2002

Direct sequential simulation algorithms in geostatistics

Robyn Robertson
Edith Cowan University

Edith Cowan University

Copyright Warning

Direct Sequential Simulation Algorithms in Geostatistics

A Thesis Submitted to the

Faculty of Communications, Health and Science

Edith Cowan University

Perth, Western Australia

by Robyn Robertson

In Partial Fulfillment of the Requirements for the Degree

of

Bachelor of Science Honours (Mathematics)

December 2002

Supervisors: Dr Ute Mueller

Associate Professor Lyn Bloom

USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

Abstract

Conditional sequential simulation algorithms have been used in geostatistics for many years but we currently find new developments are being made in this field. This thesis presents two new direct sequential simulation with histogram reproduction algorithms and compares them with the efficient and widely used sequential Gaussian simulation algorithm and the original direct sequential simulation algorithm. We explore the possibility of reproducing both the semivariogram and the histogram without the need for a transformation to normal space, through optimising an objective function and placing linear constraints on the local conditional distributions. Programs from the GSLIB Fortran library are expanded to provide a simulation environment. An isotropic and an anisotropic data set are analysed. Both sets are positively skewed and the exhaustive data is available to define global target distributions and for comparing the cumulative distribution functions of the simulated values.

To

Nathan, Kaylee, Stephen & Louise.

Declaration

I certify that this thesis does not, to the best of my knowledge and belief:

(i) *incorporate without acknowledgement any material previously submitted for a degree or diploma in any institution of higher education;*

(ii) *contain any material previously published or written by another person except where due reference is made in the text; or*

(iii) *contain any defamatory material.*

Signature: .....

Date: 28/2/03

Acknowledgements

I wish to express my deepest thanks to my children Nathan, Kaylee, Stephen and Louise for their love, support and patience which has enabled me to complete my degree. I also wish to thank my supervisors Ute Mueller and Lyn Bloom for their help and guidance in completing my degree and thesis at Edith Cowan University.

Table of Contents

1 Introduction

1.1 Background and Significance

Geostatistics developed from a need to evaluate recoverable reserves in mineral deposits and provides statistical tools for the description and modelling of spatial and spatiotemporal variables. It takes into account both the structure and the randomness inherent in any deposit. In 1962 G. Matheron defined geostatistics as "the application of the formalism of random functions to the reconnaissance and estimation of natural phenomena." Geostatistics can be considered as a set of statistical procedures that deal with the characterisation of spatial attributes. Geostatistics is now used in many different fields, wherever there is a need for evaluating spatially correlated data, such as in mining, petroleum, oceanography, hydrogeology and environmental science.

The two principal components of geostatistics are estimation and simulation. Estimation is used to infer attribute values at unsampled locations from the (known) values at sampled locations. The most common geostatistical estimation method is kriging, which is a generalised linear regression technique that provides at each location a best linear unbiased estimator (BLUE) for the unknown attribute studied. Many variants of kriging have been developed, but all rely on the same concepts. Three types of parametric kriging for a single attribute can be differentiated depending on the model used for the mean (Remy et al, 2001). These are termed simple kriging (SK), ordinary kriging (OK) and kriging with a trend model (KT). A non-parametric type of kriging is indicator kriging (IK).

Least-squares interpolation algorithms tend to smooth out local details of the spatial variation of the attribute (Goovaerts, 1998). Kriging tends to unevenly smooth the data, that is, kriging estimates have less spatial variability than the real values and the smoothing is inversely proportional to the data density. Consequently values below the sample mean are overestimated and values above the sample mean are underestimated. The smoothing distortion is also evidenced by the experimental semivariogram of the estimates differing from the sampling experimental semivariogram, and the histogram of the sample differing from the histogram

of the estimated values (Olea, 1999).

Conditional simulation was initially developed to correct the smoothing effect produced by kriging (Deutsch & Journel, 1998). Conditional simulations are spatially consistent Monte Carlo simulations (Chilès and Delfiner, 1999) and are used to characterise the uncertainty associated with the prediction of attribute values at unsampled locations while honouring the data values at sample locations. A large number of equiprobable realisations are generated so as to obtain global accuracy by the reproduction of properties such as histograms and semivariograms. Conditional simulations are used qualitatively to obtain maps of spatial variability, and quantitatively to evaluate the effect of spatial uncertainty on the results of complex procedures, allowing for sensitivity and risk analysis.

Sequential simulation is a widely accepted and computationally efficient tool used to obtain simulations that reproduce desired properties through the use of conditional distributions. Sequential Gaussian simulation is one of the main methods that rely on the multiGaussian approach. At each unsampled location, an observation is randomly drawn from the normal distribution with mean equal to the simple kriging mean and variance equal to the simple kriging variance. Journel (1994) showed that this normality assumption can be relaxed, and any type of local conditional distribution can be used to simulate the values, as long as its mean and variance are equal to the simple kriging parameters. This led to the development of direct sequential simulation (dss), which ensures variogram reproduction but not necessarily global histogram reproduction.

In this study we explore two direct sequential simulation algorithms with histogram reproduction, the first using the one norm (dssℓ1) and the second the two norm (dssℓ2). We examine the possibility of a simulation algorithm being able to reproduce both the histogram and the experimental semivariogram model without the need for a transformation to normal space. This project allows us to link numerical analysis, operations research and computer programming with geostatistics. Fortran code was developed by modifying and adding to GSLIB code, and other programs were incorporated with this to achieve the programs required to run the simulation algorithms.

1.2 Objective of the Study

In this study we develop and investigate in detail, a new direct sequential simulation technique dssℓ2 and compare it to earlier direct sequential simulation methods dss and dssℓ1 and also to sequential Gaussian simulation. This new direct sequential simulation algorithm that uses quadratic programming to determine local conditional probability distributions from which the resulting realisations will depend on. In addition, the algorithms for sequential Gaussian simulation and the earlier direct sequential simulation methods will be outlined and discussed.

These simulation algorithms are applied to sample data sets that have different statistical and spatial features, and the results are evaluated and compared. Two data sets, *Permeability* and *Potassium*, have been selected for analysis, in order to present comparisons of the simulation methods. Both data sets are positively skewed, albeit only slightly for *Potassium*. The two data sets also exhibit different patterns of spatial continuity. The *Permeability* data are isotropic, while the *Potassium* data are anisotropic.

1.3 Thesis Outline

Chapter 2 presents the theoretical background of the random function model, stationarity and simple kriging. In Chapter 3 we look at sequential simulation algorithms, outline the mathematical background of optimisation and discuss the implementation of the simulation algorithms. In Chapter 4 we discuss the programming algorithms developed in the research. In Chapter 5 and Chapter 6 the isotropic data set and the anisotropic data set respectively, are presented. We present the quantitative and qualitative performance assessments used in this study in Chapter 7. In Chapter 8 we discuss the results and findings of the study.

1.4 Notation

The geostatistical notation used in this thesis follows Goovaerts (1997) and the GSLIB user's manual (Deutsch & Journel, 1998). In particular:

$\forall$	for all
$\mathcal{A}$	study region
a	range parameter
$C(0)$	stationary variance of the random variable $Z(\mathbf{u})$
$C(\mathbf{h})$	stationary covariace of the random function $Z(\mathbf{u})$ for lag $\mathbf{h}$
$Cov\{\cdot\}$	covariance
$E\{\cdot\}$	expected value
$F(\mathbf{u};z)$	cumulative distribution function of the random variable $Z(\mathbf{u})$
$F(\mathbf{u};z\vert(n))$	conditional cumulative distribution function of a random variable $Z(\mathbf{u})$
$F(\mathbf{u}_1,\ldots,\mathbf{u}_N;z_1,\ldots,z_N)$	multivariate cumulative distribution function
$g(\mathbf{h})$	model semivariogram at lag vector $\mathbf{h}$
$\gamma(\mathbf{h})$	semivariogram at lag vector $\mathbf{h}$
$\hat{\gamma}(\mathbf{h})$	experimental semivariogram at lag vector $\mathbf{h}$
$h=\vert\mathbf{h}\vert$	separation distance or lag
$\mathbf{h}$	separation vector
K	number of threshold values z_k
$\lambda_\alpha^{SK}(\mathbf{u})$	Simple Kriging weight of attribute value at sampled location $\mathbf{u}_\alpha$ for estimation of the attribute value at location $\mathbf{u}$
$\mathbf{m}(\mathbf{u})$	expected value of the random variable $Z(\mathbf{u})$
m	constant mean of the random variable $Z(\mathbf{u})$
$N(\mathbf{h})$	number of sample data pairs separated by lag vector $\mathbf{h}$

n	number of data values $z(\mathbf{u}_\alpha)$ available over the region A
$\rho(\mathbf{h})$	correlogram of the random function $Z(\mathbf{u})$ at lag vector $\mathbf{h}$
$\|\mathbf{u}\|_1$	one-norm
$\|\mathbf{u}\|_2$	two-norm
$Var\{\bullet\}$	variance
$Z(\mathbf{u})$	random variable at sample location $\mathbf{u}$
$z(\mathbf{u})$	actual attribute value at location $\mathbf{u}$
$z(\mathbf{u}_\alpha)$	sample attribute value at location $\mathbf{u}$
$Z(\mathbf{u}_\alpha)$	random variable at location $\mathbf{u}_\alpha$
$Z^*(\mathbf{u})$	random variable of estimated value at location $\mathbf{u}$
$Z^*_{SK}(\mathbf{u})$	Simple Kriging estimator of $Z(\mathbf{u})$

2 Geostatistical Framework

2.1 Random Functions

Geostatistics deals with the characterisation of spatial attributes in a given region $\mathcal{A}$ in two- or three-dimensional space. The attribute values are usually only known at some locations in the region $\mathcal{A}$ and in order to carry out any statistical inference it is necessary to impose a conceptual model that will allow one to obtain a realisation of the attribute over the entire region. This conceptual model is known as the random function model. Suppose that the attribute values are known at the locations $\mathbf{u}_\alpha \in \mathcal{A}$, $\alpha = 1, \ldots, n$. A known sample value $z(\mathbf{u}_\alpha)$ is considered as one particular realisation of a random variable $Z(\mathbf{u}_\alpha)$. Any unknown attribute $z(\mathbf{u})$ is regarded as one realisation of a random variable $Z(\mathbf{u})$ (Armstrong, 1998; Chilès & Delfiner, 1999; Goovaerts, 1997). The random variable $Z(\mathbf{u})$ is completely defined by its cumulative distribution function given by

$$F(\mathbf{u}; z) = \Pr\{Z(\mathbf{u}) \leq z\} \quad \text{for all } z \in \mathbb{R}. \tag{1}$$

The family of (usually) dependent random variables $\{Z(\mathbf{u}), \mathbf{u} \in \mathcal{A}\}$ is called a random function. The random function is fully characterised by the set of all its N-variate cumulative distribution functions, for any number N and any choice of the N locations $\mathbf{u}_n$, $n = 1, \ldots, N$:

$$F(\mathbf{u}_1, \ldots, \mathbf{u}_N; z_1, \ldots, z_N) = \Pr\{Z(\mathbf{u}_1) \leq z_1, \ldots, Z(\mathbf{u}_N) \leq z_N\} \tag{2}$$

A multivariate cumulative density function is used to model the joint uncertainty about the N values $z(\mathbf{u}_1), \ldots, z(\mathbf{u}_N)$. Generally, the number of data available is insufficient to infer the joint distribution function, so in practice the spatial analysis is limited to cumulative density functions involving no more than two locations at a time, and their corresponding moments. The first two moments of the distribution provide an acceptable approximate solution.

2.2 Distributions

Most of the theoretical concepts in geostatistics rely on data that follow a particular probability distribution. The most widely used of these is the Gaussian or 'normal' distribution whose probability density function, called the normal curve, is the symmetric bell-shaped curve with positive and negative tails that stretch to infinity in both directions. The probability density function of the normal probability distribution (Walpole & Myers, 1989) with mean μ and variance σ^2, is given by

$$g(z) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{1}{2}\left(\frac{z-\mu}{\sigma}\right)^2\right), \qquad \text{where} \quad -\infty < z < \infty. \tag{3}$$

The standard normal distribution has mean $\mu = 0$ and variance $\sigma^2 = 1$, and in this case Equation (3) becomes

$$g(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) \qquad \text{where} \quad -\infty < z < \infty. \tag{4}$$

Any normal random variable Z with mean μ and standard deviation σcan be transformed to a standard normal random variable Y by letting

$$Y = \frac{Z-\mu}{\sigma}.$$

Another distribution often used in geostatistics is the lognormal distribution, where the logarithms of the data values are normally distributed. The lognormal model is a natural choice for positively skewed data such as gold grades, pollution levels and permeability. The random variable Z is lognormal if $Y = \log Z$ is normal. A logarithmic transformation can convert a skewed variable into a more symmetric form, and it may also be useful in stabilising the variance. When the variance is proportional to the mean, a logarithmic transformation may be able to correct this condition.

2.3 Stationarity

In the case of the data under consideration in geostatistics, repeated measurements at any one location are usually impossible so a structure needs to be imposed on

the random function that enables us to carry out statistical inference. The observed data $z(\mathbf{u}_\alpha)$, $\alpha = 1, \cdots, n$ are considered as a single realisation of the process $\{Z(\mathbf{u}) : \mathbf{u} \in \mathcal{A}\}$. When replication of data is not available, this can be overcome with assumptions concerning the spatial behaviour. A random function is said to be stationary if the probabilistic structure looks similar in different parts of the study region $\mathcal{A}$. Replication within a single set of data is then possible from different subregions.

A random function is stationary within a study region $\mathcal{A}$ if its multivariate cumulative density function is invariant under translation, that is, the characteristics of a random function stay the same when shifting a given set of N points from one part of the study region to another. A random function is said to be strictly stationary if for any set of N locations $\mathbf{u}_1, \ldots, \mathbf{u}_N$ and any translation vector $\mathbf{h}$

$$F(\mathbf{u}_1, \ldots, \mathbf{u}_N; z_1, \ldots, z_N) = F(\mathbf{u}_1 + \mathbf{h}, \ldots, \mathbf{u}_N + \mathbf{h}; z_1, \ldots, z_N) \tag{5}$$

As long as two pairs of observations have the same separation vector $\mathbf{h}$, they both can contribute in the estimation of $z(\mathbf{u})$. The vector $\mathbf{h}$ is called the lag vector between two spatial locations.

A random function is said to be second-order stationary when the mean $E\{Z(\mathbf{u})\}$ exists and does not depend on the location $\mathbf{u}$, and the covariance function $C(\mathbf{h})$ exists and depends only on the separation vector $\mathbf{h}$:

$$E\{Z(\mathbf{u})\} = E\{Z(\mathbf{u}+\mathbf{h})\} \tag{6}$$

$$Cov\{Z(\mathbf{u}), Z(\mathbf{u}+\mathbf{h})\} = C(\mathbf{h}) \tag{7}$$

$$C(0) = Var\{Z(\mathbf{u})\} \tag{8}$$

Second-order stationarity assumes the existence of a finite variance. There are many physical phenomena, for example Brownian Motion (Serway & Beichner, 2000), and associated random functions that do not have a finite variance or covariance, so the assumption of strict stationarity is replaced by the weaker hypothesis of second-order intrinsic stationarity. An intrinsic random function assumes that for every vector $\mathbf{h}$ the increment $Z(\mathbf{u}+\mathbf{h}) - Z(\mathbf{u})$ is second-order stationary and is characterised by the relationships

$$E\{Z(\mathbf{u}+\mathbf{h}) - Z(\mathbf{u})\} = 0 \tag{9}$$

and

$$Var\{Z(\mathbf{u}+\mathbf{h}) - Z(\mathbf{u})\} = 2\gamma(\mathbf{h}) \tag{10}$$

where $2\gamma(\mathbf{h})$ is the variogram function. The semivariogram $\gamma(\mathbf{h})$ shows how the dissimilarity between $Z(\mathbf{u})$ and $Z(\mathbf{u}+\mathbf{h})$ changes with separation $\mathbf{h}$. The greater the value of $\gamma(\mathbf{h})$, the less close the relationship between values at points separated by h. The semivariogram is an even, nonnegative function equal to 0 at $\mathbf{h} = \mathbf{0}$:

$$\gamma(\mathbf{h}) = \gamma(-\mathbf{h}) \qquad \gamma(\mathbf{h}) \geq 0 \qquad \gamma(\mathbf{0}) = 0 \tag{11}$$

The parameters commonly used to summarise the bivariate behaviour of a stationary random function are the covariance function $C(\mathbf{h})$, correlogram $\rho(\mathbf{h})$, and semivariogram $\gamma(\mathbf{h})$ and these are related by:

$$\gamma(\mathbf{h}) = C(0) - C(\mathbf{h}) \tag{12}$$

$$\rho(\mathbf{h}) = \frac{C(\mathbf{h})}{C(0)} = 1 - \frac{\gamma(\mathbf{h})}{C(0)} \tag{13}$$

The correlogram expresses how the correlation between locations changes with spatial separation.

If a random process is second-order stationary, then it is also intinsically stationary, but the converse is not true. That is, if $C(\mathbf{h})$ is defined, then the semivariogram is necessarily defined, but the existence of the semivariogram does not imply the existence of $C(\mathbf{h})$. If the process is intrinsically stationary but not second-order stationary, the covariance function does not exist. This is evident in the power variogram $\gamma(\mathbf{h}) = b|\mathbf{h}|^p$ with $0 < p < 2$ and $b > 0$, which cannot be obtained from a covariance function as it is unbounded.

2.4 Relationship Between Covariogram and Semivariogram

Assuming that the process is second-order stationary so that $C(0)$ is defined, then $C(0) = Var\{Z(\mathbf{u})\}$. A second order stationary process has $C(\mathbf{h}) \geq 0$, from which $|C(\mathbf{h})| \leq C(0)$ and $C(0) \geq 0$. As $\|\mathbf{h}\|$ increases $C(\mathbf{h})$ tends to zero, so the semivariogram of a second-order stationary process has an asymptote equal to $C(0)$. This helps to provide a way of checking for stationarity. The semivariogram of the process

should flatten out with increasing separation distance of data points. If the semivariogram steadily increases then the process is not second-order stationary. The semivariogram is intrinsically stationary if

$$\frac{2\gamma(\mathbf{h})}{\|\mathbf{h}\|^2} \to 0 \text{ as } \|\mathbf{h}\| \to \infty \tag{14}$$

2.5 Inference and Modelling

Once a random function model is chosen, its parameters, the mean and covariance, are inferred from the sample information available over the study region $\mathcal{A}$. The sample statistics are used as estimates of population parameters, so the sample needs to be representative of the study region.

The semivariogram, rather than the covariance, is commonly used to model spatial variability, although kriging systems are more easily solved with covariance matrices (Deutsch & Journel, 1998). The semivariogram measures the average dissimilarity between data separated by a vector $\mathbf{h}$ and is inferred by the sample (experimental) semivariogram, whereas the covariance measures similarity. The sample semivariogram used for modelling is computed as half the average squared difference between the attribute values of every data pair:

$$\hat{\gamma}(\mathbf{h}) = \frac{1}{2N(\mathbf{h})} \sum_{n=1}^{N(\mathbf{h})} [z(\mathbf{u}_a) - z(\mathbf{u}_a + \mathbf{h})]^2 \tag{15}$$

where $z(\mathbf{u}_a)$ and $z(\mathbf{u}_a+\mathbf{h})$ are the data values at locations $\mathbf{u}_a$ and $\mathbf{u}_a+\mathbf{h}$ respectively, and $N(\mathbf{h})$ is the number of pairs of data values separated by the vector $\mathbf{h}$. The sample semivariogram may not tend to zero when $\mathbf{h}$ tends to zero, although by definition $\gamma(0) = 0$.

2.6 Range and Sill

The rate of increase of the sample semivariogram with distance indicates how quickly the influence of a sample reduces with distance. The sample semivariogram can increase indefinitely if the variability of the attribute has no limit at large distances,

and this is indicative of nonstationary behaviour. If the random function is second-order stationary, the sample semivariogram fluctuates about a limiting value, and the range of the spatial process is the distance at which this limit is reached. This limit is called the sill of the semivariogram and it signifies that after a certain separation distance there is no longer any correlation between samples. (Armstrong, 1998). If the semivariogram approaches its sill asymptotically, then the practical range is the value at which the semivariogram reaches 95 % of its sill. The variogram can reveal nested structures, each characterised by its own range (Chilès & Delfiner, 1999).

The sample semivariogram provides a set of experimental values for a finite number of lags, $\mathbf{h}_k$, $k = 1,...,K$, and directions. Continuous functions must be fitted to these experimental values so as to deduce semivariogram or covariance values for any possible lag $\mathbf{h}$ required by kriging, and also to smooth out sample fluctuations. (Goovaerts, 1997).

2.7 The Nugget Effect

From the definition of the semivariogram we have:

$$\gamma(\mathbf{h}) = \gamma(-\mathbf{h}) \text{ and } \gamma(0) = 0 \tag{16}$$

In some applications $\gamma(\mathbf{h})$ tends to $c_0 \neq 0$ as $\mathbf{h}$ tends to $\mathbf{0}$. This implies that observation differences at the same location have a positive variance. This is due to measurement error and/or a spatial process operating at lag distances shorter than the smallest lag observed in the data set. If this micro-scale process has sill c_{MS} and if σ^2_{ME} denotes the variance of the measurement error, then

$$c_0 = \sigma^2_{ME} + c_{MS}. \tag{17}$$

When either of the two components is not zero, the semivariogram exhibits a discontinuity at the origin. This discontinuity at the origin is called the nugget effect. The term originated from the idea that gold nuggets are dispersed thoughout a larger body of rock but (possibly) at distances smaller than the smallest sampling distance. When a semivariogram has nugget c_0 and sill $C(0)$, the difference $C(0) - c_0$ is called

the partial sill of the semivariogram. The nugget effect is obvious in many data sets. In the absence of measurement error, the nugget effect is an indication that the sampling interval was not small enough.

2.8 Isotropy and Anisotropy

The covariance and the semivariogram are said to be anisotropic if they depend on both distance and direction. They are said to be isotropic if they depend only on the magnitude of **h**. When experimental semivariograms exhibit anisotropy, a coordinate transformation can be applied to obtain an isotropic model. (Goovaerts, 1997; Wackernagel, 1998). To determine the presence of anisotropy we need to look at directional experimental semivariograms. A semivariogram surface, which is essentially a contour plot of the directional semivariograms, visually indicates the direction of greatest spatial continuity. It is important that any pronounced anisotropy is modelled and not ignored. Anisotropy can be classified as either geometric anisotropy or zonal anisotropy.

2.8.1 Geometric Anisotropy

A semivariogram has a geometric anisotropy when it has the same sill in all directions but different ranges in at least two directions. A plot of the calculated range of the semivariogram in various directions appears ellipsoidal, and this ellipse can be transformed to a circle with radius equal to the minor axis via a rotation and subsequent dilation.

2.8.2 Zonal Anisotropy

A semivariogram exhibits zonal anisotropy when its sill values vary with direction. This type of anisotropy can be modelled as the sum of two components; an isotropic semivariogram in both coordinates and a one-dimensional semivariogram that depends only on the distance in the direction of greater variance. The coordinate

system is rotated so that the y-axis coincides with the direction of maximum continuity.

Thus a semivariogram model is completely specified by stating the direction of greatest continuity, and the anisotropy ratio (minor/major axis in the case of geometric anisotropy, and zero in the case of zonal anisotropy) and a suitable isotropic model function In the next section we will consider isotropic semivariogram models. These basic models are used to form a linear model that can be isotropic or display either type of anisotropy.

2.9 Isotropic Semivariogram Models

Only certain functions can be used as models for semivariograms and covariances. Covariances must be positive definite functions, and so semivariograms have to be conditionally negative semi-definite, that is

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j 2\gamma\left(\mathbf{u}_i - \mathbf{u}_j\right) \leq 0 \tag{18}$$

for any set of locations $\mathbf{u}_1, ..., \mathbf{u}_n$ and constants $a_1, ..., a_n$. It is common practice to fit a positive linear combination of basic models that are known to be permissible (Goovaerts, 1997). This eliminates the need to test the permissibility of a semivariogram model after it has been constructed. The following isotropic semivariogram models depend only on scalar differences between the locations, $h = |\mathbf{h}|$, not directions.

2.9.1 Nugget effect model

The nugget effect model is a semivariogram for a pure white-noise process. It is defined by

$$g(h) = \begin{cases} 0 & \text{for } h = 0 \\ c & \text{for } h > 0 \end{cases} \tag{19}$$

where $c \geq 0$. The nugget effect is used to model a discontinuity at the origin of the semivariogram and since it reaches the sill as soon as $h > 0$, it is bounded.

2.9.2 Spherical model

The spherical model has range a and sill c. It is defined by

$$g(h) = \begin{cases} c\left(\frac{3}{2}\frac{h}{a} - \frac{1}{2}\left(\frac{h}{a}\right)^3\right) & \text{for } 0 \leq h < a \\ c & \text{for } h \geq a \end{cases} \tag{20}$$

where $c_s \geq 0$. The semivariogram exhibits linear behaviour near the origin, and once the range is reached, the semivariogram is bounded and remains constant.

2.9.3 Exponential model

The exponential model reaches its sill asymptotically and has a practical range a. The model is defined by

$$g(h) = c\left(1 - \exp\left(\frac{-3h}{a}\right)\right) \quad \text{for } h \geq 0 \tag{21}$$

The exponential model is bounded and exhibits linear behaviour near the origin.

Differentiating the spherical and exponential model functions with respect to h, we find the gradient of the spherical model at the origin is

$$\begin{aligned} g'(0) &= \left(\frac{3}{2a} - \frac{3h^2}{2a^3}\right)\Big|_{h=0} \\ &= \frac{3}{2a} \end{aligned} \tag{22}$$

and the gradient of the exponential model is

$$\begin{aligned} g'(0) &= \frac{3}{a}\exp\left(\frac{-3h}{a}\right)\Big|_{h=0} \\ &= \frac{3}{a} \end{aligned} \tag{23}$$

Clearly we have $\frac{3}{2a} < \frac{3}{a}$ for all values of $a \in \mathbb{R}$, so the exponential model is steeper near the origin.

2.9.4 Gaussian model

The Gaussian model has practical range a and is bounded as it reaches the sill c asymptotically. It is defined by

$$g(h) = c\left(1 - \exp\left(-\frac{3h^2}{a^2}\right)\right) \quad \text{for } h \geq 0 \tag{24}$$

The model exhibits parabolic behaviour near the origin, and is infinitely differentiable everywhere. It is characteristic of highly regular attributes.

2.9.5 Power model

The power model is unbounded and has no sill. It is defined by

$$g(h) = ch^{\omega} \quad \text{for } h \geq 0 \tag{25}$$

where $0 < \omega < 2, \omega \in \mathbb{R}$. The power model plays an important role in the theory of turbulence and its application to meteorology.

2.10 Kriging

Kriging is a local estimation technique which provides a best linear unbiased estimator of the attribute z at location $\mathbf{u}$. This method uses the modelled spatial correlation estimated from the sample data. The estimator used in kriging is $Z^*(\mathbf{u})$ which is defined as

$$Z^*(\mathbf{u}) = m(\mathbf{u}) + \sum_{\alpha=1}^{n(\mathbf{u})} \lambda_\alpha(\mathbf{u}) \left[Z(\mathbf{u}_\alpha) - m(\mathbf{u}_\alpha)\right] \tag{26}$$

where $m(\mathbf{u})$ and $m(\mathbf{u}_\alpha)$ are the expected values of the random variables $Z(\mathbf{u})$ and $Z(\mathbf{u}_\alpha)$, and λ_α is the weight given to the sample value at location $\mathbf{u}_\alpha$. The number of data used in the estimation, as well as their weights, may change from one location to another. Generally only the $n(\mathbf{u})$ data closest to the location $\mathbf{u}$ being estimated are retained. The weights are chosen so as to minimise the error variance

$$\sigma_E^2(\mathbf{u}) = Var\left\{Z^*(\mathbf{u}) - Z(\mathbf{u})\right\} \tag{27}$$

under the unbiasedness constraint that

$$E\{Z^*(\mathbf{u}) - Z(\mathbf{u})\} = 0 \tag{28}$$

This means that kriging is a best linear unbiased estimation (BLUE) method. The kriging estimator is an exact interpolator because it honours the data values $z(\mathbf{u}_\alpha)$ at their locations. Different kriging methods are used according to the model considered for the trend $m(\mathbf{u})$.

The simulation algorithms that we outline in the next chapter use simple kriging (SK), which considers the mean $m(\mathbf{u})$ to be known and constant throughout the study region, that is

$$m(\mathbf{u}) = m \quad \text{for all } \mathbf{u} \in \mathcal{A}. \tag{29}$$

2.10.1 Simple Kriging

The simple kriging estimator is a linear combination of the n random variables $z(\mathbf{u}_\alpha)$ and the mean value m. In this case equation (26) becomes

$$Z^*_{SK}(\mathbf{u}) = \sum_{\alpha=1}^{n(\mathbf{u})} \lambda^{SK}_\alpha(\mathbf{u}) Z(\mathbf{u}_\alpha) + \left[1 - \sum_{\alpha=1}^{n(\mathbf{u})} \lambda^{SK}_\alpha(\mathbf{u})\right] m \tag{30}$$

where the n weights $\lambda^{SK}_\alpha(\mathbf{u})$ are the simple kriging weights determined to minimise the error variance, $Z^*(\mathbf{u})$ is the random variable of the estimated value and $Z(\mathbf{u}_\alpha)$ is the random variable at the sample location $\mathbf{u}_\alpha$. This minimisation results in the following set of $n(\mathbf{u})$ normal equations:

$$\sum_{\beta=1}^{n(\mathbf{u})} \lambda^{SK}_\beta(\mathbf{u}) C(\mathbf{u}_\alpha - \mathbf{u}_\beta) = C(\mathbf{u}_\alpha - \mathbf{u}) \quad \text{for } \alpha = 1, ..., n(\mathbf{u}). \tag{31}$$

The corresponding simple kriging variance is:

$$\sigma^2_{SK}(\mathbf{u}) = Var[Z^*_{SK}(\mathbf{u}) - Z(\mathbf{u})] \tag{32}$$

$$= C(0) - \sum_{\alpha=1}^{n(\mathbf{u})} \lambda^{SK}_\alpha(\mathbf{u}) C(\mathbf{u} - \mathbf{u}_\alpha) \geq 0 \tag{33}$$

Simple kriging will be applied in the sequential simulation algorithms discussed in the next chapter to obtain estimates of the first two moments of the local distributions used in the simulation.

2.11 Simulation

Simulation is often preferred to estimation because it allows the generation of maps, or realisations, that reproduce the sample variability. By generating many realisations that reproduce global statistics such as the histogram and the semivariogram the uncertainty about the spatial distribution of the attribute values can be assessed. The set of geostatistical realisations allows local uncertainty, spatial uncertainty and response uncertainty to be modelled. The models of uncertainty and subsequent risk quantification is influenced by the decisions made along the modelling process. These decisions include the choice of conceptual model, the selection of simulation algorithm and the number of realisations generated to explore the space of uncertainty, and the inference of the parameters of the random function model (Goovaerts, 2001).

There are numerous simulation algorithms used in geostatistical applications. These differ in the underlying random function model, the amount and type of information accounted for, and computational requirements. Sequential simulation is based on Monte Carlo simulation which generates realisations of random processes. For the purpose of this research, we are interested in sequential Gaussian simulation and direct sequential simulation. Both these methods involve the sequential sampling of a conditional cumulative distribution function. In sequential simulation, a random path visiting all locations once and only once is defined and each location is simulated when it is visited. With conditional simulation, the resulting realisations honour the data values at their locations.

Sequential Gaussian simulation assumes that the given random field is multivariate normal, which implies that the given data are normally distributed. Before sequential Gaussian simulation is applied, the original data usually require a transformation into normal score data to honour the normality requirement. Direct sequential simulation does not rely on the multi-Gaussian assumption, so it does not require such a transformation and the simulation is performed directly in the original data space. Variogram reproduction is ensured by Journel's result (Journel, 1994) which states that for the sequential simulation algorithm to reproduce a specific covariance model, it suffices that all conditional cumulative distribution functions used in sequential simulation have the mean and variance equal to the corresponding

simple kriging mean and simple kriging variance. We discuss the algorithm in more detail in the next chapter.

The limitations of sequential Gaussian simulation (Caers, 2000b) are that it:

- assumes a multivariate Gaussian field, which can never be fully checked in practice.
- requires a back-transformation after simulation if a normal score transform was applied.
- does not reproduce the original semivariogram model, only the normal score semivariogram model.

The limitations of direct sequential simulation are that it does not always reproduce the histogram, only the mean and variance (covariance model). A post processing algorithm may be necessary to identify the target histogram, but this may destroy the variogram reproduction (Caers, 2000b).

3 Sequential Simulation Methods and their Implementation

In this chapter we discuss the sequential simulation methods we considered, provide the relevant mathematical background and explain their implementation algorithms.

3.1 Sequential Simulation Algorithm

The simulation algorithms we use in this study all belong to a class of simulation algorithms known as sequential simulation algorithms. A conditional cumulative distribution function is modelled and sampled at each of the N nodes visited along a random path. Reproduction of the semivariogram model is ensured by making each conditional cumulative distribution function conditional on both the original n data and the values simulated at previously visited locations.

The sequential simulation process consists of the following steps:

- Define a random path through all nodes to be simulated, visiting each node once and only once.
- For each node $\mathbf{u}_i, i = 1, ..., N$:
  - Determine the parameters for the local conditional cumulative distribution function at the node such that its mean and variance equals the simple kriging mean and simple kriging variance respectively.
  - Draw a simulated value from the conditional cumulative distribution function at location $\mathbf{u}_i$.
  - Add the simulated value to the data set.
- Loop until all N nodes have been simulated.

Each of the sequential simulation algorithms we use follow these steps but they take different approaches to determining the local conditional cumulative distribution functions. The algorithms used to determine the conditional cumulative

distribution functions can be divided into two main categories - parametric and non-parametric.

3.2 Parametric Algorithms

In this section we discuss simulation algorithms for which the local conditional distribution can be written as a function of the mean and variance.

3.2.1 Sequential Gaussian Simulation

The main assumption in sequential Gaussian simulation is that the local conditional cumulative distribution function is from a standard normal distribution. If the original z-data are not standard normal, or even normal, they need to be transformed into y-values with a standard normal distribution. This can be done by associating to the percentiles of the cumulative distribution of Z the corresponding percentiles of the standard normal distribution. This is called the normal score (nscore) transformation or Gaussian anamorphosis and it preserves the rank of the sample data. The simulation is then carried out in normal score space where the random normal score deviate is calculated by

$$y = \mu_{SK} + r\sigma_{SK} \tag{34}$$

where μ_{SK} is the kriging mean, σ_{SK} is the kriging variance and r is a random number in $[0, 1]$. The resulting realisations are then back-transformed to the original variable.

For the back-transformation the program performs a linear interpolation separately within each of the middle classes. The lower tail is extrapolated towards a minimum value using a power model with a strictly positive parameter, ω that represents the power. When $\omega = 1$ the power model corresponds to a linear model. The upper tail is extrapolated by using a hyperbolic model as this allows the cumulative distribution function values to go beyond the largest threshold value z_k, and the parameter $\omega \geq 1$ controls how fast the cumulative distribution function model reaches its limiting value 1. The smaller ω is, the longer the tail of the distribution will be (Goovaerts, 1997).

3.2.2 Direct Sequential Simulation

As shown by Journel (1994), the conditional distribution $F(\mathbf{u}_i; z \mid (n+i-1))$ can be of any type and need not be the same at each location, as long as its parameters are calculated from the simple kriging mean and simple kriging variance. In this study we will use a lognormal distribution as the local conditional cumulative distribution function where the logarithmic variance $\sigma^2(\mathbf{u})$ is given by

$$\sigma^2(\mathbf{u}) = \log\left(\frac{\sigma^2_{SK}(\mathbf{u})}{(Z^*_{SK}(\mathbf{u}))^2} + 1\right) \tag{35}$$

and the logarithmic mean $\mu(\mathbf{u})$ is

$$\mu(\mathbf{u}) = \log(Z^*_{SK}(\mathbf{u})) - \frac{\sigma^2(\mathbf{u})}{2}. \tag{36}$$

The random deviate is given by

$$z = \exp(\mu(\mathbf{u}) + r\sigma^2(\mathbf{u})) \tag{37}$$

where r is a random number in $[0, 1]$.

3.3 Non-parametric Algorithms

Unlike the Gaussian approach, non-parametric algorithms do not assume any particular shape or analytical expression for the local conditional distributions (Goovaerts, 2001).

3.3.1 Direct Sequential Simulation with Histogram Identification

Caers (2000b) proposed a direct sequential simulation method that tries to overcome the shortcomings of sequential Gaussian simulation and the original direct sequential simulation by directly matching the target histogram associated with each simulation node. This target histogram is defined through a set of thresholds $\{t_k, k = 0, ..., K\}$ that discretise the range of values for the attribute, and probabilities p^g_k, where $p^g_k = \Pr\{t_{k-1} < Z(\mathbf{u}) \leq t_k\}$ denotes the global proportion for the target histogram. The following describes the principles of this method.

For each location $\mathbf{u}$ the value of the local conditional probability distribution function corresponding to a given threshold $k = 1, ..., K$, is denoted by $p_k(\mathbf{u} \mid (n))$ and defined as

$$p_k(\mathbf{u} \mid (n)) = \Pr\{t_{k-1} < Z(\mathbf{u}) \leq t_k \mid (n)\}. \tag{38}$$

The aim of the algorithm is to locally match the global target histogram as closely as possible, while at the same time, requiring that:

1. The mean of the local conditional cumulative distribution function is equal to the simple kriging mean, and so

$$z^*_{SK}(\mathbf{u}) = \sum_{k=1}^{K} \left(\frac{t_{k-1} + t_k}{2} \right) p_k(\mathbf{u} \mid (n)) \tag{39}$$

2. The variance of the local conditional cumulative distribution function is equal to the simple kriging variance, and so

$$\sigma^2_{SK} + (z^*_{SK}(\mathbf{u}))^2 = \sum_{k=1}^{K} \left(\frac{t_{k-1} + t_k}{2} \right)^2 p_k(\mathbf{u} \mid (n)). \tag{40}$$

3. The sum of the probabilities equals one, and so

$$\sum_{k=1}^{K} p_k(\mathbf{u} \mid (n)) = 1. \tag{41}$$

4. The consistency condition

$$0 \leq p_k(\mathbf{u} \mid (n)) \leq 1, \qquad k = 1, ..., K \tag{42}$$

holds for the probabilities.

There are many different ways to measure the match between the local conditional histogram and the global target histogram. We will discuss two possibilities, the first, used by Caers (2000b), is to minimise the absolute deviation between the target, that is global, and the local conditional probabilities. This can be achieved by requiring the absolute deviation

$$O_1 = \|\mathbf{p}(\mathbf{u} \mid (n)) - \mathbf{p}^g\|_1 = \sum_{k=1}^{K} |p_k(\mathbf{u} \mid (n)) - p^g_k| \tag{43}$$

to be minimised.

As a result, a nonlinear constrained minimisation problem needs to be solved at each location where the objective is to minimise.

$$O_1 = \|\mathbf{p} - \mathbf{p}^g\|_1 \tag{44}$$

where the vector $\mathbf{p} = \mathbf{p}(\mathbf{u} \mid (n))$ and the vector $\mathbf{p}^g$ is constant. Because of the nature of the objective function in (43) we call the resulting algorithm dssℓ1. This objective function is piecewise linear.

A more natural approach is investigated in this thesis. We use the least squares principle and minimise the sum of the squares of the deviations between the local and global histograms:

$$O_2 = \|\mathbf{p}(\mathbf{u} \mid (n)) - \mathbf{p}^g\|_2^2 = \sum_{k=1}^{K} (p_k(\mathbf{u} \mid (n)) - p_k^g)^2 . \tag{45}$$

The measure O_2 is a differentiable function with respect to $p_k(\mathbf{u} \mid (n))$. The objective function in (45) minimises the squared difference between the local and global probabilities and hence we call this algorithm dssℓ2. At each location we are required to solve a constrained least squares problem. We need to minimise O_2 subject to the constraints (39)-(42). Equation (45) can be rewritten as

$$O_2 = \sum_{k=1}^{K} \left((p_k(\mathbf{u} \mid (n)))^2 - 2p_k^g p_k(\mathbf{u} \mid (n)) + (p_k^g)^2 \right) \tag{46}$$

which, after dropping the constant term, $\sum_{k=1}^{K} (p_k^g)^2$ gives us the new objective function

$$O_2 = \sum_{k=1}^{K} (p_k(\mathbf{u} \mid (n)))^2 - \sum_{k=1}^{K} 2p_k^g p_k(\mathbf{u} \mid (n)) . \tag{47}$$

The resulting problem can be formulated as a quadratic programming problem where the objective is to minimise

$$O_2 = -2(\mathbf{p}^g)^T \mathbf{p} + \frac{1}{2}\mathbf{p}^T \mathbf{Q}\mathbf{p} \tag{48}$$

where the vector $\mathbf{p} = \mathbf{p}(\mathbf{u} \mid (n))$, the vector $\mathbf{p}^g$ is constant, and the $(K \times K)$ dimensional matrix $\mathbf{Q} = 2\mathbf{I}$.

For both problems the optimal solution, where it exists, results in local probability density function values

$$p^*(\mathbf{u}) = \{p_k^*(\mathbf{u} \mid (n))\} \qquad , k = 1, ..., K. \tag{49}$$

These values are regarded as the frequencies of a histogram that has the same threshold classes as the target histogram, and this new histogram will be used to draw random deviates. Given a random number $r \in [0, 1]$ and a threshold class $(z_i, z_{i+1}]$ with cumulative distribution function values $F(z_i)$ and $F(z_{i+1})$, the random deviate x is linearly interpolated using the definition

$$x = \frac{(r - F(z_i))(z_{i+1} - z_i)}{(F(z_{i+1}) - F(z_i))} + z_i. \tag{50}$$

These two simulation algorithms both encounter convergence problems when the simple kriging mean is less than the midpoint of the first threshold. When this occurs an optimal solution does not exist. A random deviate needs to be found in an alternative way. This could be done by using a different local conditional cumulative distribution function, for example a normal distribution, or as in the case of this study, by setting the random deviate equal to the simple kriging mean.

3.4 Constrained Optimisation

In the previous section we have identified two constrained optimisation problems that need to be solved in order to determine the local conditional cumulative distribution function. The constraints are linear equations and/or inequalities and the objective function is either piecewise linear or quadratic. The two problems are called an ℓ^1 approximation problem and a quadratic programming problem respectively. They can both be rewritten as linear programming problems, that is, as problems with a linear objective function and linear constraints.

3.5 Linear Programming Problems

A linear programming problem is characterised by a linear objective function and linear constraints. The standard form of a linear program is
Minimise

$$f(\mathbf{x}) = \mathbf{c}^T\mathbf{x} \tag{51}$$

subject to

$$\mathbf{Ax} \leq \mathbf{b} \text{ and } \mathbf{x} \geq \mathbf{0} \tag{52}$$

where $\mathbf{x}$ is an n-dimensional column vector, $\mathbf{c}^T$ is an n-dimensional row vector, $\mathbf{A}$ is an $m \times n$ matrix, and $\mathbf{b}$ is an m-dimensional column vector. Inequalities are converted to equality equations by introducing new positive variables y_i known as slack variables. This allows us to rewrite the inequalities in (52) as a system of m linear equations in $n+m$ unknowns

$$\mathbf{A}'\mathbf{x}' = \mathbf{b} \tag{53}$$

where $\mathbf{A}' = [\mathbf{A}, \mathbf{I}_m]$ and $(\mathbf{x}')^T = [\mathbf{x}, \mathbf{y}]$. If $\mathbf{B}$ is a nonsingular $m \times m$ submatrix of $\mathbf{A}'$, then the solution to $\mathbf{Bx}_B = \mathbf{b}$ is called a basic solution of Equation (53). The basic variables are the components of $\mathbf{x}$ associated with columns of $\mathbf{B}$.

A feasible solution of the linear programming problem is a solution for which all the constraints are satisfied. A vector that satisfies Equation (52) is said to be feasible for these constraints. A feasible constraint that is also basic is known as a basic feasible solution. An infeasible solution is a solution for which at least one constraint is violated. We call the collection of all feasible solutions the feasible region. If the feasible region is bounded, the optimisation problem is bounded, otherwise it is said to be unbounded. The optimal solution is a feasible solution that results in the objective function having the smallest value when minimising. When the constraints are inconsistent, an optimisation problem has no solution, and the problem is said to be infeasible.

Feasible regions that are defined by linear constraints are convex. In general, a set $S \in \mathbb{R}^n$ is convex if, given any two points in the set, every point on the line segment joining these two points is also a member of the set. A hyperplane in $\mathbb{R}^n$ is the set of points $H = \left\{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T\mathbf{x} = c\right\}$, where $\mathbf{a} \neq \mathbf{0}$ is an n-dimensional column vector in $\mathbb{R}^n$ and c is a real number. A hyperplane is a set of solutions to a single linear equation. The closed half spaces are defined by $H = \left\{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T\mathbf{x} \geq c\right\}$ and $H = \left\{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T\mathbf{x} \leq c\right\}$. The open half spaces are defined by $H = \left\{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T\mathbf{x} > c\right\}$ and $H = \left\{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T\mathbf{x} < c\right\}$. A convex polytope is a set which can be expressed as the intersection of a finite number of closed half spaces. Convex polytopes are the sets of solutions obtained from a system of linear inequalities. Each inequality

defines a half space and the solution is the intersection of these half spaces. A polytope may be empty, bounded or unbounded. A nonempty polytope is called a polyhedron.

An extreme point of a convex set is a point $\mathbf{x}$ in the convex set that does not lie strictly within the line segment connecting two other points of the set. Adjacent extreme points are points that lie on a common edge. Any polytope has at most a finite number of extreme points (Luenberger, 1984; Wismer & Chattergy, 1978).

A function $f(x)$ is called convex in $\mathbb{R}^n$ if

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) \tag{54}$$

for all $x, y \in \mathbb{R}^n$ and $\lambda \in [0,1]$. A function is strictly convex if this definition holds with strict inequality when $0 < \lambda < 1$ and $x \neq y$. A convex function is defined only over the domain of a convex set. The definition does not require that f be either continuous or differentiable.

A vector $\mathbf{x}$ is an extreme point of a polytope K if and only if $\mathbf{x}$ is a basic feasible solution to Equation (52).

Denote by K the polytope of all (feasible) solutions of (52). The relationship between extreme points and basic feasible solutions is as follows:

1. If the convex set K corresponding to Equation (52) is nonempty, it has at least one extreme point.

2. If there is a finite optimal solution to a linear programming problem, there is a finite optimal solution which is an extreme point of the constraint set.

3. The constraint set K corresponding to Equation (52) possesses at most a finite number of extreme points.

4. If the convex polytope K corresponding to Equation (52) is bounded, then K is a convex polyhedron and K consists of points that are convex combinations of a finite number of points.

The optimal solution for a linear programming problem must lie on the boundary of the feasible region. Any point on the boundary of the feasible region lies on one

or more of the hyperplanes defined by the respective constraint boundary equations. The hyperplanes define a polytope with vertices at which at least n of these planes meet. At least one member of the optimal set is at a vertex, and in general the number of vertices can be prohibitively large, even for small problems.

The simplex method, originally formulated by Dantzig in 1947 (Gill et al., 1984), is an algebraic procedure for determining the optimal solution of a linear programming problem that has underlying geometric concepts. The set of all feasible solutions to Equations (51)-(52) is defined by the set $K = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ and the linear programming problem consists of finding an extremum of $f(\mathbf{x})$ on K. When the objective is linear, and when an optimal solution exists then there is at least one vertex of K at which this optimum is attained.

There are two phases to the Simplex Method. Phase I is the process of locating a vertex of the polytope. Extra slack variables and constant offsets are added to all of the inequalities to help find a feasible vertex. Phase I concludes when a basic feasible solution is obtained for the artificial vectors, and this solution is used as the initial basic feasible solution for applying the simplex method to the objective function in Phase II. Once we reach a vertex for which the slack variable is zero, we have found a vertex of the original polytope and we then continue with Phase II on that polytope. We then move from one vertex to an adjacent one, checking the objective function after each move to determine if further improvement is possible. The algorithm proceeds to move on the surface defined by the working set of constraints to an improved point until the optimal vertex is reached. The vector to enter the basis is chosen as that with the greatest nonnegative marginal cost. The vector leaving the basis is chosen from among all basic vectors by selecting that which causes the maximum reduction in the objective function, allowing many intermediate simplex vertices to be bypassed.

The objective functions we will be concerned with are piecewise linear and quadratic respectively, and both problems can be rewritten in such a way that the problem becomes a linear programming problem, which can be solved using a two phase simplex algorithm.

3.6 One-norm Approximation with Linear Constraints

The constrained one-norm linear approximation problem is to

Minimise

$$\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_1 = \sum_{i=1}^{m} |b_i - A_i\mathbf{x}| \tag{55}$$

subject to the linear constraints

$$\mathbf{C}\mathbf{x} = \mathbf{d} \tag{56}$$

$$\mathbf{E}\mathbf{x} \leq \mathbf{f} \tag{57}$$

where the vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$ and we are given the vector $\mathbf{b} = [b_1, b_2, \ldots, b_m]^T$ and the $m \times n$ matrix $\mathbf{A}$, the $k \times n$ matrix $\mathbf{C}$ and the $l \times n$ matrix $\mathbf{E}$.

The problem (55)-(57) can be formulated as the linear programming problem (Barrodale and Roberts, 1978):

Minimise

$$\mathbf{e}(\mathbf{u} + \mathbf{v}) \tag{58}$$

subject to

$$\begin{aligned} \mathbf{A}(\mathbf{x}' - \mathbf{x}'') + \mathbf{u} - \mathbf{v} &= \mathbf{b} \\ \mathbf{C}(\mathbf{x}' - \mathbf{x}'') &= \mathbf{d} \\ \mathbf{E}(\mathbf{x}' - \mathbf{x}'') + \mathbf{u}'' &= \mathbf{f} \\ \mathbf{x}',\ \mathbf{x}'',\ \mathbf{u},\ \mathbf{u}'',\ \mathbf{v} &\geq \mathbf{0} \end{aligned} \tag{59}$$

where $\mathbf{e} = [1, 1, \ldots, 1] \in \mathbb{R}^m$, $\mathbf{u} = [u_1, u_2, \ldots, u_m]^T$, $\mathbf{v} = [v_1, v_2, \ldots, v_m]^T$ and $\mathbf{u}'' = [u_1'', u_2'', \ldots, u_l'']^T$. The vector $\mathbf{v}$ is introduced as a slack variable to convert the inequality constraint $\mathbf{E}\mathbf{x} \leq \mathbf{f}$ to an equivalent equality constraint. This augmented form is needed in order to apply the simplex method.

To start the simplex iterations, artificial variables need to be introduced for the purpose of being the initial basic variable for their respective equation. These variables have the usual nonnegativity constraints placed on them, and the objective function is modified so an exorbitant penalty is imposed if their values are larger than zero.

After introducing the artificial vectors $\mathbf{u}'$,$\mathbf{v}'$, and $\mathbf{v}''$, we can restate problem (58)-(59) in the form:
Minimise

$$\mathbf{e}(\mathbf{u}+\mathbf{v})+M\mathbf{e}'(\mathbf{u}'+\mathbf{v}')+M\mathbf{e}''\mathbf{v}'' \tag{60}$$

subject to

$$\begin{aligned} \mathbf{A}(\mathbf{x}'-\mathbf{x}'')+\mathbf{u}-\mathbf{v} &= \mathbf{b} \\ \mathbf{C}(\mathbf{x}'-\mathbf{x}'')+\mathbf{u}'-\mathbf{v}' &= \mathbf{d} \\ \mathbf{E}(\mathbf{x}'-\mathbf{x}'')+\mathbf{u}''-\mathbf{v}'' &= \mathbf{f} \\ \mathbf{x}',\ \mathbf{x}'',\ \mathbf{u},\ \mathbf{u}',\ \mathbf{u}'',\ \mathbf{v},\ \mathbf{v}',\ \mathbf{v}'' &\geq \mathbf{0} \end{aligned} \tag{61}$$

where $\mathbf{u}=[u_1',u_2',\ldots,u_k']^T$, $\mathbf{v}'=[v_1',v_2',\ldots,v_k]^T$, $\mathbf{v}''=[v_1'',v_2'',\ldots,v_l'']^T$ and $\mathbf{e}'$ and $\mathbf{e}''$ are row vectors of 1's of dimensions k and l respectively. The quantity M in the objective function is a large positive number which represents the cost of each artificial vector.

The iterations of the simplex method automatically force the artificial variables to become zero, one at a time. When all the artificial variables are zero, the real problem is solved (Hillier and Lieberman, 1995). The initial basis normally includes some of the artificial vectors so the algorithm is implemented using the two-phase simplex method. The objective function $\mathbf{e}'(\mathbf{u}'+\mathbf{v}')+\mathbf{e}''\mathbf{v}''$ is used in phase I, and if the optimal solution to this problem is positive, then no feasible solution to the constraints (56) and (57) exists, and the algorithm terminates. If the optimal solution is zero, the algorithm proceeds with Phase II using the objective function $\mathbf{e}(\mathbf{u}+\mathbf{v})$.

3.7 Quadratic Programming

A linearly constrained optimisation problem with a quadratic objective function is called a quadratic program. The general quadratic program can be written as
Minimise

$$f(\mathbf{x})=\mathbf{c}^T\mathbf{x}+\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} \tag{62}$$

subject to

$$\mathbf{A}\mathbf{x}\leq\mathbf{b} \qquad \text{and} \qquad \mathbf{x}\geq\mathbf{0},$$

where $\mathbf{c}^T$ is an n-dimensional row vector containing the coefficients of the linear terms in the objective function, and $\mathbf{Q}$ is a $(n \times n)$ symmetric matrix containing the coefficients of the quadratic terms. The decision variables are denoted by the n-dimensional column vector $\mathbf{x}$, and the constraints are defined by an $(m \times n)$ matrix $\mathbf{A}$ and an m-dimensional column vector $\mathbf{b}$ of right-hand-side coefficients.

The Lagrangian function for the quadratic program is

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \boldsymbol{\lambda}(\mathbf{A}\mathbf{x} - \mathbf{b}) \tag{63}$$

where the vector $\boldsymbol{\lambda}$ is called the vector of Lagrange multipliers. The Karush-Kuhn-Tucker conditions (Wismer & Chattergy, 1978) for the quadratic program are first-order necessary conditions for optimality that are sufficient for a global minimum when $\mathbf{Q}$ is positive definite. The Karush-Kuhn-Tucker conditions for a local minimum are:

$$\mathbf{c}^T + \mathbf{x}^T\mathbf{Q} + \boldsymbol{\lambda}\mathbf{A} \geq \mathbf{0} \tag{64}$$

$$\mathbf{A}\mathbf{x} - \mathbf{b} \leq \mathbf{0} \tag{65}$$

$$\mathbf{x}^T(\mathbf{c} + \mathbf{Q}\mathbf{x} + \mathbf{A}^T\boldsymbol{\lambda}) = \mathbf{0} \tag{66}$$

$$\boldsymbol{\lambda}(\mathbf{A}\mathbf{x} - \mathbf{b}) = \mathbf{0} \tag{67}$$

$$\mathbf{x}, \boldsymbol{\lambda} \geq \mathbf{0}. \tag{68}$$

We then introduce surplus variables $\mathbf{y} \in \mathbb{R}^n$ to the inequalities in Equation(64) and nonnegative surplus variables $\mathbf{v} \in \mathbb{R}^n$ to the inequalities in (65). The Karush-Kuhn-Tucker conditions (64)-(68) can now be expressed in a form that closely resembles linear programming:

$$\mathbf{Q}\mathbf{x} + \mathbf{A}^T\boldsymbol{\lambda} - \mathbf{y} = -\mathbf{c} \tag{69}$$

$$\mathbf{A}\mathbf{x} + \mathbf{v} = \mathbf{b} \tag{70}$$

$$\mathbf{x} \geq \mathbf{0}, \boldsymbol{\lambda} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}, \mathbf{v} \geq \mathbf{0} \tag{71}$$

$$\mathbf{y}^T\mathbf{x} = 0, \boldsymbol{\lambda}\mathbf{v} = 0 \tag{72}$$

where equations (69)-(70) are linear equalities, condition(71) restricts all the variables to be nonnegative, and condition(72) is called the complementary slackness condition and it ensures that all λs are zero for inactive constraints and positive for active constraints (Wismer & Chattergy,1978; Gill et al, 1984).

Introducing n slack variables $\mathbf{z} \geq \mathbf{0}$ we can rewrite the quadratic problem as the linear programming problem. This problem is given as

Minimise

$$Z = \sum_{j=1}^{n} z_j \tag{73}$$

subject to

$$\mathbf{Qx} + \mathbf{A}^T\boldsymbol{\lambda} - \mathbf{y} + \mathbf{z} = -\mathbf{c} \tag{74}$$

$$\mathbf{Ax} + \mathbf{v} = \mathbf{b} \tag{75}$$

$$\mathbf{x},\ \mathbf{v},\ \mathbf{y},\ \boldsymbol{\lambda} \geq \mathbf{0} \tag{76}$$

The goal is to find the solution that minimises Equation (73) whilst ensuring that the complementary slackness conditions are also satisfied at each iteration. The rule for selecting the entering variable is modified to accommodate this condition. If the sum is zero, the solution will satisfy (69) to (72). The entering variable will be the one whose coefficient is most negative provided that its complementary variable is not in the basis or would leave the basis on the same iteration. At the conclusion of the algorithm, the vector $\mathbf{x}$ defines the optimal solution.

This algorithm works well when the objective function is positive definite, and the computational effort required is comparative to the linear programming problem with $m + n$ constraints, where m is the number of constraints and n is the number of variables in the quadratic program.

3.8 Implementation and Specifications

In this section we describe in detail the specific algorithms used to implement the sequential simulation methods discussed in Sections 3.2 and 3.3. In order to implement these algorithms we first had to create a simulation environment comprising a set of Fortran programs. The flowchart shown in Figure 3.1 outlines the steps involved in the sequential simulation algorithm discussed in Section 3.1. The main difference between the simulation algorithms is associated with the subroutine in which a random deviate is drawn from a local conditional distribution.

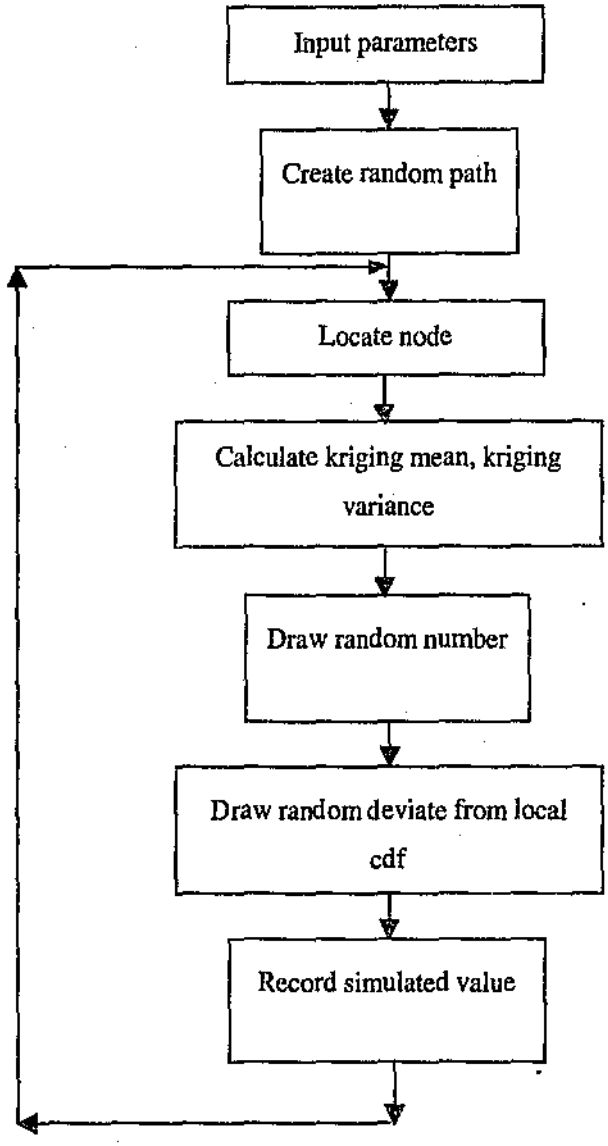

Figure 3.1. Flow chart for the sequential simulation algorithm.

3.8.1 Sequential Gaussian simulation-SGSIM

The SGSIM algorithm used the program *sgsim.exe* from the Geostatistical Software Library (GSLIB). The only modification that was required to run this program was an adjustment to the dimension of some matrices, since the *Permeability* data set was larger than the preset default values. The parameter files for the *Permeability* and *Potassium* data sets are included in Appendices A1 and B1 respectively. The program requires a semivariogram model for the normal scores, and the kriging variance is directly interpreted as the variance of the conditional distribution, so the nugget constant and the sill parameters must add to 1.0. (Deutsch & Journel, 1998).

3.8.2 Direct sequential simulation

In the case of the three direct sequential simulation algorithms we need to input a non-standardised semivariogram model derived from the original sample data. The programs do not require that the data be transformed to normal scores. Our implementation is based on a modification of SGSIM. As a first step the subroutine *krige* was changed to incorporate the sample mean into the formula for the kriging mean, as SGSIM calculates the simple kriging mean using a global mean of zero. This program will be referred to as *dssim.exe*. Additional requirements for the particular direct sequential simulation algorithms were then added where necessary.

3.8.3 DSSIM - Original direct sequential simulation

This program is a modification of *dssim.exe* so that the local conditional probability distribution is no longer assumed to be normally distributed. The code was amended to allow the random deviate associated with a location to be drawn from a lognormal distribution. The parameters for this distribution are calculated from the simple kriging mean and simple kriging variance using Equations (35)-(36). The parameter files are for the *Permeability* and *Potassium* data sets are seen in Appendices A2 and B2 respectively.

3.8.4 Direct sequential simulation with histogram reproduction

The two algorithms, DSSL1 and DSSL2 require a global target histogram at the start of the programs. These histograms consist of 40 equiprobable classes with a maximum and minimum value determined by the parameters zmin and zmax. The thresholds and global probabilities for the *Permeability* and *Potassium* data sets are given in Appendices A5 and B5 respectively. These programs differ from SGSIM and DSSIM in that they make use of special subroutines which return a local probability distribution from which a random deviate is drawn, as shown in Figure 3.2.

3.8.5 DSSL1 - Direct sequential simulation with histogram reproduction using the one-norm

For this algorithm we included Algorithm 552 from the Association for Computing Machinery (ACM) Transactions of Mathematical Software (Barrodale & Roberts, 1980) as a subroutine in the modified *dssim.exe* program. The parameter file for *dssℓ1.exe* for the *Permeability* and *Potassium* data sets are given in Appendices A3 and B3 respectively. From Equations (55)-(57) the parameters listed in Table 3.1 must be passed to the Algorithm 552 subroutine at execution time.

The only parameters that are continuously updated from location to location are **C** and **d**. The subroutine returns the solution vector through an array and a logical flag which indicates if an optimal solution was found.
The program also requires values for the following three parameters:

- Iter - an upper bound on the maximum number of iterations allowed. It is set to the suggested value of $10\,(k+l+m)$. This parameter is actually calculated in the program after k is input.

- Kode - a parameter that on exit informs the main program if an optimal solution has been found. On entry though, if set equal to one, the nonnegativity constraints on the probabilities are included implicitly in the constraints. This has been coded into the program and no further input is required. If the flag returned with the solution vector informs the main program that a solution

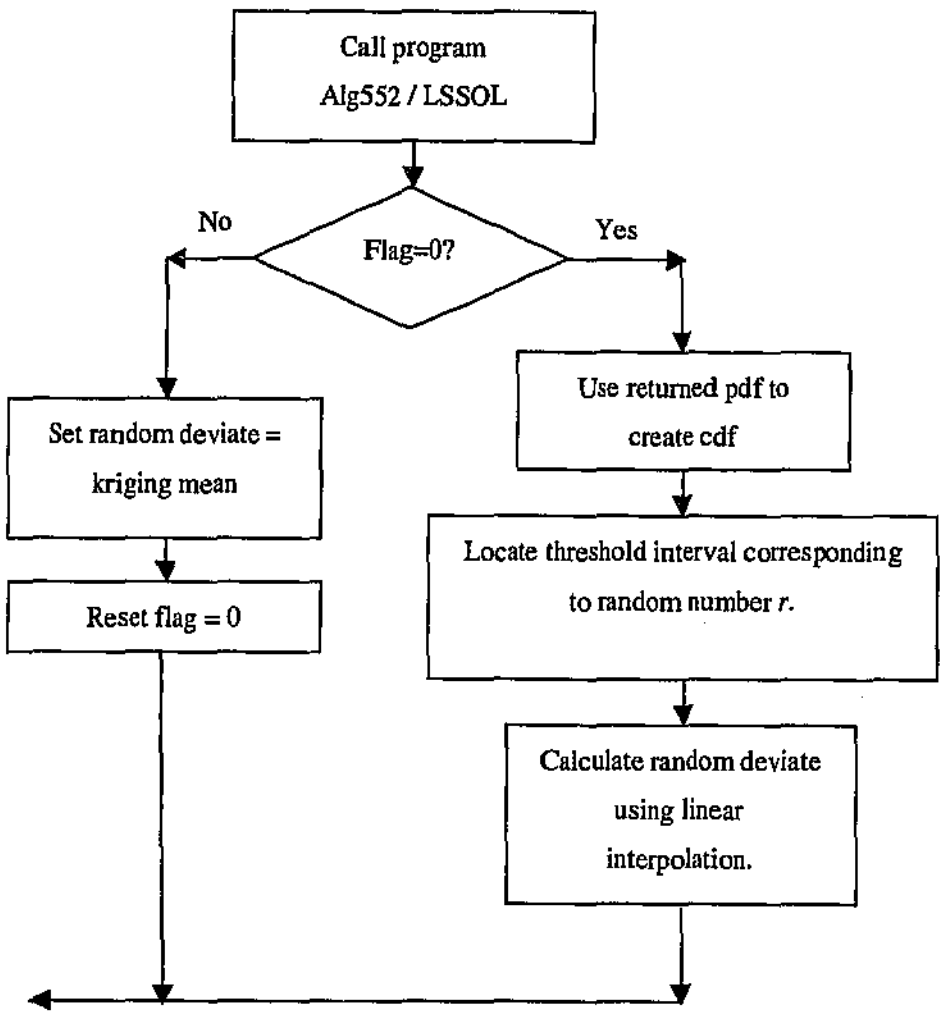

Figure 3.2. Flowchart for calculating a random deviate with DSSL1/DSSL2 .

was not found, the estimate is set equal to the simple kriging mean, and the program continues.

- Toler - a small positive tolerance for which empirical evidence suggests be set as $toler = 10^{\left(\frac{-2d}{3}\right)}$ where d represents the number of decimal digits of accuracy available. The subroutine cannot distinguish between zero and any quantity whose magnitude does not exceed *toler*. It will not pivot on any number whose magnitude does not exceed *toler*. The tolerance is preset at a value of 10^{-5} in the program code.

Table 3.1. Parameters for Algorithm 552.

Parameter	Description and input
K	Number of rows of matrix A = 40
L	Number of rows of matrix C = 3
M	Number of rows of matrix E = 0
N	Number of columns of the matrices A,C,E = 40
A	$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$
C	$\begin{bmatrix} \left(\frac{t_0+t_1}{2}\right) & \left(\frac{t_1+t_2}{2}\right) & \cdots & \left(\frac{t_{k-1}+t_k}{2}\right) \\ \left(\frac{t_0+t_1}{2}\right)^2 & \left(\frac{t_1+t_2}{2}\right)^2 & \cdots & \left(\frac{t_{k-1}+t_k}{2}\right)^2 \\ 1 & 1 & \cdots & 1 \end{bmatrix}$
E	**0**
b	$(p_1^g, p_2^g, \cdots, p_k^g)^T$
d	$\left(z_{SK}^*, \sigma_{SK}^2 + (z_{SK}^*)^2, 1\right)^T$
f	**0**

3.8.6 DSSL2 - Direct sequential simulation with histogram reproduction using the two-norm

This algorithm solves the quadratic programming problem in Equation (48). In order to accomplish this we use the software package LSSOL, version 1.0. This is a set of Fortran subroutines for linearly constrained linear least-squares and convex quadratic programming. LSSOL uses the two-phase, active-set type method. (Gill et al, 1986). The reader is referred to the user's guide for an in-depth discussion of the program and parameters.

The LSSOL subroutine is included in a modified *dssim.exe* program. The parameter files for *dssℓ2.exe* for the *Permeability* and *Potassium* data sets are given in Appendices A4 and B4 respectively. The LSSOL program states the quadratic programming in the general form
Minimise

$$F(\mathbf{x}) = \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} \tag{77}$$

subject to

$$\mathbf{l} \leq \left\{ \begin{array}{c} \mathbf{x} \\ \mathbf{Cx} \end{array} \right\} \leq \mathbf{u} \tag{78}$$

where l and u are the lower (BL) and upper (BU) bounds respectively.

The program requires an initial estimate (X) of the solution be entered. The LSSOL subroutine requires the following parameters to be input at execution time. The only parameters that are continuously updated from location to location are C, BL and BU. The subroutine returns the solution vector through an array and a logical flag which indicates if an optimal solution was found. Before calling the *lsmain* subroutine with the required inputs, we call the subroutine *lsoptn* to select a programming problem of type QP2.

Table 3.2. Parameters for LSSOL.

Parameter	Description and input
M	Number of rows of matrix A = 40
N	Number of variables = 40
NCLIN	Number of general linear constraints = 3
NROWC	Row dimension of C = 3
NROWA	Row dimension of A = 40
C	$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \left(\frac{t_0+t_1}{2}\right) & \left(\frac{t_1+t_2}{2}\right) & \cdots & \left(\frac{t_{k-1}+t_k}{2}\right) \\ \left(\frac{t_0+t_1}{2}\right)^2 & \left(\frac{t_1+t_2}{2}\right)^2 & \cdots & \left(\frac{t_{k-1}+t_k}{2}\right)^2 \end{bmatrix}$
BL	$\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 1 & z^*_{SK} & \sigma^2_{SK}+(z^*_{SK})^2 \end{bmatrix}$
BU	$\begin{bmatrix} 1 & 1 & 1 & \cdots & 0 & 1 & z^*_{SK} & \sigma^2_{SK}+(z^*_{SK})^2 \end{bmatrix}$
X	Initial estimate $= (p^g_1, p^g_2, \cdots, p^g_k)^T$
A	$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$
c	$(p^g_1, p^g_2, \cdots, p^g_k)^T$

4 Performance Assessment

Multiple realisations generated by simulation algorithms provide a measure of the uncertainty about the spatial distribution of attribute estimates. This uncertainty arises from our imperfect knowledge of the phenomenon under study. It is dependent on both the data and the model specifying our prior decisions about the phenomenon (Goovaerts, 1997). There are several ways in which the spatial uncertainty can be assessed. Qualitative assessment includes visualisation of realisations and various types of displays. Quantitative assessment focusses on the reproduction of key statistics such as the target histogram and semivariogram.

4.1 Qualitative Assessment

For each simulation algorithm, we generate a set of L realisations. These sets can be post-processed and the spatial uncertainty can be visualised through different displays, including probability maps, quantile maps and conditional variance.

4.1.1 Probability Maps

At each simulated grid node $\mathbf{u}'_j$, the probability of exceeding a given threshold z_k is evaluated as the proportion of the L simulated values that exceed that threshold. The map of such probabilities is referred to as a probability map.

4.1.2 Quantile Maps

The p-quantile of the distribution $F(x)$ is the value x_p such that

$$F(x_p) = \Pr(X \leq x_p) = p. \tag{79}$$

Quantile maps display the p-quantile values corresponding to any given probability p. In this study we will be comparing $x_{0.1}$,the median $x_{0.5}$ and $x_{0.9}$. Local differences between realisations can be depicted through the changes in the quantile maps.

4.1.3 Conditional Variance

The conditional variance $\sigma^2(\mathbf{u})$ measures the spread of the conditional probability distribution around its mean $z_E^*(\mathbf{u})$:

$$\sigma^2(\mathbf{u}) = \int_{-\infty}^{\infty} [z - z_E^*(\mathbf{u})]^2 f(\mathbf{u}|(n))\, dz. \tag{80}$$

In practice, this is approximated by the discrete sum

$$\sigma^2(\mathbf{u}) \simeq \sum_{k=1}^{K+1} [\bar{z} - z_E^*(\mathbf{u})]^2 [F(\mathbf{u}; z_k(n)) - F(\mathbf{u}; z_{k-1}(n))] \tag{81}$$

where z_k, $k = 1, ..., K$, are the threshold values discretising the range of variation of z-values, $\bar{z}_k$ is the mean of the class $(z_{k-1}, z_k]$, which depends on the within-class interpolation model, and $z_E^*(\mathbf{u})$ is the expected value of the conditional cumulative distribution function approximated by the discrete sum

$$z_E^*(\mathbf{u}) \simeq \sum_{k=1}^{K+1} \bar{z} [F(\mathbf{u}; z_k(n)) - F(\mathbf{u}; z_{k-1}(n))]. \tag{82}$$

The variance estimate in (81) depends on the K within-class means $\bar{z}_k$. The conditional variance $\sigma^2(\mathbf{u})$, conditional cumulative distribution function mean and upper tail mean can be very sensitive to the choice of extrapolation model. Local differences between realisations can be depicted through mapping a measure of the spread of the distribution of L simulated values at each simulated grid node.

4.2 Quantitative Assessment

The quality of a realisation could be defined as its ability to match *a priori* knowledge about the spatial distribution of the attributes values (Goovaerts, 2001). The structural characteristics of a simulation, which include the histogram and the semivariogram, are evaluated from its values at the nodes of the discretisation grid, which differ from the characteristics of the theoretical model or the sample data. (Chilès & Delfiner, 1999).

The performance of the simulations can be checked both visually and quantitatively by comparing the histogram and the semivariogram reproduction for each

realisation. These allow us to calculate the fluctuation variance of the spatial mean of the simulation and, in the Gaussian case, the spatial variance of the simulation and of its variogram. This enables us to see if the deviations of these characteristics from their theoretical values are acceptable.

4.2.1 Histogram Reproduction

Two quantitative measures used to compare the target histogram and the histogram of the realisations is given by the mean absolute deviation (MAD) and the mean squared deviation (MSD) between the target quantiles and the quantiles of the realisation. If there are K classes the mean absolute deviation is given by

$$H_{MAD} = \frac{\sum_{i=1}^{K} \left| z_i - \hat{z}_i^{(\ell)} \right|}{K} \tag{83}$$

and the mean squared deviation as

$$H_{MSD} = \frac{\sum_{i=1}^{K} \left(z_i - \hat{z}_i^{(\ell)} \right)^2}{K} \tag{84}$$

where z_i and $\hat{z}_i$ denotes the ith quantile of the target distribution and the ℓ^{th} realisation respectively. When the mean squared deviation is very large, the natural way to reduce the magnitude of the measure is to take the square root. The magnitude of the discrepancies between realisation and model statistics are referred to as ergodic fluctuations and they depend on several factors, including the density of the conditioning data, the semivariogram parameters and the size of the simulation grid.

Both measures of accuracy are used in comparing the histogram reproduction for the simulation algorithms as the mean average deviation is comparative to DSSL1 and the mean squared difference is comparative to DSSL2. By considering both measure we can eliminate any bias towards a particular algorithm.

4.2.2 Semivariogram Reproduction

To compare the semivariogram of the realisations with the theoretical model we calculate the mean square deviation between the theoretical model and the experimental semivariogram values. For an isotropic data set, this is done by using an omnidirectional model and omnidirectional experimental semivariogram for each realisation and the mean squared deviation is given by

$$MSD_I = \frac{\sum_{i=1}^{L} (\gamma(\mathbf{h}_i) - \hat{\gamma}(\mathbf{h}_i))^2}{L} \tag{85}$$

where $\mathbf{h}_i$ denotes the ith lag vector, L denotes the number of lags and γ and $\hat{\gamma}$ denote the theoretical semivariogram and the experimental semivariogram of the realisation respectively.

When the data set is anisotropic we sum the mean squared deviation in the directions of maximum and minimum continuity. This mean squared deviation is given by

$$MSD_A = \frac{\sum_{i=1}^{L} (\gamma(\mathbf{h}_i)_{\max} - \hat{\gamma}(\mathbf{h}_i)_{\max})^2 + \sum_{i=1}^{L} (\gamma(\mathbf{h}_i)\min - \hat{\gamma}(\mathbf{h}_i)_{\min})^2}{L} \tag{86}$$

where $\hat{\gamma}_{\max}$ and $\hat{\gamma}_{\min}$ are the experimental semivariograms in the directions of maximum and minimum continuity respectively. When the mean squared deviation is very large, the natural way to reduce the magnitude of the measure is to take the square root.

5 Application to the Isotropic Case

The *Permeability* data set comes from Goovaerts (2001) and consists of of permeability values in a 2-D section of a reservoir. The 10404 data measurements are located on a 102 x 102 regular grid which has a grid spacing of 1.0 unit. The sample set *Perm50* also comes from Goovaerts (2001) and consists of fifty permeability values randomly drawn from *Permeability*.

5.1 Exploratory Data Analysis

Descriptive statistics for the permeability variable from *Permeability* and *Perm50* are listed in Table 5.1.

Table 5.1. Descriptive statistics for the *Permeability* and *Perm50* data sets.

	Permeability	*Perm50*
n	10404	50
Mean	582.95	592.53
Std Deviation	502.5	481.653
Variance	252502	231989
Skewness	1.265	1.184
Kurtosis	1.370	1.417
Minimum	1.03	4.47
1st quartile	194.21	194.28
Median	480.69	576.30
3rd quartile	869.17	853.42
Maximum	2498.87	2081.85
IQR	674.96	659.14
Range	2497.84	2077.38

The data sets have similar means and standard deviations, but there is a noticeable difference in minimum, median and maximum values. The sample data have a higher minimum and median and a lower maximum than the exhaustive data. The

positive skewness evident in the histograms, and seen in Figure 5.1, is confirmed by the summary statistics, but overall the sample data appear to reflect the summary statistics of the exhaustive data. Neither set is normally distributed.

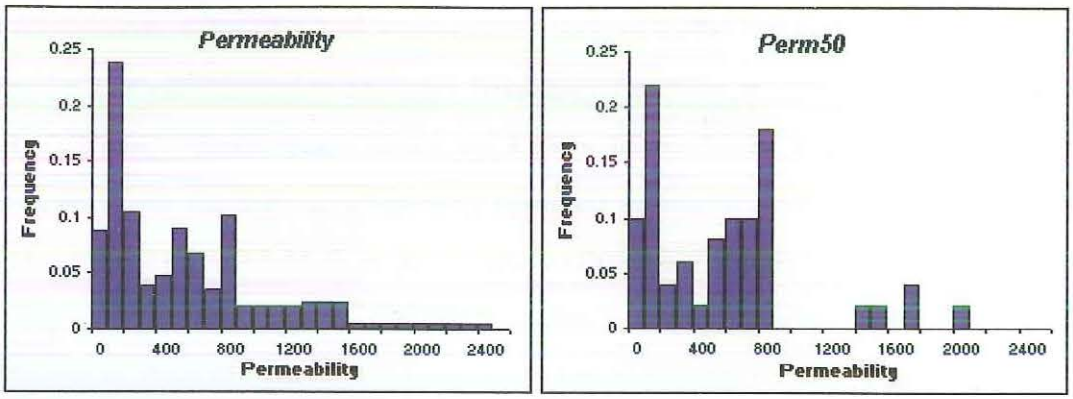

Figure 5.1. Histograms for *Permeability* (left) and *Perm50* (right).

There are several extreme values in the data sets. The maximum *Permeability* and *Perm50* values are approximately equal to four times the mean. These extreme values will influence the simple kriging means. These values are sometimes deleted from a sample data set or adjusted to reduce their influence, but since they originated from an exhaustive data set, they must be incorporated into the analysis in this case.

The cumulative frequency distributions and the Q-Q plot comparing the exhaustive and sample permeability data sets are shown in Figure 5.2.

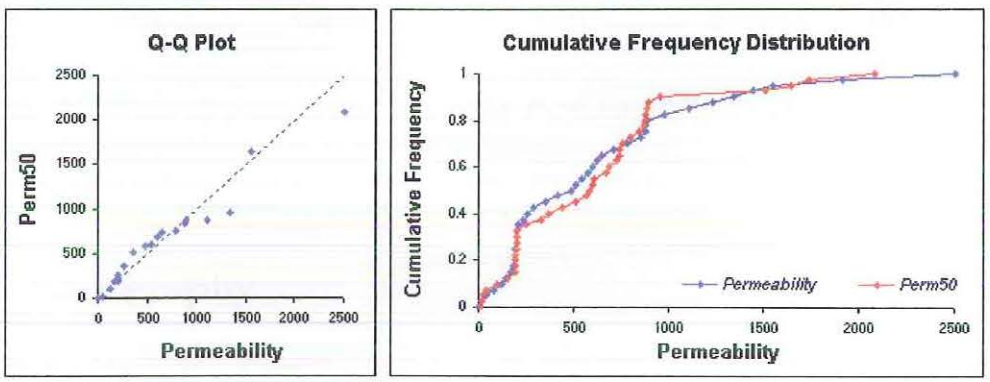

Figure 5.2 Q-Q plot (left) and cumulative frequency distribution (right).

We notice the cumulative frequency curves are almost identical to begin with and then there are some deviations in their shapes. The sample does not contain

the extreme high values present in the exhaustive data set, but the fit is reasonably good considering the sample size is relatively small.

The forty permeability thresholds are used to define the global probability distribution in DSSL1 and DSSL2 are given in Appendix A5. These values are bounded by a minimum value of 1 and a maximum value of 2500. For each simulation we will consider the conditional cumulative frequency function at two locations $\mathbf{u}_1\,(46, 94)$ and $\mathbf{u}_2(88, 52)$ whose exact values are known to be 421.02 and 26.11 respectively. Location $\mathbf{u}_1$ is situated in a sparsely sampled region of above average values and location $\mathbf{u}_2$ is situated in close proximity to three low values.

Figure 5.3 shows plots of the permeability values for *Permeability* and *Perm50*. The sample data are randomly located within the study region. The *Perm50* data are sparsely situated in the southern and north-eastern regions and do not capture the spread of high values evident in the exhaustive *Permeability* data.

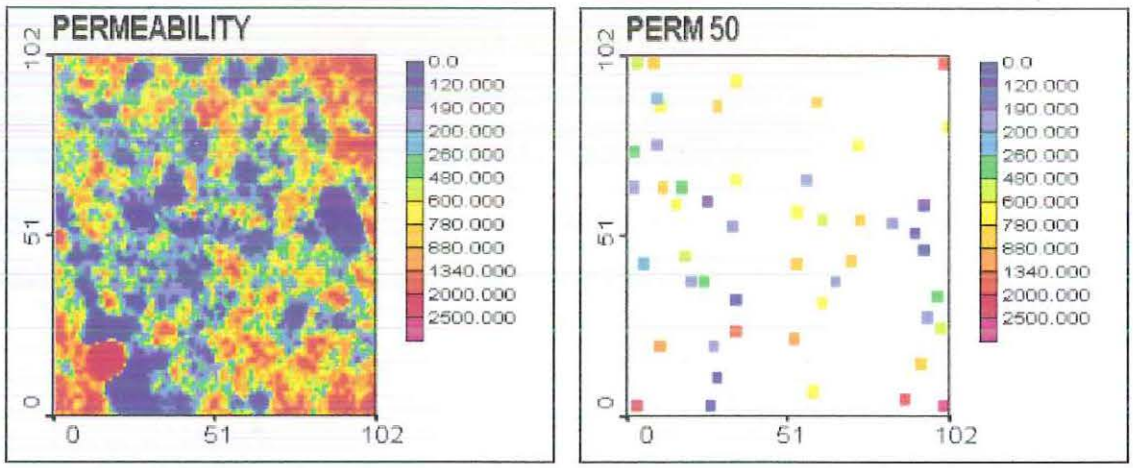

Figure 5.3. Plots of permeability values for *Permeability* (left) and *Perm50* (right).

5.2 Variography

The direct simulation algorithms we investigate all make use of simple kriging, and for this to be applied we must first calculate an experimental semivariogram and fit an appropriate model. The direct sequential simulation algorithms do not require the data to be transformed, but in order to apply sequential Gaussian simulation the *Perm50* data must first be transformed to obtain standard normal scores. The

variogram surfaces from *Perm50* and the *Perm50 Normal Scores* data sets were created using 4 lags with a lag spacing of 10 and are shown in Figure 5.4. There appears to be no strong evidence of anisotropy evident in either data set, so an isotropic model was fitted in both instances.

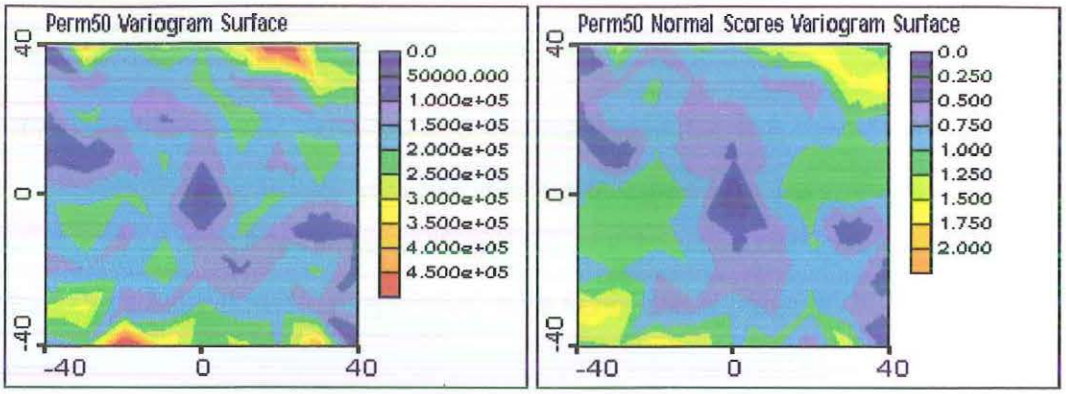

Figure 5.4. Variogram surface for *Perm50* and *Perm50 Normal Scores*.

The omnidirectional experimental semivariogram for *Perm50* was calculated using 12 lags at a lag spacing of 5.5. The fitted model and its parameters are shown in Figure 5.5 and Table 5.2 respectively. The model consists of a nugget effect and two spherical structures. The contribution of the nugget is approximately 19% of the total sill and the two spherical structures have ranges of 10 and 50 respectively.

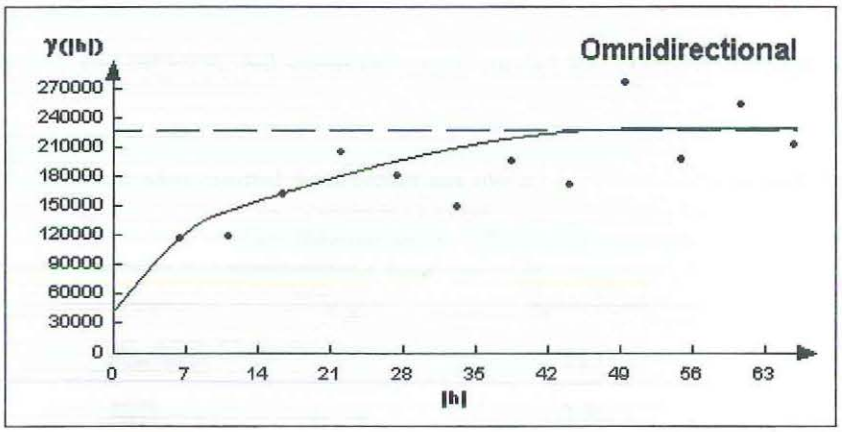

Figure 5.5. Omnidirectional experimental semivariogram model for *Perm50*.

Table 5.2. Omnidirectional semivariogram model for *Perm50*.

	1st Structure	2nd Structure	3rd Structure
Type	Nugget	Spherical	Spherical
Range	-	10.0	50.0
Sill	43680	63000	124000

The omnidirectional experimental semivariogram for the *Perm50 Normal Scores* was calculated using 8 lags at a lag spacing of 10 and its parameters are given in Table 5.3. The model is shown in Figure 5.6 and it consists of a nugget contributing 14% of the total sill and one spherical structure with a range of 24.

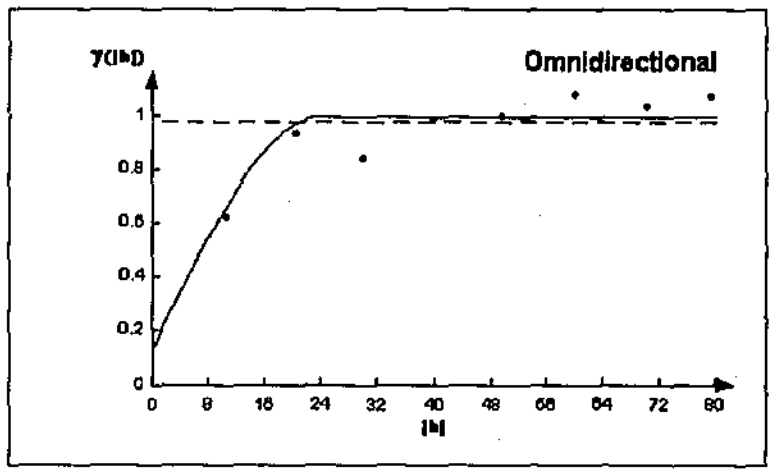

Figure 5.6. Omndirectional semivariogram model for *Perm50 Normal Scores*.

Table 5.3. Omnidirectional semivarigram model for *Perm50 Normal Scores*.

	1st Structure	2nd Structure
Type	Nugget	Spherical
Range	-	24.0
Sill	0.14	0.86

5.3 Simulation

The *Permeability* data set is positively skewed, so when using SGSIM, the lower tail is extrapolated using a negatively skewed power model with $\omega = 2.5$ and for the upper tail a hyperbolic model with $\omega = 2.0$. The cumulative frequenccy distribution for *Permeability*, shown in Figure 5.2, has a relatively long tail, and we do not want to understate the probability of occurrence of very large values. The minimum value for the permeability values is set to 1. The DSSIM algorithm does not have a maximum value set but the other algorithms have their maximum value set to 2500. The positive skewness visible in the *Permeability* data set suggests considering a lognormal local conditional distribution, by transforming the mean and variance of the data using Equations (35)-(36).

To reduce computational effort, the conditioning data is located within a neighbourhood of the location being simulated. At least four and at most twenty original data values, and up to twelve simulated values are used in each simulation. A multiple-grid concept is used, whereby a coarse grid is simulated first and then used to condition a second, finer grid simulation. The grid refinement is performed three times and this results in better reproduction of the long range variogram structure.

The random number generator draws independent seed values uniformly distributed in $[0, 1]$. Each realisation is identified by its random number seed, and this number remains the same for each simulation algorithm. This means, for example, that realisation #1 using SGSIM was generated using the same random path as realisation #1 using DSSIM. For each algorithm we generate 100 realisations.

In Figure 5.7 and Figure 5.8 we compare the mosaic maps for the realisations with the minimum, median and maximum mean absolute deviation and mean squared deviation from the *Permeability* data set respectively. The realisations with the maximum deviations are the same for both accuracy measures for all simulation algorithms except DSSL1. The realisations with the minimum deviations are the same for both accuracy measures when using DSSIM and DSSL2. There does not appear to be any significant differences between the corresponding realisations of the different measures.

The SGSIM and DSSIM algorithms appear to overestimate the values in most

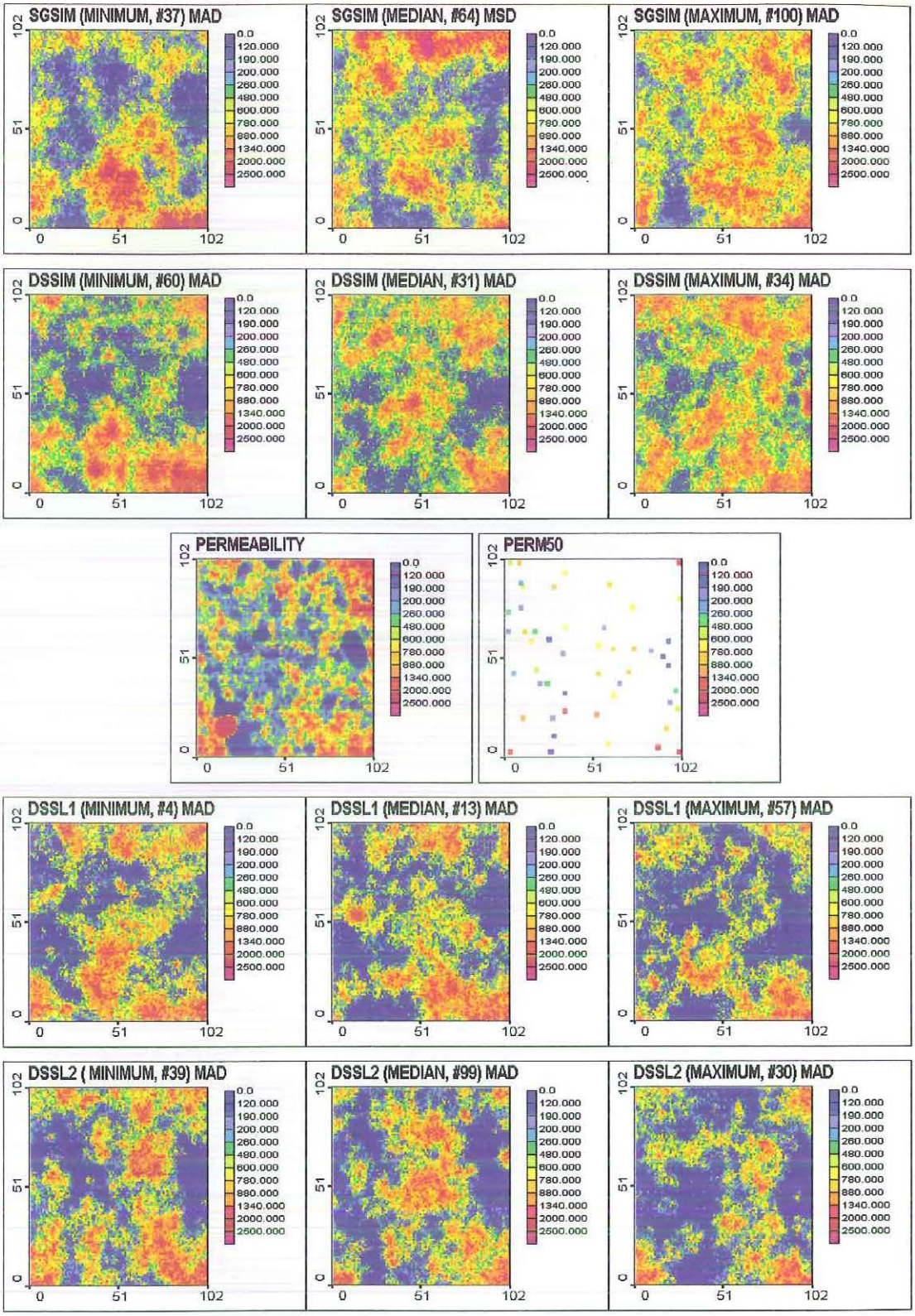

Figure 5.7. Post plots for realisations with minimum, median and maximum mean absolute deviation.

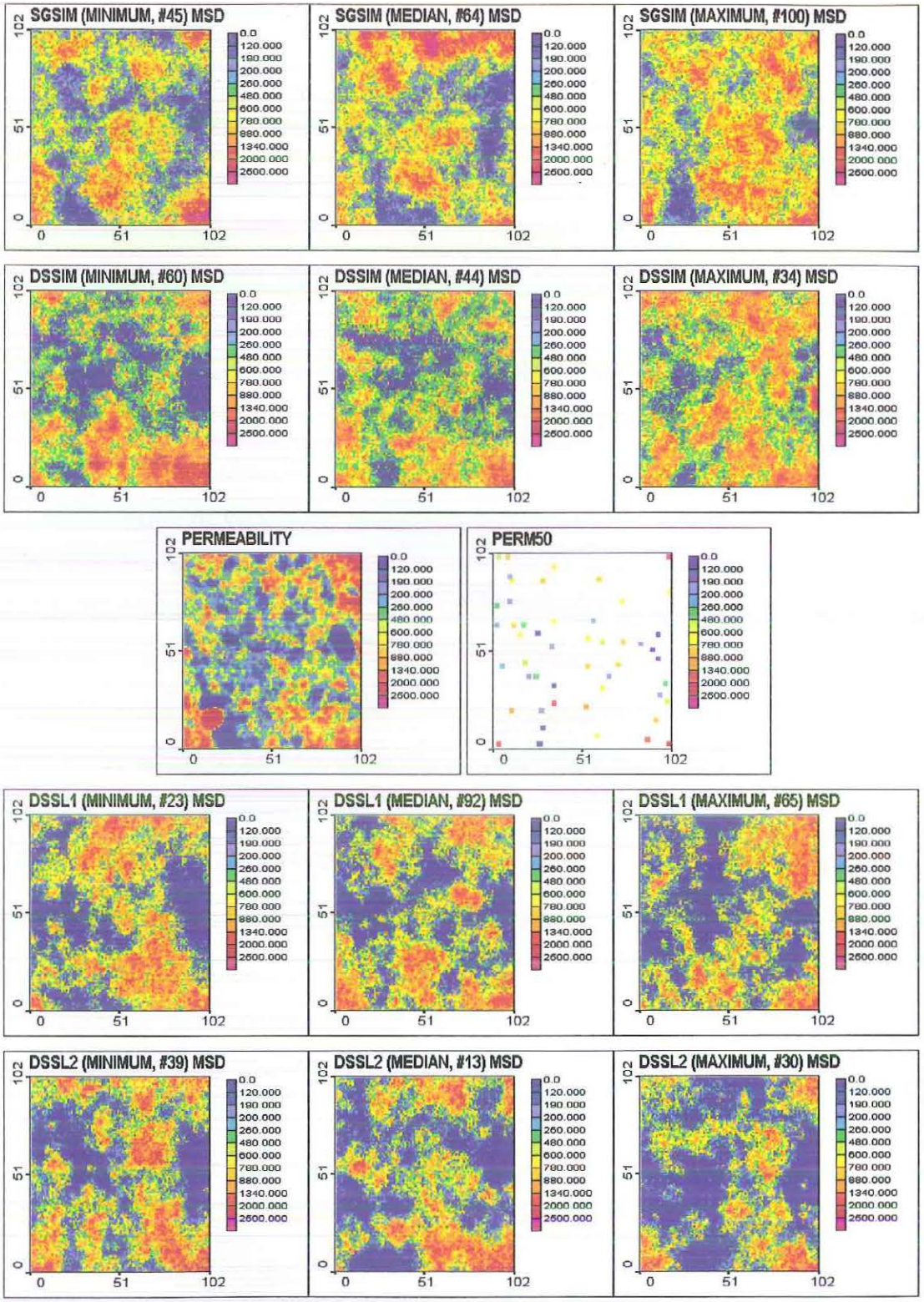

Figure 5.8. Post plots for realisations with minimum, median and maximum mean squared deviation.

regions and the spatial features are not reproduced very well. The maps show that the high regions seen in the *Permeability* map have not been reproduced. The algorithms have attempted to replicate the region of higher values in the southeast corner and the region of low values near the central eastern border. The DSSL1 and DSSL2 algorithms reproduce the spatial features well and indicate that there are high values in the southwest and northeast corners but they tend to underestimate the lower values. They also indicate the presence of high values in the southeastern corner and low values near the central eastern border. There is far better reproduction of the low values within the western half using the DSSL1 and DSSL2 algorithms than with SGSIM and DSSIM.

The region of very high values seen the the southwest corner in the *Permeability* map has not been reproduced by any of the simulation algorithms but this is understandable since the this area was not sampled from. The realisations relating to DSSL1 and DSSL2 appear more disjointed and scattered than the SGSIM and DSSIM realisations. The DSSIM realisation appear to be smoother than the others, and there seems to be a lot of locations with values closer to the mean.

5.4 Histogram Reproduction

The histograms for the realisations with the best, worst and median reproduction are shown in Figure 5.9. The SGSIM realisations match closely to the *Perm50* distribution, but they do not reproduce the *Permeability* distribution. The DSSIM realisations are influenced by the local conditional distribution used in the simulation process, and this lognormal distribution is clearly evident in the graphs. As the mean absolute deviation increases there is a noticeable change in the shape of the DSSIM distributions. The distribution of the realisation with the maximum deviation appears to be almost normally distributed. The distributions for DSSL1 and DSSL2 are very similar and both reproduce the *Permeability* distribution very well. The large spikes of low values are due to the random deviate being set equal to the simple kriging mean when the algorithms fail to find a solution. This problem occurs when the simple kriging mean is less than the midpoint for the first global probability threshold. The spike increases in size as the mean absolute deviation increases.

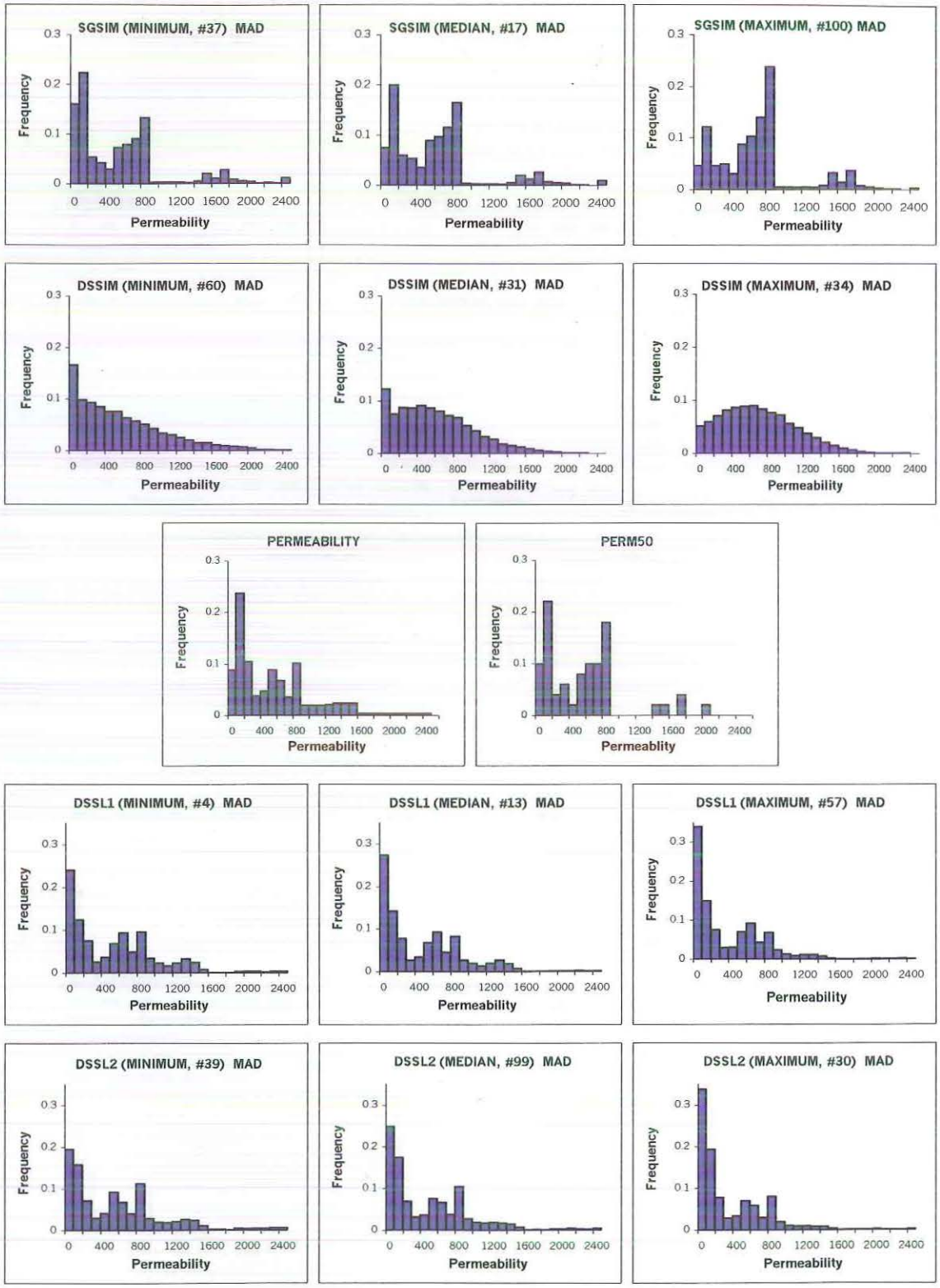

Figure 5.9. Histograms for minimum, median and maximum mean absolute deviation.

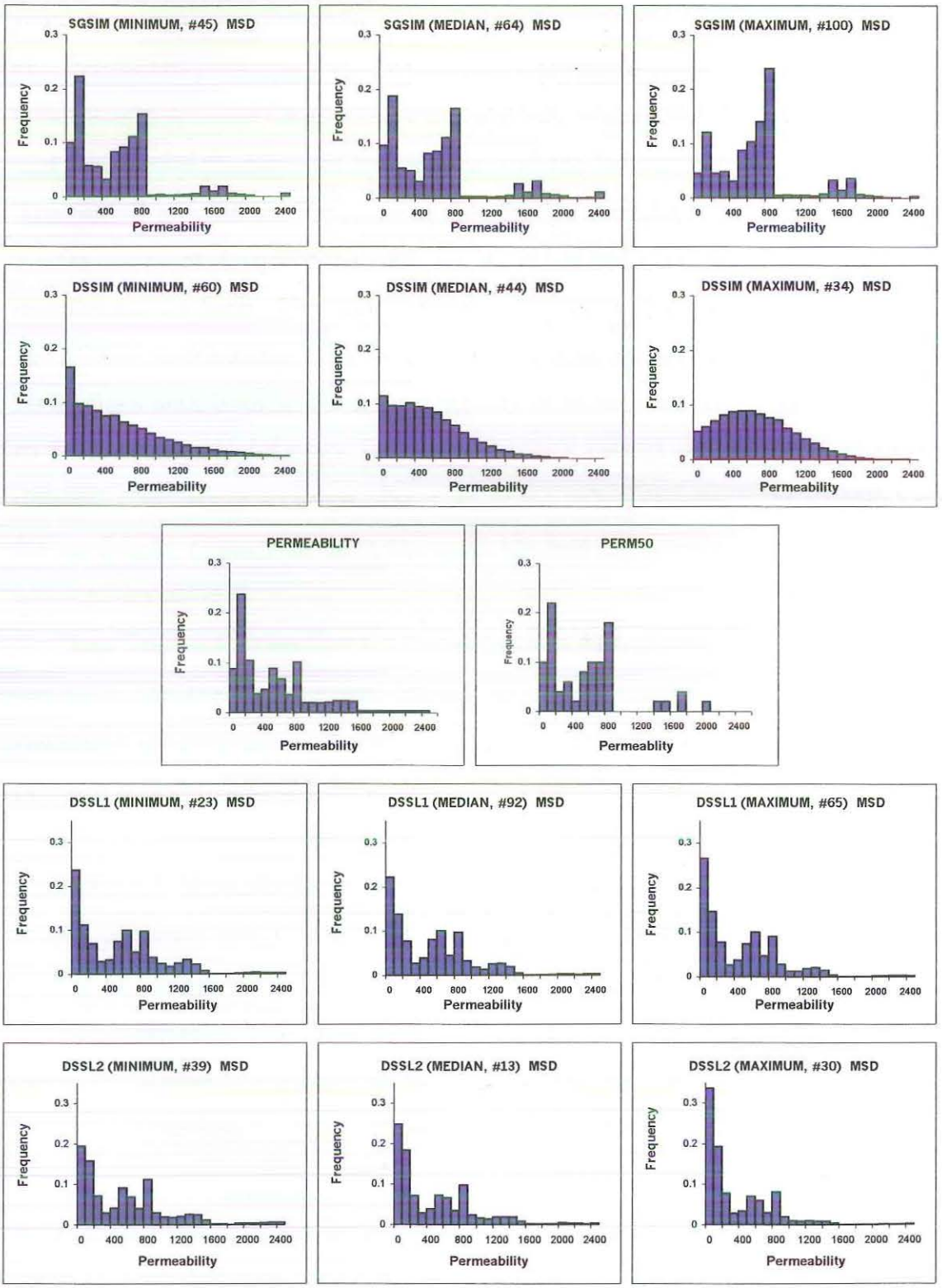

Figure 5.10. Histograms for minimum, median and maximum mean squared deviation.

Figure 5.10 compares the probability distribution functions for the realisations with the minimum, median and maximum mean squared deviation. The SGSIM realisations again match closely to the *Perm50* distribution and they do not match the *Permeability* distribution. The DSSIM realisations are influenced by the local conditional lognormal distribution with relatively smooth distributions. The DSSL1 and DSSL2 distributions are very similar and the low value spikes are noticeable. The reproduction of the *Permeability* distribution is evident.

For the SGSIM algorithm realisation #100 has the worst match with the global distribution for both. The choice of measure does not have a significant effect on the SGSIM distributions. The DSSIM distributions are noticeably different for the realisations with both minimum and maximum deviations. There does not appear to be a significant difference in the DSSL1 and DSSL2 realisations between the different measures of accuracy. The spike of low values appears to be slightly lower for the DSSL1 realisations using the mean squared deviation rather than the mean absolute deviation.

From Table 5.4 we see that for the realisations with the minimum mean absolute deviation, the DSSL2 algorithm appears to perform better, but when we look at those with the median and maximum mean absolute deviation, the DSSIM algorithm provides a better fit to the *Permeability* data set.

Table 5.4. Mean absolute deviation between realisation and target distribution.

Histogram MAD	SGSIM		DSSIM		DSSL1		DSSL2	
Rank	#	MAD	#	MAD	#	MAD	#	MAD
Minimum	37	53.47	60	41.51	4	49.79	39	33.70
Median	17	87.28	31	75.79	13	96.99	99	88.05
Maximum	100	170.00	34	158.91	57	183.10	30	207.51

A similar result is seen in Table 5.5 when comparing the realisations using the mean squared deviation. The difference between the two deviation methods is seen in the realisations with the median deviations. For the mean absolute deviation we find DSSIM <SGSIM<DSSL2<DSSL1 but for the mean squared deviation the order

has changed to DSSIM<DSSL2<DSSL1<SGSIM. The DSSL1 performs better than the DSSL2 algorithm for the realisations with the minimum and median deviations.

Table 5.5. Mean sqared deviation between realisation and target distribution.

Histogram MSD	SGSIM		DSSIM		DSSL1		DSSL2	
Rank	#	MSD	#	MSD	#	MSD	#	MSD
Minimum	45	79.03	60	52.26	23	67.53	39	45.17
Median	64	128.46	44	99.7	92	124.04	13	106.30
Maximum	100	203.48	34	176.81	65	213.60	30	234.65

Table 5.6 and Table 5.7 compare the realisations with the minimum mean absolute deviation and the mean squared deviation respectively. For DSSL1 and DSSL2, the median values are higher and the lower quartiles are less than the target values. The upper quartile for SGSIM with both accuracy measures is significantly lower than the upper quartile for *Permeability*.

Table 5.6. Comparison of realisations with minimum mean absolute deviation.

Minimum MAD	SGSIM	DSSIM	DSSL1	DSSL2	*Permeability*	*Perm50*
Mean	556.24	588.19	542.52	570.63	582.95	592.53
Std Deviation	520.47	500.73	495.70	517.48	502.5	481.65
Skewness	1.45	1.06	1.05	1.23	1.26	1.18
Minimum	1.84	1.00	1.00	1.00	1.03	4.47
1st quartile	192.80	187.27	127.51	174.97	194.21	194.28
Median	469.76	476.76	496.90	503.56	480.69	576.30
3rd quartile	794.00	867.67	866.35	870.32	869.17	853.42
Maximum	2500	3559.29	2498.92	2497.78	2498.87	2081.85

No restriction was placed on the maximum value for the DSSIM algorithm and this value is very high. The SGSIM has a higher standard deviation than *Permeability* when using the mean absolute deviation, but when using the mean squared deviation the standard deviation is less. Compared to *Permeability*, both DSSIM and DSSL2 have a lower degree of skewness for both accuracy measures.

Table 5.7. Comparison of realisations with minimum mean squared deviation.

Minimum MSD	SGSIM	DSSIM	DSSL1	DSSL2	*Permeability*	*Perm50*
Mean	557.90	588.19	553.67	570.63	582.95	592.53
Std Deviation	446.91	500.73	492.13	517.48	502.5	481.65
Skewness	1.35	1.06	1.00	1.23	1.26	1.18
Minimum	2.00	1.00	1.01	1.00	1.03	4.47
1st quartile	194.31	187.27	137.06	174.97	194.21	194.28
Median	562.02	476.76	525.43	503.56	480.69	576.30
3rd quartile	785.01	867.67	870.81	870.32	869.17	853.42
Maximum	2500	3559.29	2499.82	2497.78	2498.87	2081.85

There are quite a few differences between the realisations with the median mean absolute deviations and median mean squared deviation, seen in Table 5.8 and Table 5.9. The most significant difference is with the lower quartile values where DSSL1 is markedly less than *Permeability* when we use the mean absolute deviation. Although not quite as low, the lower quartile value for DSSL2 is also significantly less than for *Permeability* with both accuracy measures.

Table 5.8. Comparison of realisations with median mean absolute deviation.

Median MAD	SGSIM	DSSIM	DSSL1	DSSL2	*Permeability*	*Perm50*
Mean	614.81	600.16	483.55	497.22	582.95	592.53
Std Deviation	479.32	433.22	473.22	487.96	502.5	481.65
Skewness	1.37	0.80	1.21	1.36	1.26	1.18
Minimum	2.61	1.00	1.03	1.02	1.03	4.47
1st quartile	198.19	259.66	57.51	101.31	194.21	194.28
Median	585.54	541.05	324.51	327.25	480.69	576.30
3rd quartile	841.35	865.26	763.03	819.11	869.17	853.42
Maximum	2500	3327.74	2495.29	2499.37	2498.87	2081.85

The median values are also lower for DSSL1 and DSSL2 when using the mean absolute deviation, and DSSL2 is also lower when using the mean squared deviation. The SGSIM values are significantly higher than those for *Permeability* for both

measures. The DSSIM realisations are not as skewed as the target distribution of the other realiastions generated by the other algorithms. Again we notice the DSSIM realisations have very high maximum values. All realisations, except for SGSIM when using the mean squared deviation, have a lower standard deviation than *Permeability*.

Table 5.9. Comparison of realisations with median mean squared deviation.

Median MSD	SGSIM	DSSIM	DSSL1	DSSL2	*Permeability*	*Perm50*
Mean	635.24	557.76	528.53	493.81	582.95	592.53
Std Deviation	517.80	401.38	474.32	491.53	502.5	481.65
Skewness	1.28	0.87	1.11	1.36	1.26	1.18
Minimum	2.05	1.00	1.00	1.03	1.03	4.47
1st quartile	195.11	240.96	156.44	104.22	194.21	194.28
Median	593.64	496.35	492.93	283.60	480.69	576.30
3rd quartile	865.27	795.96	846.61	807.29	869.17	853.42
Maximum	2500	2988.07	2494.21	2499.09	2498.87	2081.85

The summary statistics for the realisations with the maximum mean absolute deviation and maximum mean squared deviation are given in Table 5.10 and Table 5.11 respectively.

Table 5.10. Comparison of realisations with maximum mean absolute deviation.

Maximum MAD	SGSIM	DSSIM	DSSL1	DSSL2	*Permeability*	*Perm50*
Mean	711.36	709.97	398.18	375.29	582.95	592.53
Std Deviation	449.30	425.24	428.36	415.05	502.5	481.65
Skewness	0.96	0.64	1.52	1.68	1.26	1.18
Minimum	1.65	1.00	1.01	1.01	1.03	4.47
1st quartile	369.75	382.90	37.15	49.21	194.21	194.28
Median	724.26	668.65	203.55	196.17	480.69	576.30
3rd quartile	874.07	983.12	643.59	607.66	869.17	583.42
Maximum	2500	3136.33	2498.34	2494.34	2498.87	2081.85

Table 5.11. Comparison of realisations with maximum mean squared deviation.

Maximum MSD	SGSIM	DSSIM	DSSL1	DSSL2	*Permeability*	*Perm50*
Mean	711.36	709.97	472.70	375.29	582.95	592.53
Std Deviation	449.30	425.24	454.40	415.05	502.5	481.65
Skewness	0.96	0.64	1.25	1.68	1.26	1.18
Minimum	1.65	1.00	1.00	1.01	1.03	4.47
1st quartile	369.75	382.90	64.59	49.21	194.21	194.28
Median	724.26	668.65	334.72	196.17	480.69	576.30
3rd quartile	874.07	983.12	737.95	607.66	869.17	583.42
Maximum	2500	3136.33	2495.16	2494.34	2498.87	2081.85

The SGSIM and DSSIM realisations have much greater means than the target distribution, whereas those from DSSL1 and DSSL2 have significantly lower means. This also appears to be the case with the lower quartile values, medians and upper quartile values. The SGSIM, DSSIM and DSSL2 algorithms have the same realisation with maximum mean absolute deviation as maximum mean squared deviation. The SGSIM and DSSIM realisations are not as skewed as the *Permeability* data set.

The cumulative distributions for the mean absolute deviation and the mean squared deviation are shown in Figure 5.11 and they indicate that DSSL1 and DSSL2 better represent the cumulative distribution function of the target distribution. The deviation in the lower half of the SGSIM and DSSIM distributions is significant, although the SGSIM realisation with the minimum deviation has a very good fit to the target distribution. There is very little difference between the two accuracy measures for DSSL2. The target cumulative distribution function for DSSL1 is well reproduced for the lower values when the mean absolute deviation is compared. The greatest deviation for DSSL1 and DSSL2 for the realisations with the smaller deviations, is seen in the upper quartile region.

Looking at the cumulative distributions for both the mean absolute deviation and the mean sqared deviation, and taking into account the difference between the best and worst fitted realisations, the DSSL1 algorithm appears to result in the best histogram reproduction.

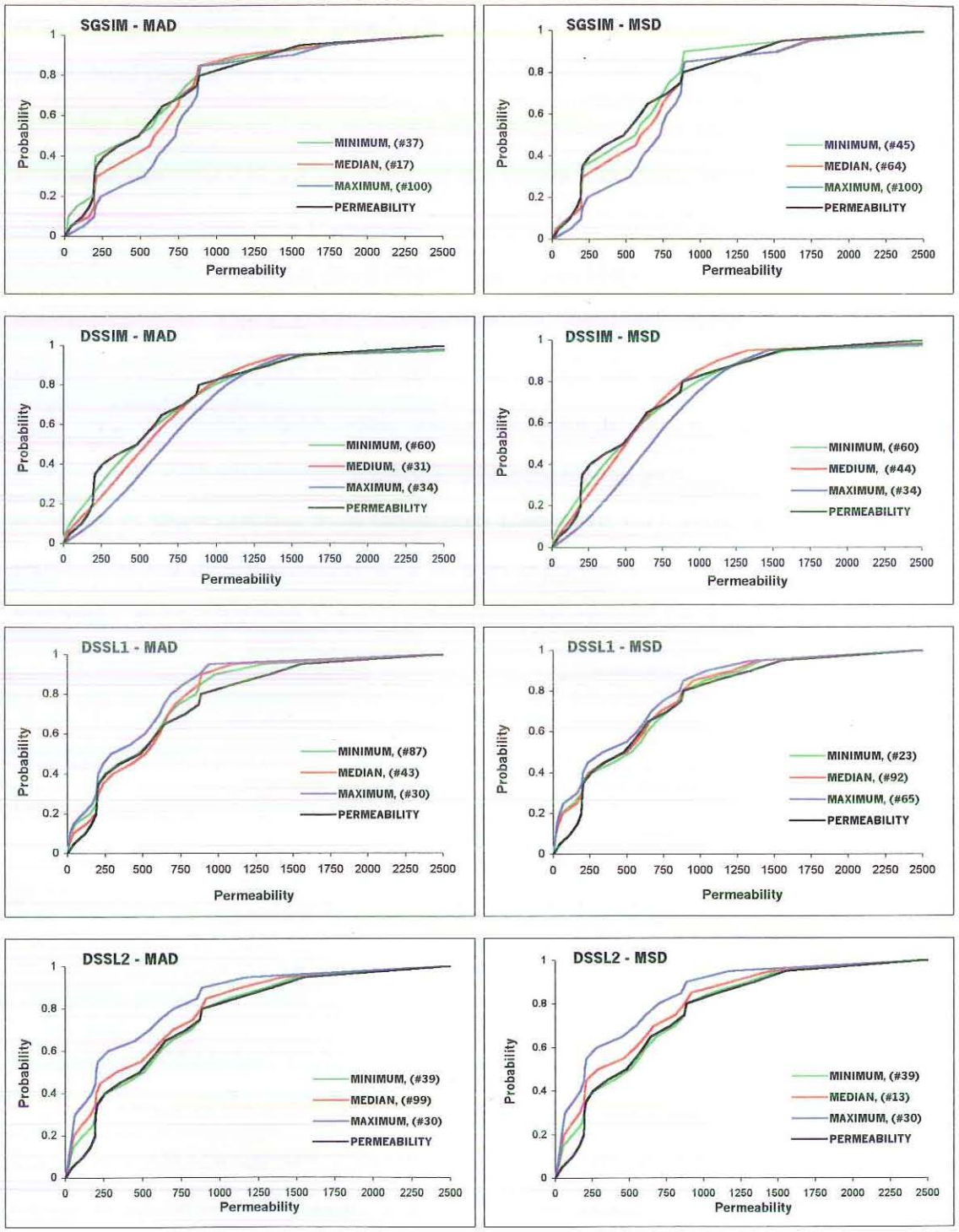

Figure 5.11. Cumulative distribution functions for minimum, median and maximum mean absolute deviation (left) and mean squared deviation (right).

5.5 Variogram Reproduction

The omnidirectional experimental semivariograms are displayed together with the semivariogram model in Figure 5.12 (left). The program used to calculate the experimental semivariograms did not start at lag 0, as the semivariogram model does, and this accounts for the difference in the graphs. The fan shape they produce indicates that there is a wide range in sill values. As expected the SGSIM semivariogram model is very well reproduced, but this is calculated using the normal scores. SGSIM appears to have a good short range coincidence of model and experimental semivariograms. The majority of experimental semivariograms however understate the variability compared to the target, although the range of the target appears to be approximately reproduced. The DSSIM model has a higher sill than all but a few of the omnidirectional experimental semivariograms calculated and there are a couple of experimental semivariograms that have very large deviations from the semivariogram model which results in a wide spread overall. The short range fit is relatively good, although there is a tendency to understate it.

We notice with the DSSL1 model that approximately seventy percent of the omnidirectional semivariograms understate the sill. The short range variability is overstated with most graphs of experimental semivariograms above the model. The DSSL1 model also has the smaller spread of sill values when compared to DSSIM and DSSL2. The DSSL2 algorithm appears to have an overall greater variability. The short range behaviour is very similar to DSSL1. Approximately half the experimental semivariograms overstsate the long range variability.

In Figure 5.12 (right) we compare the experimental semivariograms with the minimum, median and maximum deviations from the target model. These graphs show the shape of the experimental semivariograms more clearly. Comparing the experimental semivariograms with the minimum deviation, we notice that the SGSIM model is reproduced the best, followed by the DSSL1 and DSSL2 models. The DSSIM algorithm has a larger spread in sill values between the omnidirectional experimental semivariograms with the minimum and maximum deviations.

Table 5.12 confirms that overall the DSSL1 and DSSL2 methods are both better than SGSIM and DSSIM in reproducing the target semivariogram model, with the

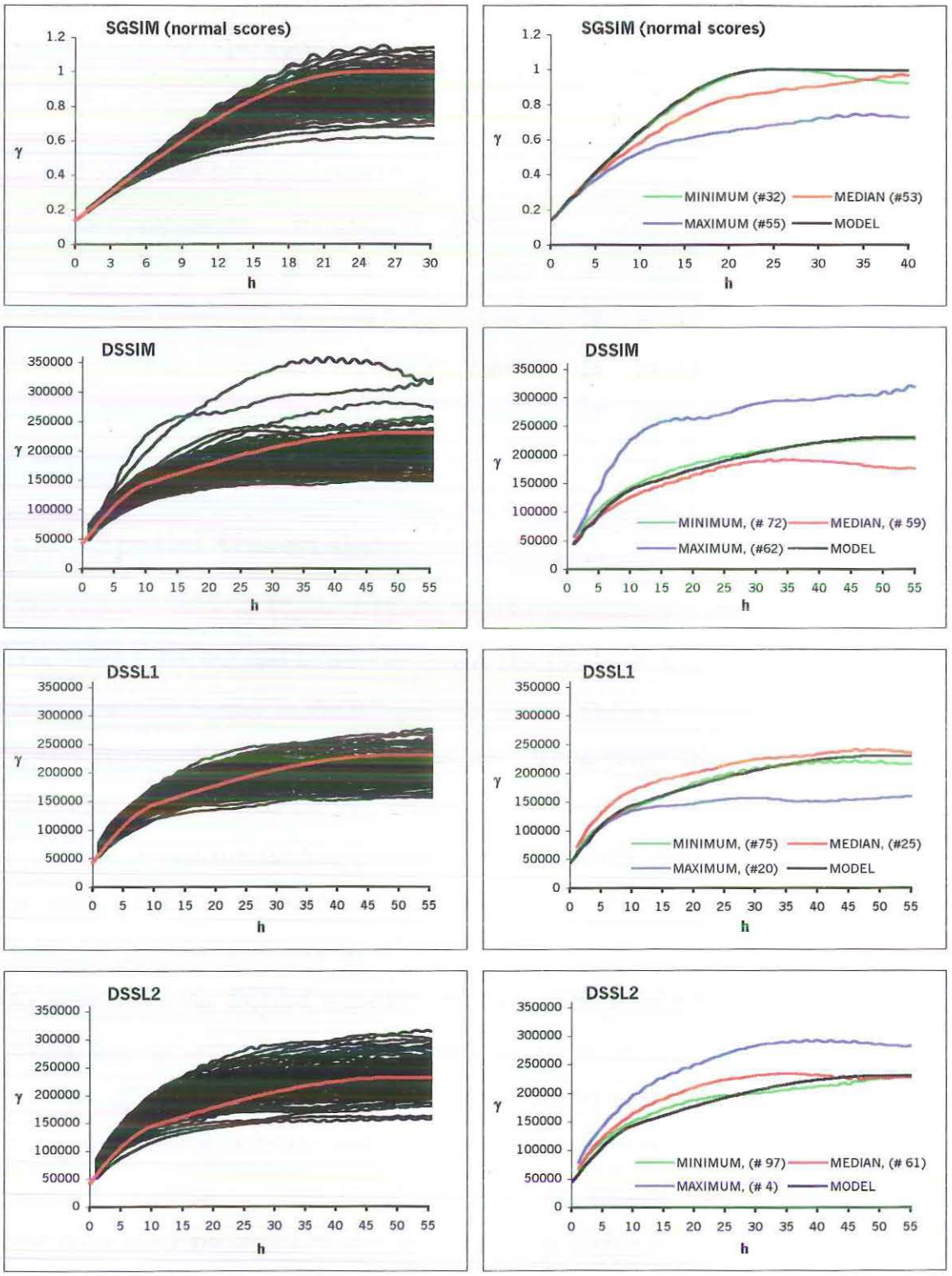

Figure 5.12. Omnidirectional experimental semivariograms for 100 realisations (left) and with minimum, median and maximum mean squared deviation (right).

DSSL1 algorithm fitting better than DSSL2, since it is more central to the fan of omnidirectional experimental semivariograms.

Table 5.12. Comparison of mean squared deviation for experimental semivariograms.

Semivariogram	SGSIM		DSSIM		DSSL1		DSSL2	
Rank	#	MSD	#	MSD	#	MSD	#	MSD
Minimum	32	0.01	72	3788.8	75	8965.2	97	9288.8
Median	53	0.09	59	25686.4	25	9413.0	61	22097.5
Maximum	55	0.25	62	105915.8	20	62951.3	4	71623.6

5.6 Spatial Uncertainty

The quantile plots in Figure 5.13 display the uncertainty in the simulated values. The DSSL1, DSSL2 and to a lesser extent DSSIM, have dark blue sections near the central eastern border in the 0.9-quantile maps. These areas indicate that there is a high degree of certainty in the simulated values being low values. The DSSIM algorithm does not indicate this area.

When comparing the 0.1-quantile and 0.9-quantile maps, the simulated values generated using the DSSIM algorithm have a lower degree of uncertainty, as they have the greatest difference in values. The highest degree of uncertainty appears to arise when the SGSIM algorithm is used. The results for DSSL1 and DSSL2 algorithms are very similar to each other.

Two grid nodes were selected from regions displaying different spatial uncertainty to compare the parametric and non-parametric algorithms. The true permeability values at locations $\mathbf{u}_1 = (88, 52)$ and $\mathbf{u}_2 = (46, 94)$ are 26.11 and 421.02 respectively and these are represented by the dashed lines in Figure 5.14.

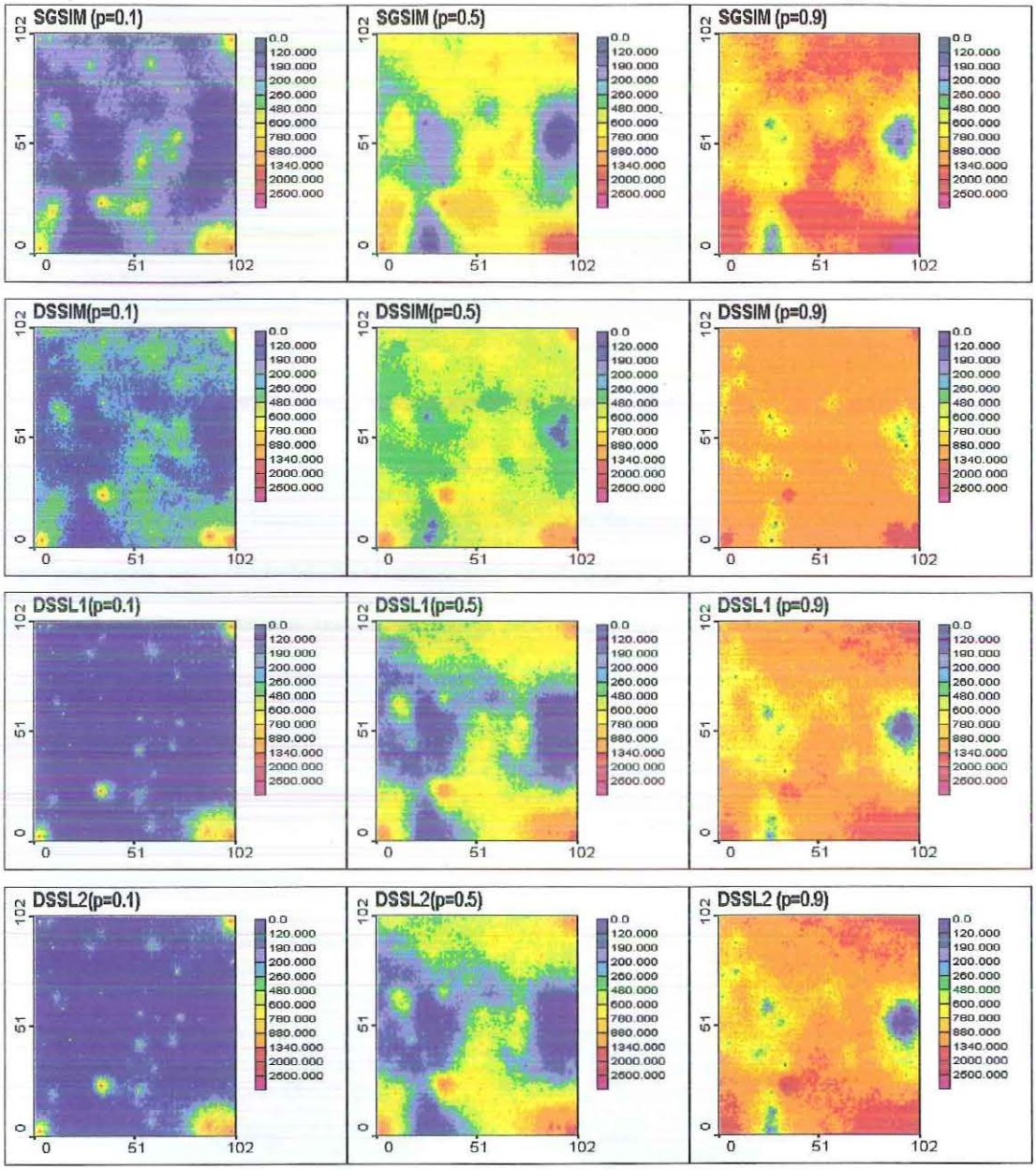

Figure 5.13. Mosaic plots comparing the 0.1-decile, 0.5-decile and 0.9-decile.

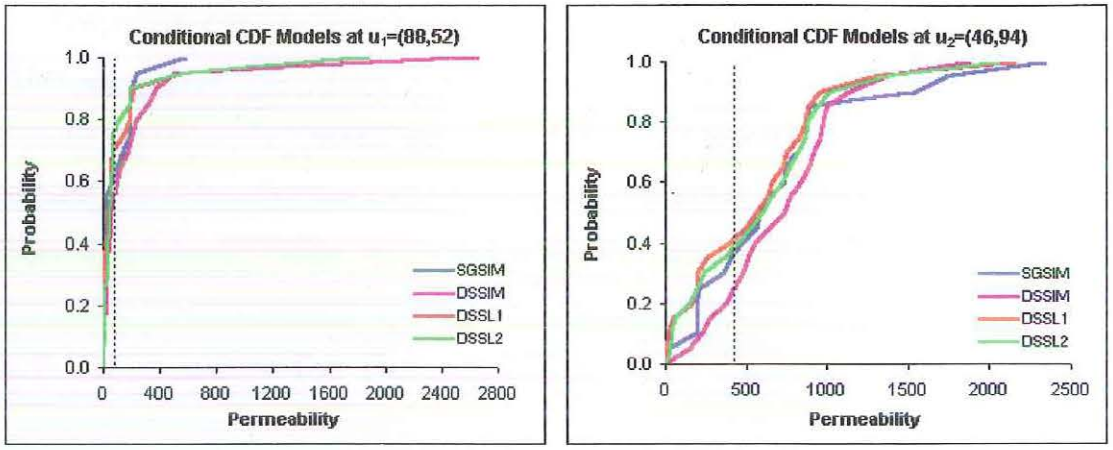

Figure 5.14. Local conditional cumulative distributions.

At location $\mathbf{u}_1$ there is a greater difference between the uncertainty models than at location $\mathbf{u}_2$. The SGSIM model is controlled by the kriging variance which is related to the distance the location is from neighbouring sample data, and this has a greater spread at location $\mathbf{u}_2$ than location $\mathbf{u}_1$ which is close to sample data. The DSSIM model has a large spread of values at both locations.

The conditional variance for each algorithm is displayed through mosaic maps in Figure 5.15.

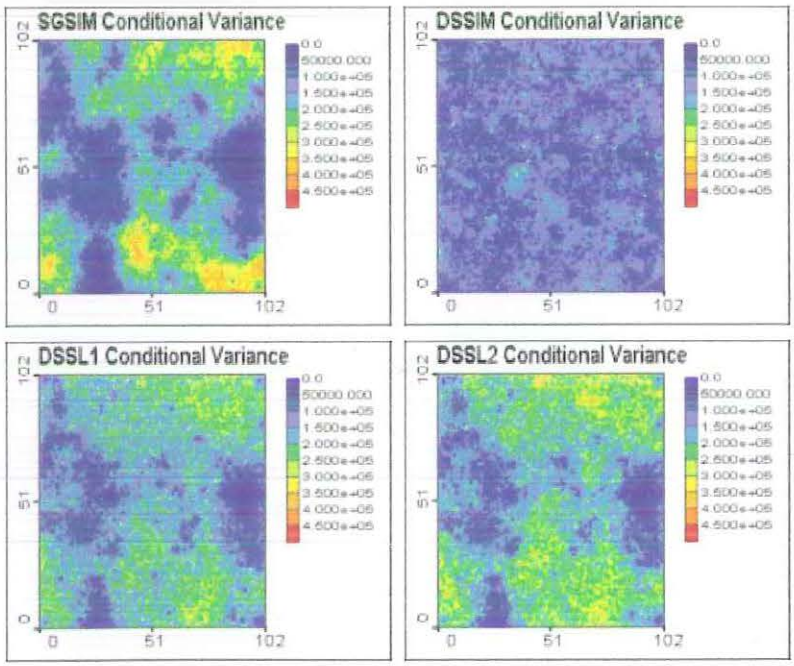

Figure 5.15. Comparison of conditional variance for the algorithms.

The conditional variance is very low for the DSSIM algorithm compared to the other algorithms. The DSSL1 and DSSL2 are very similar, although there are some regions for which the DSSL2 algorithm results in a greater variation in values. The SGSIM algorithm has the highest variation in simulated values. These regions are located in areas of higher values and where the data were sparsely sampled.

5.7 Summary for Permeability

Comparing the histogram and semivariogram reproduction for the four algorithms, we find that, as expected, SGSIM reproduces the variogram for the normal score data and the histogram of the sample data set. DSSIM reproduced the local log-normal distribution used in the algorithm, but the semivariogram model was only reproduced well over a short range. The DSSL1 and DSSL2 both reproduced the target distribution but the DSSL1 algorithm reproduced the semivariogram model better than all the other algorithms, so it would appear to be the better algorithm for this isotropic case.

6 Application to the Anisotropic Case

The *Potassium* data set is a two-dimensional simulated exhaustive data set based on soil samples taken originally from an uncropped field in the Jimperding Brook catchment region in the south-west of Western Australia (Bloom & Kentwell, 1999). The data comprise potassium concentrations in parts per million (ppm). The 3600 data measurements are located on a 60×60 regular grid which has a grid spacing of one metre. Data at one hundred locations were drawn at random and form the sample data set *K100*.

6.1 Exploratory Data Analysis

Descriptive statistics for the potassium variable from *Potassium* and *K100* are listed in Table 6.1.

Table 6.1. Descriptive statistics for *Potassium* and *K100* data sets.

	Potassium	*K100*
n	3600	100
Mean	87.097	89.447
Std Deviation	34.703	34.416
Variance	1204.31	1184.46
Skewness	1.035	0.603
Kurtosis	1.962	0.08
Minimum	23.872	34.6
1st quartile	59.941	63.875
Median	82.470	86.85
3rd quartile	106.867	110.575
Maximum	331.978	189.0

The data sets have similar descriptive statistics, although there is a noticeable difference in the maximum values. The sample has a significantly lower maximum value than the exhaustive set. The positive skewness evident in the exhaustive data histogram (Figure 6.1) is confirmed by the summary statistics. We notice the sample

data set. Overall, the sample data appear to reflect the summary statistics of the exhaustive data.

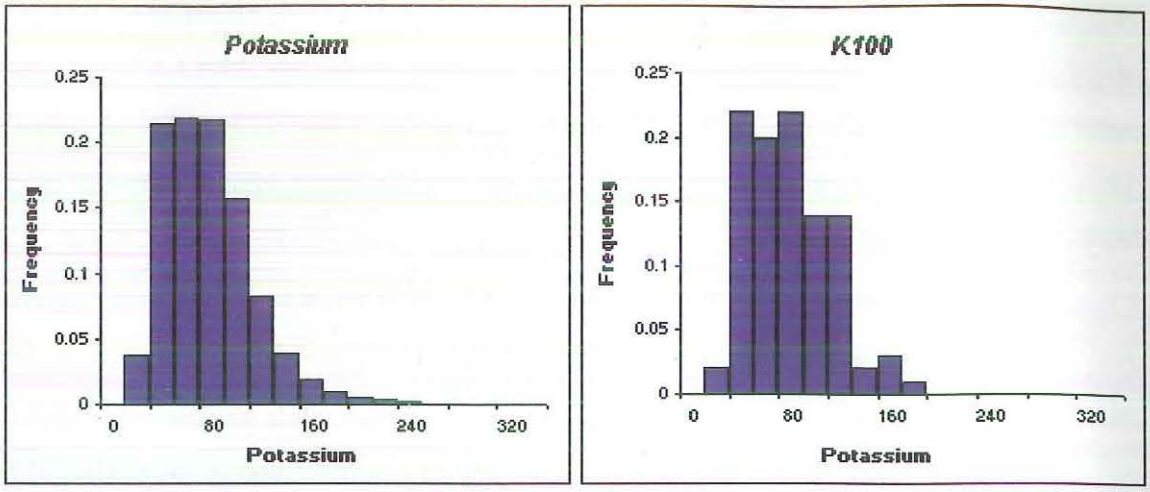

Figure 6.1. Histograms for *Potassium* (left) and *K100* (right).

There are several extreme values in the data sets. The maximum *Potassium* and *K100* values are approximately four times and twice the mean respectively. These extreme values will inflate estimated values in their neighbourhood. The Q-Q plot and cumulative distribution functions of the *Potassium* and the *K100* data sets are shown in Figure 6.2.

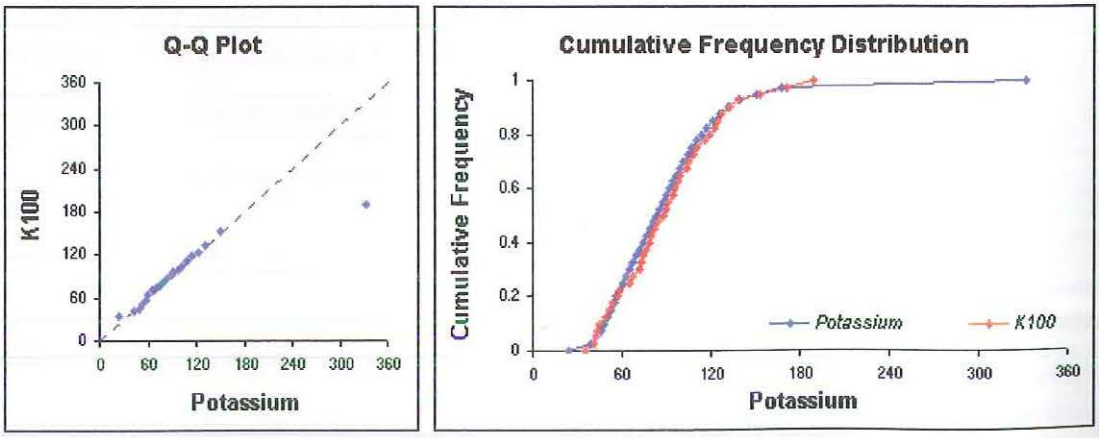

Figure 6.2. Q-Q plot (left) and cumulative frequency function (right) for *Potassium* and *K100*.

Comparing the distributions we notice that the *K100* data are representative of the *Potassium* data. The Q-Q plot reveals the effectiveness of the sampling. The

two distributions are almost identical as they follow the dashed line fairly closely, the only significant exception being a single extreme value.

The forty permeability thresholds used to define the global probability distribution in DSSL1 and DSSL2 are given in Appendix B5. These values are bounded by a minimum value of 20 and a maximum value of 360. For each simulation we will consider the conditional cumulative frequency function at two locations $\mathbf{u}_1$ $(28.5, 44.5)$ and $\mathbf{u}_2(28.5, 22.5)$ whose exact values are known to be 57.975 and 202.381 respectively. Location $\mathbf{u}_1$ is in an area of low values and location $\mathbf{u}_2$ is situated near high values. Figure 6.3 shows mosaic plots of the potassium values for *Potassium* and *K100*. The sample data appear to have a good coverage of the study region. The *K100* data set seems to capture the spread of high and low values evident in the *Potassium* data set.

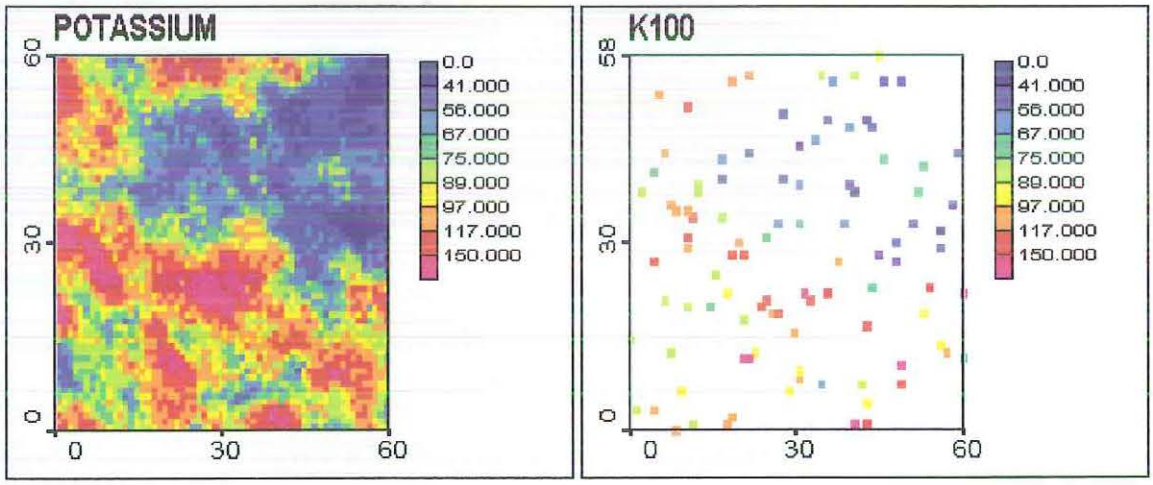

Figure 6.3. Plots of potassium values for *Potassium* (left) and *K100* (right).

6.2 Variography

An experimental semivariogram was calculated and an appropriate model fitted so simple kriging could be used in the direct simulation algorithms. The direct sequential simulation algorithms do not require any data transformation, but in order to apply sequential Gaussian simulation, the *K100* data must first be transformed to obtain standard normal scores. The variogram surfaces of the *K100* and *K100*

Normal Score data sets were created using 7 lags with a lag spacing of 3.0 and these are shown in Figure 6.4.

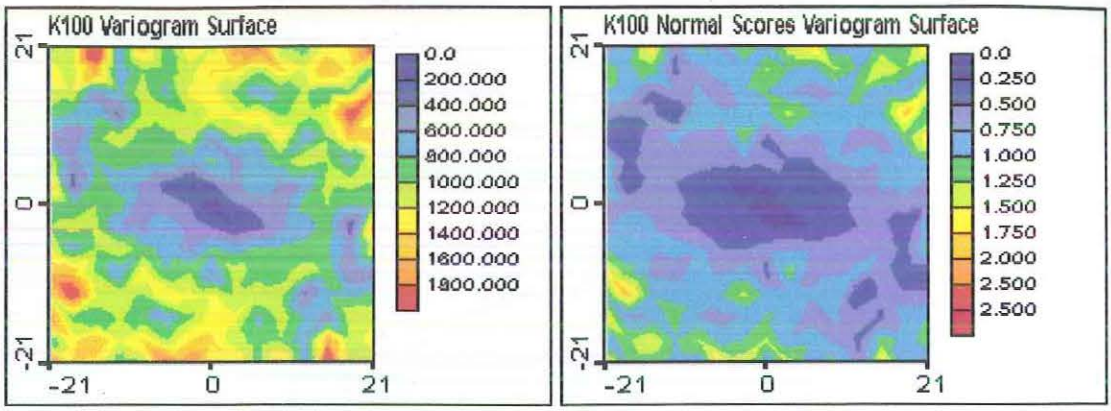

Figure 6.4. Variogram surface for *K100* and *K100 Normal Scores.*

The data sets both appear to exhibit anisotropy, with greater continuity in the N65W direction, and minimum continuity in the N25E direction. The experimental semivariograms for both data sets were calculated using 10 lags with a lag spacing of 3. A geometric anisotropic model was fitted to the experimental semivariograms of the *K100* and *K100 normal scores.* In both cases, it consists of a nugget effect and one spherical structure.

For K100, the nugget has a relative nugget effect of approximately 7% and the spherical structure has a range of 22 and anisotropy factor of 0.76. The anisotropic model is shown in Figure 6.5 with the experimental semivariograms and the parameters are given in Table 6.2. The model is fitted using the program *Variowin* and the direction is given in degrees anticlockwise from East, so direction 65 and direction 155 relate to N25E and N65W respectively.

Table 6.2. Geometric anisotropic model for *K100.*

	1st Structure	**2nd Structure**
Type	Nugget	Spherical
Range	-	22.0
Sill	84.0	1104
Anisotropic ratio	-	0.76

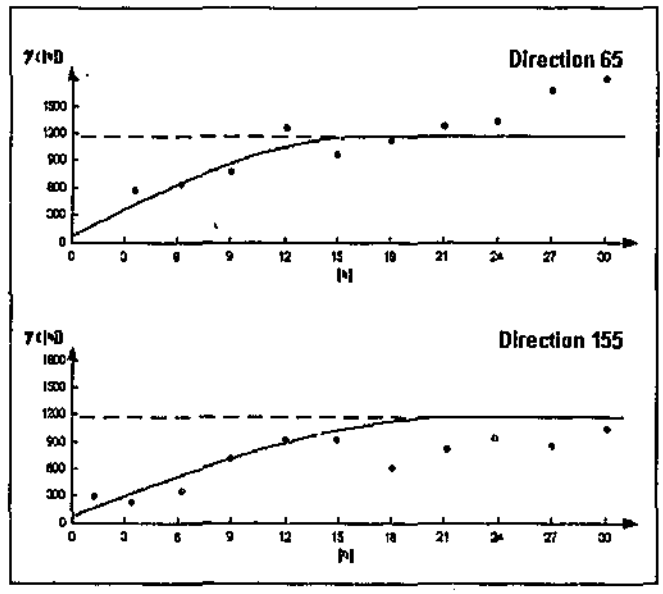

Figure 6.5. Geometric anisotropic model fitted to experimental semivariogram for *K100.*

For K100, the nugget has a relative nugget effect of approximately 8% and the spherical structure has a range of 25.0 and anisotropy factor of 0.64. The anisotropic model is fitted to the experimental semivariogram in Figure 6.6, and the parameters are given in Table 6.3.

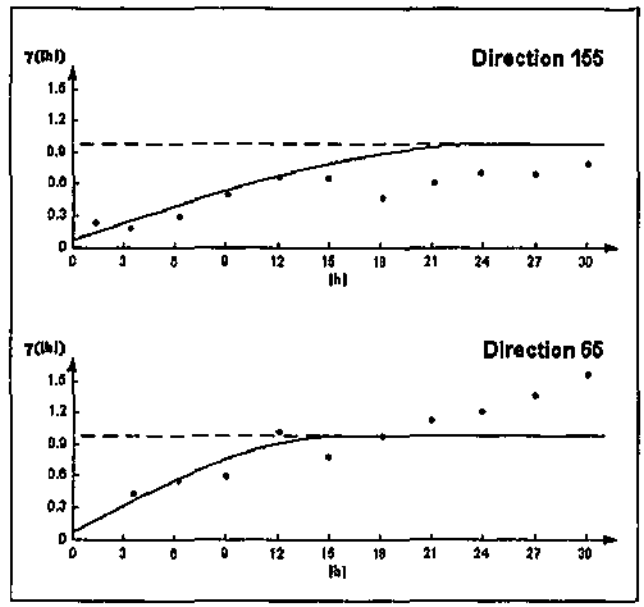

Figure 6.6. Geometric anisotropic model fitted to experimental semivariogram for *K100 Normal Scores.*

Table 6.3. Geometric anisotropic model for *K100 Normal Scores*.

	1st Structure	2nd Structure
Type	Nugget	Spherical
Range	-	25.0
Sill	0.08	0.92
Anisotropic ratio	-	0.64

6.3 Simulation

The parameter files for the simulation algorithms are listed in Appendix B. The simulated values are located on a regular grid of size 60 × 60 and they all work with a limited data search neighbourhood of 40.0 units. A minimum of four and a maximum of twenty original sample data, and a maximum of twelve simulated values are used in each simulation. For SGSIM, the cumulative distribution function (Figure 6.2) indicates extrapolating the lower tail to the minimum value of 20, using a linear model and the upper tail using a hyperbolic model with $\omega = 1.5$. One hundred realisations were generated for each algorithm, and then ranked in increasing order according to both their mean absolute deviation (MAD) and their mean squared deviation (MSD) from the exhaustive *Potassium* data set. The minimum and maximum values, *zmin* and *zmax*, for the potassium values are set to 20 and 360 respectively for SGSIM, DSSL1 and DSSL2.

The positive skewness visible in the *Potassium* data set suggests considering a lognormal local conditional distribution, by transforming the mean and variance of the data using equations (35)-(36). To reduce computational effort, the conditioning data is located within a neighbourhood of the location being simulated. At least four and at most twenty original data values, and up to twelve simulated values are used in each simulation. A multiple-grid concept is used, whereby a coarse grid is simulated first and then used to condition a second, finer grid simulation. The grid refinement is performed three times and this results in better reproduction of the long range variogram structure.

The random number generator draws independent seed values uniformly distributed in $[0, 1]$. Each realisation is identified by its random number seed, and this number remains the same for each simulation algorithm. This means, for example, that realisation #1 using SGSIM was generated using the same random path as realisation #1 using DSSIM.

6.4 Histogram Reproduction

In Table 6.4 and Table 6.5 we compare the mean absolute deviation and mean squared deviation values for the realisations with the minimum, median and maximum deviations, and from these it appears SGSIM has the closest fit to the *Potassium* data set, with DSSIM actually giving a better fit for the realisations with the maximum deviations. The DSSL2 algorithm has a better fit than DSSL1 except for the maximum mean squared deviation.

Table 6.4. Comparison of the mean absolute deviations.

Histogram MAD	SGSIM		DSSIM		DSSL1		DSSL2	
Rank	#	MAD	#	MAD	#	MAD	#	MAD
Minimum	2	2.05	73	2.60	69	5.11	52	4.18
Median	7	4.28	20	5.28	90	9.10	5	8.53
Maximum	93	9.64	69	8.78	95	13.40	95	13.23

Table 6.5. Comparison of the mean squared deviations.

Histogram MSD	SGSIM		DSSIM		DSSL1		DSSL2	
Rank	#	MSD	#	MSD	#	MSD	#	MSD
Minimum	53	2.60	29	3.35	36	7.29	93	5.77
Median	5	5.12	17	5.79	64	11.13	20	10.17
Maximum	93	11.58	69	9.58	72	14.74	33	16.01

Figure 6.7 and Figure 6.8 compare the realisations ranked according to mean absolute deviation and mean squared deviation respectively. We notice immediately when viewing the postplots that the SGSIM and DSSIM realisations appear very similar, as do DSSL1 and DSSL2.

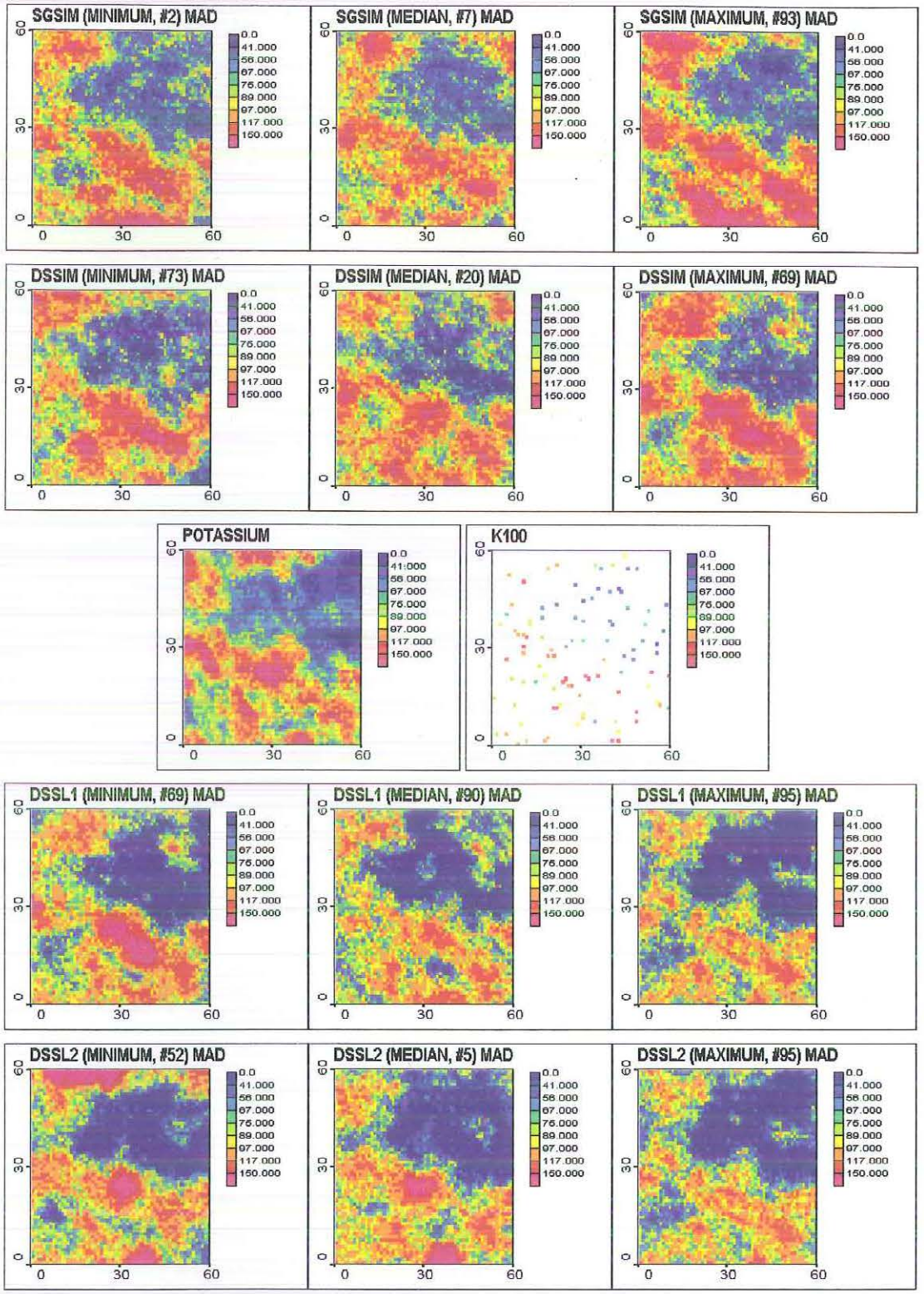

Figure 7.7. Post plots for realisations with minimum, median and maximum mean absolute deviation.

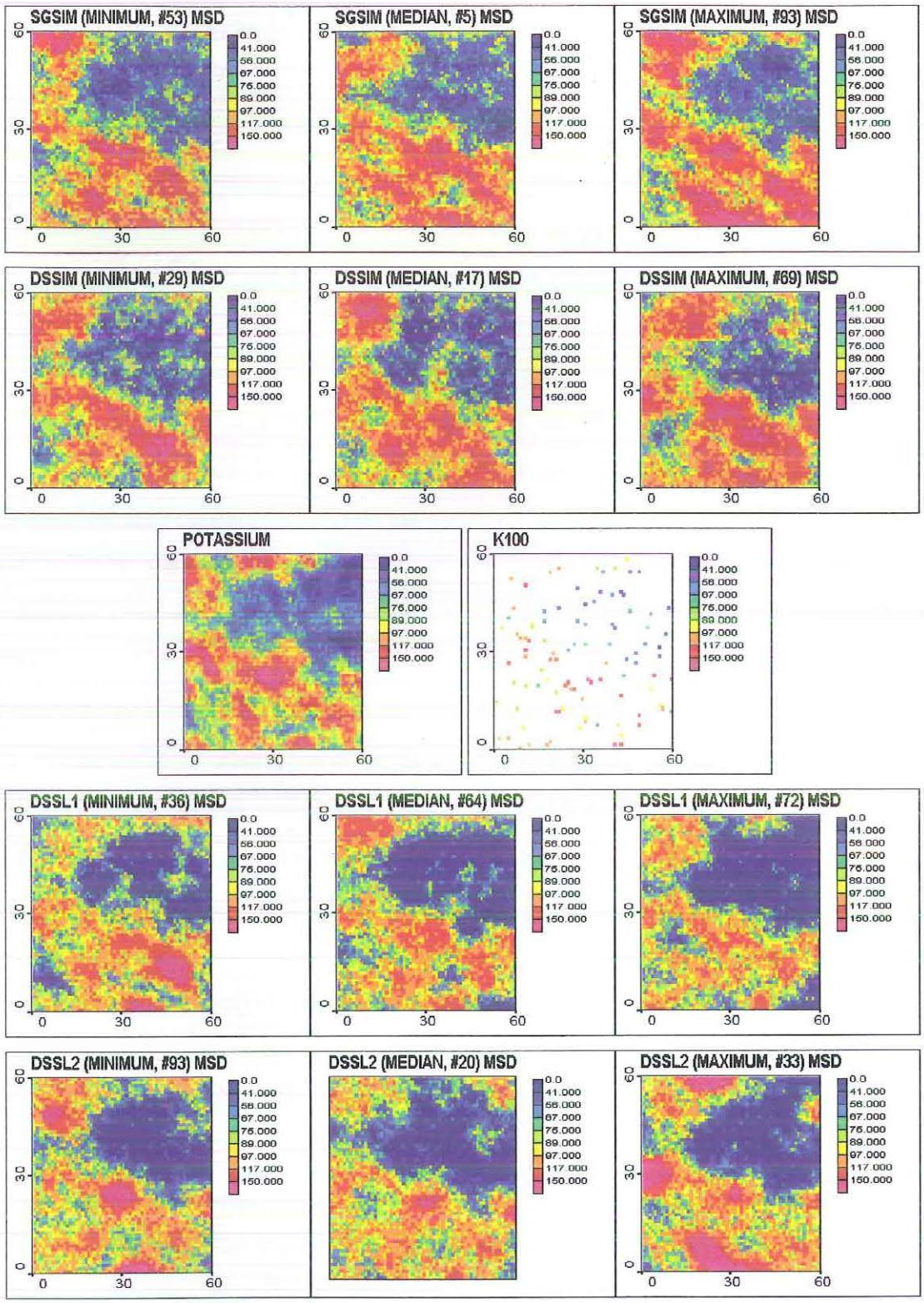

Figure 7.8. Post plots for realisations with minimum, median and maximum mean squared deviation.

There is a noticeable difference between these two distinct groups. The DSSL1 and DSSL2 realisations have a similar coverage of low values, but the values are within the lowest interval scale. The SGSIM and DSSIM realisations appear to have a rather large spread of high values that are well connected to each other. The high values from DSSL1 and DSSL2 are more disjointed and scattered in appearance.

Neither SGSIM nor DSSIM really capture the low values in the NE corner, but this is not surprising since the sampling is very sparse in this region. The DSSL1 and DSSL2 realisations capture these low values best when the deviation is a maximum. DSSL1 and DSSL2 underestimates the low values. All four algorithms seem to recognise that there is a small area of higher values within the region of low values near the NE corner. The high values near the centre of the region are captured by all the algorithms, although it appears to be more evident in the SGSIM and DSSIM post plots. It appears that SGSIM and DSSIM may overestimate the high values and DSSL1 and DSSL2 both underestimate the high valued regions. The anisotropy is visually evident from the mosaic maps.

The different measures of accuracy do not appear to differentiate between the SGSIM and DSSIM algorithms. Both the mean absolute deviation and the mean squared deviation result in realisation #93 having the maximum deviation for SGSIM. Similarly realisation #96 has the maximum deviation for the DSSIM algorithm regardless of the measure of accuracy. Using the mean absolute deviation we also notice that the realisation #95 has the maximum deviation for both DSSL1 and DSSL2.

The histograms in Figure 6.9 and Figure 6.10 highlight the differences in the distributions of the realisations. There is very little difference between the two accuracy measures for both SGSIM and DSSIM realisations. The DSSIM histograms are reproducing the local conditional lognormal distribution that was used in the simulation algorithm. Large spikes of low values are seen in the DSSL1 and DSSL2 distributions and this is due to the algorithms inability to find an optimal solution and the random deviate was set equal to a low-valued simple kriging mean. This appears to have a significant affect on the shape of the histograms.

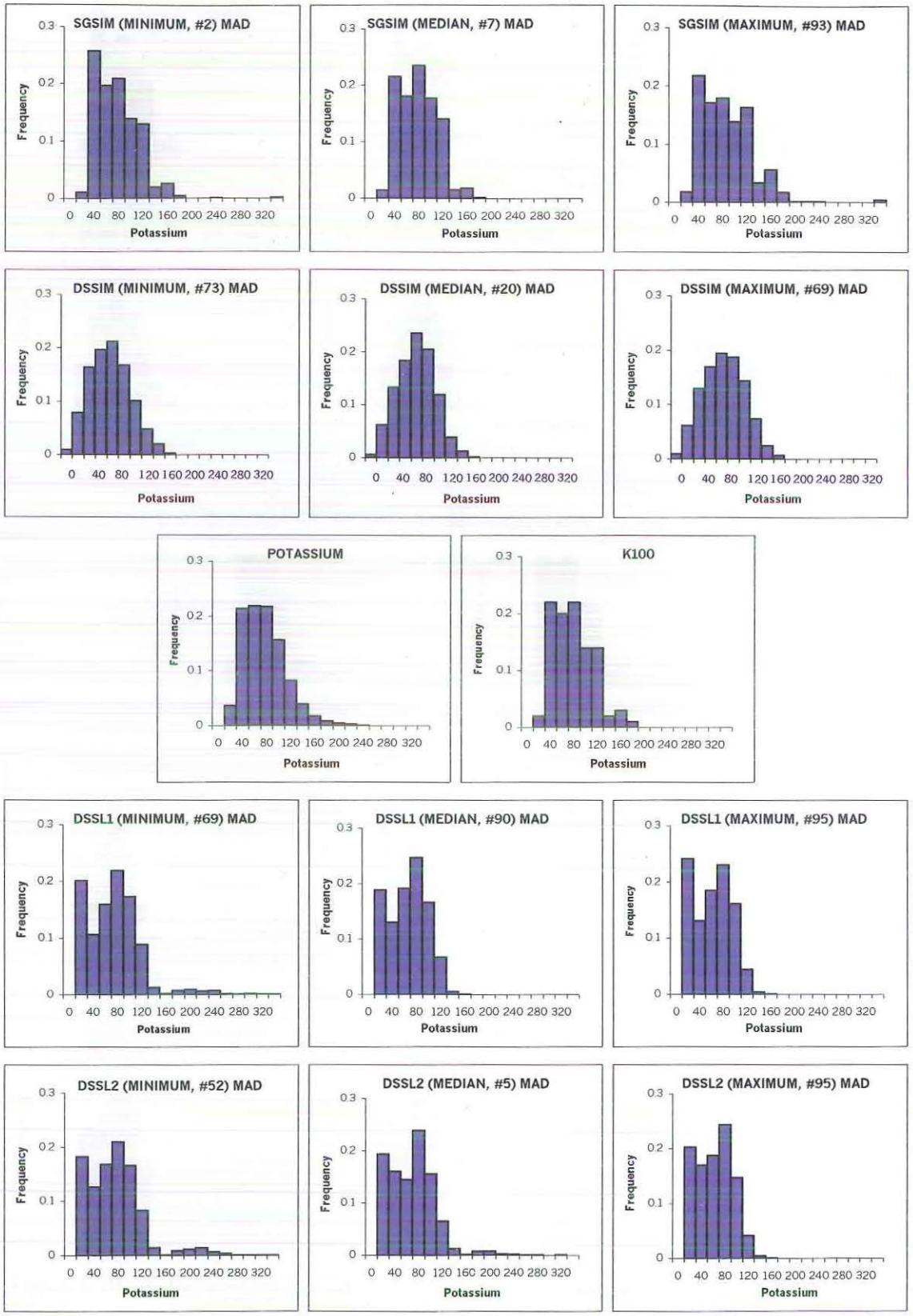

Figure 6.9. Histograms for realisations with minimum, median and maximum mean absolute deviation.

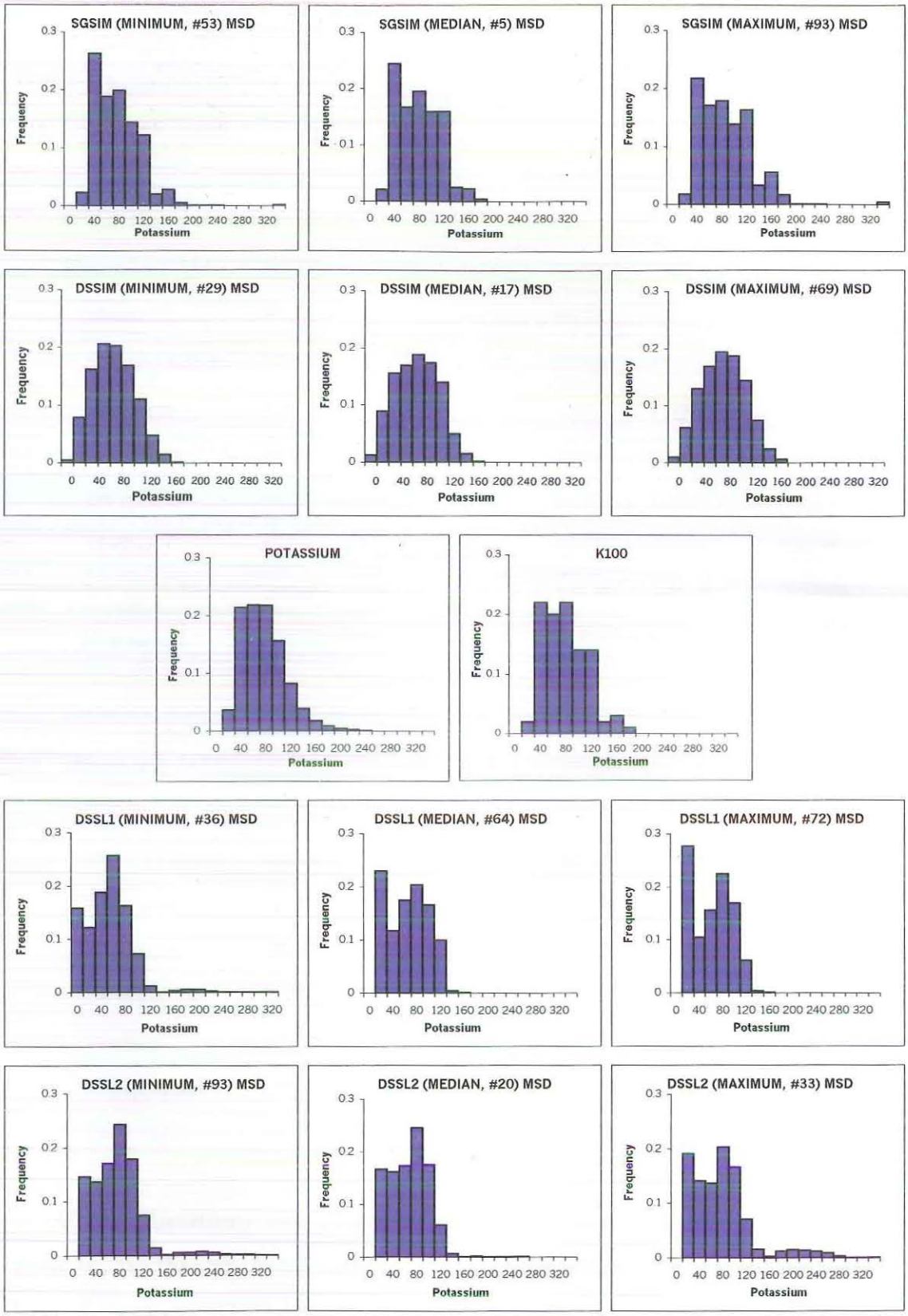

Figure 6.10. Histograms for realisations with minimum, median and maximum mean square deviation.

In Table 6.6 and Table 6.7 we compare the summary statistics for the minimum deviation realisations for the four algorithms using both the mean absolute deviation and the mean squared deviation.

Table 6.6. Comparison of realisations with mininum mean absolute deviation.

Minimum MAD	SGSIM	DSSIM	DSSL1	DSSL2	*Potassium*	*K100*
Mean	87.32	86.08	84.21	85.37	87.1	89.45
Std Deviation	35.05	34.49	44.32	45.43	34.7	34.42
Skewness	1.27	0.29	1.57	1.61	1.04	0.60
Minimum	29.80	20.00	20.23	20.01	23.87	34.6
1st quartile	56.74	59.71	49.23	49.46	59.94	63.88
Median	82.41	84.45	83.12	82.09	82.47	86.85
3rd quartile	107.91	109.78	107.28	106.89	106.87	110.58
Maximum	360.00	194.97	359.42	353.69	331.98	189.0

Table 6.7. Comparison of realisations with minimum mean squared deviation.

Minimum MSD	SGSIM	DSSIM	DSSL1	DSSL2	*Potassium*	*K100*
Mean	86.89	86.56	82.73	85.21	87.1	89.45
Std Deviation	36.56	33.91	38.24	42.13	34.7	34.42
Skewness	1.25	0.28	1.58	1.84	1.04	0.60
Minimum	25.05	20.00	20.23	20.02	23.87	34.6
1st quartile	56.13	60.29	55.93	55.13	59.94	63.88
Median	82.22	84.57	82.30	83.94	82.47	86.85
3rd quartile	108.23	111.34	102.64	104.70	106.87	110.58
Maximum	360.00	208.74	358.53	353.11	331.98	189.0

All the algorithms have generated realisations with positively skewed distributions greater than the target distribution but the DSSIM algorithm has a very small level of skewness. The average and median values are close to the *Potassium* mean and median. The most significant difference is that the maximum value for DSSIM is a lot smaller than the maximum *Potassium* value, but it is vey close to the *K100* value. The DSSL1 and DSSL2 algorithms have higher standard deviations than

both the *Potassium* and *K100* data sets and the minimum and in particular, the lower quartile for these two algorithms is distinctly lower.

In Table 6.8 and Table 6.9 we compare the realisations with the median mean absolute deviation and the median mean squared deviation. All the algorithms have less skewed distributions than the previous realisations with minimum deviations.

Table 6.8. Comparison of realisations with median mean absolute deviation.

Median MAD	SGSIM	DSSIM	DSSL1	DSSL2	*Potassium*	*K100*
Mean	88.69	89.25	76.74	79.39	87.1	89.45
Std Deviation	31.21	32.01	30.35	38.53	34.7	34.42
Skewness	0.36	0.09	0.00	1.40	1.04	0.60
Minimum	29.18	20.00	20.35	20.08	23.87	34.6
1st quartile	65.45	65.98	50.21	45.09	59.94	63.88
Median	88.36	89.61	79.30	79.89	82.47	86.85
3rd quartile	108.86	111.64	99.54	100.86	106.87	110.58
Maximum	257.01	203.63	189.02	352.00	331.98	189.0

Table 6.9. Comparison of realisations with median mean squared deviation.

Median MSD	SGSIM	DSSIM	DSSL1	DSSL2	*Potassium*	*K100*
Mean	88.75	87.75	76.42	77.79	87.1	89.45
Std Deviation	34.18	35.53	32.69	32.21	34.7	34.42
Skewness	0.41	0.13	0.10	0.78	1.04	0.60
Minimum	29.64	20.00	20.01	20.04	23.87	34.6
1st quartile	56.77	59.21	42.10	49.20	59.94	63.88
Median	88.12	87.34	78.02	79.79	82.47	86.85
3rd quartile	115.08	114.94	102.76	100.02	106.87	110.58
Maximum	273.16	204.16	214.25	358.74	331.98	189.0

DSSL2 is the only algorithm to reproduce the maximum value of the *Potassium* data set. SGSIM overestimates the minimum and underestimates the maximum values, although both DSSIM and DSSL1 underestimate the maximum value substantially. Both SGSIM and DSSIM have higher medians than *Potassium*, and DSSL1 and DSSL2 have significantly lower median values.

The realisations with the maximum deviations are compared in Table 6.10 and Table 6.11. The SGSIM and DSSIM mean and median values have increased and the DSSL1 and DSSL2 mean and median values have decreased. The standard deviation for SGSIM is higher than that for *Potassium*. This is also a lot higher for DSSL2 when using the mean squared deviation.

Table 6.10. Comparison of realisations with maximum mean absolute deviation.

Maximum MAD	SGSIM	DSSIM	DSSL1	DSSL2	*Potassium*	*K100*
Mean	95.85	93.68	72.63	72.82	87.1	89.45
Std Deviation	42.74	36.13	30.39	29.21	34.7	34.42
Skewness	1.39	0.10	0.17	0.19	1.04	0.60
Minimum	21.02	20.00	20.07	20.14	23.87	34.6
1st quartile	64.00	66.42	40.56	44.15	59.94	63.88
Median	90.39	93.83	74.77	73.82	82.47	86.85
3rd quartile	122.96	119.95	96.81	95.55	106.87	110.58
Maximum	360.00	197.41	252.29	189.02	331.98	189.0

Table 6.11. Comparison of realisations with maximum mean squared deviation.

Maximum MSD	SGSIM	DSSIM	DSSL1	DSSL2	*Potassium*	*K100*
Mean	95.85	93.68	72.65	88.56	87.1	89.45
Std Deviation	42.74	36.13	32.27	52.15	34.7	34.42
Skewness	1.39	0.10	0.08	1.69	1.04	0.60
Minimum	21.02	20.00	20.13	20.12	23.87	34.6
1st quartile	64.00	66.42	38.15	47.16	59.94	63.88
Median	90.39	93.83	76.39	83.10	82.47	86.85
3rd quartile	122.96	119.95	98.72	107.85	106.87	110.58
Maximum	360.00	197.41	189.02	358.84	331.98	189.0

In Figure 6.11 we compare the cumulative frequency functions for the realisations with minimum, median and maximum mean absolute deviation and mean square deviation.

The distributions appear very similar regardless of which accuracy measure is applied. The spike of low values that result for DSSL1 and DSSL2 is again seen

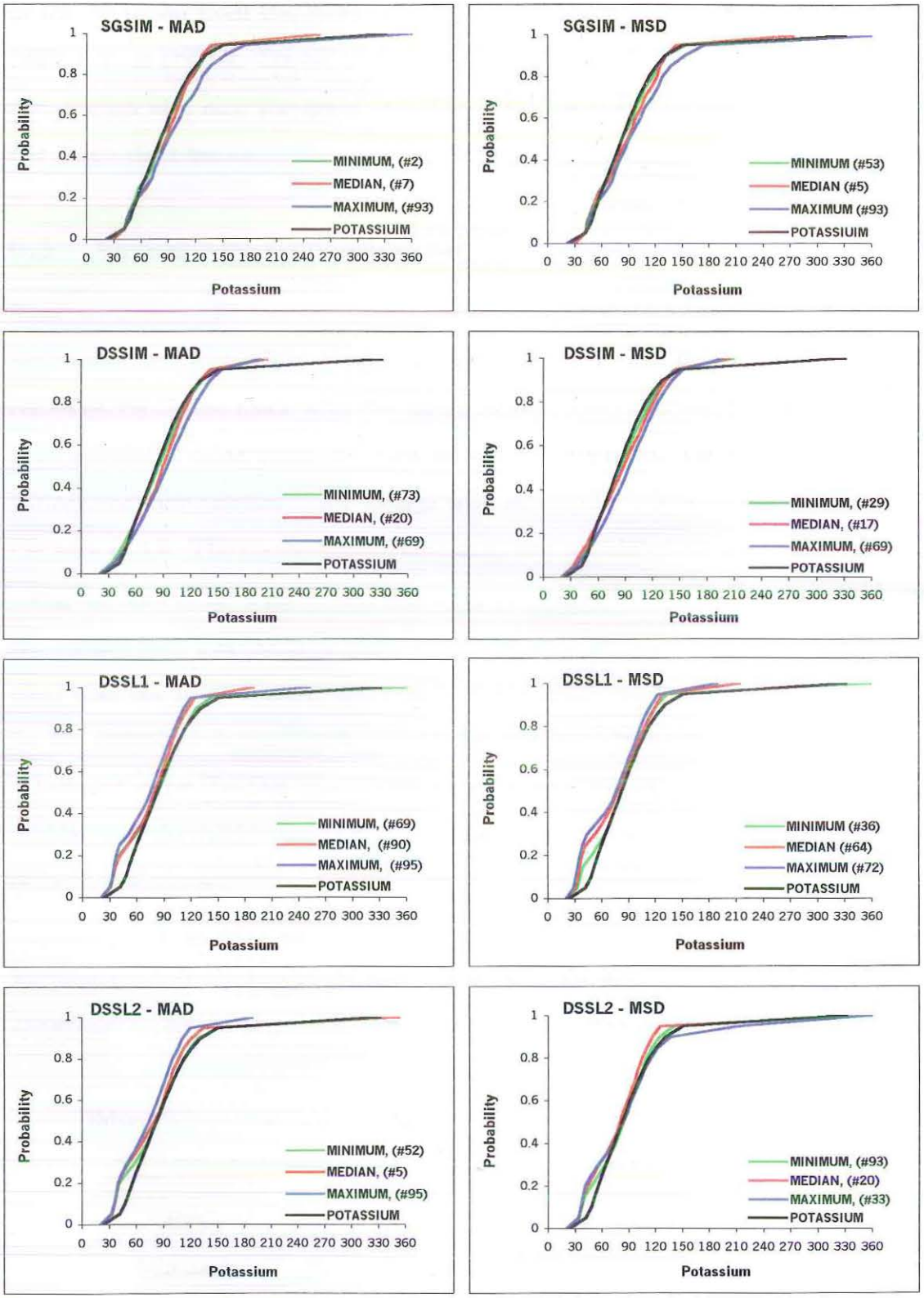

Figure 6.11. Cumulative distribution functions for realisations with minimum, median and maximum mean absolute deviation (left) and mean squared deviation (right).

in the deviation from the lower tail of the target distribution, but it is greater for DSSL1 thtan DSSL2. The SGSIM and DSSIM algorithms both reproduce the target distribution well near the lower tail. The DSSL1 and DSSL2 algorithms reproduce the target distribution better near the median.

6.5 Variogram Reproduction

Since an anisotropic semivariogram model is required, we calculated experimental semivariograms for each realisation in both the N65W and the N25E directions, so comparisons can be made with the target model. The experimental semivariograms were calculated using *gamv.exe* from the GSLIB programs. The experimental semivariograms were calculated for 40 lags with an angular tolerance of 22.5° and a lag distance of 1.0. The mean squared deviation was calculated for each semivariogram using the sum of the squared deviations in both directions as in Equation (86), and the square root was taken to make comparisons. The SGSIM algorithm was modelled with the normal score data before it was back-transformed and this accounts for the difference in magnitude seen in the graphs. The mean squared deviation is calculated using the first fifteen lags in both the direction of maximum continuity, $N65W$, and the $N25E$ direction as this enables us to compare the same realisations in both directions.

Table 6.12 displays the mean squared deviation for the experimental semivariograms using both directions, and it appears that after SGSIM's fit, the DSSIM algorithm results in the closest experimental semivariograms.

Table 6.12. Comparison of mean squared deviations for Potassium - N65W.

Semivariogram	SGSIM		DSSIM		DSSL1		DSSL2	
Rank	#	MSD	#	MSD	#	MSD	#	MSD
Minimum	93	0.02	19	57.8	3	107.6	88	85.4
Median	84	0.14	79	209.7	23	335.7	47	379.9
Maximum	40	0.24	86	352.7	66	1221.6	40	1461.4

In Figure 6.12 we see the experimental semivariograms for the 100 realisations.

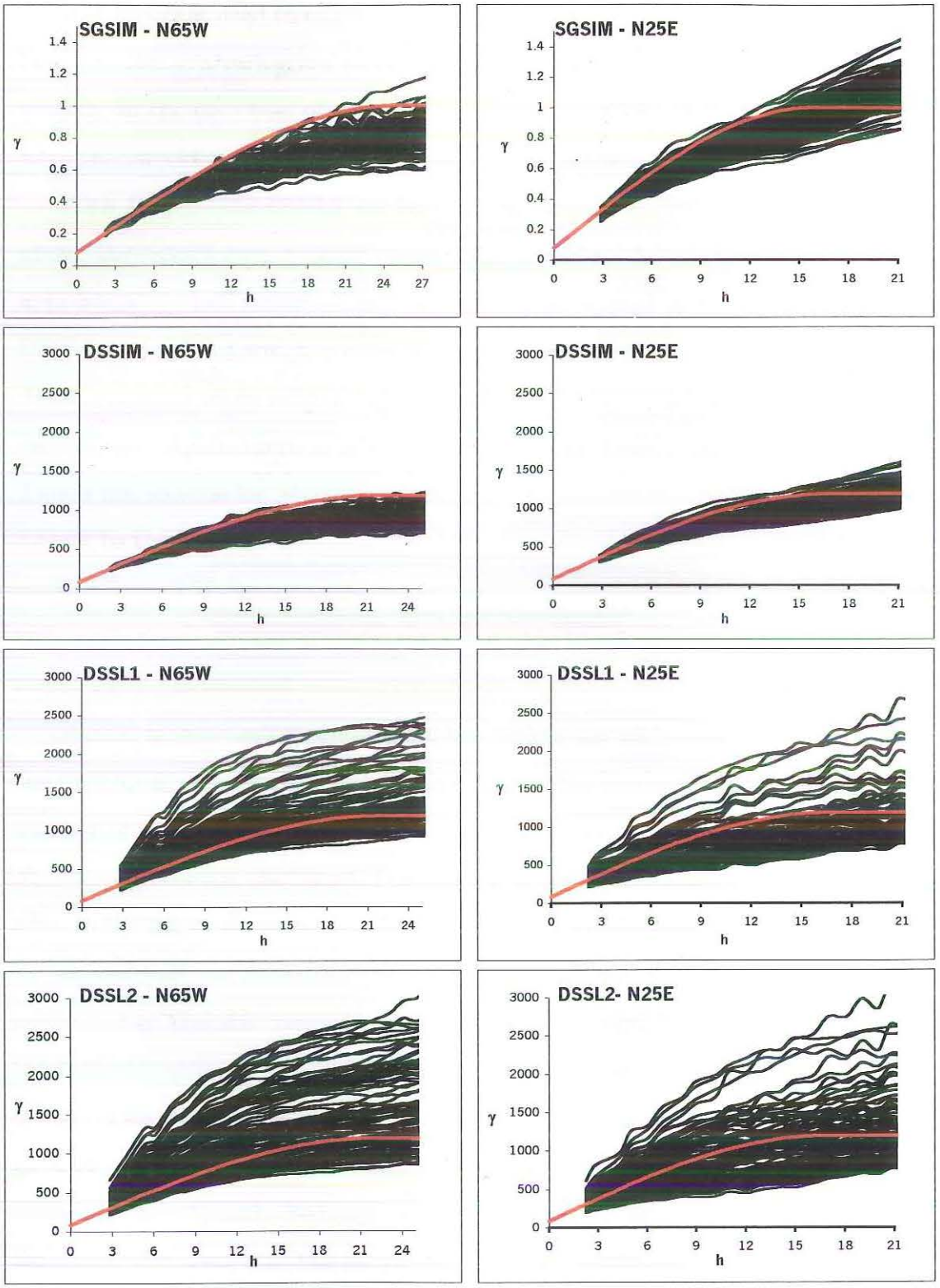

Figure 6.12. Variograms for 100 realisations in direction of maximum continuity (left) and minimum continuity (right).

The program used to calculate the experimental semivariograms did not start at lag 0, as the semivariogram model does, and this accounts for the difference in the graphs. In the direction of maximum continuity, $N65W$, both SGSIM and DSSIM have the majority of semivariograms below the model.

With DSSL1 and DSSL2 we find the majority of experimental semivariograms above the model over a short range, but as the model reaches its sill, there are approximately half below. There is a very large spread in the semivariograms for DSSL1 and DSSL2 which is does not happen with DSSIM. In the minor direction, $N25E$, we find more semivariograms fall below the model than above for all the algorithms. Again there is a wide spread for the DSSL1 and DSSL2 algorithms. This is not so wide for SGSIM and DSSIM. The algorithms appear to have a similar nugget to the model.

In Figure 6.13 we see that in the major direction, DSSIM is reproduced fairly well for a small number of realisations but this is only over a short range and the target sill is not reached.

SGSIM is very well reproduce for the realisation with minimum deviation, but the median is well under the sill, so this reasonable reproduction is only for a small number of semivariograms. DSSL1 and DSSL2 have a greater number of semivariograms reproducing the target, but there is a large percentage that have very high sills. In the minor direction SGSIM is reasonably well reproduced over a short range for almost half the experimental semivariograms, but they do not reproduce the overall range and sill. DSSIM appears to be reasonably reproduced for about half the semivariograms as the minimum and median deviations are fairly well reproduced. This also appears to be the case for DSSL1 and DSSL2 which have very good reproduction with the minimum and median semivariograms, but the maximum has an extremely high sill. DSSL1 and DSSL2 appear to have better fits in the minor direction than the major direction.

6.6 Spatial Uncertainty

The spatial uncertainty of a set of realisations is visualized in the decile maps of Figure 6.14.

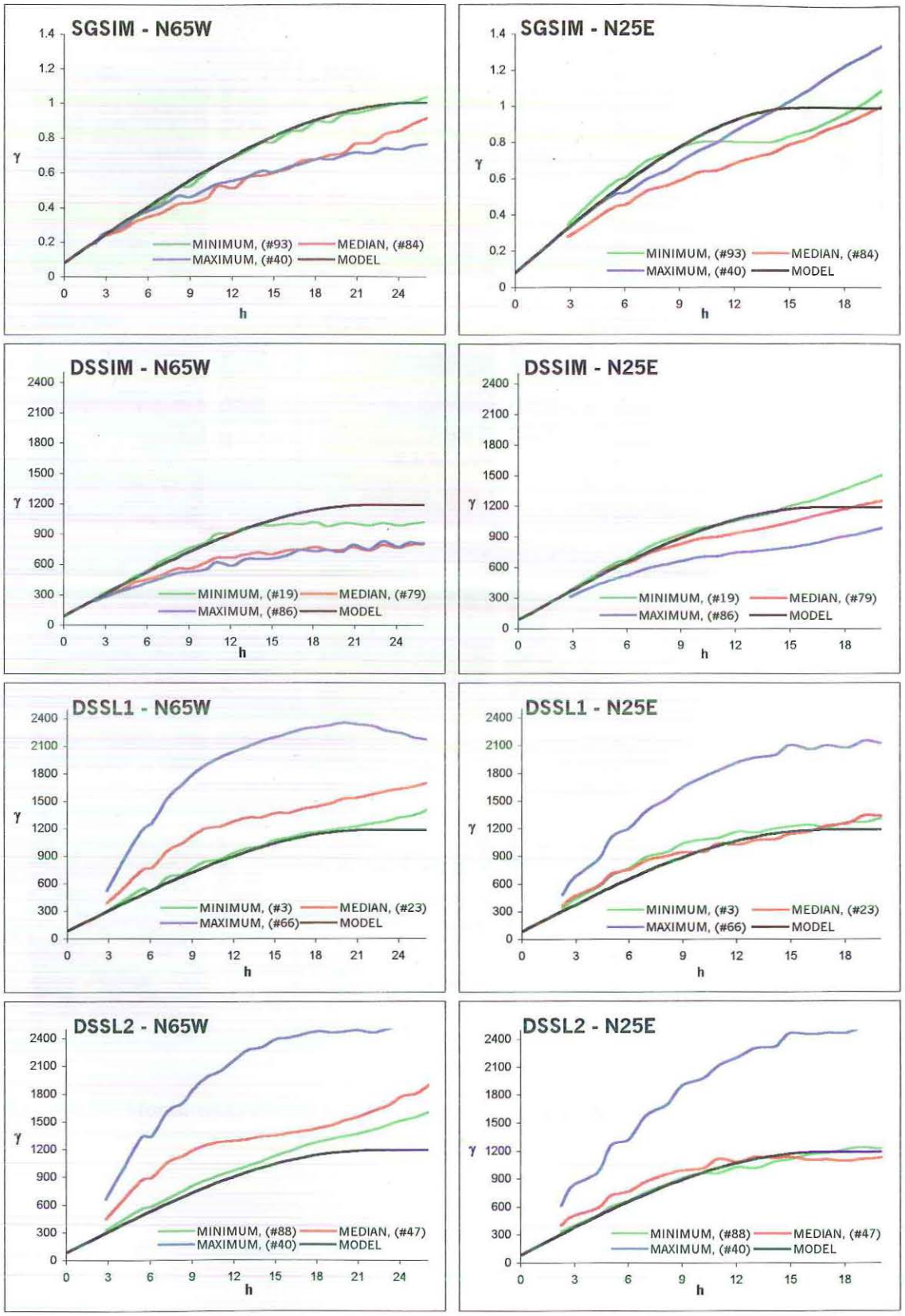

Figure 6.13. Variograms for realisations with minimum, median and maximum mean squared deviation in direction of maximum continuity (left) and minimum continuity (right).

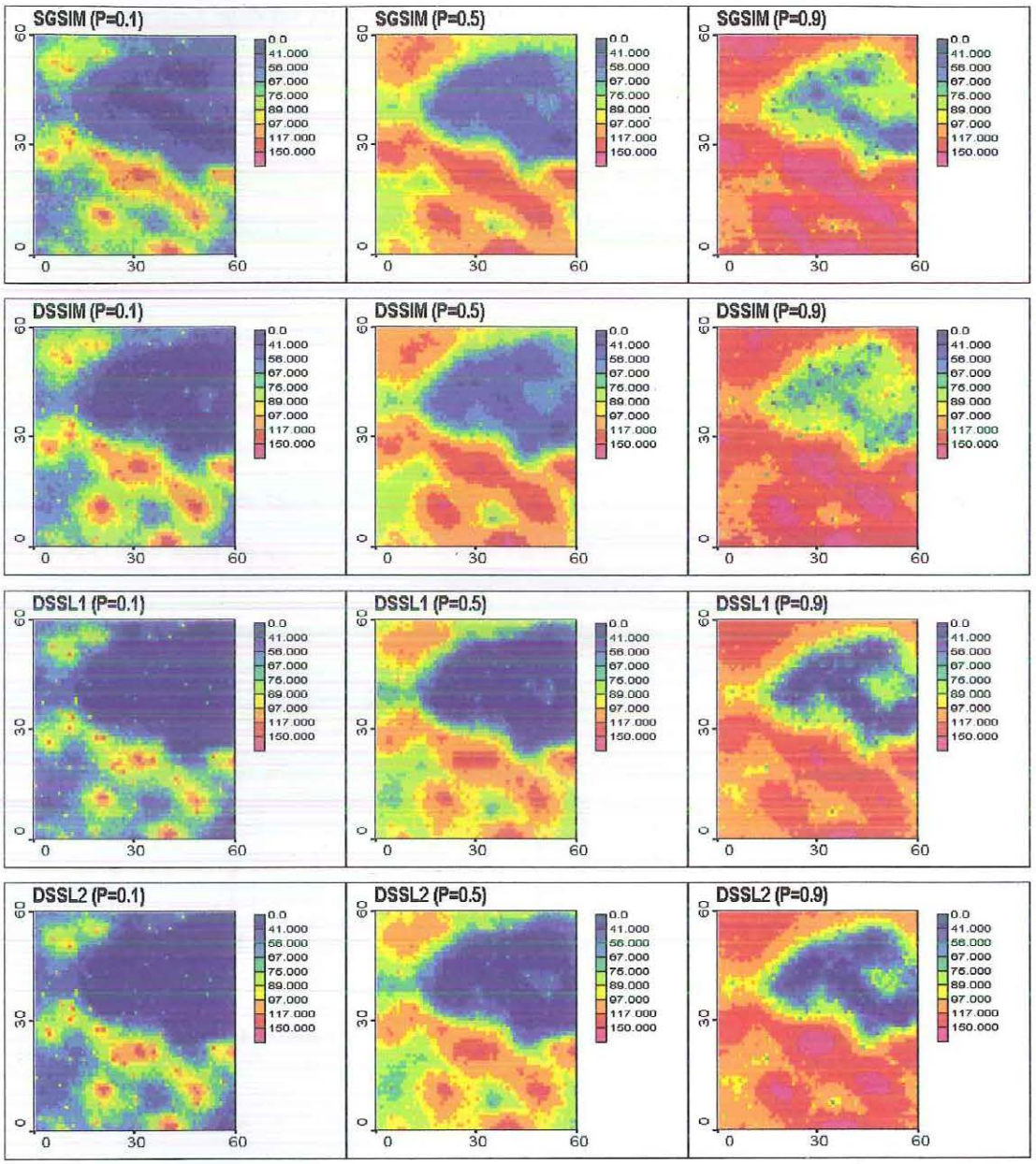

Figure 6.14. Mosaic plots comparing the 0.1-decile, 0.5-decile and 0.9-decile.

The 0.1, 0.5 and 0.9 deciles are displayed for each algorithm. There appears to be a greater spread in the region of low values for SGSIM and DSSIM. All the algorithms have a similar distribution in the southwest. In the southeast ther are more high values for SGSIM and DSSIM. The red areas in the 0.1-quantile plots indicate that there is a high possiblity that the unknown permeability values in this region are high, since these values are exceeded by ninety percent of the simulated values. SGSIM and DSSIM have a similar uncertainty associated with there spatial distributions as do DSSL1 and DSSL2.

Two grid nodes were selected from regions displaying different spatial uncertainty to compare the parametric and non-parametric algorithms. The true potassium values at locations $\mathbf{u}_1 = (28.5, 44.5)$ and $\mathbf{u}_2 = (28.5, 22.5)$ are 57.975 and 202.381 respectively and these are represented by the dashed lines in Figure 6.15.

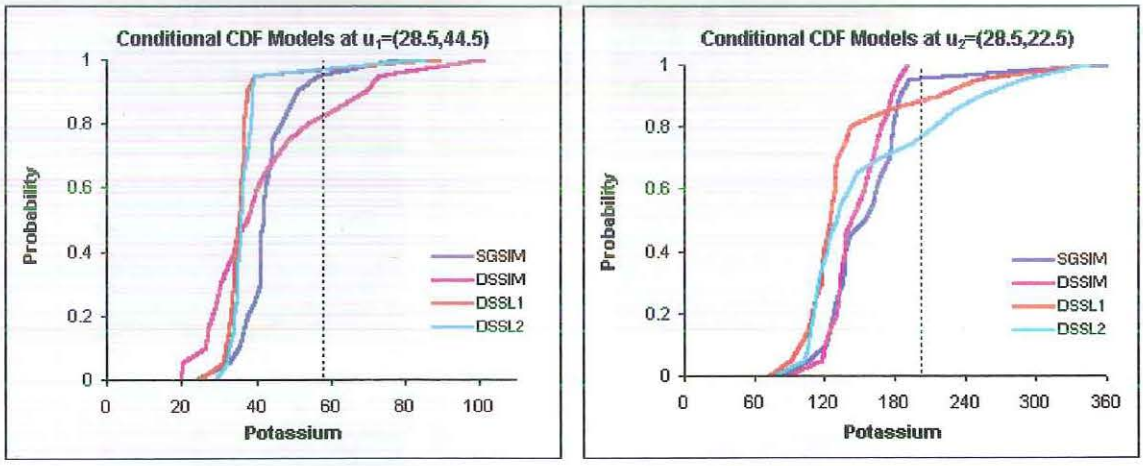

Figure 6.15. Local conditional cumulative distributions.

At location $\mathbf{u}_1$ the cumulative histograms for DSSL1 and DSSL2 are almost identical, and very steep, indicating the value is likely to be less than 40. This is considerably less than the true value of 57.975, but all the algorithms include the true value in their range of potassium values. These differ markedly from the SGSIM and DSSIM cumulative histograms. The DSSIM algorithm has the highest probability of attaining the true value at this location.

At location $\mathbf{u}_1$ the cumulative histograms are different for all algorithms, although SGSIM and DSSIM are almost identical for the lower fifty percent of values.

SGSIM only has a very small probability of attaining the true potassium value, less than five percent. The DSSIM algorithm does not include the true value of 202.381 in its range of potassium values. The greatest probability of the true potassium value being attained, approximately twenty five percent, is with the DSSL2 algorithm.

The conditional variance for each algorithm is displayed through mosaic maps in Figure 6.16. The DSSIM algorithm does not indicate any regions of significantly high uncertainty. The SGSIM, DSSL1 and DSSL2 all indicate some areas where the uncertainty is very high. This appears to be greatest in the band of high values where sample data are sparsely located. The greatest uncertainty within the low valued areas is in the NW corner, but none of the algorithms indicate this to be a significant amount.

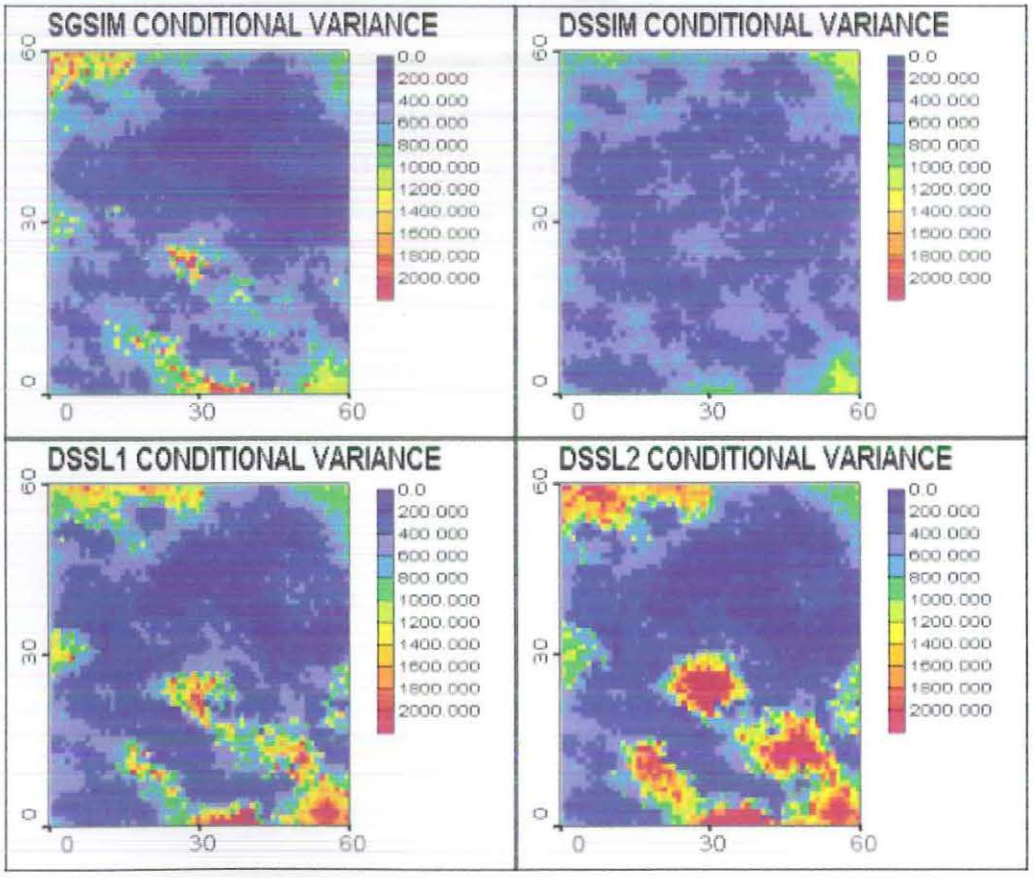

Figure 6.16. Comparison of conditional variance for the algorithms.

6.7 Summary for Potassium

These results for *Potassium* have confirmed what we already knew about the SGSIM algorithm. This algorithm has very good reproduction of the semivariogram model when using normal score data and it reproduces the sample distribution since these were the values used to produce the normal score data. The DSSIM algorithm did not reproduce the target semivariogram very well in either the direction of maximum continuity, N65W, or the minor direction, N25E. DSSIM reproduces the local conditional distribution that was used, in this case a Gaussian distribution. With the exception of the lower tail which is affected by the inability of the algorithm to converge and find a feasible solution, the DSSL1 and DSSL2 both reproduced the global distribution very well. There is very little difference between the results of the DSSL1 and DSSL2 algorithms. The DSSL1 and DSSL2 algorithms both had trouble reproducing the model semivariogram for a large number of realisations. Both algorithms have some very extreme semivariograms, but for those that do show a reasonable reproduction, DSSL2 appears to have a larger proportion of experimental semivariograms closer to the target. For this anisotropic model, the DSSL2 algorithm would therefore appear to provide the best reproduction of both the histogram and the semivariogram.

7 Results and Conclusions

In this study we investigated four sequential simulation algorithms, namely sequential Gaussian simulation (SGSIM), the original direct sequential simulation (DSSIM), and two direct sequential simulation with histogram reproduction algorithms, one using the one-norm (DSSL1) and the other the two-norm (DSSL2). We were interested to see how the algorithms with histogram reproduction compared to each other and to the more traditional SGSIM and DSSIM algorithms.

This study used two data sets displaying different characteristics. The permeability data display a strong positive skewness and are from a 2-D section of a reservoir. The potassium data exhibit a slight positive skewness and the values come from soil samples. Both data sets comprise a sample data set and an exhaustive data set which establishes a target distribution and a reference for comparing the different simulation algorithms.

An important aim of this study was to determine if it is possible for a simulation algorithm to reproduce both the histogram and the experimental semivariogram model without the need for a normal score transformation. Secondly, we wanted to compare two algorithms, one based on the one-norm and the other on the two-norm, to see if the natural measure produced better results. Thirdly, we were interested in the computational effort required for both algorithms.

Each simulation algorithm generated 100 realisations and the simulation results are produced in both tabloid and graphical form. The histograms and experimental semivariograms for each realisation were compared to the exhaustive data sets. Two measures of accuracy were used in analysing the histogram reproduction. This was done because the mean absolute deviation is related to the DSSL1 algorithm and the mean squared deviation is related to the DSSL2 algorithm. By using both measurements in the comparisons, a preference is not being unduly given to a particular algorithm. For the experimental semivariograms, the mean squared deviation from the target semivariogram model was computed.

The realisations were ranked according to their increasing deviation from the target values and the minimum, median and maximum realisations from each algorithm were used in the comparison of results. The multiple realisations generated

help depict the uncertainty about the spatial distribution of the data. The results were analysed graphically through the use of histograms, cumulative distribution functions, mosaic maps and semivariograms and with tables using the accuracy measurements. These results were compared between individual realisations and between the different algorithms.

Histogram reproduction without the need for a transformation into normal space or the multi-Gaussian assumptions was found to be possible using either the DSSL1 or DSSL2 algorithms, although we will discuss later in this chapter some problems that were encountered with these two simulation algorithms. The DSSIM algorithm reproduced the local conditional distribution and not the target distribution, and SGSIM reproduced the sample distribution and not the target distribution. These two algorithms would require a transformation in order to reproduce the required target distribution.

Semivariogram reproduction was best when SGSIM was used, but this was in normal score space. The normal score data must be back-transformed and the excellent variogram reproduction may then destroyed (Caers, 2000b). DSSIM was unable to reproduce the model experimental semivariogram. The experimental semivariograms calculated for the DSSIM realisations were not at all representative of the experimental semivariogram model. Both the DSSL1 and DSSL2 algorithms were able to reproduce the experimental semivariogram model. This was better with the omnidirectional Permeability data set than with the anisotropic Potassium data set. The results for the DSSL1 and DSSL2 algorithms are very similar.

Mosaic maps displaying the 0.1, 0.5 and 0.9 deciles were generated for each simulation algorithm, along with conditional variance maps. These display the uncertainty in a distribution and may influence the choice of algorithm used. These characteristics are also very similar with the DSSL1 and DSSL2 algorithms, but the DSSL2 has a slightly higher variation in its higher valued areas than DSSL1.

The DSSL1 and DSSL2 algorithms both encountered situations when a feasible solution could not be found to the objective function and a decision was made to set the random deviate equal to the simple kriging mean when this occurred. Figure 7.1 compares the number of infeasible solutions encountered by both algorithms for the data sets.

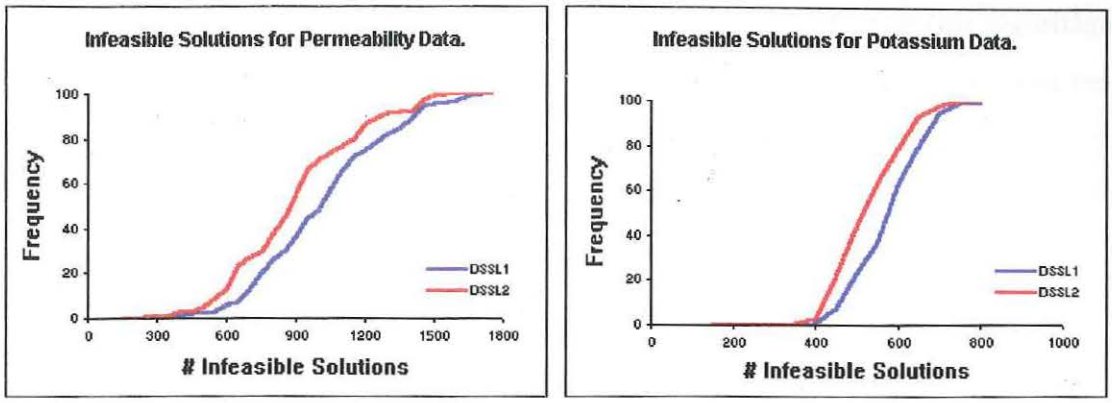

Figure 7.1. Comparison of infeasible solutions encountered for *Permeability* (left) and *Potassium* (right).

In Table 7.1 we see that the average percentage of solutions that did not converge was significantly high, especially for the potassium data set. In both cases DSSL2 had fewer infeasible solutions occurring but the number is still very high. This appears to have had an impact on the number of low values within a realisation. The number of very low values may be artificially high and improving this situation should result in better histogram and variogram reproduction.

Table 7.1. Perecentage of infeasible solutions encountered by DSSL1 and DSSL2.

Percentage of Infeasible Solutions	DSSL1	DSSL2
Permeability	10.3	9.1
Potassium	17.7	16.3

Further research is needed to find a way to overcome this problem. Adding local accuracy to the simulation (Caers, 2000a) by adding a spike in the local conditional distribution increases the connectivity of extreme values and may be a way to improve the histogram reproduction and reduce the number of infeasible solutions encountered by the algorithm. It is likely this will affect the variogram reproduction and a compromise may have to be found.

The global distribution used in this analysis was taken to be the exhaustive data set. When the sample data are not truly representative of the distribution, it may help to use a smoothed distribution based on the exhaustive data set to create the threshold values required for the DSSL1 and DSSL2 algorithms.

Another factor to take into account when deciding which of the two algorithms is better, is the time taken to execute the simulations. This time factor was very noticeable in our study. The time will vary depending on the specifications of the computer system, but we found on average that the DSSL2 algorithm takes approximately 3.5 times as long as the DSSL1 algorithm for the Permeability data set and almost 6 times as long for the Potassium data set, as seen in Table 7.2.

Table 7.2. Computation time for DSSL1 and DSSL2.

Computation Time per Simulation	DSSL1	DSSL2
Permeability	24	82
Potassium	6	35

We believe that improvements in the current programming structure will decrease this slightly, but the DSSL2 will still take longer than the DSSL1 algorithm. These simulation methods are both considerably slower than sequential Gaussian simulation and direct sequential simulation which give almost instantaneous results, so when one of those methods is suitable to the analysis required, it is recommended that it be used.

In summary, this analysis has shown that it is possible to reproduce both the histogram (cumulative distribution function) and the semivariogram using direct sequential simulation and without the need for a transformation into normal space or the assumption of normality. This was found to be possible using a choice of objective functions based on either the one-norm or the two-norm. The decision as to which algorithm to use may be influenced by the computational time involved and the characteristics of the data. The DSSL1 algorithm provided the better results for the isotropic data set and the DSSL2 algorithm performed better with the anisotropic data set.

8 References

Armstrong, M. (1998). *Basic Linear Geostatistics.* Berlin: Springer-Verlag.

Barrodale, I. & Roberts, F.D.K. (1978). An efficient algorithm for discrete ℓ_1 linear approximation with linear constraints. *SIAM J. Num. Anal., 15*(3), 603-611.

Barrodale, I. & Roberts, F.D.K. (1980). Algorithm 552. Solution of the constrained ℓ_1 linear approximation problem. *ACM Transactions on Mathematical Software, 6*(2), 231-235.

Bloom, L.M. & Kentwell, D.J. (1999). A geostatistical analysis of cropped and uncropped soil in the Jimperding Brook catchment area of Western Australia. In J. Gomez-Hernandez et al (Eds.), *geoENVII - Geostatistics for environmental applications* (pp. 369-380). Dordrecht: Kluwer Academic Publishers.

Caers, J. (2000a). Adding Local Accuracy to Direct Sequential Simulation. *Mathematical Geology, 32*(7), 815-850.

Caers, J. (2000b). Direct sequential indicator simulation. In Kleingeld, W. & Krige, D. (Eds.), *Geostats 2000: Cape Town, Proceedings of the 6th International Geostatistics Congress* (pp. 39-48), Cape Town, South Africa, April 10-14, 2000.

Chiles, J-P. & Delfiner, P. (1999). *Geostatistics: Modelling Spatial Uncertainty.* New York: Wiley-Interscience.

Deutsch, C.V. & Journel, A.G. (1998). *GSLIB: Geostatistical Software Library and User's Guide. Second Edition.* New York: Oxford University Press.

Gill, P.E., Murray, W. & Wright, M.H. (1984). *Practical Optimisation.* London: Academic Press.

Gill, P.E., Hammarling, S.J., Murray, W., Saunders, M.A. and Wright, M.H. (1986). *User's guide for LSSOL (Version 1.0): a Fortran package for constrained linear least-squares and convex quadratic programming*, Report SOL 84-6, Department of Operations Research, Stanford University, California.

Goovaerts, P. (1997). *Geostatistics for Natural Resources Evaluation.* New York: Oxford University Press.

Goovaerts, P. (2001). *Geostatistical assessment of the spaces of local, spatial and response uncertainty for continuous petrophysical properties.* Unpublished paper, University of Michigan, Ann Arbor.

Hillier, F.S. & Lieberman, G.J. (1995). *Introduction to Operations Research.* (6^{th} ed.). New York: McGraw-Hill, Inc.

Journel, A.G. (1994). Modelling Uncertainty: Some conceptual thoughts. In R. Dimitrakopoulos (Ed.), *Geostatistics for the Next Century*, (pp. 30-43). Dordrecht: Kluwer Academic.

Journel, A.G. & Huijbregts, C.J. (1978). *Mining Geostatistics.* New York: Academic Press.

Luenberger, D.G. (1989). *Linear and Nonlinear Programming.* (2^{nd} ed.) Menlo Park, California: Addison-Wesley.

Remy, N., Shtuka, A., Levy, B. & Caers, J. (2001). *GsTL: the Geostatistical Template Library in C++.* Unpublished paper, Stanford University, California.

Serway, R.A. & Beichner, R.J. (2000). *Physics for Scientists and Engineers.* (5^{th} ed.). Fort Worth: Saunders College Publishing.

Wackernagel, H. (1995). *Multivariate Geostatistics.* Berlin: Springer.

Walpole, R.E. & Myers, R.H. (1993). *Probability and Statistics for Engineers and Scientists.* (5^{th} ed.). New York: Macmillan.

Wismer, D.A. & Chattergy, R. (1978). *Introduction to Nonlinear Optimization. A Problem Solving Approach.* New York: Elsevier North-Holland, Inc.

Xu, W. & Journel, A.G. (1994). DSSIM: A general sequential simulation algorithm. In *Report 7, Stanford Center for Reservoir Forecasting*, Stanford, CA.

9 Appendices

APPENDIX A.

Parameter files for Isotropic Case

A1 sgsim.par

```
                  Parameters for SGSIM
                  ********************

START OF PARAMETERS:
perm50.dat                        -file with data
1  2  0  3  0  0                  -  columns for X,Y,Z,vr,wt,sec.var.
-1.0       1.0e21                 -  trimming limits
1                                 -transform the data (0=no, 1=yes)
sgsim.trn                         -  file for output trans table
0                                 -  consider ref. dist (0=no, 1=yes)
ref.dat                           -  file with ref. dist distribution
3  0                              -  columns for vr and wt
1.0    2500.0                     -  zmin,zmax(tail extrapolation)
2       2.5                       -  lower tail option, parameter
4       2.0                       -  upper tail option, parameter
0                                 -debugging level: 0,1,2,3
sgsim.dbg                         -file for debugging output
sgsim.dat                         -file for simulation output
100                               -number of realizations to generate
102    1.0    1.0                 -nx,xmn,xsiz
102    1.0    1.0                 -ny,ymn,ysiz
1      0.0    1.0                 -nz,zmn,zsiz
69069                             -random number seed
4      20                         -min and max original data for sim
12                                -number of simulated nodes to use
1                                 -assign data to nodes (0=no, 1=yes)
1      3                          -multiple grid search (0=no, 1=yes),num
0                                 -maximum data per octant (0=not used)
60.0  60.0  1.0                   -maximum search radii (hmax,hmin,vert)
 0.0   0.0   0.0                  -angles for search ellipsoid
0      0.0   1.0                  -ktype: 0=SK,1=OK,2=LVM,3=EXDR,4=COLC
no.dat                            -  file with LVM, EXDR, or COLC variable
4                                 -  column for secondary variable
1    0.14                         -nst, nugget effect
1    0.86  90.0   0.0   0.0       -it,cc,ang1,ang2,ang3
           24.0  24.0  0.0        -a_hmax, a_hmin, a_vert
```

A2 dssim.par

```
                  Parameters for DSSIM
                  ********************

START OF PARAMETERS:
perm50.dat                          -file with data
1  2  0  3  0  0                    -columns for X,Y,Z,vr,wt,sec.var.
-1.0       1.0e21                   -trimming limits
0                                   -transform the data (0=no, 1=yes)
dssim.trn                           -file for output trans table
0                                   -consider ref. dist (0=no, 1=yes)
histsmth.out                        -file with ref. dist distribution
1  2                                -columns for vr and wt
1.0    2500.0                       -zmin,zmax(tail extrapolation)
1        1.0                        -lower tail option, parameter
4        2.0                        -upper tail option, parameter
0                                   -debugging level: 0,1,2,3
dssim.dbg                           -file for debugging output
dssim.dat                           -file for simulation output
100                                 -number of realizations to generate
102     1.0     1.0                 -nx,xmn,xsiz
102     1.0     1.0                 -ny,ymn,ysiz
1      0.0     1.0                  -nz,zmn,zsiz
69069                               -random number seed
4      20                           -min and max original data for sim
12                                  -number of simulated nodes to use
1                                   -assign data to nodes (0=no, 1=yes)
1      3                            -multiple grid search (0=no, 1=yes),num
0                                   -maximum data per octant (0=not used)
60.0   60.0   0.0                  -maximum search radii (hmax,hmin,vert)
 0.0    0.0    0.0                 -angles for search ellipsoid
0      0.60    1.0                  -ktype: 0=SK,1=OK,2=LVM,3=EXDR,4=COLC
592.53      582.95                  -global mean, standard deviation
2                                   - local dist:1-normal,2-lognormal
nodata.dat                          -  file with LVM, EXDR, or COLC variable
4                                   -  column for secondary variable
2    43680                          -nst, nugget effect
1    63000   90.0    0.0    0.0    -it,cc,ang1,ang2,ang3
         10.0  10.0    0.0         -a_hmax, a_hmin, a_vert
1    124000  90.0    0.0    0.0    -it,cc,ang1,ang2,ang3
         50.0  50.0    0.0         -a_hmax, a_hmin, a_vert
```

A3 dssl1.par

```
                  Parameters for DSSL1
                  ********************

START OF PARAMETERS:
perm50.dat                          -file with data
1  2  0  3  0  0                    -columns for X,Y,Z,vr,wt,sec.var.
-1.0        1.0e21                  -trimming limits
0                                   -transform the data (0=no, 1=yes)
dssL1.trn                           -file for output trans table
0                                   -consider ref. dist (0=no, 1=yes)
histsmth.out                        -file with ref. dist distribution
1  2                                -columns for vr and wt
p40.dat                             -cdf file
dssl1.inf                           -info file
2   1                               -select(1-DSSIM,2-DSSL1)opt(1-skmean,
                                       2-logsim,3-normal)
1.0     2500.0                      -zmin,zmax(tail extrapolation)
1        1.0                        -lower tail option, parameter
1       2500.0                      -upper tail option, parameter
1                                   -debugging level: 0,1,2,3
dssL1.dbg                           -file for debugging output
dssL1.dat                           -file for simulation output
100                                 -number of realizations to generate
102     1.0     1.0                 -nx,xmn,xsiz
102     1.0     1.0                 -ny,ymn,ysiz
1      0.0     1.0                  -nz,zmn,zsiz
69069                               -random number seed
4      20                           -min and max original data for sim
12                                  -number of simulated nodes to use
1                                   -assign data to nodes (0=no, 1=yes)
1      3                            -multiple grid search (0=no, 1=yes),num
0                                   -maximum data per octant (0=not used)
60.0  60.0  1.0                     -maximum search radii (hmax,hmin,vert)
 0.0   0.0   0.0                    -angles for search ellipsoid
0      0.60   1.0                   -ktype: 0=SK,1=OK,2=LVM,3=EXDR,4=COLC
592.53      582.95                  -global mean, standard deviation
nodata.dat                          -file with LVM, EXDR, or COLC variable
4                                   -column for secondary variable
2   43680                           -nst, nugget effect
1   63000  90.0   0.0   0.0        -it,cc,ang1,ang2,ang3
         10.0  10.0   0.0          -a_hmax, a_hmin, a_vert
1   124000  90.0   0.0   0.0       -it,cc,ang1,ang2,ang3
         50.0  50.0   0.0          -a_hmax, a_hmin, a_vert
```

A4 dssl2.par

```
                  Parameters for DSSL2
                  ********************

START OF PARAMETERS:
perm50.dat                        -file with data
1  2  0  3  0  0                  -  columns for X,Y,Z,vr,wt,sec.var.
-1.0        1.0e21                -  trimming limits
0                                 -transform the data (0=no, 1=yes)
dssL2.trn                         -file for output trans table
0                                 -consider ref. dist (0=no, 1=yes)
histsmth.out                      -file with ref. dist distribution
1  2                              -columns for vr and wt
p40.dat                           -cdf file
dssL2.inf                         -output info file
2 10                              -select 1-DSSIM,2-DSSL2 opt: 1-QP2,2-LS1
1.0     2500.0                    -zmin,zmax(tail extrapolation)
1        1.0                      -lower tail option, parameter
1       2500.0                    -upper tail option, parameter
1                                 -debugging level: 0,1,2,3
dssL2.dbg                         -file for debugging output
dssL2.dat                         -file for simulation output
100                               -number of realizations to generate
102     1.0     1.0               -nx,xmn,xsiz
102     1.0     1.0               -ny,ymn,ysiz
1       0.0     1.0               -nz,zmn,zsiz
69069                             -random number seed
4       20                        -min and max original data for sim
12                                -number of simulated nodes to use
1                                 -assign data to nodes (0=no, 1=yes)
1       3                         -multiple grid search (0=no, 1=yes),num
0                                 -maximum data per octant (0=not used)
60.0  60.0  1.0                   -maximum search radii (hmax,hmin,vert)
 0.0   0.0   0.0                  -angles for search ellipsoid
0       0.60    1.0               -ktype: 0=SK,1=OK,2=LVM,3=EXDR,4=COLC
592.53      582.95                -global mean, standard deviation
nodata.dat                        -file with LVM, EXDR, or COLC variable
4                                 -column for secondary variable
2   43680                         -nst, nugget effect
1   63000  90.0    0.0    0.0    -it,cc,ang1,ang2,ang3
         10.0  10.0    0.0       -a_hmax, a_hmin, a_vert
1   124000  90.0    0.0    0.0   -it,cc,ang1,ang2,ang3
          50.0  50.0    0.0      -a_hmax, a_hmin, a_vert
```

A5 p40.dat

```
p40
3       40
i
k
prob
1       17.5869775  0.025
2       43.547355   0.025
3       80.4025825  0.025
4       117.25781   0.025
5       146.372725  0.025
6       163.18037   0.025
7       179.987915  0.025
8       191.95234   0.025
9       193.0798025 0.025
10      194.207325  0.025
11      195.5754475 0.025
12      197.74167   0.025
13      199.9079925 0.025
14      205.601605  0.025
15      230.6076125 0.025
16      255.61362   0.025
17      283.75861   0.025
18      349.414575  0.025
19      415.07057   0.025
20      480.6922    0.025
21      510.626525  0.025
22      540.52645   0.025
23      570.426375  0.025
24      595.14352   0.025
25      619.426925  0.025
26      643.71033   0.025
27      706.7461425 0.025
28      776.85997   0.025
29      846.973945  0.025
30      869.159725  0.025
31      876.8952525 0.025
32      884.63078   0.025
33      974.5320775 0.025
34      1101.154505 0.025
35      1227.776875 0.025
36      1340.76718  0.025
37      1445.18373  0.025
38      1549.600405 0.025
39      1912.615898 0.025
40      2498.8733   0.025
```

APPENDIX B.

Parameter files for Anisotropic Case

B1 sgsim.par

```
                  Parameters for SGSIM
                  ********************

START OF PARAMETERS:
k100.dat                          -file with data
1  2  0  3  0  0                  -  columns for X,Y,Z,vr,wt,sec.var.
-1.0       1.0e21                 -  trimming limits
1                                 -transform the data (0=no, 1=yes)
sgsim.trn                         -file for output trans table
0                                 -consider ref. dist (0=no, 1=yes)
histsmth.out                      -file with ref. dist distribution
1  2                              -columns for vr and wt
20.0     360.0                    -zmin,zmax(tail extrapolation)
1        20.0                     -lower tail option, parameter
4        1.5                      -upper tail option, parameter
1                                 -debugging level: 0,1,2,3
sgsim.dbg                         -file for debugging output
sgsim.dat                         -file for simulation output
100                               -number of realizations to generate
60     0.5     1.0                -nx,xmn,xsiz
60     0.5     1.0                -ny,ymn,ysiz
1      0.0     1.0                -nz,zmn,zsiz
69069                             -random number seed
4      20                         -min and max original data for sim
12                                -number of simulated nodes to use
1                                 -assign data to nodes (0=no, 1=yes)
1      3                          -multiple grid search (0=no, 1=yes),num
0                                 -maximum data per octant (0=not used)
40.0   40.0   1.0                -maximum search radii (hmax,hmin,vert)
 0.0    0.0    0.0                -angles for search ellipsoid
0      0.0    1.0                -ktype: 0=SK,1=OK,2=LVM,3=EXDR,4=COLC
87.097      34.703                -global mean, standard deviation
nodata.dat                        -file with LVM, EXDR, or COLC variable
4                                 -column for secondary variable
1    0.08                         -nst, nugget effect
1    0.92  115.0    0.0    0.0    -it,cc,ang1,ang2,ang3
         25.0   16.0   0.0        -a_hmax, a_hmin, a_vert
```

B2 dssim.par

```
                  Parameters for DSSIM
                  ********************

START OF PARAMETERS:
k100.dat                          -file with data
1  2  0  3  0  0                  -  columns for X,Y,Z,vr,wt,sec.var.
-1.0       1.0e21                 -  trimming limits
0                                 -transform the data (0=no, 1=yes)
direct_k.trn                      -  file for output trans table
0                                 -  consider ref. dist (0=no, 1=yes)
histsmth.out                      -  file with ref. dist distribution
1  2                              -  columns for vr and wt
20.0   360.0                      -  zmin,zmax(tail extrapolation)
1       20.0                      -  lower tail option, parameter
4       1.5                       -  upper tail option, parameter
0                                 -debugging level: 0,1,2,3
direct_k.dbg                      -file for debugging output
direct_k.dat                      -file for simulation output
100                               -number of realizations to generate
60    0.5    1.0                  -nx,xmn,xsiz
60    0.5    1.0                  -ny,ymn,ysiz
1     0.0    1.0                  -nz,zmn,zsiz
69069                             -random number seed
4     20                          -min and max original data for sim
12                                -number of simulated nodes to use
1                                 -assign data to nodes (0=no, 1=yes)
1     3                           -multiple grid search (0=no, 1=yes),num
0                                 -maximum data per octant (0=not used)
40.0  40.0  0.0                   -maximum search radii (hmax,hmin,vert)
 0.0   0.0   0.0                  -angles for search ellipsoid
0     0.60   1.0                  -ktype: 0=SK,1=OK,2=LVM,3=EXDR,4=COLC
89.45     34.42                   -global mean, standard deviation
nodata.dat                        -  file with LVM, EXDR, or COLC variable
4                                 -  column for secondary variable
1    84                           -nst, nugget effect
1    1104  115.0   0.0   0.0     -it,cc,ang1,ang2,ang3
         22.0  16.72  0.0        -a_hmax, a_hmin, a_vert
```

B3 dssl1.par

```
                  Parameters for DSSL1
                  ********************

START OF PARAMETERS:
k100.dat                          -file with data
1  2  0  3  0  0                  -columns for X,Y,Z,vr,wt,sec.var.
-1.0       1.0e21                 -trimming limits
0                                 -transform the data (0=no, 1=yes)
dssim.trn                         -file for output trans table
0                                 -consider ref. dist (0=no, 1=yes)
histsmth.out                      -file with ref. dist distribution
1  2                              -columns for vr and wt
k40.dat                           -cdf file
dssim.inf                         -info file
2  1                              -select(1-DSSIM,2-DSSL1)opt(1-skmean,
                                      2-logsim,3-normal)
20.0     360.0                    -zmin,zmax(tail extrapolation)
1        20.0                     -lower tail option, parameter
1       360.0                     -upper tail option, parameter
1                                 -debugging level: 0,1,2,3
dssim.dbg                         -file for debugging output
dssim.dat                         -file for simulation output
100                               -number of realizations to generate
60     0.5     1.0                -nx,xmn,xsiz
60     0.5     1.0                -ny,ymn,ysiz
1      0.0     1.0                -nz,zmn,zsiz
69069                             -random number seed
4      20                         -min and max original data for sim
12                                -number of simulated nodes to use
1                                 -assign data to nodes (0=no, 1=yes)
1      3                          -multiple grid search (0=no, 1=yes),num
0                                 -maximum data per octant (0=not used)
40.0   40.0   1.0                -maximum search radii (hmax,hmin,vert)
 0.0    0.0    0.0                -angles for search ellipsoid
0      0.60    1.0                -ktype: 0=SK,1=OK,2=LVM,3=EXDR,4=COLC
89.45      34.42                  -global mean, standard deviation
nodata.dat                        -  file with LVM, EXDR, or COLC variable
4                                 -  column for secondary variable
1   84                            -nst, nugget effect
1   1104  115.0    0.0    0.0    -it,cc,ang1,ang2,ang3
        22.0  16.72  0.0          -a_hmax, a_hmin, a_vert
```

B4 dssl2.par

```
                  Parameters for DSSL2
                  ********************

START OF PARAMETERS:
k100.dat                          -file with data
1  2  0  3  0  0                  -  columns for X,Y,Z,vr,wt,sec.var.
-1.0       1.0e21                 -  trimming limits
0                                 -transform the data (0=no, 1=yes)
dssL2.trn                         -file for output trans table
0                                 -consider ref. dist (0=no, 1=yes)
histsmth.out                      -file with ref. dist distribution
1  2                              -columns for vr and wt
k40.dat                           -cdf file
dssL2.inf                         -output info file
2 10                              -select 1-dssim,2-dssl2 opt:10-QP2,20-LS1
20.0     360.0                    -zmin,zmax(tail extrapolation)
1        20.0                     -lower tail option, parameter
1        360.0                    -upper tail option, parameter
1                                 -debugging level: 0,1,2,3
dssL2.dbg                         -file for debugging output
dssL2.dat                         -file for simulation output
100                               -number of realizations to generate
60     0.5     1.0                -nx,xmn,xsiz
60     0.5     1.0                -ny,ymn,ysiz
1      0.0     1.0                -nz,zmn,zsiz
69069                             -random number seed
4      20                         -min and max original data for sim
12                                -number of simulated nodes to use
1                                 -assign data to nodes (0=no, 1=yes)
1      3                          -multiple grid search (0=no, 1=yes),num
0                                 -maximum data per octant (0=not used)
40.0  40.0  1.0                   -maximum search radii (hmax,hmin,vert)
 0.0   0.0   0.0                  -angles for search ellipsoid
0      0.60   1.0                 -ktype: 0=SK,1=OK,2=LVM,3=EXDR,4=COLC
89.45      34.42                  -global mean, standard deviation
nodata.dat                        -file with LVM, EXDR, or COLC variable
4                                 -column for secondary variable
1   84                            -nst, nugget effect
1   1104  115.0    0.0    0.0    -it,cc,ang1,ang2,ang3
        22.0  16.72  0.0          -a_hmax, a_hmin, a_vert
```

B5 k40.dat

```
k40
3       40
i
k
prob
1       38.15775        0.025
2       41.8968         0.025
3       45.1628         0.025
4       48              0.025
5       49.57725        0.025
6       51.78715        0.025
7       53.9086         0.025
8       55.872          0.025
9       57.79575        0.025
10      59.85275        0.025
11      62.525          0.025
12      64.7698         0.025
13      67.0457         0.025
14      69.2315         0.025
15      71.778          0.025
16      73.6688         0.025
17      76.21795        0.025
18      78.12165        0.025
19      80.387          0.025
20      82.326          0.025
21      84.85635        0.025
22      87.22815        0.025
23      89.323          0.025
24      91.367          0.025
25      93.94825        0.025
26      96.38045        0.025
27      98.5661         0.025
28      101.1474        0.025
29      104.115         0.025
30      106.8665        0.025
31      109.6984        0.025
32      112.9?8         0.025
33      117.162         0.025
34      121.31225       0.025
35      125.7195        0.025
36      130.7551        0.025
37      139.59615       0.025
38      150.5525        0.025
39      168.418         0.025
40      331.978         0.025
```