Wide Array Declustering for Representative Distributions
(The Ultimate DECLUS Program)

Brandon Wilde and Clayton V. Deutsch
Centre for Computational Geostatistics (CCG) – University of Alberta

An essential step in geostatistical simulation and model validation is establishing representative distributions. A representative distribution is required for rock type proportions and for each variable within each rock type. These distributions must be constructed accounting for preferential clustering of sample locations, large-scale geologic trends and uncertainty in the sample statistics. There are a number of techniques for declustering including cell declustering and accumulation of estimation weights. Each technique has pros and cons. We propose a unified approach that implements a wide array of declustering methods and integrates the results by an expert system. There are four methods with a series of options – these results form the basis for the recommended representative distribution and the allowable tolerance.

Introduction

Geostatistical simulation is being used increasingly to assess uncertainty and the impact of heterogeneity on process performance/design. Geostatistical simulation amounts to drawing realizations from a multivariate distribution model. The histogram or univariate distribution is the most important input parameter of a multivariate distribution. We also require a representative distribution to check alternative models and to ensure that inappropriate implementation decisions are not introducing bias.

Review of Existing Methods

There are many techniques for declustering. They have pros and cons. Some are useful in certain settings and some are not. A central feature of all declustering schemes is the assignment of a non-negative weight to each data within the population. Then, the weights are standardized to sum to one. The cdf and all summary statistics are calculated with the weights. Consider *n* data. Equal weighting would amount to set each weight proportional to 1:

$$\lambda_i \propto 1,\ i = 1,\ldots,n \tag{1}$$

The weights are standardized to sum to one before being used for cdf generation. The weights are sometimes standardized to sum to the number of data for visualization and checking.

Cell Declustering

The technique of cell declustering is commonly used (Journel, 1983; Deutsch, 1989). Cell declustering starts by dividing the volume of interest into a grid of cells $l=1,\ldots,L$ (see Figure 1). The number of occupied cells are counted: L_o, $L_o \le L$, and the number of data in each occupied cell n_{lo}, $l_o = l=1,\ldots,L_o$ are also counted where

$$\sum_{lo=1}^{Lo} n_{lo} = n = \text{the number of data} .$$

Each data is assigned a weight inversely proportional to the number of data in the same cell:

$$\lambda_i \propto \frac{1}{n_l} \tag{2}$$

The weights will sum to the number of occupied cells L_o and are standardized as appropriate. The choice of cell size is critical. A good rule is to choose the cell size at the spacing of the data in the widely spaced areas. The shape of the cells depends on the geometric configuration of the data. The shape is adjusted to conform to major directions of preferential sampling. Fixing the cell size and changing the origin often leads to different declustering weights. This artifact is avoided by considering a number of different origin locations for each cell size. The declustering weights are averaged for each origin offset. Cell declustering is robust and stable in most situations. Its drawbacks stem from the subjectivity of cell size selection, and its sensitivity to trends and boundaries. People tend to choose a too-large cell size. Conventional practice of taking the cell size that leads to the minimum declustered mean may lead to a biased low result. It is common to plot the declustered mean vs. the cell size for many different cell sizes. An example of this plot is shown in Figure 2. We use this plot in creating the output results for wide array declustering.

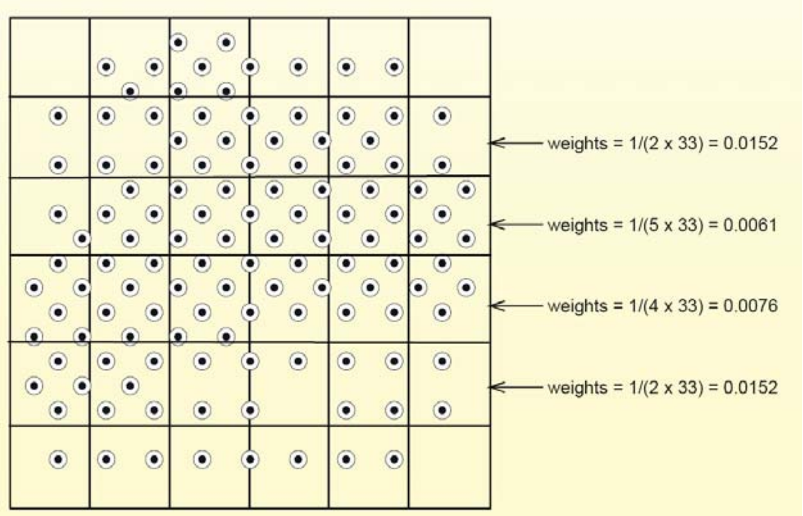

weights = 1/(2 x 33) = 0.0152
weights = 1/(5 x 33) = 0.0061
weights = 1/(4 x 33) = 0.0076
weights = 1/(2 x 33) = 0.0152

Figure 1: Example of 2-D cell declustering. The area of interest is divided into a grid of cells. The number of occupied cells is determined, in this case, 33. Each data is then weighted inversely by the number of data in the cell. This weight is standardized by the number of occupied cells.

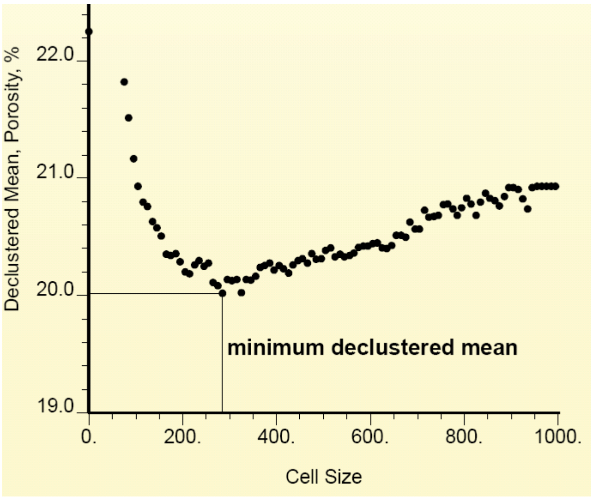


Figure 2: Declustered mean vs. cell size. This plot is commonly generated as a means to determine the most appropriate cell size for cell declustering. This plot will form the basis for our wide array declustering output.

Volume of Influence

The volume of influence could be calculated analytically; however, we normally consider a fine grid over the domain of interest and accumulate the number of times a data is closer to the nodel than any other data. The weight assigned to that data is proportional to the number of nodes closest to that data.

Polygonal declustering, as this method is often called, is simple and easy to understand. Though it appears objective, it is only truly applicable when the boundaries or limits of the volume of interest are well known. When these boundaries are not well known, the weights given to the edge or end samples are not accurate. This phenomenon is demonstrated in Figure 3.

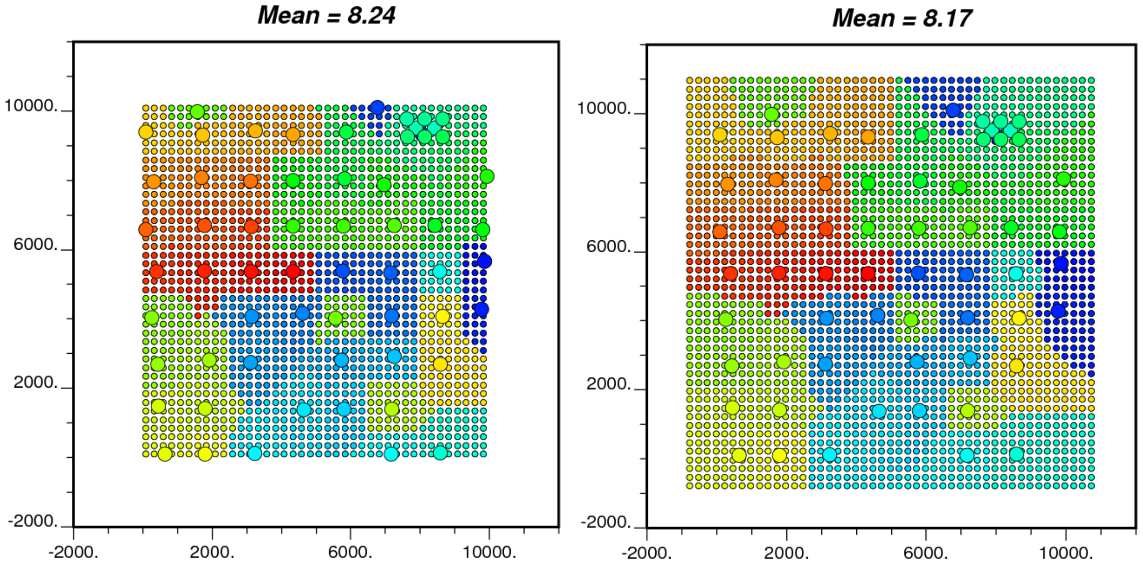


Figure 3: An example of the weakness in polygonal declustering when the boundaries/limits are not well defined. As the fine grid expands, more weight is given to those data located at the edge of the area of interest. In this case, the edge data have low values meaning that as the grid expands the mean decreases.

The declustered distribution is sensitive to both the grid size and the grid density. In the wide array declustering program proposed herein, the user specifies each of these parameters. The grid density can be any positive value, but it is recommended that it be from 1 to 3 where 1 creates a sparse grid and 3 produces a denser grid. Where computational time is not an issue, it is recommended that a denser grid be utilized.

There are two ways for specifying the declustering grid boundaries. The first way is to use a square grid with its size being limited by specifying a grid expansion factor which indicates how far the fine grid extends beyond the extents of the data. Figure 3 was generated using grid expansion factors of 1.0 and 1.2 respectively. A grid expansion factor of 1.0 will align the edges of the grid precisely with the extents of the data. As the expansion factor increases, so too does the size of the grid. A larger grid gives more weight to the data near the edge of the field. It has been observed that a grid expansion factor of 1.05 gives the best results.

The second way to limit or specify the declustering grid size is to find the ‘loneliest’ or most outlying data and the distance from that data to its nearest neighbor as shown in Figure 4. This distance value is then used to limit the distance at which a grid node can be assigned to a data point as shown in Figure 5. Though a grid node may be closest to a certain data, if it is farther away than this arbitrary distance, it will not be added to the weighting of that data. A more realistic example using more data is shown in Figure 6. Note that the mean is higher than the mean calculated in Figure 3 with the expanded grid.

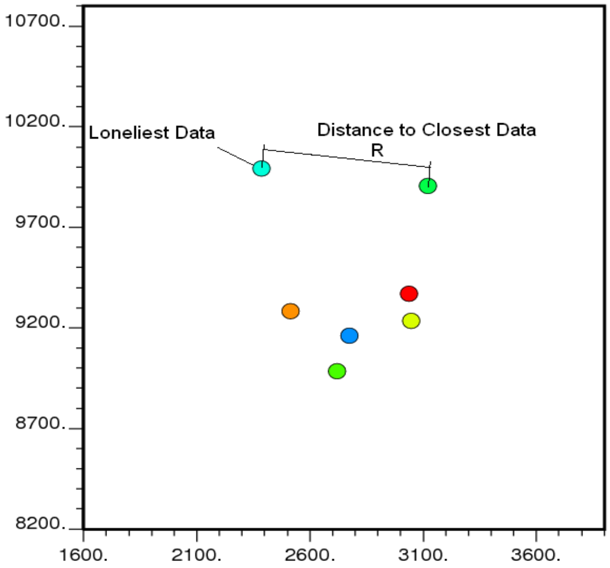


Figure 4: The distance at which a node is assigned to a data is limited by finding the 'loneliest' data and the distance from that data to its nearest neighbor.

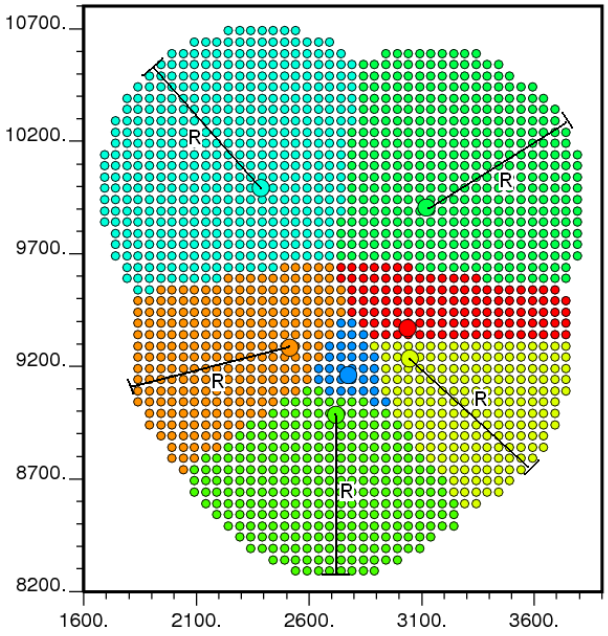


Figure 5: Only those nodes that are within the calculated distance are assigned to the data point.

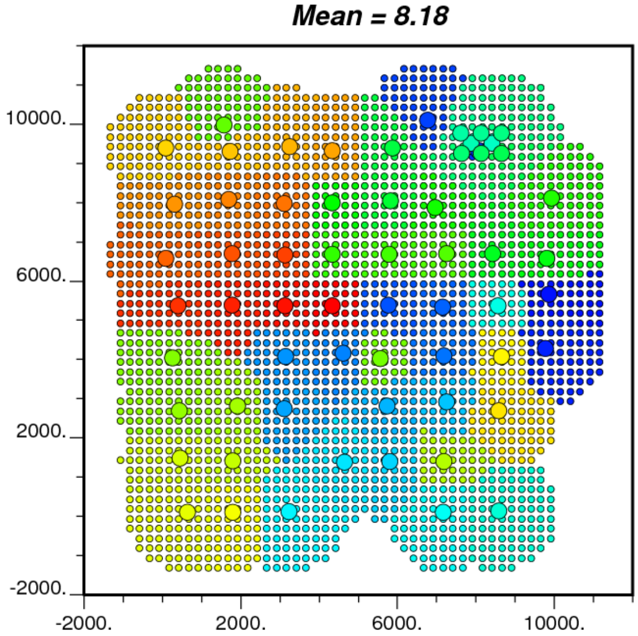


Figure 6: The distance at which a declustering grid node can be assigned to a data point is limited by the shortest distance to the 'loneliest' data.

Ordinary Kriging

Each location in the domain of interest is estimated with a local search neighborhood and ordinary kriging tuned for reasonable estimates. The declustering weights are taken as the accumulated kriging weight that each data receives.

$$\lambda_i \propto \sum_{\substack{\text{all } \mathbf{u} \text{ locations} \\ \text{within domain}}} \lambda_{OK}(\mathbf{u}_i) \tag{5}$$

Different amounts for the maximum amount of data used to generate an estimate are used to give an idea how the number of data used to perform estimation effects the declustered mean. This is specified by the user.

Global Ordinary Kriging

A reasonable large scale trend is obtained by global ordinary kriging (or very large search neighborhoods) with a variogram arbitrarily set to have a 30-50% nugget effect and a range about 30-50% of the size of the domain (accounting for major anisotropy). These weights could also be accumulated for declustering weights.

`WA_DECLUS` Program

The `WA_DECLUS` program was assembled from the `kt3d` and `declus` codes in GSLIB with significant modifications to proceed in parallel and post process the results. The parameter file for the program is shown in Figure 7. There are six groups of parameters. The first group contains those parameters which are common to all four types of declustering. This includes the data file, column numbers, trimming limits, a yes/no indicator for each declustering type, whether we are seeking a minimum or maximum declustered mean (depending on whether low-valued or high-valued areas were preferentially sampled), and the output files. These parameters are specified in lines 6-13 respectively.

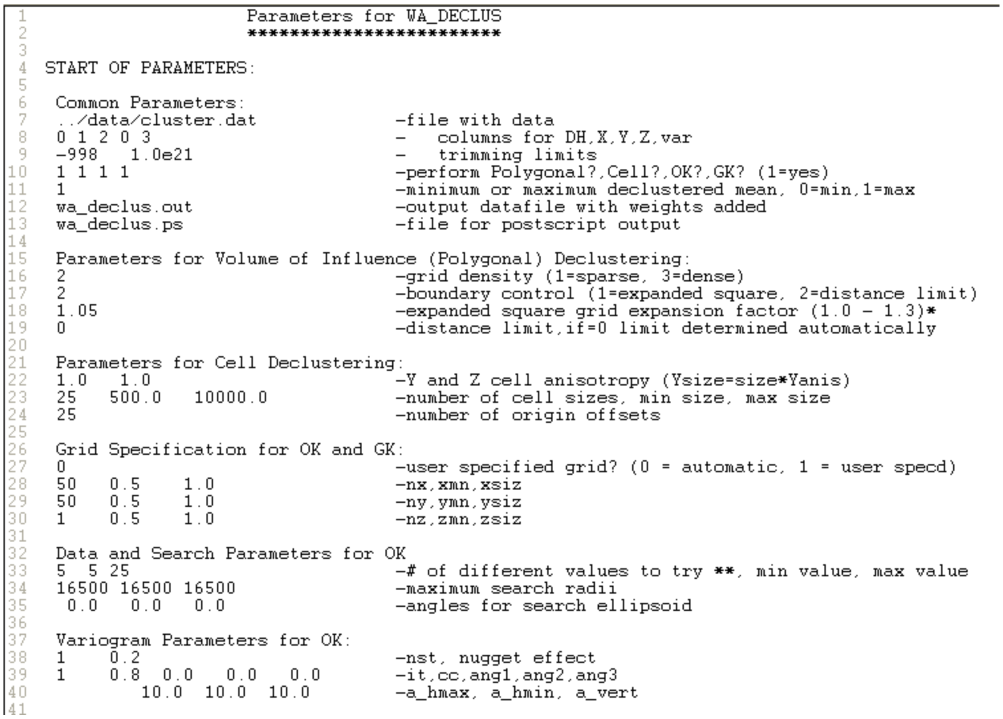
The second group of parameters relate to the volume of influence or polygonal method of declustering. The grid density is specified first followed by the fine grid use parameter which tells the program whether the expanded square grid or a distance limit will be used for limiting the nodes assigned to ‘edge’ data points. The third parameter in this section relates to the expanded square grid and specifies the grid expansion factor. This number is in the range of 1.0-1.3. It has been observed that a value of 1.05 gives optimal results. The final parameter relates to the distance limited technique for assigning grid nodes to data points. The user can either let the program determine the distance limit or can input the distance limit themselves. These parameters are input in lines 16-19 respectively.

The third group of parameters relates to cell declustering. These are the same parameters utilized by the GSLIB program `declus`. Input here are the Y and Z cell anisotropy, the number and range of cell sizes, and the number of origin offsets. These parameters are input in lines 22-24 respectively.

The fourth group of parameters specifies the grid necessary for the ordinary kriging and global ordinary kriging methods. There is an option for the program to create this grid automatically which it will do if the first parameter in this section is set to zero. If this parameter is not set to zero, the grid specification in the proceeding lines is used. This is the typical GSLIB grid specification. These parameters are specified in lines 27-30 respectively.

The fifth group of parameters relates to the amount of data used to generate an estimate and the search used to identify these data for the ordinary kriging method. The effect that using different amounts of data to estimate has on the declustering results can be observed by specifying different ‘maximum number of data’ cutoffs. The search radii and angles for the search ellipsoid are common GSLIB parameters. These parameters are specified in lines 33-35 respectively.

The sixth group of parameters is simply the variogram specification as per typical GSLIB format. The variogram specification begins on line 37. The number of lines it takes to specify the variogram depends on the number of structures.



```
1                      Parameters for WA_DECLUS
2                      ************************
3
4   START OF PARAMETERS:
5
6    Common Parameters:
7    ../data/cluster.dat             -file with data
8    0 1 2 0 3                       -   columns for DH,X,Y,Z,var
9    -998   1.0e21                   -   trimming limits
10   1 1 1 1                         -perform Polygonal?,Cell?,OK?,GK? (1=yes)
11   1                               -minimum or maximum declustered mean, 0=min,1=max
12   wa_declus.out                   -output datafile with weights added
13   wa_declus.ps                    -file for postscript output
14
15   Parameters for Volume of Influence (Polygonal) Declustering:
16   2                               -grid density (1=sparse, 3=dense)
17   2                               -boundary control (1=expanded square, 2=distance limit)
18   1.05                            -expanded square grid expansion factor (1.0 - 1.3)*
19   0                               -distance limit,if=0 limit determined automatically
20
21   Parameters for Cell Declustering:
22   1.0   1.0                       -Y and Z cell anisotropy (Ysize=size*Yanis)
23   25   500.0   10000.0            -number of cell sizes, min size, max size
24   25                              -number of origin offsets
25
26   Grid Specification for OK and GK:
27   0                               -user specified grid? (0 = automatic, 1 = user specd)
28   50   0.5    1.0                 -nx,xmn,xsiz
29   50   0.5    1.0                 -ny,ymn,ysiz
30   1    0.5    1.0                 -nz,zmn,zsiz
31
32   Data and Search Parameters for OK
33   5  5 25                         -# of different values to try **, min value, max value
34   16500 16500 16500               -maximum search radii
35    0.0   0.0   0.0                -angles for search ellipsoid
36
37   Variogram Parameters for OK:
38   1    0.2                        -nst, nugget effect
39   1    0.8  0.0   0.0   0.0       -it,cc,ang1,ang2,ang3
40           10.0  10.0  10.0        -a_hmax, a_hmin, a_vert
41
```

Figure 7: Parameter file for WA_DECLUS program.

Examples

This program has been tested on many different data sets, all which gave expected results. We show the results from three of those data sets herein.

The first example comes from porosity data contained in the Amoco.dat data file. There is a cluster of data in the north-east corner of the lease with the clustered data being in an area of high porosity. The non-declustered mean is therefore high and any declustered means will be lower than this. By performing declustering we see that the mean was inflated by about 3% due to the clustered data. This information is all displayed in Figure 8.

The second example is based on gold grades from the red.dat data file. The clustering in this area is not so obvious, but there are definite areas which have been more densely sampled, again in high-grades. This has resulted in a higher-than-expected mean. Using the `wa_declus` program shows us that the non-declustered mean was overstated by about 22%. The true mean is around 1.0g/t as opposed to 1.36g/t.

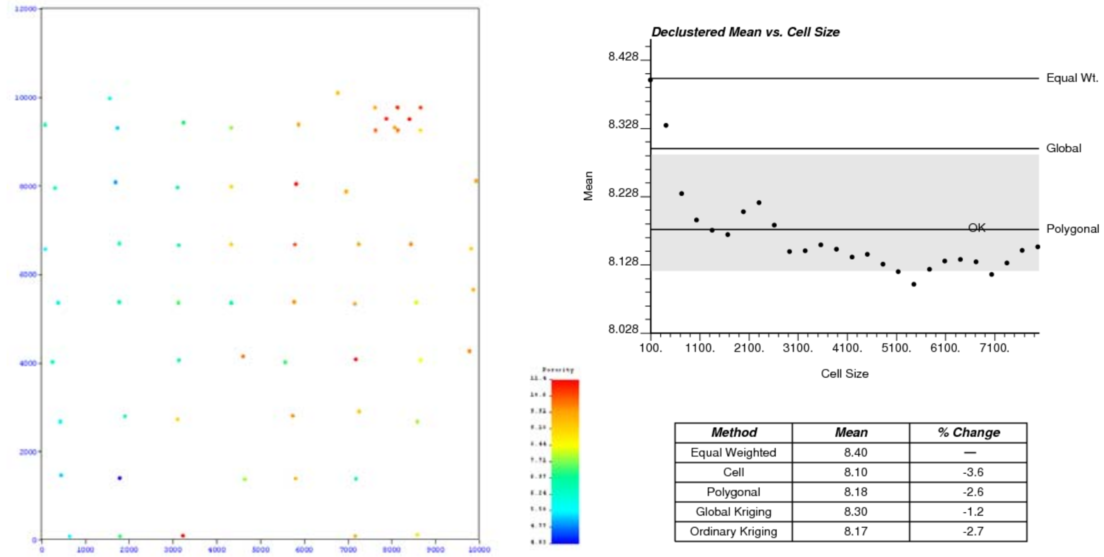
The third and final example comes from the cluster.dat data file. The clustering is, once again, quite obvious, but is now located throughout the area of interest. The data is clustered in areas of high grade creating an exaggerated mean value. Performing the different declustering techniques demonstrates that the non-declustered mean is indeed too high. It is about 35% higher than the declustered means. We can choose the most appropriate method and use the associated weights for generating a representative distribution.

Conclusions

Establishing the representative distribution for every variable is a longstanding problem. The wide array solution presented here is practical and useful for this purpose. A distribution of declustered means is created against which simulated realization averages and kriged model averages may be checked.

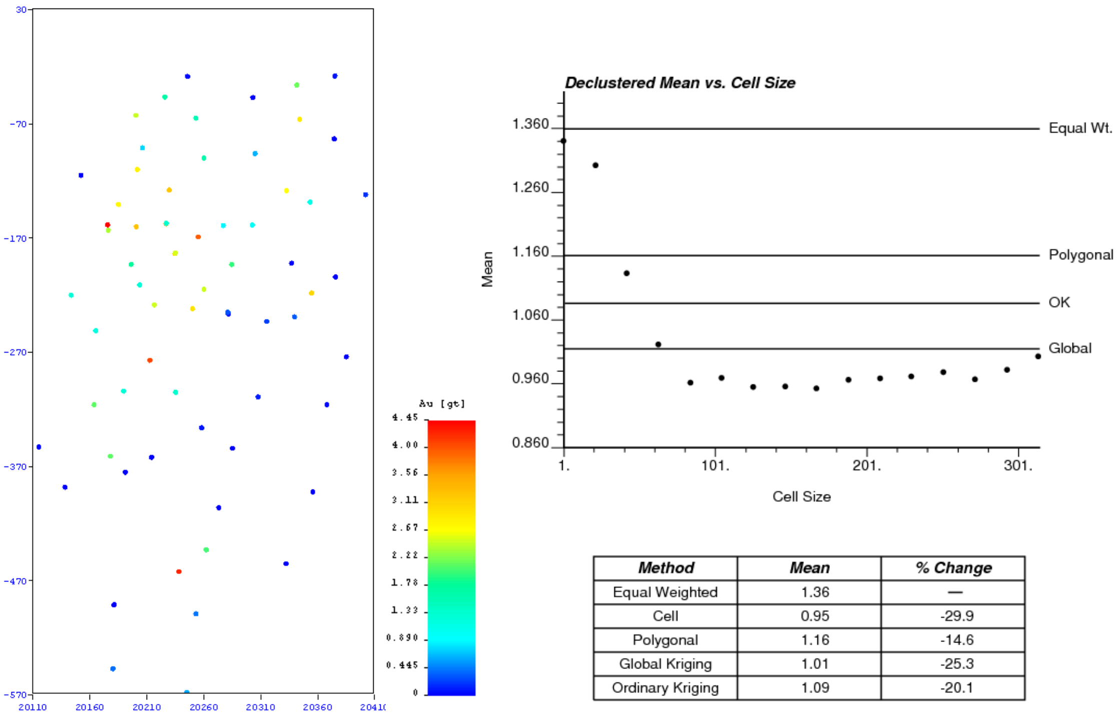
References

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



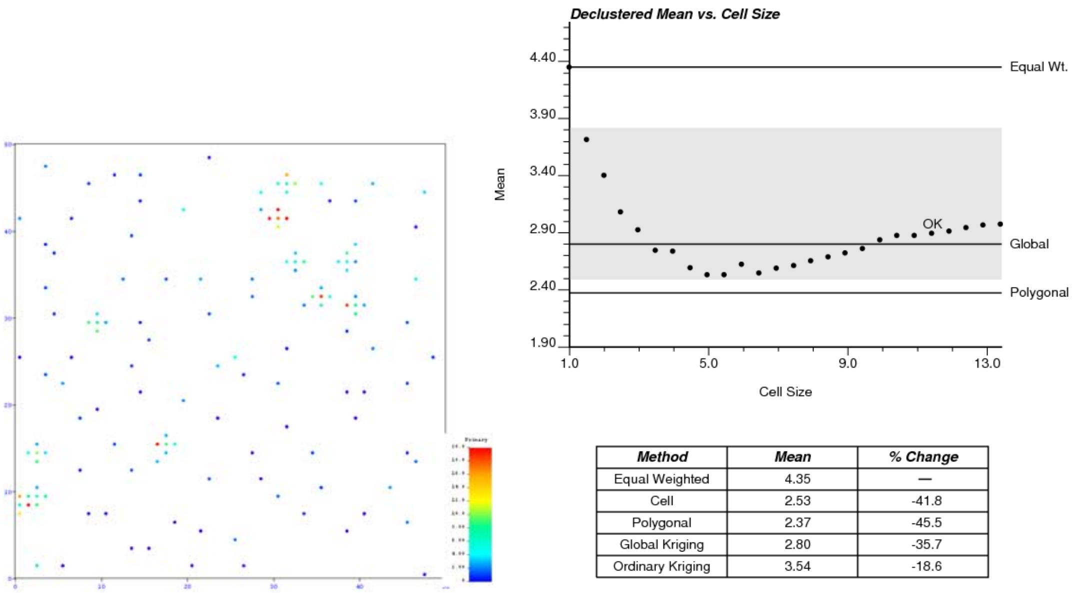
Method	Mean	% Change
Equal Weighted	8.40	—
Cell	8.10	-3.6
Polygonal	8.18	-2.6
Global Kriging	8.30	-1.2
Ordinary Kriging	8.17	-2.7

Figure 8: The location map and declustering results from the Amoco.dat data set. There group of clustered data within high-valued areas has exaggerated the mean. The declustering results have calculated more realistic mean values.



Method	*Mean*	*% Change*
Equal Weighted	1.36	—
Cell	0.95	-29.9
Polygonal	1.16	-14.6
Global Kriging	1.01	-25.3
Ordinary Kriging	1.09	-20.1

Figure 9: The location map and declustering results from the red.dat data set. The clustering in the high-grade areas has inflated the mean. Declustering has weighted the data to produce a more practical mean.



Method	*Mean*	*% Change*
Equal Weighted	4.35	—
Cell	2.53	-41.8
Polygonal	2.37	-45.5
Global Kriging	2.80	-35.7
Ordinary Kriging	3.54	-18.6

Figure 10: Location map and declustering results for the cluster.dat data file. The high-grade clusters have exaggerated the mean. Declustering has generated a better distribution with a lower mean.