre For Computational Geostatistics, Report 8*, 2006.

[3] J. Boisvert, M. J. Pyrcz, and C. V. Deutsch. Choosing training images and checking realizations with multiple point statistics — Paper 202. In *Centre For Computational Geostatistics, Report 8*, 2006.

[4] J. B. Boisvert and Clayton V. Deutsch. Multivariate block lu simulation — Paper 305. In *Centre For Computational Geostatistics, Report 9*, 2007.

[5] J. B. Boisvert, J. G. Manchuk, and Clayton V. Deutsch. Calculating distance in the presence of locally varying anisotropy — Paper 102. In *Centre For Computational Geostatistics, Report 9*, 2007.

[6] J. B. Boisvert, J. G. Manchuk, and Clayton V. Deutsch. Kriging and simulation in the presence of locally varying anisotropy — Paper 103. In *Centre For Computational Geostatistics, Report 9*, 2007.

[7] Jeff Boisvert, Peter Oshust, and Clayton V. Deutsch. A methodology for geometric modeling of kimberlite deposits with uncertainty — Paper 307. In *Centre For Computational Geostatistics, Report 9*, 2007.

[8] C. V. Deutsch. Direct assessment of local accuracy and precision. In E. Y. Baafi and N. A. Schofield, editors, *Geostatistics Wollongong '96*, volume 1, pages 115–125. Kluwer Academic Publishers, 1996.

[9] C. V. Deutsch, P. Frykman, and Y-L Xie. Declustering with seismic or "soft" geological data — Paper 01. In *Centre For Computational Geostatistics, Report 1*, 1999.

[10] C. V. Deutsch and A. G. Journel. *GSLIB: Geostatistical Software Library and User's Guide*. Oxford University Press, New York, 2nd edition, 1998.

[11] C. V. Deutsch and Stefan Zanon. Ultimatesgsim: Non-stationary sequential gaussian cosimulation by rock type — Paper 51. In *Centre For Computational Geostatistics, Report 4*, 2002.

[12] Clayton V. Deutsch. A statistical resampling program for spatially correlated data: `Spatial_Boostrap` — Paper 401. In *Centre For Computational Geostatistics, Report 6*, 2004.

[13] Clayton V. Deutsch. Optimal well placement in presence of multiple geostatistical realizations — Paper 207. In *Centre For Computational Geostatistics, Report 7*, 2005.

[14] Clayton V. Deutsch. Order relations correction and tail extrapolation for stepwise conditional transformation — Paper 109. In *Centre For Computational Geostatistics, Report 7*, 2005.

[15] Clayton V. Deutsch. Post processing object based models to reproduce well data: `MAPSpp` — Paper 204. In *Centre For Computational Geostatistics, Report 7*, 2005.

[16] Clayton V. Deutsch. A sequential indicator simulation program for categorical variables with point and block data: `BlockSIS` — Paper 402. In *Centre For Computational Geostatistics, Report 7*, 2005.

[17] Clayton V. Deutsch. A short note on: The uncertainty matrix for assessment of risk and uncertainty — Paper 307. In *Centre For Computational Geostatistics, Report 7*, 2005.

[18] Clayton V. Deutsch. A recall of factorial kriging with examples and a modified version of kt3d — Paper 404. In *Centre For Computational Geostatistics, Report 9*, 2007.

[19] Clayton V. Deutsch and Thies Dose. Programs for debiasing and cloud transformation: `bimodel` and `cltrans` — Paper 404. In *Centre For Computational Geostatistics, Report 7*, 2005.

[20] P. Goovaerts. *Geostatistics for Natural Resources Evaluation*. Oxford University Press, New York, 1997.

[21] R.M. Hassanpour and O. Leuangthong. Calculation of full permeability tensor in an unstructured grid block — Paper 207. In *Centre For Computational Geostatistics, Report 8*, 2006.

[22] R.M. Hassanpour and O. Leuangthong. Calculation of permeability tensors for unstructured grid blocks — Paper 201. In *Centre For Computational Geostatistics, Report 9*, 2007.

[23] R.M. Hassanpour and O. Leuangthong. Determination of locally varying directions through mass moment of inertia tensor — Paper 112. In *Centre For Computational Geostatistics, Report 9*, 2007.

[24] S. Hong and Clayton V. Deutsch. Data integration for 2-d areal proportions of facies — Paper 205. In *Centre For Computational Geostatistics, Report 9*, 2007.

[25] S. Hong and Clayton V. Deutsch. High resolution facies modeling in presence of large-scale probabilistic data — Paper 206. In *Centre For Computational Geostatistics, Report 9*, 2007.

[26] S. Hong and Clayton V. Deutsch. Methods for integrating conditional probabilities for geostatistical modeling — Paper 105. In *Centre For Computational Geostatistics, Report 9*, 2007.

[27] E. H. Isaaks and R. M. Srivastava. *An Introduction to Applied Geostatistics*. Oxford University Press, New York, 1989.

[28] Paula Larrondo and Clayton V. Deutsch. An update on kriging near boundaries — Paper 304. In *Centre For Computational Geostatistics, Report 7*, 2005.

[29] Paula F. Larrondo, Chad T. Neufeld, and Clayton V. Deutsch. `VARFIT`: A program for semi-automatic variogram modelling — Paper 61. In *Centre For Computational Geostatistics, Report 5*, 2003.

[30] Oy Leuangthong. Advances in experimental design matrix for establishing sensitivity — Paper 102. In *Centre For Computational Geostatistics, Report 7*, 2005.

[31] Oy Leuangthong, Chris Prins, and Clayton V. Deutsch. SGSIM_LVA: gaussian simulation with locally varying angles — Paper 408. In *Centre For Computational Geostatistics, Report 8*, 2006.

[32] S. Lyster and C.V. Deutsch. Short note: An entropy-based approach to establish mps templates — Paper 114. In *Centre For Computational Geostatistics, Report 8*, 2006.

[33] S. Lyster and C.V. Deutsch. Tisis: A program to perform full indicator cosimulation using a training image — Paper 406. In *Centre For Computational Geostatistics, Report 8*, 2006.

[34] S. Lyster, C.V. Deutsch, and T. Dose. A new mps simulation algorithm based on gibbs sampling — Paper 101. In *Centre For Computational Geostatistics, Report 8*, 2006.

[35] S. Lyster, J.M. Ortiz C., and C.V. Deutsch. `MPASIM`: A multiple-point annealing simulation program — Paper 402. In *Centre For Computational Geostatistics, Report 6*, 2004.

[36] David F. Machuca-Mory and Clayton V. Deutsch. Estimation with non stationary first and second moments — Paper 118. In *Centre For Computational Geostatistics, Report 9*, 2007.

[37] David F. Machuca-Mory and Clayton V. Deutsch. Flexible change of support model suitable for a wide range of mineralization styles — Paper 303. In *Centre For Computational Geostatistics, Report 9*, 2007.

[38] David F. Machuca-Mory and Clayton V. Deutsch. Inference of the nugget effect and variogram range with sample compositing — Paper 123. In *Centre For Computational Geostatistics, Report 9*, 2007.

[39] David F. Machuca-Mory and Clayton V. Deutsch. Non stationary variograms based on continuously varying weights — Paper 117. In *Centre For Computational Geostatistics, Report 9*, 2007.

[40] J. Manchuk. Flexible optimization of stope boundaries — Paper 303. In *Centre For Computational Geostatistics, Report 8*, 2006.

[41] John Manchuk and Clayton V. Deutsch. Sensitivity analysis and the value of information in gaussian multivariate prediction — Paper 115. In *Centre For Computational Geostatistics, Report 7*, 2005.

[42] John Manchuk, Oy Leuangthong, and Clayton V. Deutsch. A generalized direct log-normal simulation algorithm with software: `slogsim` — Paper 108. In *Centre For Computational Geostatistics, Report 7*, 2005.

[43] J. A. McLennan and C. V. Deutsch. Implicit boundary modeling — Paper 102. In *Centre For Computational Geostatistics, Report 8*, 2006.

[44] Jason A. McLennan, Weishan Ren, Oy Leuangthong, and C. V. Deutsch. Optimization of well positions for sagd — Paper 206. In *Centre For Computational Geostatistics, Report 7*, 2005.

[45] D. F. Machuca Mory and C. V. Deutsch. Deriving indicator direct and cross variograms from a normal scores variogram model — Paper 404. In *Centre For Computational Geostatistics, Report 8*, 2006.

[46] D. F. Machuca Mory and C. V. Deutsch. A program for robust calculation of drillhole spacing in three dimensions — Paper 309. In *Centre For Computational Geostatistics, Report 8*, 2006.

[47] C. Neufeld. *Guide to Recoverable Reserves with Uniform Conditioning*. Centre for Computational Geostatistics, Edmonton, Canada, 2005.

[48] C. Neufeld and C.V. Deutsch. Data integration with non-parametric bayesian updating — Paper 105. In *Centre For Computational Geostatistics, Report 8*, 2006.

[49] Chad Neufeld and Clayton V. Deutsch. Angle rotations in GSLIB — Paper 403. In *Centre For Computational Geostatistics, Report 6*, 2004.

[50] Chad Neufeld and Clayton V. Deutsch. Developments in semiautomatic variogram fitting — Paper 404. In *Centre For Computational Geostatistics, Report 6*, 2004.

[51] Chad Neufeld, Julián Ortiz C., and Clayton V. Deutsch. A non-stationary correction of the probability field covariance bias — Paper 103. In *Centre For Computational Geostatistics, Report 7*, 2005.

[52] Chad Neufeld and Brandon Wilde. A global kriging program for artifact-free maps — Paper 403. In *Centre For Computational Geostatistics, Report 7*, 2005.

[53] Julián Ortiz C. and Clayton V. Deutsch. Testing pseudo-random number generators — Paper 17. In *Centre For Computational Geostatistics, Report 3*, 2001.

[54] Julián Ortiz C. and Clayton V. Deutsch. Local recoverable reserves prediction with block lu simulation — Paper 304. In *Centre For Computational Geostatistics, Report 9*, 2007.

[55] Julián Ortiz C., Clayton V. Deutsch, and Chad Neufeld. New software for fitting indicator covariances and for indicator kriging/simulation — Paper 401. In *Centre For Computational Geostatistics, Report 7*, 2005.

[56] Julián Ortiz C., Steven Lyster, and Clayton V. Deutsch. Scaling multiple point statistics for non-stationary geostatistical modeling — Paper 111. In *Centre For Computational Geostatistics, Report 7*, 2005.

[57] Weishan Ren. Short note on conditioning turning bands realizations — Paper 405. In *Centre For Computational Geostatistics, Report 7*, 2005.

[58] Weishan Ren and Clayton V. Deutsch. Exact downscaling in geostatistical modeling — Paper 101. In *Centre For Computational Geostatistics, Report 7*, 2005.

[59] B. Wilde and Clayton V. Deutsch. A new approach to calculate a robust variogram for volume variance calculations and kriging (deriving the z-variogram from the y-variogram) — Paper 308. In *Centre For Computational Geostatistics, Report 7*, 2005.

[60] Brandon Wilde and Clayton V. Deutsch. A comparison of z variograms obtained by transformation using hermite polynomials and monte carlo simulation — Paper 407. In *Centre For Computational Geostatistics, Report 9*, 2007.

[61] Brandon Wilde and Clayton V. Deutsch. Feasibility grade control (`FGC`): Simulation of grade control on geostatistical realizations — Paper 301. In *Centre For Computational Geostatistics, Report 9*, 2007.

[62] Brandon Wilde and Clayton V. Deutsch. Modeling multiple coal seams using volume functions — Paper 108. In *Centre For Computational Geostatistics, Report 9*, 2007.

[63] Brandon Wilde and Clayton V. Deutsch. Wide array declustering for representative distributions — Paper 110. In *Centre For Computational Geostatistics, Report 9*, 2007.

[64] Fuenglarb Zabel, Michael J. Pyrcz, and Clayton V. Deutsch. Conditioning event-based channelized facies models to multiple well data — Paper 202. In *Centre For Computational Geostatistics, Report 7*, 2005.

[65] S. Zanon and C. V. Deutsch. Predicting reservoir performance with multiple geological, geophysical, and engineering variables: Bayesian updating under a multivariate gaussian model — Paper 50. In *Centre For Computational Geostatistics, Report 5*, 2003.

[66] Xingquan Zhang and Michael J. Pyrcz. Advances in stochastic surface modeling: conditioning, facies and non stationarity — Paper 205. In *Centre For Computational Geostatistics, Report 8*, 2006.