# VISIM : Sequential simulation for linear inverse problems.*

# *Code is available from server at http://www.iamg.org/CGEditor/index.htm

Thomas Mejer Hansen and Klaus Mosegaard

*Niels Bohr Institute, Juliane Maries Vej 28, 2100 København Ø, Denmark*

**Abstract**

Linear inverse Gaussian problems is traditionally solved using least squares based inversion. The center of the posterior Gaussian probability distribution is often chosen as the solution to such problems, while the solution is in fact the posterior Gaussian probability distribution itself. We present an algorithm, based on direct sequential simulation, which can be used to efficiently draw samples of the posterior probability distribution for linear inverse problems. There is no Gaussian restriction on the distribution in the model parameter space, as inherent in traditional least squares based algorithms.

As data for linear inverse problems can be seen as weighed linear averages over some volume, block kriging can be used to perform both estimation (i.e., finding the center of the posterior Gaussian pdf) and simulation (drawing samples the posterior Gaussian pdf). We present the kriging system we use to implement a flexible GSLIB based algorithm for solving linear inverse problems.

We show how we implement such a simulation program conditioned to linear

average data. The program is called VISIM as an acronym for Volume average Integration SIMulation. An effort has been made to make the program efficient, even for larger scale problems, and the computational efficiency and accuracy of the code is investigated.

Using a synthetic cross-borehole tomography case study, we show how the program can be used to generate realizations of the a posteriori distributions (i.e. solutions) from a linear tomography problem. Both Gaussian and non-Gaussian a priori model parameter distributions are considered.

*Key words:* kriging, sequential simulation, linear inverse theory

## 1 Introduction

Some data $\mathbf{d}$ are indirect measurements, and hence a function of some model parameters $\mathbf{m}$ (typically describing subsurface structure). Let the forward problem be the problem of calculating data from a given set of model parameters using a function $\mathbf{g}$, typically related to some physical problem, such that $\mathbf{d} = \mathbf{g}(\mathbf{m})$. We refer to the problem of inferring properties of $\mathbf{m}$ from measurements $\mathbf{d}$ as the inverse problem.

In a Bayesian formulation the solution to an inverse problem is a probability density function (pdf) (the posterior pdf, $\sigma$) which is given as a normalized (K is a normalization factor) product of the pdf describing the a priori information (the prior pdf, $\rho$) and the likelihood function ($\Delta$), related to the forward

*Email address:* tmh@gfy.ku.dk (Thomas Mejer Hansen).
*URL:* http://imgp.gfy.ku.dk/ (Thomas Mejer Hansen).

operator $\mathbf{g}$ (which is related to some physical law) (Tarantola, 2005) :

$$\sigma(\mathbf{d},\mathbf{m}) \ = \mathrm{K}\ \rho(\mathbf{d},\mathbf{m})\ \boldsymbol{\Delta}(\mathbf{d},\mathbf{m}) \tag{1}$$

In case the inverse problem is linear, such that the operator $\mathbf{g}$ is linear, and Gaussian, such that the prior pdf and the noise distribution is Gaussian, the solution to the inverse problems is a Gaussian pdf described by a mean and a covariance.

The center of this Gaussian a posteriori pdf is often chosen as the 'solution' to the inverse problem, but this point cannot adequately describe the a posteriori pdf. To do this a representative sample (a set of realizations) of the posteriori pdf must be generated, from which a posteriori statistics can be obtained. This can be done using for example Markov chain Monte Carlo methods (Mosegaard and Tarantola, 1995). This is, however, computationally very expensive.

Hansen et al. (2006) show how the a posteriori Gaussian distribution can be exactly described by a simple kriging system with noisy data of mixed support (that is, point- as well as volume support). Therefore sequential simulation techniques can be used to efficiently draw samples of the a posteriori pdf, as described by Hansen et al. (2006) and Gómez-Hernández and Cassiraga (2000).

In this manuscript we will introduce the kriging system to deal with noisy data of mixed support. Using this kriging system we show how to implement a sequential simulation program that generates realizations of a random field with a chosen a priori mean, variance, covariance and histogram, honoring observations of point- as well as volume support.

This approach can be used to solve linear inverse problems with an a priori 2-point covariance model, in the sense that actual samples from the a posteriori distribution of a linear inverse problem can be drawn in a computationally efficient manner. This is an improvement over traditional least-squares-based linear inversion where only the center of the Gaussian posterior pdf is chosen as the solution.

Making use of direct sequential simulation, the method we propose is not restricted to the assumption of a Gaussian distribution over the model parameter space as in traditional least-squares-based linear inversion. Any histogram can be used to describe the prior distribution over the model parameter space.

Using a synthetic cross-borehole tomography example we will illustrate the use of the program as well as investigate the computational efficiency and accuracy of a number of available simulation and estimation options.

## 2 Theory

Hansen et al. (2006) describes how samples of the a posteriori pdf of linear Gaussian inverse problems can be generated using sequential simulation. The least squares system used by Hansen et al. (2006) as part of sequential simulation, can be formulated as a simple kriging system with weighed linear average data with associated Gaussian measurement error. This is the kriging system we use as part of the presented sequential simulation algorithm, and it will briefly be introduced here.

Let $Z(\mathbf{u})$ denote the measurement of some random variable $Z$ at the location $\mathbf{u}$ with measurement error R. This is the definition of a datum of 'point sup-

port'. For the remainder of this paper we will refer to measurement errors as errors that are non-systematic, uncorrelated with the random function $Z$ and possibly correlated among themselves, following Chiles and Delfiner (1999, page 211).

Let $Z_v$ be the measurement of a weighted linear average of $Z$ over the block $v$, with measurement error R, such that

$$\mathrm{Z}_v = \frac{1}{|v|}\int_v \mathrm{w}(\mathbf{u})\ Z(\mathbf{u})\ d\mathbf{u}\ + \mathrm{R} \tag{2}$$

$$\approx \frac{1}{N\ \sum \mathrm{w}(\mathrm{u}_i)} \sum_{i=1}^{N} \mathrm{w}(\mathrm{u}_i)\ Z(\mathrm{u}_i)\ + \mathrm{R} \tag{3}$$

$\mathrm{w}(\mathbf{u})$ is an averaging kernel allowing variable weight within the defined support. $|v|$ is the volume of the support. This is the definition of a datum with 'volume support'. Any datum of a Gaussian linear inverse problem can be given by $\mathrm{Z}_v$.

In the discrete case, eqn. 3, $\mathrm{w}(\mathrm{u}_i)$ are averaging weights for each of the $N$ points, $\mathbf{u}$=$[\mathrm{u}_1, \mathrm{u}_2, ..., \mathrm{u}_N]$ in the support $v$. If $N = 1$ and $\mathrm{w}(\mathrm{u}_1) = 1$ eqn. 3 reduces to an expression for a measurement of point support, and therefore data of any support is described by eqn. 2-3.

Given n data, $Z_{v,i}$ where $i = 1, ..., \mathrm{n}$, interpreted as average measurements of a realization of a stationary random function, $RF$, and an a priori model of the mean $m$, variance $\sigma_0^2$, and covariance $C$ (2-point spatial connectivity) of the $RF$, the mean and variance of the Gaussian probability density function, $N(\mu_k, \sigma_k^2)$, describing the outcome of the $RF$ over any support $v$, conditioned to data, can be estimated. This mean and variance is referred to as the kriging mean $\mu_K$ and variance $\sigma_K^2$. $N(\mu_k, \sigma_k^2)$ is strictly identical to the local a

posteriori pdf of a Gaussian linear inverse problem.

$\mu_k$ and $\sigma_k^2$ can be obtained by solving the system of normal equations equivalent to the 'kriging system' (see Journel and Huijbregts, 1978, chapter V):

$$\sum_{\beta=1}^{n(\mathbf{u})} \lambda_\beta \, [\mathrm{C}(v_\alpha, v_\beta) \; + \; \mathrm{R}(v_\alpha, v_\beta) \,] \;\; = \;\; \mathrm{C}(v_\alpha, v) \quad \forall \alpha = 1, ..., \mathrm{n} \tag{4}$$

and in matrix notation :

$$[\, \mathbf{K} + \mathbf{C}_D \,] \, \boldsymbol{\lambda} \;\; = \;\; \mathbf{k} \tag{5}$$

where $\mathbf{K} = \mathrm{C}(v_\alpha, v_\beta)$ is the data to data covariance matrix, i.e. the covariance between the support of all pairs of observations. $\mathbf{C}_D = \mathrm{R}$ is the data to data measurement error covariance matrix. $\mathbf{k} = \mathrm{C}(v_\alpha, v)$ is the data to unknown covariance matrix, i.e. the covariance between the unknown and the observed data.

$\boldsymbol{\lambda} \;\; = \;\; [\lambda_1, \lambda_2, ..., \lambda_n]$ is the kriging weights that can be obtained by solving eqn. 5 as $\boldsymbol{\lambda} = [\mathbf{K} \; + \mathbf{C}_D]^{-1} \; \mathbf{k}$. Having solved this linear system of equations the kriging mean, $\mu_K$, and variance, $\sigma_K^2$, can be calculated as (see for example Journel and Huijbregts, 1978)

$$\mu_K = \sum_{\alpha=1}^{n(\mathbf{u})} \lambda_\alpha \; (Z_{v,\alpha} - m) + m \tag{6}$$

$$\sigma_K^2 = C(0) \;\; - \;\; \boldsymbol{\lambda} \;\cdot\; \mathbf{k} \tag{7}$$

Let $\mathrm{C}(\mathbf{h})$ be the covariance as a function of distance, $\mathbf{h}$, and $\overline{\mathrm{C}}(v_\alpha, v_\beta)$ be the covariance between two data of support $v_\alpha$ and $v_\beta$, then $\mathbf{k}$ and $\mathbf{K}$ can be

calculated using:

$$\begin{aligned}\overline{\mathrm{C}}(v_\alpha, v_\beta) &= \frac{1}{|v_\alpha|\ |v_\beta|} \int_{v_\alpha}\int_{v_\beta} \mathrm{w}_\alpha(\mathbf{u}')\ \mathrm{w}_\beta(\mathbf{u}'')\ \mathrm{C}(\mathbf{u}'-\mathbf{u}'')\ d\mathbf{u}'\ d\mathbf{u}'' \\ &\approx \frac{1}{\sum \mathrm{w}(\mathbf{u}_\alpha) \sum \mathrm{w}(\mathbf{u}_\beta)} \sum_{i=1}^{N_\alpha}\sum_{j=1}^{N_\beta} \mathrm{w}_\alpha(\mathbf{u}_i)\ \mathrm{w}_\beta(\mathbf{u}_j)\ \mathrm{C}(\mathbf{u}_i-\mathbf{u}_j)\end{aligned} \tag{8}$$

The general simple kriging system and solution in eqns. 4-7 is completely general in that both observations and estimations can be of any support, with any linear averaging kernel $\mathbf{w} = [\mathrm{w}_1, ..., \mathrm{w}_N]$, and data uncertainty given by $\mathbf{C}_D$.

Essential for the application of simple kriging to linear inverse Gaussian problems, is the specification of measurement error, through $\mathbf{C}_D$, and the use of a variable weight function, $\mathbf{w}$, within the support.

### *2.1 Sequential Simulation with noisy data of mixed support*

Sequential simulation is a method used to generate samples of point values of the $RF$ described above, given a local conditional probability density function. $N(\mu_k, \sigma_k^2)$ is precisely the local Gaussian pdf conditional to data observations, $\mathrm{Z}_v$, and can therefore be used as part of sequential simulation, to generate realization of the $RF$ in question.

In essence sequential simulation programs randomly visit all nodes in the model grid. At each node a data value is simulated from a local conditional distribution (conditional to all observed data and previously sampled data). Once all nodes in a model grid has been visited, one realization is generated. A new realization is generated simply by rerunning the process with a new random seed. See for example Goovaerts (1997) for a general introduction

to sequential simulation and Gómez-Hernández and Cassiraga (2000) for the first application of sequential simulation conditioned to linear average data observations.

Because we make use of the general simple kriging system with Gaussian measurement errors and a variable weight function within the support, equivalent to linear least squares inversion, the realizations generated are samples of the posterior pdf of a linear, Gaussian inverse problem.

It is thus conceptually straightforward to implement sequential simulation based on the given kriging system. However, simple substitution of the traditional point-based kriging system in existing sequential simulation algorithms with the kriging system in eqns. 4-7, will lead to a CPU intensive algorithm that will not honor data of volume support, Gómez-Hernández and Cassiraga (2000); Hansen et al. (2006). The remainder of this manuscript will deal with the practical implementation of the sequential simulation algorithm allowing samples of the a posteriori pdf of linear, Gaussian inverse problems to be generated.

### *2.2 Alternate method*

An alternative to sequential simulation to condition observations to linear average data is conditional estimation/simulation through error simulation, as suggested by Journel and Huijbregts (1978, page 495) in a co-kriging formulation. The method makes use of a combination of conditional kriging, unconditional simulation, and estimation of the kriging error. Carr and Myers (1985) extends this approach to co-conditional simulation. This approach has been

used by Gloaguen et al. (2004), Gloaguen et al. (2005) and Gómez-Hernández et al. (2004) to perform conditional simulation, conditioned to linear average data.

We shall investigate the difference in computational efficiency and accuracy between the methods later.

## 3 Implementation

We have implemented a sequential simulation algorithm, based on the general simple kriging system conditioned to noisy data of mixed support as given in eqn. 4, in Fortran 77. The program is based on and follow the style of the GSLIB geostatistical software package, Deutsch and Journel (1998). We label the code VISIM, as an acronym for Volume Integration SIMulation.

Figure 1 shows a flowchart of the implementation of VISIM. The structure of the flowchart is similar to any other implementation of sequential simulation except for the sections labeled in boldface, indicating sub-functions and options that have been implemented specifically to enable conditioning to the linear average data.

The implementation of VISIM can be done on top of the framework of an existing sequential simulation algorithm using data of point support only, as indicated by Figure 1 . Here we will discuss the implementation aspects of VISIM that are not part of conventional sequential simulation algorithms, and thus specific to VISIM.

### *3.1 Kriging system*

As shown in the previous section, the kriging system to be solved in presence of volume average data is only slightly more complex than a conventional kriging system. The only difference is in calculating the covariance matrices. We have implemented a kriging system considering data of any (both point and volume) support and labeled the function `krig_volume`. In case only data of point support are available the kriging system reduces to the conventional point-based kriging algorithm and makes VISIM behave like a traditional sequential simulation program.

The kriging weights obtained at each node can be optionally saved to and read from disk. This has dramatic effect of the computational efficiency when using the method of conditional simulation through error simulation. This will be discussed further as part of the following case study.

### *3.2 Direct or Gaussian sequential simulation*

Two of the most widely used sequential simulation techniques are sequential Gaussian simulation (sgsim) and direct sequential simulation (dssim). The implementation of sgsim and dssim is very similar, except for a few steps. VISIM can be used to perform both sgsim and dssim.

#### *3.2.1 Sequential Gaussian simulation*

sgsim requires that the distribution of data is Gaussian. This can be ensured using for example a normal score transform of the original data. Sequential

Gaussian simulation is performed in normal score space, and the realizations are subsequently transformed back into the original space.

With respect to linear average measurements as considered in this paper, the (non-linear) normal score transformation makes it impossible to honor the data of volume support. Therefore one cannot make use of the normal score transformation, and only make use of sgsim in case all data of any volume support can be considered Gaussian, such that no normal score transformation is needed.

### *3.2.2 Direct sequential simulation*

As the Gaussian assumption over the model parameter space is rarely met, direct sequential simulation is attractive for our purpose. Journel (1994) found that one can draw samples from an arbitrarily shaped local probability density function, and still ensure reproduction of the mean, variance and semivariogram, as long as the mean and variance of the local conditioning distribution corresponds to the local kriging mean and variance.

Consider for example a lognormal global distribution as target distribution. During simulation one can draw samples from a lognormal distribution with mean and variance according to the kriging mean and kriging variance. However, a reason for the relative lack of success of dssim has been that even though the a priori chosen mean, variance and semivariogram are reproduced, the a priori chosen histogram is not. Post-processing of generated realizations has been suggested to transform data into a distribution with correct histogram, while honoring the hard data, Journel and Xu (1994). In the context of this paper, post processing of the realizations will lead to corrected data realiza-

tions that do not honor the observed volume average measurements. Therefore we do not consider using post processing of the generated realizations.

Deutsch et al. (2000) and Soares (2001) show how histogram reproduction using dssim can be ensured without any post processing of data realizations. They show how the local shape of the conditional distribution, centered on the simple kriging mean estimate, can be calculated. Drawing from such a correctly shaped conditional distribution ensures that the a priori chosen histogram will be matched.

We have implemented direct sequential simulation with histogram reproduction based on the approach proposed by Deutsch et al. (2000) and Oz et al. (2003), who suggest to use a lookup table of the mean, variance and shape of the local conditional distribution. Prior to running sequential simulation a lookup table is calculated containing the correctly shaped distribution for a limited set of mean and variances.

This is performed in a three-step process and implemented in the `create_condtab` function. First the target histogram is converted into normal score space, using a normal score transformation. Second, from a number of Gaussian distributions in normal score space, with given means and variances, a number of samples are drawn and back transformed to the original space, using the target distribution. The distribution of these back transformed samples in original space is the local probability density function with mean and variance of the back transformed data values. This local probability distribution, as well as the associated mean and variance, are kept in a lookup table. This distribution is now the correctly shaped probability distribution, given the mean and variance computed from the back transformed data, Deutsch et al. (2000). Each

set of pairs of mean and variance values in normal score space corresponds to a set of local pdf, mean and variance in the original back transformed space.

The function `draw_from_condtab` is used to draw samples from the local conditional distribution whose mean and variance are closest to kriging mean and variance.

The calculation of the lookup table need only be done once before the simulation starts. During simulation no normal score transformation occurs. The approach ensures that one draws from a properly shaped conditional distribution that will lead to target histogram reproduction (Oz et al., 2003).

### *3.3 Allowing noisy data*

As mentioned, in geostatistical literature data measurements are typically considered hard data, i.e. noise free. Especially when many linear average measurements are available it becomes crucial to be able to specify data uncertainty. First of all because such data typically are associated with significant measurement error, and secondly because the assumption of noise free data leads to unstable matrix systems, both factors leading to unreliable sequential simulation results.

As we have laid out the theory previously it is trivial to include measurement uncertainty in the kriging system through $C_D$. Covariance matrices can be specified for both correlated and uncorrelated data errors. Accounting for data uncertainty in the kriging system is done in the `krig_volume` function as part of setting up the kriging system.

### *3.4 Neighborhood*

The use of data 'neighborhoods' is probably one of the main keys to the success of many geostatistical methods. A neighborhood defines the data locations to be used for a specific location to be kriged. There are two main effects of selecting a neighborhood. 1) For methods based on ordinary kriging and kriging with a trend, the trend is evaluated within the specified neighborhood only, allowing the utilization of a non-stationary mean. The method we present here is based on simple kriging, and thus assumes a stationary mean, so the trend considerations are not valid here. 2) Only data locations within the neighborhood are included in the kriging systems to be solved. Often this reduces the size of the kriging system to be solved dramatically, as opposed to using all data, and thus allows for much faster computation time. At the same time, if the neighborhood is selected properly, the computed realizations can be practically identical to those calculated without using a neighborhood. This is practical when using sequential simulation where the number of conditioning data increases as simulation progresses. Three types of neighborhoods are available to reduce the dimension of the kriging system, and implemented through the function `nhoodvol`:

#### *3.4.1 point data neighborhood*

The 'point data neighborhood' is the traditional kriging neighborhood containing a number of previously simulated values. Internally in `visim` no distinction is made between hard original data and previously simulated values. The point data neighborhood contains the previously simulated values and original data with the highest covariance to the point to be simulated. This neighborhood

concept is well understood, see for example the Deutsch and Journel (1998, page 32) for details.

### *3.4.2 volume average neighborhood*

Using all volume average data all the time can become extremely CPU-expensive, and therefore it is necessary to explore the effects of using a 'volume average neighborhood'. By volume average data in the volume average neighborhood we refer to those volume average data actually used in setting up the kriging system.

There are three options for selecting the volume average neighborhood: 1) using all volume average data all the time, 2) using the $N$ volume average measurements with the highest covariance to the point being simulated, and 3) using all volume average measurements with a covariance to the point being simulated above some threshold level. The case study to follow will illustrate the difference between choice of volume-average neighborhood.

### *3.4.3 point-in-volume neighborhood*

During simulation the number of previously simulated nodes increases. In order to exactly match the observed volume average data within their uncertainty, all previously simulated data within the volume average neighborhood must be used when kriging points within that volume, Gómez-Hernández and Cassiraga (2000) and Journel (1999) and . We refer to such previously simulated data within a volume average as point-in-volume data.

As the number of previously simulated nodes increases during simulation the

number of point-in-volume data can become quite large, and therefore CPU demanding. Therefore one can consider to use only part of the available simulated point-in-volume data. We refer to the selection of used point-in-volume data, as the 'point-in-volume neighborhood'. By default VISIM uses the exhaustive (all data) point-in-volume data neighborhood to ensure data reproduction.

Optionally one can, at compile time, choose to select only a specified number of randomly selected data within the point-in-volume neighborhood. This will not ensure exact data reproduction, and is only implemented in case of very large problems where this will allow an approximate but fast approach.

As part of the case study to follow, the effect of the size of the volume average neighborhood will be illustrated.

### *3.5 The random path*

One has to specify in which order data locations in the grid to be simulated are visited. This is referred to as determining the path. There are two ways to setup the path in VISIM.

The traditional way of defining the path is by using a random path where the likelihood of visiting each point in the grid at any time is the same for all grid nodes. We refer to this path as the 'independent random path'. This is the type of random path used by most geostatistical simulation algorithms. See Goovaerts (1997, p. 379) for details on using random paths with geostatistical simulation.

To improve computational efficiency, we suggest an alternate approach where

more informed nodes are visited prior to less informed nodes. A more informed node has a significantly stronger covariance to a larger number of volume average measurements than a less informed node.

In the early phase of a simulation process, only few data locations have been simulated, and thus the number of data in the point-in-volume neighborhood will be relatively small. At the end of the simulation process most data locations have been visited and thus the number of point data within the point-in-volume neighborhood will be very large, requiring a large kriging system to be solved. Therefore visiting more informed nodes prior to less informed nodes is expected to enhance the computational efficiency. We refer to this choice of path as the 'preferential path'.

The calculation of the random paths is performed by the `rayrand_path` subfunction. The effect of choosing each type of random paths is discussed later as part of a case study.

### *3.6 Covariance lookup tables*

As VISIM is based on GSLIB, a point-to-point covariance lookup table is generated each time `visim` is started. This contains covariance values below a selected separation distance, and this makes any simulation code run much faster, by avoiding many recalculations of the same value. In addition, we have implemented lookup tables for both point-to-volume covariances and volume-to-volume covariances through `cov_data2vol` and `cov_vol2vol`. When any of the two functions are called, requesting a covariance value, the function checks if the value has allready been calculated. If this is the case the covariance

`visim_volgeom.eas` value is looked up in memory, otherwise it is calculated and stored in memory.

In contrast to the point covariance lookup table that is only specified for a limited range of separation distances, all point-to-volume and volume-to-volume covariances are, as simulation progresses, stored in memory. This is possible because the number of averaging data is typically much smaller than the dimension of the model parameter space, and because there is significant computational improvements in doing so, which we shall demonstrate later as part of the case study.

Both the volume-to-volume and data-to-volume covariance lookup tables can be saved to disk at the end of simulation and read from disk prior to simulation. This enables the use of external methods for calculating the covariance lookup tables, using for example the computationally very efficient fast Fourier transform method proposed by Liu et al (2006).

## 4 Case Study: Cross-borehole tomography

A synthetic cross-bore hole tomography problem is chosen as a case study to illustrate the computational efficiency and accuracy of VISIM, and also to serve as an illustration of how to use VISIM. The ray geometry and subsurface model size is inspired by an actual ground-penetrating-radar tomography investigation in North Sealand, Denmark. Tomography is a typical inverse problem, that can be formulated as a linear or linearizable inverse problem and therefore provides a good case study for running VISIM.

All computations to follow were performed on a Pentium 3.2GHz with 3Gb

RAM and the Fortran 77 source code was compiled using the Intel Fortran Compiler for Linux (The source codes compiles using the GNU Fortran 77 compiler as well).

### *4.1 Reference model*

Consider a reference velocity model bounded by two vertical bore holes, 12 meter deep with 5 meter separation. To infer the subsurface velocity structure a cross-borehole tomography experiment is carried out, measuring observed travel times between sources and receivers placed in the boreholes. 14 sources are evenly spread down through one borehole, and 14 receivers are evenly spread down through the other borehole. All receivers measure the arrival time for rays traveling from all sources. All in all 196 rays are considered. See Figure 2 for the geometry of the rays used. Only linear average data are considered, thus no data of point support (such as direct measurements in the bore hole) are considered.

We consider straight rays traversing the model. Thus, given the geometry of the ray, found using ray tracing, both the travel time delay along the ray, $dt$, and the distance traversed, $dx$, are known, and we can compute the measured average velocity along each ray as $v = dx/dt$. These measurements are linear averages over the model parameters with weights proportional to the length traveled in each grid cell. Therefore we deal with a linear inverse problem of estimating the underground velocity model based on linear average measurements, and thus, VISIM can be used to generate samples of the a posteriori distribution of this inverse problem.

We make use of straight rays only to keep the case study simple. It is trivial to include for example bending 'fat' rays, in which case the tomography problem becomes non-linear. In case such a problem is weakly non-linear, iterative inversion techniques can be used to solve a linearized version of the problem (Tarantola, 2005). Once linearized, samples of the a posteriori pdf can be generated using the proposed method as discussed by Hansen et al. (2006).

Figure 2b-c shows the weight function associated to two specific rays. A ray travelling horizontal, Figure 2b, have a constant weight for each cell it passes through, while dipping rays will have varying weight for each cell it passes through according to the length the ray traveling in each cell, see for example Figure 2c.

Two reference models are generated using unconditional simulation in both sgsim and dssim mode. For both models a spherical semivariogram model is chosen as reference, with the principal axis of continuity dipping $6.5^o$ downward to the right. The range in the principal direction is chosen to by 4 meters and the range perpendicular to the principal direction is chosen to be 1 meter. The mean and variance is chosen to be $m_0$ =0.13 m/ns, and $\sigma^2$ =0.0002 respectively. In the dssim case a bimodal distribution (the sum of two Gaussian distributions) is chosen.

Figure 3a and 3b show the unconditional realization chosen as the reference model in the sgsim and dssim case, respectively. Figure 3c-f show the histogram and semivariogram of the chosen reference model, compared to the a priori chosen model. The observed fluctuation around the a priori chosen parameters is due the ergodic fluctuations caused by the finite size of the model grid.

### *4.2 VISIM in sgsim mode*

As mentioned earlier, running VISIM in sgsim estimation mode is identical to solving a traditional, Gaussian linear inverse problem considering the center of the posterior Gaussian pdf as the solution, when using an exhaustive volume average neighborhood. Therefore we will initially investigate the computational accuracy and efficiency in detail while running in sgsim mode, as the results can be compared to the results of least squares linear inversion.

#### *4.2.1 Conditional estimation*

Figure 4 shows the estimated kriging mean and variance at each grid node as a result of running VISIM in estimation mode. This result is identical to the smooth least squares linear inversion result. As we shall see below, any use of a non-exhaustive volume average neighborhood will provide an approximation of the least squares solution.

Note that there is no difference in the results running VISIM in 'estimation' mode using sgsim or dssim, as no actual sampling from the estimated local probability density function is performed. Only the kriged mean and variance are computed, and no data are added as conditioning data in the sequential simulation process. Using dssim, the estimated mean will only be the true mean in case the target distribution is symmetrical around $m_0$. When using any non-symmetrical target histogram one has to perform simulation to obtain the point-wise mean and variance value.

### *4.2.2 Unconditional simulation*

It is good practice to test that the a priori model of the mean, variance, semivariogram and distribution is reproduced using unconditional simulation and that the observed fluctuation of the semivariogram and data distribution are ergodic fluctuations, caused by using a finite size model parameter space, and not the effect of some implementation error/option.

If the fluctuations are ergodic, and not due to some systematic implementation error, then the mean of the mean, variance, semivariogram and histogram of the generated realizations should tend to the a priori chosen mean, variance, semivariogram and histogram as the number of simulations increase.

Figure 5a shows seven unconditional realizations and the E-type of 100 unconditional realizations. As expected, the E-type tends to the constant value of the a priori chosen mean value (0.13).

The computed experimental semivariogram fluctuate around the a priori chosen semivariogram model, Figure 5b. The average semivariogram computed from the experimental semivariogram of all 100 realizations matches the a priori chosen model.

Likewise we notice that the histogram for each realization shows ergodic fluctuation around the reference Gaussian distribution, and that the distribution of all 100 realizations match the reference distribution almost exactly. See Figure 5c.

As expected, Figure 5d shows that the data misfit computed for all rays and all 100 realizations is quite large, as simulations are unconditional to the observed data.

### *4.2.3 Conditional simulation*

Figure 6 shows the result of running conditional simulation using VISIM in sgsim mode.

Figure 6a shows seven conditional realizations and the E-type of 100 conditional realizations, conditioned to volume average data only (the point values down through the bore holes are considered unknown). As can be seen, most of the larger features of the reference model, Figure 3a, can be identified in each of the realizations, and also on the E-type.

Figure 6b shows the experimental variogram and histogram calculated from each of 100 conditional simulations. The ergodic fluctuations are noticeable smaller as compared to those observed for unconditional simulation, Figure 5b.

Note how the ergodic fluctuations of the semivariogram in both the principal ($6.5^o$ dipping downward towards right) and secondary direction, Figure 6b, fluctuate around the semivariogram model of the reference model and not that of the a priori chosen semivariogram model. This is simply caused by the conditioning to the volume average data.

Thus, adding volume average data to VISIM will cause the ergodic fluctuations of the experimental semivariogram to decrease in amplitude, and as the volume average coverage of the model increases the ergodic fluctuations will tend to vary around the actual semivariogram model of the subsurface. In the extreme case where volume average data completely resolve the subsurface model, there will be no ergodic fluctuations.

Figure 6d shows the data misfit computed for all rays and all 100 realizations.

Recall that the Gaussian measurement errors were chosen as Gaussian noise with a variance of $4 \cdot 10^{-6}$, which is very close to the actual measured variance of the data misfit ($3.9 \cdot 10^{-6}$).

The E-type mean of the 100 conditional simulations is shown on Figure 6a, and shows that indeed, as expected, the E-type mean is very close to the least squares inversion result, Figure 4b.

### *4.3 VISIM in dssim mode*

Figure 7 summarizes the result of running visim in dssim mode. As seen on Figure 7b the reproduction is good though not as perfect as in the sgsim case (Figure 6). The bimodal target histogram, Figure 7c, is well reproduced. Even better reproduction could be obtained by using a more densely sampled lookup table for estimating the local pdf to draw from during simulation. The reproduction of the data uncertainty is also quite good, as the variance of the distribution of the observed linear average misfit of 100 realizations is found at $4.5 \cdot 10^{-6}$ as opposed to the true $4.0 \cdot 10^{-6}$.

The use of dssim is thus highly applicable, and enables us to generate samples, efficiently and non-iteratively, of solutions to a linear inverse problem with no Gaussian assumption for the prior distribution in the model space.

### *4.4 The benefit of simulation vs. estimation*

In case one is interested only in the local (ie. point-wise) distribution of the a posteriori distribution, and if the multi-Gaussian model is appropriate, then

there is no need to perform conditional simulation. In that case conditional Gaussian estimation, through the kriging mean and variance, exactly describes the Gaussian a posteriori distribution of a value at a single point in space. However, often one is interested in joint variability at several locations. This is where the sequential simulation approach to solving linear inverse problems becomes highly applicable.

The case study above is inspired by a real ground penetrating radar cross-borehole setup. Here the result of the tomography inversion is to be fed into a flow modeling engine via a transfer function converting velocity to hydraulic conductivity, to model water transportation. In this case it makes little sense to use the minimum variance model (the kriging mean) as this is the smoothest of all possible samples from the a posteriori distribution. Extreme values will be severely underestimated, and thus zones of high and low hydraulic conductivity will not be modelled properly. This will certainly result in a biased flow model. When possible one should generate several realizations of the a posteriori distribution of the cross-borehole tomography problem, and feed them into a flow modeling engine. This will result in a number of different flow models. The variation between each model is a measure of the uncertainty of the flow model.

Another example of the use of samples of the posteriori pdf is the ability to answer complex questions like : "What is the probability that locations A and B are continuously connected with a low velocity layer ?" Such questions are important when dealing with propagation of pollution. The answer can be obtained by counting the proportion of all generated realizations within a sample of the posterior that shows a connected low velocity zone between locations A and B. Such questions cannot be answered using least squares- or

kriging estimation.

## 5 Computational efficiency and accuracy

Using the link to least squares linear inversion as comparison for running VISIM in sgsim mode we shall investigate the computational accuracy and efficiency of a number of features of VISIM.

### *5.1 Effect of using covariance lookup tables*

Table 1 documents the computational speedups associated with the use of data-to-volume and volume-to-volume covariance lookup tables.

With a specific choice of the number of simulations performed and the size of the volume average neighborhood, listed in Table 1, the use of data-to-volume and volume-to-volume covariance lookup tables are in turn enabled and disabled, and the computation time compared. The traditional data-to-data lookup table, as implemented in GSLIB, is enabled for all simulations. Simulated data are identical both when disabling or enabling any lookup tables, which is why no figures need to be shown to illustrate the computational accuracy.

Table 1 shows that the computational efficiency achieved using lookup tables is crucial to any practical use of VISIM. All listed speed-up factors refer to the computation time using no covariance lookup tables, divided by the computation time found using covariance lookup tables. Using 10 volume average data, a computational speedup of about 9 is found. As the number of volume

average data increases, so does the computational speedup. Using 50 volume average data, we observe a speedup of about a factor of 39 and using 100 volume average data, we observe a speedup of about a factor of 100. Thus, the bigger the simulation problem, the larger the effect of using covariance lookup tables.

Also note that the point-to-volume provide much bigger computational efficiency than the volume-to-volume covariance table. This is caused by the fact that the volume-to-volume covariance can quite easily be calculated from the point-to-volume covariance using numerical integration.

Using covariance lookup tables comes at no expense with respect to computational accuracy. The expense paid is to keep the lookup tables stored in physical memory. For the present case the memory needed to store the volume-to-volume covariance lookup tables is nvols*nvols*8 byte = 735*735*8 byte $\approx$ 0.3Mb , and to store the data to volume covariance table nx*ny*nz*nvols*8 byte $\approx$ 1.6 Mb.

In case of very large or many-volume averages the computation time needed to calculate the values for the covariance lookup table may be very large. Then one could make use of the CPU efficient algorithm for calculation of the covariance between data of point support and data of volume support, as proposed by Liu et al. (2006). They make use of the fast Fourier transform to efficiently calculate the point-to-volume lookup table. The volume-to-volume covariance lookup table is then easily calculated using integration of the point-to-volume lookup table.

### *5.2 Effect of choice of random path*

Figure 8 illustrates the two types of random paths. For this example only 14 rays were used, see Figure 8a, and the spatial sampling distance was set to 0.1m. This was chosen only to make the difference between the random paths visually clear.

Figures 8b-c show the first 200 steps of the random walk for the 'independent' and 'preferential' paths. Large black dots denote locations in space visited before locations indicated by smaller black dots.

The two-point statistics obtained from 100 realizations using each of the random paths are very similar and therefore we choose only to show the comparison of the E-type mean of 100 realizations using each of the random paths, Figure 9. We can conclude that, for the setup used here, there is little to no difference between using either of the proposed random paths.

For the case considered we find a computational speedup about 1.7 using the preferential path as compared to using the independent path.

Based on these results the preferential path is the default choice of random path using VISIM. It is suggested to test if the preferential path, in another geometric setup, will produce biased simulations.

### *5.3 Effect of choice of Neighborhood*

The choice of neighborhood has significant implications on the drawn realizations. Using any geostatistical algorithm conditioning to data of point sup-

port, one can choose point neighborhoods that in most cases can be restricted to data within about twice the correlation length of the point to be estimated/simulated.

In the present case the existence of linear average data makes the choice of neighborhoods less trivial. Using all point and volume data at all times (i.e. using exhaustive search neighborhoods) will, when using VISIM in sequential Gaussian mode, produce realizations whose E-type tends exactly to the least squares mean estimate of the corresponding inverse problem, Hansen et al. (2006). Thus, linear inverse problems can be solved exactly using an exhaustive search neighborhood (both point and volume neighborhoods). Any choice of non-exhaustive neighborhood will result in an approximate solution. However, in the following we will show that using well designed data neighborhoods, realizations that are virtually indistinguishable from realizations generated using an exhaustive neighborhood can be generated. By using search neighborhoods, significant computational gains can be obtained.

Here we will only consider the effect of the volume average neighborhood. Thus, we use an exhaustive point-in-volume neighborhood, as discussed previously, to ensure that linear average data observations can be matched within the given uncertainty. The point-neighborhood is chosen as the closest 28 data to the point being simulated. Tests not shown here indicate that the size of the volume average neighborhood has much larger computational implications than the choice of the size of the point-data neighborhood.

Using the setup of the case study above, we generate 100 realizations using 9 different choices of volume average neighborhood selection, using exactly 1, 3, 5, 10, 25, 50, 100 and 196 linear average data, and using a covariance threshold

of 0.1.

Figure 10a shows the exact least squares mean estimate and the E-type using 1, 3, 5, 10, 25, 50, 100 and 196 linear average data. Using few linear average data results in a relatively smooth E-type. Using more linear average data, the E-type tends to resemble the least squares mean estimate, Figure 10j. It is virtually impossible to distinguish the E-type for using a volume average neighborhood with more than 25 linear average data, 10e-h, from the least squares mean result, 10j.

Figure 10i, shows the result of using a covariance threshold of 0.1 to select which linear average data to use in the volume average neighborhood. The E-type using this neighborhood seems to match the E-type using around 10-25 volume average data.

Figure 11 shows how the use of few volume average data results in a semivariogram relatively close to the a priori chosen semivariogram model, and how the average experimental semivariogram tends to the semivariogram model of the reference model as the size of the volume average neighborhood increases. Using a volume average neighborhood of 50 linear average data and above seems to result in realizations with similar experimental semivariograms, and similar to the semivariograms computed using an exhaustive volume average data neighborhood.

We have chosen not to show the histogram for the realizations computed for a different choice of neighborhood, as all choices of neighborhood nicely reproduces the chosen a priori Gaussian model with correct mean and variance.

Figure 12 shows how well the linear average observations are honored by the

simulated realizations in form of the variance of the distribution of the observed difference between true average velocity along rays and the average velocity along the rays computed from realizations. Recall that Gaussian noise with zero mean and variance $4 \cdot 10^{-6}$ was added to the computed average velocities of the reference model, and that the measurement uncertainty used in the simulation was also chosen as $4 \cdot 10^{-6}$. Figure 12 shows that one must use an volume average neighborhood of size 50 to ensure that the distribution of the a priori given measurement uncertainty is matched.

Based on the calculated statistics in Figures 10-12 we conclude that for the present case study it is sufficient to use a volume average data neighborhood of 50 volume average data, in order to reproduce ergodic fluctuations of the mean, variance, semivariogram and histogram, and the observed velocity average data similar to using an exhaustive volume average neighborhood.

Table 2 lists the CPU time used to calculate the 100 realizations for each considered type of neighborhood. As can be seen there is a substantial computational speedup to be gained, using a non-exhaustive volume average neighborhood. In fact, the computational speedup obtained using a volume average neighborhood, as opposed to an exhaustive neighborhood, is about 6.3 using 50 volume average data, and around 16 using 25 volume average data. As the number of volume average data increases, the computational speedup, gained using the volume average neighborhood, dramatically increases. Using for example 25 sources and receivers, as opposed to 14 in the present case study, and thus having 625 linear average measurements, will increase the computation time using an exhaustive volume average neighborhood to around 3500 minutes, while using a volume average neighborhood of 50 volume average data can be performed in 10 minutes, thus with a CPU time speedup of about 35.

The more volume average data are available the more crucial it becomes to use a well selected volume average neighborhood.

The use of volume average neighborhoods is essential, in order to be able to deal with real large-scale problems.

### *5.4 Alternate approach - conditional Gaussian simulation through error simulation*

As described earlier an alternate approach to perform conditional simulation is to perform conditional Gaussian simulation through error simulation (subsequently referred to as the alternate approach). The method is well described in the literature (Carr and Myers (1985); Deutsch and Journel (1998) ; Journel and Huijbregts (1978) page 495) and also applied to linear average data (Gloaguen et al. (2004); Gómez-Hernández et al. (2004)). Therefore we will only provide a brief introduction necessary for later discussion.

A conditional simulation, $\mathbf{m}_{cs}$ can be generated as the sum of the conditional kriging estimator, $\mathbf{m}_{ck}$, and a simulated kriging error, $\mathbf{K}_e = \mathbf{m}_u - \mathbf{m}_{uke}$, as

$$\mathbf{m}_{cs} = \mathbf{m}_{ck} + [\mathbf{m}_u - \mathbf{m}_{uke}] \tag{9}$$

$\mathbf{m}_{ck}$ is the result of kriging estimation conditioned to observed data. $\mathbf{m}_{uke}$ is the kriging estimator conditioned to the data observations obtained using the linear forward operator $\mathbf{g}$ on the unconditional realization ($\mathbf{d}_u = \mathbf{g}\,\mathbf{m}_u$). The difference, $\mathbf{K}_e = [\mathbf{m}_u - \mathbf{m}_{uke}]$, is the simulated kriging error.

This approach is valid under two conditions (Deutsch and Journel, 1998): 1) the error component must be independent or orthogonal to the conditional

kriging, $\mathbf{m}_{ck}$. 2) The error component, $\mathbf{K}_e$, must have the same covariance as the actual error.

### *5.4.1 Adherence to the multi-Gaussian model*

A major limitation of the alternative approach is the implicit assumption of the multi-Gaussian model. Using direct sequential simulation there is no multi-Gaussian restrictions in the distribution of the subsurface model parameters.

### *5.4.2 The alternate approach and noise free data*

In case of using simple kriging and noise free data the above approach has some novel features. The data configuration is then the same for calculating both $\mathbf{m}_{ck}$ and $\mathbf{m}_{uke}$. In such a case one needs only calculate the kriging weights, i.e. $\lambda$ of eqn. 5, once (when calculating $\mathbf{m}_{ck}$). These kriging weights can then be applied when kriging $\mathbf{d}_u$, with no need to perform a matrix inversion to obtain the kriging weights. This implies a major computational speedup, as also noted by Gloaguen et al. (2005). As noted previously VISIM can export and import these kriging weights.

Another advantage of the alternate methods is that the use of the 'point-in-volume' can be ignored (or left empty) as conditioning to linear average data is considered only for kriging estimation and not simulation. In cases where the linear averages span a large area, the 'point-in-volume' neighborhood can become very large. This can result in a very large kriging system to be solved during sequential simulation, and thus the solution of the kriging system will be computationally costly, as considered by Gómez-Hernández et al. (2004) and Hansen et al. (2006)

### *5.4.3 The alternate approach and noisy data*

As noted earlier, geophysical data measurements are associated with an uncertainty, and therefore one can rarely make the assumption of noise free data. Data are then usually given by a Gaussian pdf with mean $\mathbf{d}$ and data uncertainty $\mathbf{C}_D$.

The effect of data uncertainty is not readily applicable to the alternate approach. Specifically when calculating the kriging error one has to perform kriging conditional to $\mathbf{d}_u$ with the associated data uncertainty $\mathbf{C}_{D,u}$, related to the unconditional realization, $\mathbf{m}_u$. As shown above this is trivial when the data observations have no uncertainty, as then $\mathbf{C}_{D,u} = 0$. In case of data uncertainty $\mathbf{C}_{D,u} \neq 0$, and $\mathbf{C}_{D,u}$ cannot be readily obtained.

Both Gloaguen et al. (2004) and Gómez-Hernández et al. (2004) make use of this alternate approach. Though they do not explicitly consider data uncertainty, we will briefly investigate the effect of applying the alternate approach to noisy linear average data, as this approach to conditional simulation is favored over sequential simulation in Gómez-Hernández et al. (2004).

In case of noisy data there are two straightforward ways of applying the alternate approach. One can either select $\mathbf{C}_{D,u}$ as the original data covariance $\mathbf{C}_{D,u} = \mathbf{C}_D$, or set $\mathbf{C}_{D,u} = 0$. Both choices result in approximate solutions, not consistent with the linear inverse problem at hand.

In case of selecting $\mathbf{C}_{D,u} = \mathbf{C}_D$, Figure 13 shows the histogram of the distribution of the observed error $(\mathbf{d}_{obs} - \mathbf{g} * \mathbf{m}_{\mathbf{cond\_sim}})$ for each of 100 realizations generated using sequential Gaussian simulation and the alternate approach. An exhaustive volume average neighborhood is used to focus only on the ef-

fect of a conditional simulation approach. We consider examples using 38, 98 and l98 linear average data, with an a priori chosen measurement error of $\mathbf{C}_D = 4 \cdot 10^{-6}$.

Using the sequential simulation approach results in realizations which honor the a priori chosen $\mathbf{C}_D$. Using 38, 98 and 196 rays respectively the distribution of error is found as [$4.2 \cdot 10^{-6}$, $4.0 \cdot 10^{-6}$, $4.2 \cdot 10^{-6}$]. Using the alternate approach results in realizations with too small error variation. Using 38, 98 and 196 rays, respectively, the distribution of observed error is [$1.1 \cdot 10^{-6}$, $2.1 \cdot 10^{-6}$, $2.6 \cdot 10^{-6}$]. Using the alternate approach imposes a bias in the conditional simulation. The average data fit is better than chosen a priori, and thus the conditional simulation may be fitting noise in addition to data. The E-type mean will be correct, but the simulated fluctuations around this mean will not. This can have dramatic effect on the solution to the inverse problem.

Choosing $\mathbf{C}_{D,u} = 0$ will generate conditional realizations with no variation in data misfit for each realization. All realizations will produce exactly the misfit as when calculated from the conditional kriging estimate $\mathbf{m}_{ck}$. This can cause a serious bias in the conditional realizations.

### *5.5 Computational efficiency of the alternate method*

If the alternate approach is applicable (i.e. data are noise free and the multi-Gaussian framework can be adopted) the alternate approach may provide a very significant computational speedup as compared to using sequential simulation.

Figure 14 show the cpu time needed to compute 10 conditional realization, as

a function of the number of volume average data considered. Both sequential Gaussian simulation and the alternate conditional simulation through error simulation is considered.

It is expected that the cpu time increases exponentially with the number of linear average data for both methods as the cpu time needed for matrix inversion increases exponentially. Figure 14 shows how the computation time actually grows exponentially using sequential simulation. Using the alternate approach of conditional simulation through error simulation there is no significant change in computational cost using any number of volume average data below 60. Using more than 60 linear average the time seems to increase exponentially.

This is caused by the reuse of the kriging weights as discussed before. These are calculated just once when calculating the conditional kriging estimate. The subsequent estimate of the kriged error, can be computed with little computational cost using product of the pre-calculated kriging weights and the data residuals. No matrix inversion is needed. Using the alternate approach, it is therefore only the computation time used to perform kriging estimation conditional to linear average data that increases exponentially in time. The rest of the simulation process uses almost the same CPU regardless of the number and geometry of the linear average data. The reason for the flat curve below 60 linear average data is thus caused by the computational overhead of running the simulation program three times as opposed to one. When considering small data sets (less than 60 linear average data) this overhead is considerably larger (and constant) compared to actually performing the conditional kriging.

In short, if one deals with noise free data, and the multi-Gaussian assumption is appropriate, then conditional simulation through error simulation is computationally much more efficient than using sequential simulation, especially when dealing with linear average data spanning large areas. VISIM can export the kriging weights calculated during one run of VISIM, to be used for another run of VISIM where the data configuration is identical. Thus in cases where circumstances are favorable for using the alternate approach VISIM is very efficient in producing both the conditional estimator and the unconditional simulation needed by eqn. 9.

If the objective is to draw samples of the a posterior pdf of linear inverse Gaussian problems, the use of the alternate approach, as described above, will impose a serious bias in the inversion result. It will in fact not be suitable to generate samples of the a posteriori pdf. For this task sequential simulation provides an 'exact' approach.

### *5.6 Memory requirements*

The memory needed to run VISIM is as for any comparable sequential simulation program, such as SGSIM of GSLIB, with few exceptions.

As implemented, the point to volume and volume to volume covariances are kept in memory. For the problem in the case study this amounts to less than 2Mb.

Making use of dssim requires the use of lookup tables for the mean, variance and shape of the local conditional distribution. The size of these lookup tables is determined by the user, and is independent of the size of the lin-

ear problem. The choices used for the presented modeling will be applicable to most problems. The mean and variance lookup table is both of size 100*100*8b=0.08 Mb, whereas the shape of the local conditional distribution requires 100*100*170*8b=13.6Mb.

Based on these consideration we believe that CPU power, rather than memory requirements, will determine the computational limitations for the application of VISIM, as suggested by Figure 14.

## 6 Conclusion

By setting up a general kriging system with data of mixed support we have proposed and implemented a geostatistical simulation algorithm that has some novel features: a) Realizations are conditioned to data of both point and volume support, b) data noise can be specified, c) both sequential Gaussian and direct sequential simulation is implemented, d) linear inverse problems can be solved, allowing samples of the posterior probability density function to be drawn efficiently.

A software code labeled VISIM has been implemented in Fortran 77, and an effort has been made to make the implementation efficient even on quite large problems. All in all a computational speedup of around 400 was obtained for the case study as compared to using no covariance lookup tables, exhaustive volume average neighborhood and traditional independent random path. At the same time nearly identical mean, variance, semivariogram, histogram and reproduced data were extracted from generated realizations.

In addition, tests have shown that the speedup will increase dramatically as

the size of the simulation problem increases.

The code is flexible to use in that it can be used to perform both sequential estimation and simulation, and can be used as traditional simulation algorithms conditioned to noise free data of point support. In addition the code can efficiently perform conditional simulation through error simulation. This provides the freedom to use whichever method that applies best to a given problem.

Through a synthetic cross-borehole tomography case study, we have shown how the method and code can be applied using both direct and Gaussian sequential simulation, and we have investigated the effects of using various random paths, and the effect of using a non-exhaustive, volume-average neighborhood.

Through the case study we have demonstrated how to include linear volume average data into both Gaussian and direct sequential simulation enabling data from linear inverse problems to be included directly into sequential simulation. Thereby efficient sampling of the a posteriori distribution of Gaussian and non-Gaussian linear inverse problems is possible.

**Acknowledgement** We thank Albert Tarantola and Andre Journel for enlightning discussions on the similarities of linear inverse theory and geostatistics, and in general for great discussions. We thank Yongshe Liu for guidance into GSLIB. Majken Looms is acknowledged for providing a realistic cross-borehole setup geometry. The source code for VISIM and compiled binaries for Linux, Sun and Windows is available from http://imgp.gfy.ku.dk/visim.php. The geostatistical Matlab toolbox, mGstat (http://mgstat.sourceforge.net/), can be used to run, control and visualize the output of VISIM.

## References


Carr, J. R., Myers, D. E., 1985. COSIM: A Fortran IV program for coconditional simulation. Computers & Geosciences 11 (6), 675–705.

Chiles, J.-P., Delfiner, P., 1999. Geostatistics, Modeling Spatial Uncertainty. Wiley series in probability and statistics. John Wiley & Sons, Inc. 770pp.

Deutsch, C. V., Journel, A. G., 1998. GSLIB, Geostatistical Software Library and User's Guide, 2nd Edition. Applied Geostatistics. Oxford University Press. 384pp.

Deutsch, C. V., Tran, T. T., Xie, Y., 2000. An approach to ensure histogram rerpoduction in direct sequential simulation. Tech. rep., Centre for Computational Geostatistics, University of Alberta, Edmonton, Alberta. 19pp.

Gloaguen, E., Marcotte, D., Chouteau, M., 2004. A non-linear tomographic inversion algorithm based on iterated cokrigriging and conditional simulations. In: Leuangthong, O., Deutsch, C. (Eds.), Geostatistics Banff 2004. Vol. 1. Springer, pp. 409–418.

Gloaguen, E., Marcotte, D., Chouteau, M., Perroud, H., 2005. Borehole radar velocity inversion using cokriging and cosimulation. Journal of Applied Geophysics 57 (4), 242–259.

Gómez-Hernández, J., Cassiraga, E. F., 2000. Sequential conditional simulations with linear constrains. In: Monestiez, P., Allard, D., Froideveaux, R. (Eds.), Geostatistics'2000 Cape Town. Geostatistical Association of Southern Africa.

Gómez-Hernández, J., Froidevaux, R., Biver, P., 2004. Exact conditioning to linear constraints in kriging and simulation. In: Leuangthong, O., Deutsch, C. (Eds.), Geostatistics Banff 2004. Vol. 2. Springer, 999–1005.

Goovaerts, P., 1997. Geostatistics for Natural Resources Evalutaion. Applied

Geostatistics Series. Oxford University Press. 496pp.

Hansen, T. M., Journel, A. G., Tarantola, A., Mosegaard, K., 2006. Linear inverse gaussian theory and geostatistics, Geophysics 71 (6), R101–R111.

Journel, A. G., 1994. Modeling uncertainty: Some conceptual thoughts. In: Dimitrakopoulos, R. (Ed.), Geostatistics for the Next Century. Kluwer, Dordrecht, Holland, 30–43.

Journel, A. G., 1999. Conditioning geostatistical operations to nonlinear volume averages,. Mathematical Geology 31, 931–953.

Journel, A. G., Huijbregts, C. J., 1978. Mining Geostatistics. Academic Press. 600pp.

Journel, A. G., Xu, W., 1994. Posterior identification of histograms conditional to local data. Mathematical Geology 26 (3), 323–359.

Liu, Y., Jiang, Y., Kyriakidis, P., 2006. Calculation of average covariance using fast fourier transform. Tech. rep., Stanford Center for Reservoir Forecasting, Petroleum Engineering Department, Stanford University. 27pp.

Mosegaard, K., Tarantola, A., 1995. Monte Carlo sampling of solutions to inverse problems. Journal of Geophysical Research 100 (B7), 12431–12447.

Oz, B., Deutsch, C. V., Tran, T. T., Xie, Y., 2003. DSSIM-HR: a FORTRAN 90 program for direct sequential simulation with histogram reproduction. Computers & Geosciences. 29 (1), 39–51.

Soares, A., November 2001. Direct sequential simulation and cosimulation. Math. Geol 33 (8), 911–926.

Tarantola, A., 2005. Inverse Problem Theory and Methods for Model Parameter Estimation. SIAM, Philadelphia. 352pp.

**List of Tables**

| # Nsim | #VA data | No Lookup | P2V | V2V | P2V + V2V |
|---|---|---|---|---|---|
| 20 | 1 | 1 (7.2/21s) | 1.3 | 6.8 | 7.6 |
| 20 | 10 | 1 (30/55s) | 6.3 | 3.3 | 8.7 |
| 20 | 50 | 1 (213/730s) | 29 | 3.8 | 39 |

Table 1
Computational efficiency related to using of covariance lookup tables. Each row correspond to one run of VISIM, with a specific choice of size of volume average (VA) neighborhood. First column lists number of generated simulations. Second column lists number of volume average data in volume average neighborhood. Columns 3-6 lists observed CPU speedup (as compared to using no lookup table) for specific use of covariance lookup tables. P2V refers to enabling only point-to-volume lookup table, while V2V refers to enabling only volume to volume (V2V) covariance lookup table. P2V+V2V means that both lookup tables are enabled. Time in parenthesis in column three is actual computation in seconds. Computation time is calculated using pre-calculated covariance lookup tables.

| Neighborhood | *196* | *100* | *50* | *25* | *10* | *5* | *3* | *1* | *C 0.1* |
|---|---|---|---|---|---|---|---|---|---|
| Relative ratio | 1(213m) | 3.0 | 6.3 | 16 | 75 | 128 | 172 | 200 | 9 |

Table 2
Computational speed up, generating 100 conditional realizations, using a specific volume average neighborhood, compared to using an exhaustive volume average neighborhood. Volume average neighborhoods using exactly [1,3,5,10,25,50,100,196] volume average measurements in a volume average neighborhood is considered, and also a case using a covariance threshold of 0.1. Time in minutes in parentheses is CPU time needed to generate 100 realizations using an exhaustive neighborhood. Compare to simulation results shows in Figure 10. Computation time is calculated using pre-calculated covariance lookup tables.

**List of Figures**

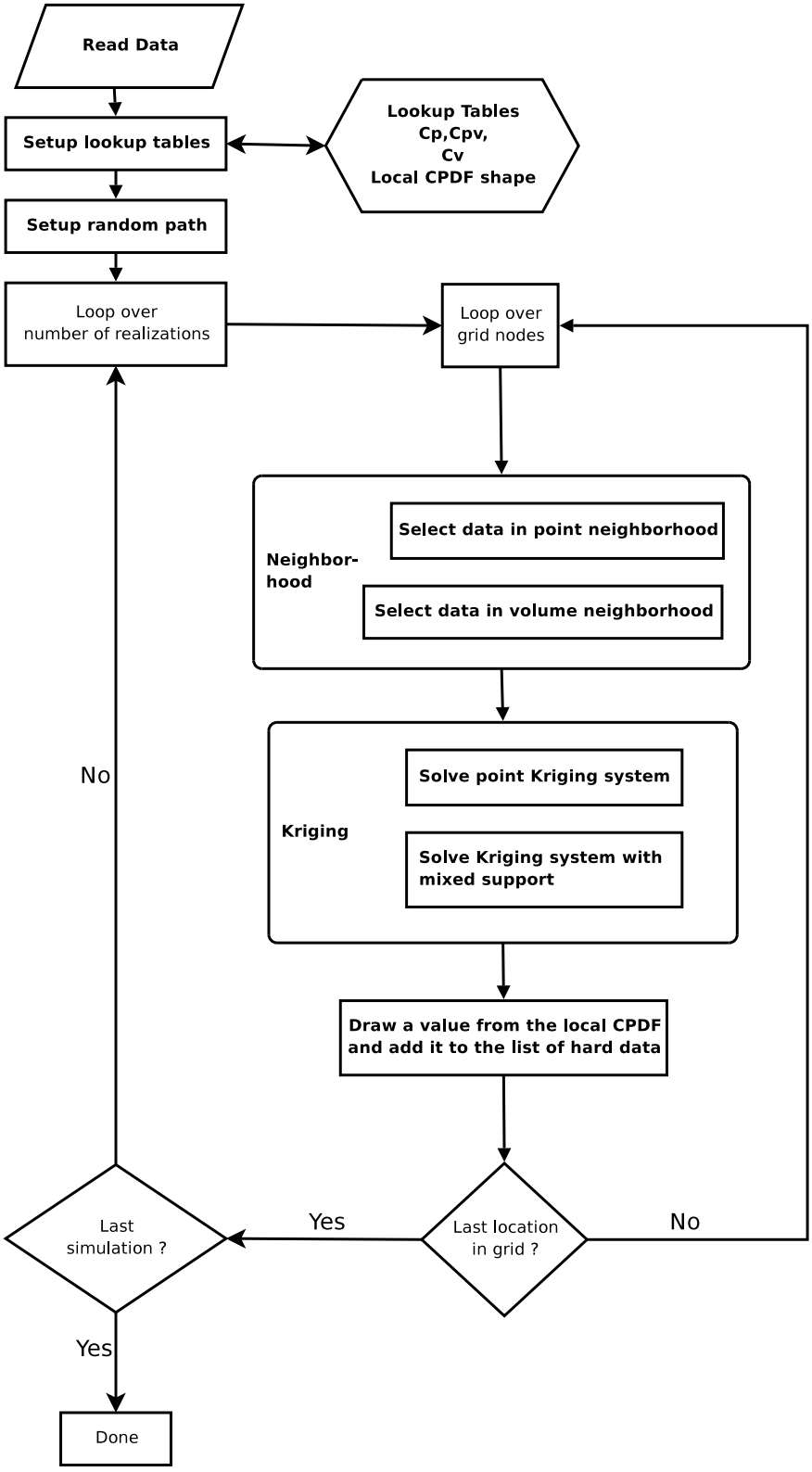

Read Data
Setup lookup tables
Lookup Tables
Cp,Cpv,
Cv
Local CPDF shape
Setup random path
Loop over
number of realizations
Loop over
grid nodes
Neighbor-
hood
Select data in point neighborhood
Select data in volume neighborhood
Kriging
Solve point Kriging system
Solve Kriging system with
mixed support
Draw a value from the local CPDF
and add it to the list of hard data
Last location
in grid ?
No
Yes
Last
simulation ?
No
Yes
Done

Fig. 1.

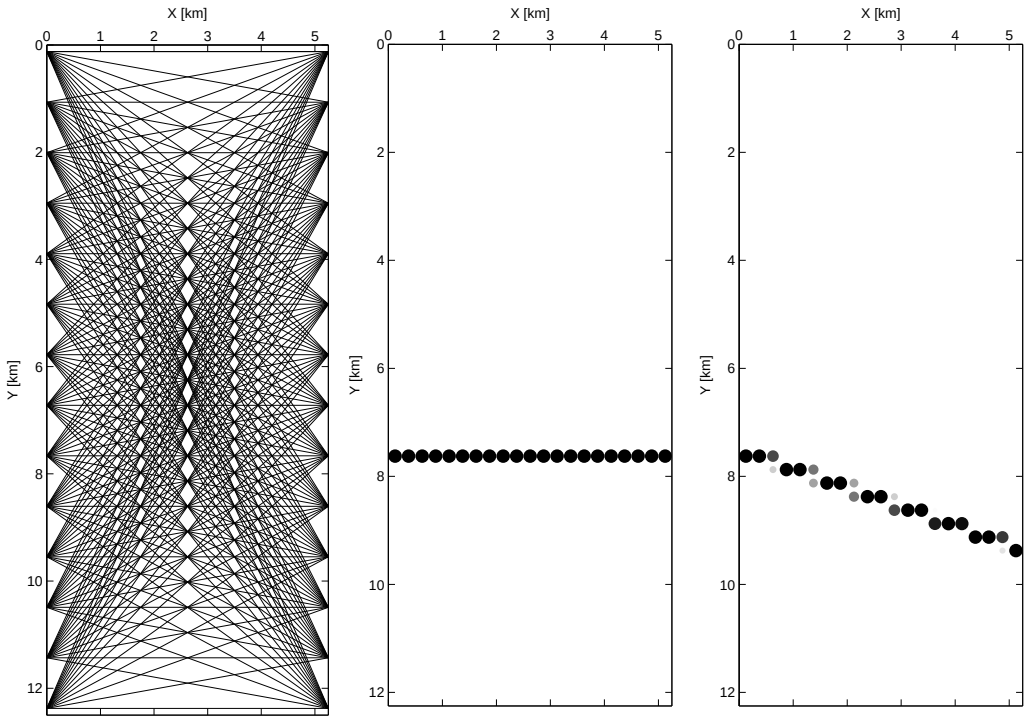
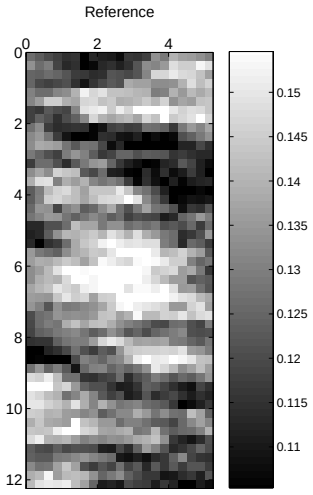
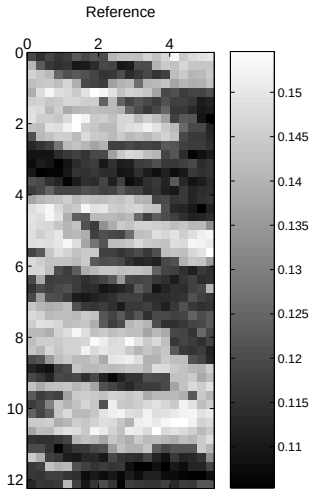


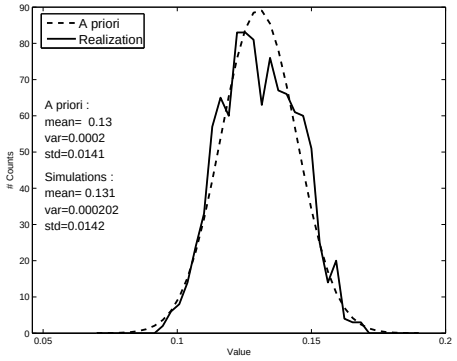

Fig. 2.



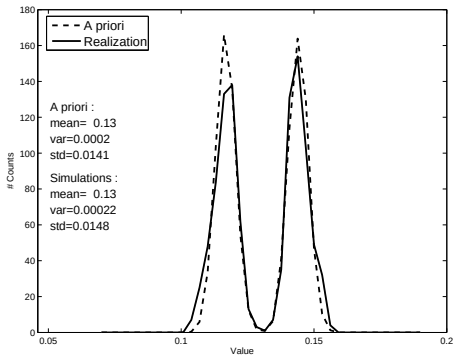

(a) sgsim reference model



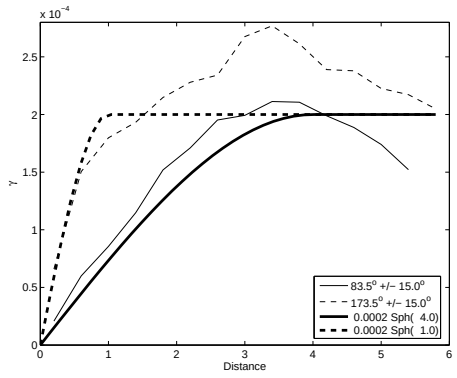

(b) dssim reference model



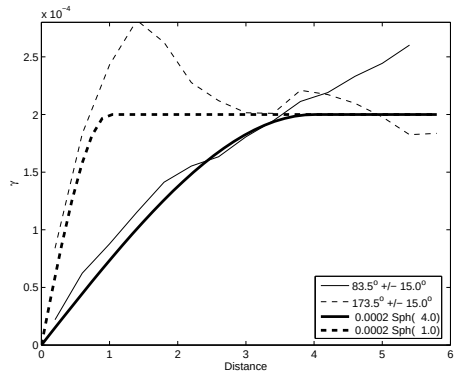

(c) sgsim reference histogram




(d) dssim reference histogram

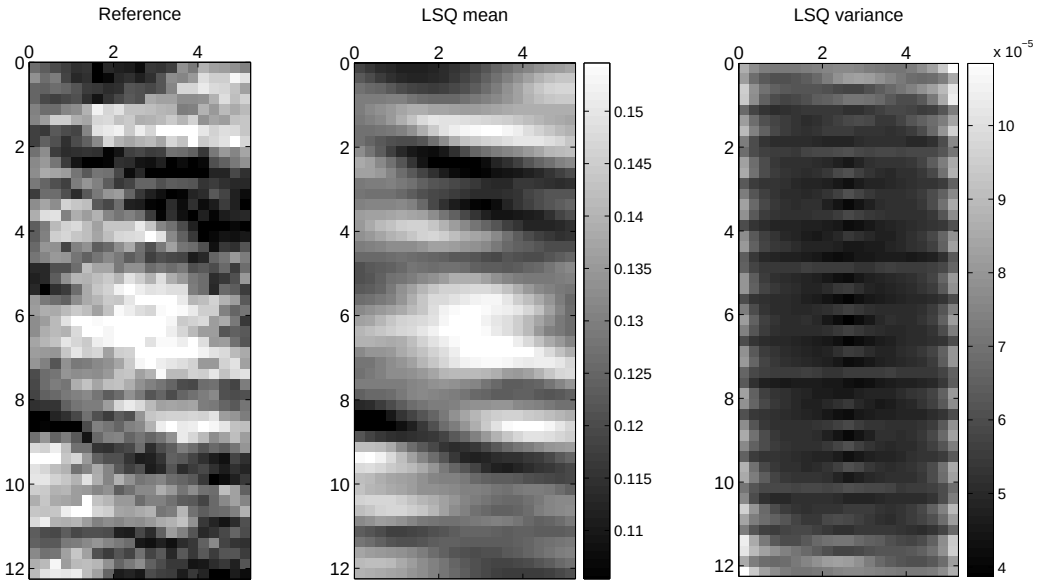



(e) sgsim reference semivariogram



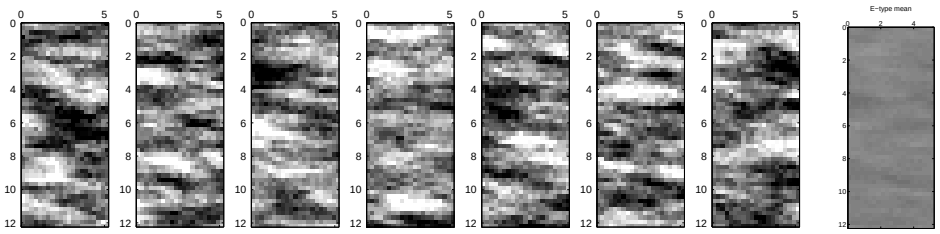

(f) dssim reference semivariogram

Fig. 3.

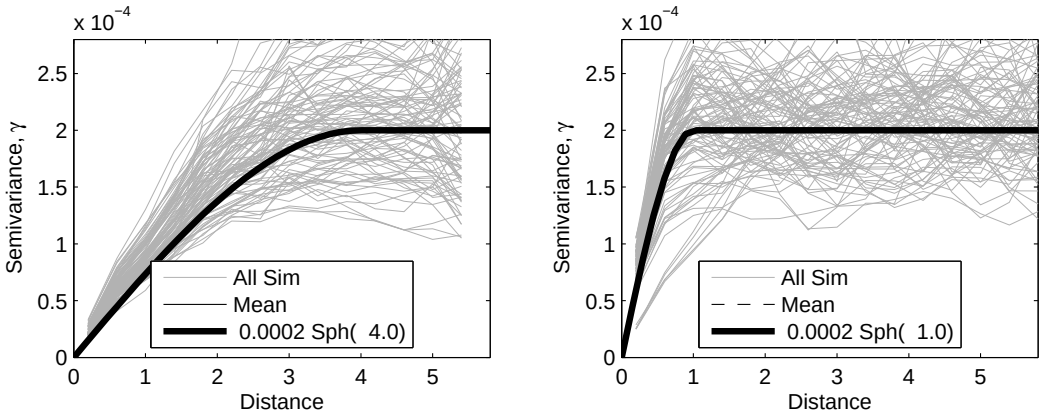

Reference
LSQ mean
LSQ variance
0
2
4
0
2
4
6
8
10
12
0.15
0.145
0.14
0.135
0.13
0.125
0.12
0.115
0.11
x 10^−5
10
9
8
7
6
5
4

(a) (b)

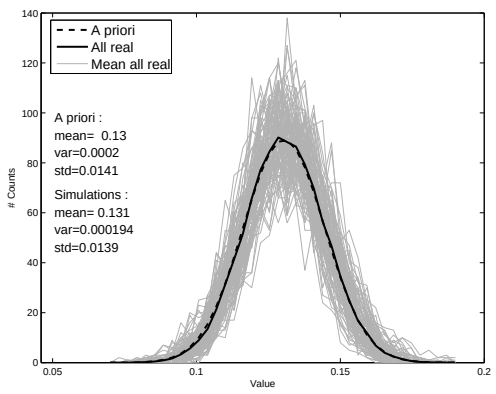
Fig. 4.



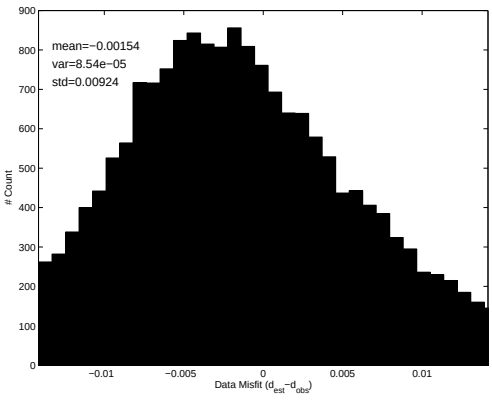

(a) Left) Relizations. Right) E-type of 100 realizations




(b) Experimental semivariogram (thin gray lines) in the principal (left) and secondary (right) direction, compared the the semivariogram of the reference model (dashed black), and the a priori chosen semivariogram model (solid black)

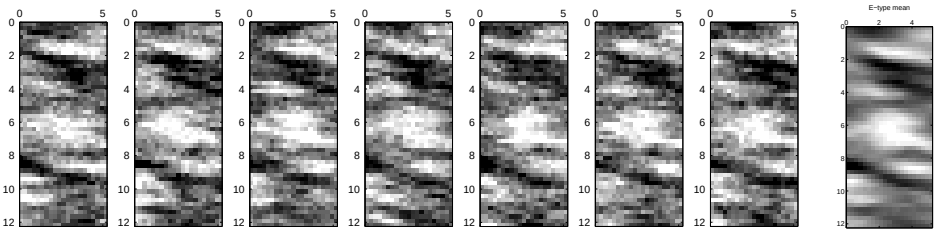



(c) Histogram of 100 realizations (black), average of 100 histograms (green), reference distribution (red).

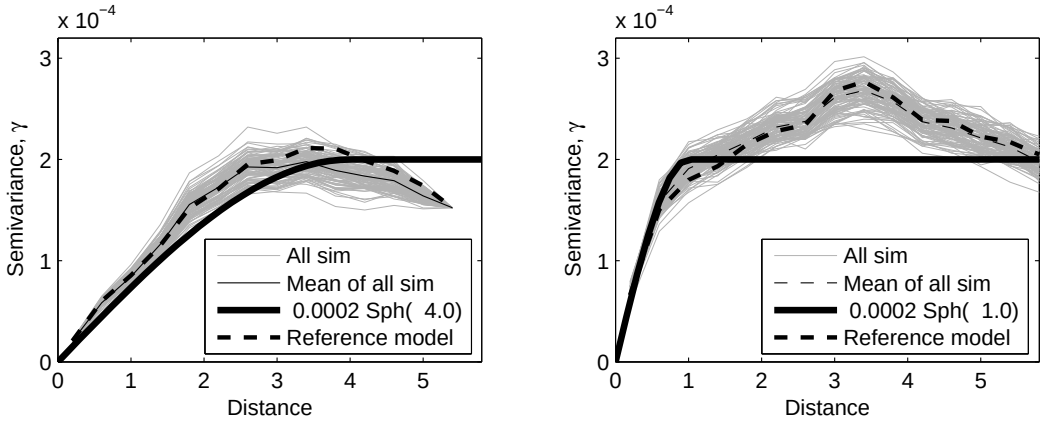



(d) Distribution of computed error of the average velcoity of the realizations and the measured average velocities.

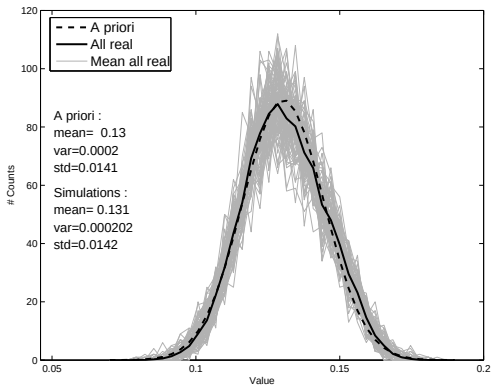
Fig. 5.



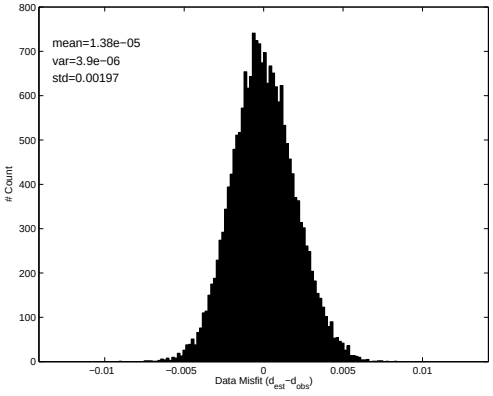

(a) Left) Relizations. Right) E-type of 100 realizations




(b) Experimental semivariogram (thin gray lines) in the principal (left) and secondary (right) direction, compared the the semivariogram of the reference model (dashed black), and the a priori chosen semivariogram model (solid black)

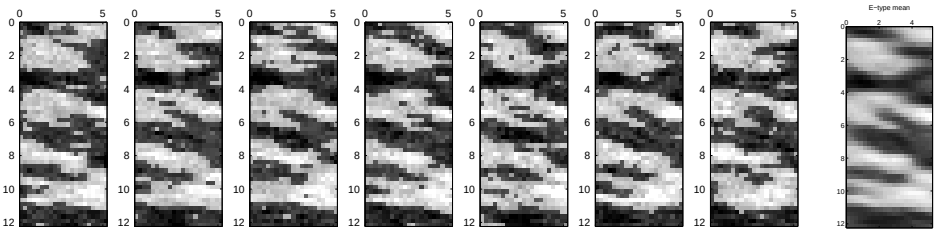



(c) Histogram of 100 realizations (black), average of 100 histograms (green), reference distribution (red).

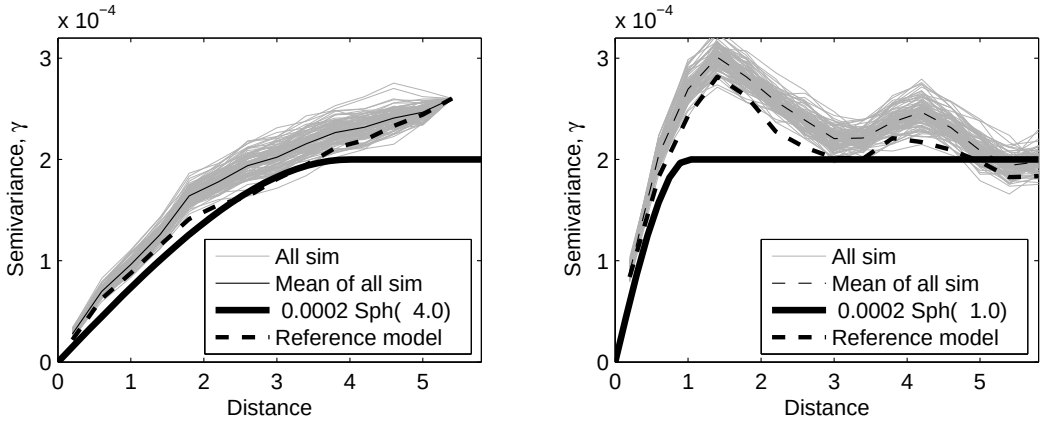



(d) Distribution of computed error of the average velcoity of the realizations and the measured average velocities.

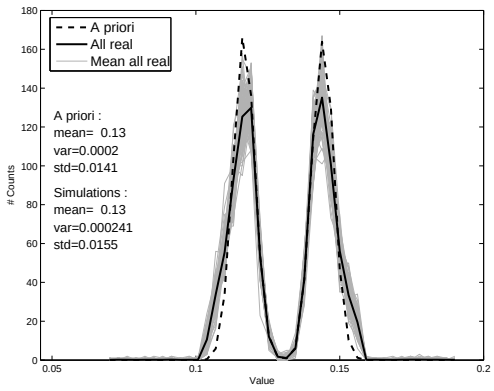
Fig. 6.



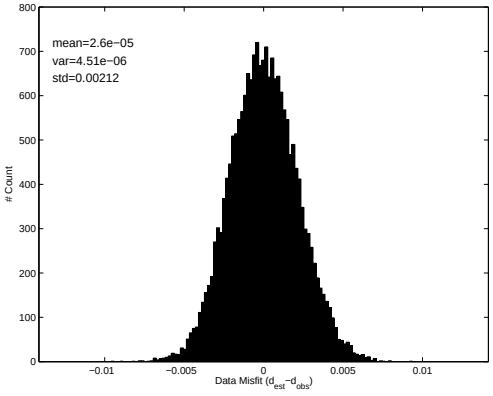

(a) Left) Relizations. Right) E-type of 100 realizations




(b) Experimental semivariogram (thin gray lines) in the principal (left) and secondary (right) direction, compared the the semivariogram of the reference model (dashed black), and the a priori chosen semivariogram model (solid black)




(c) Histogram of 100 realizations (black), average of 100 histograms (green), reference distribution (red).




(d) Distribution of computed error of the average velcoity of the realizations and the measured average velocities.

Fig. 7.

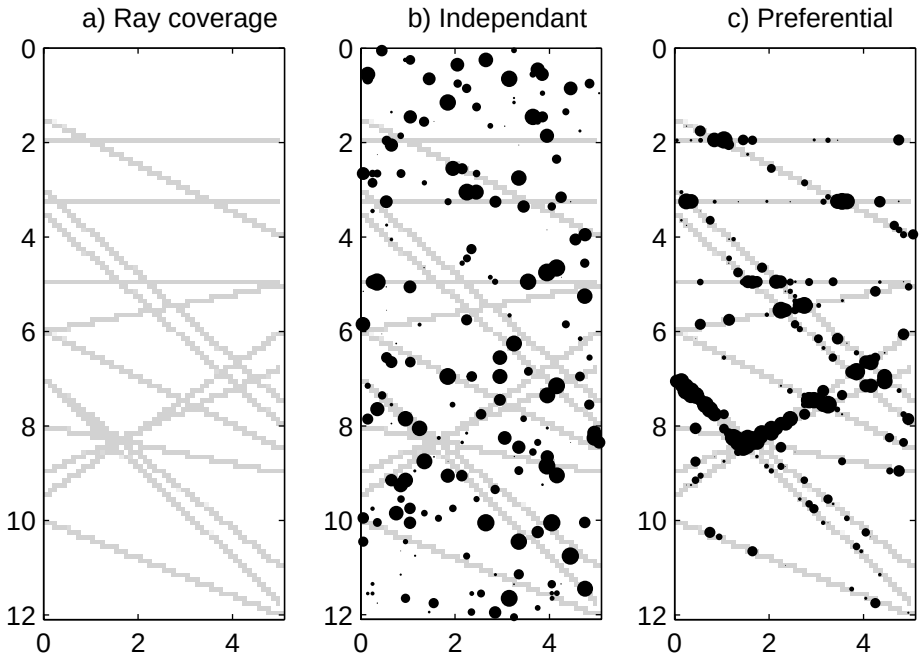



Fig. 8.

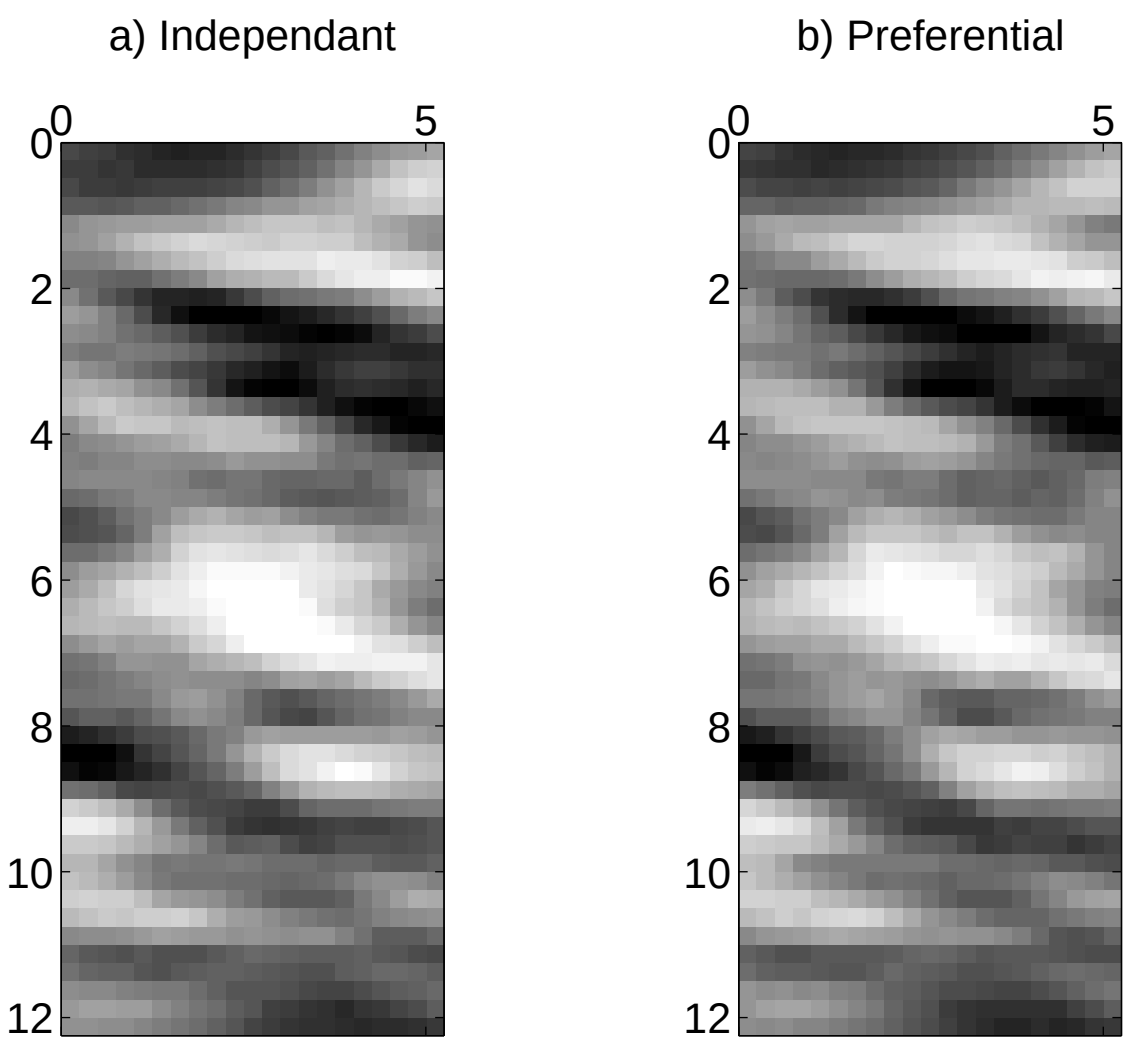



Fig. 9.

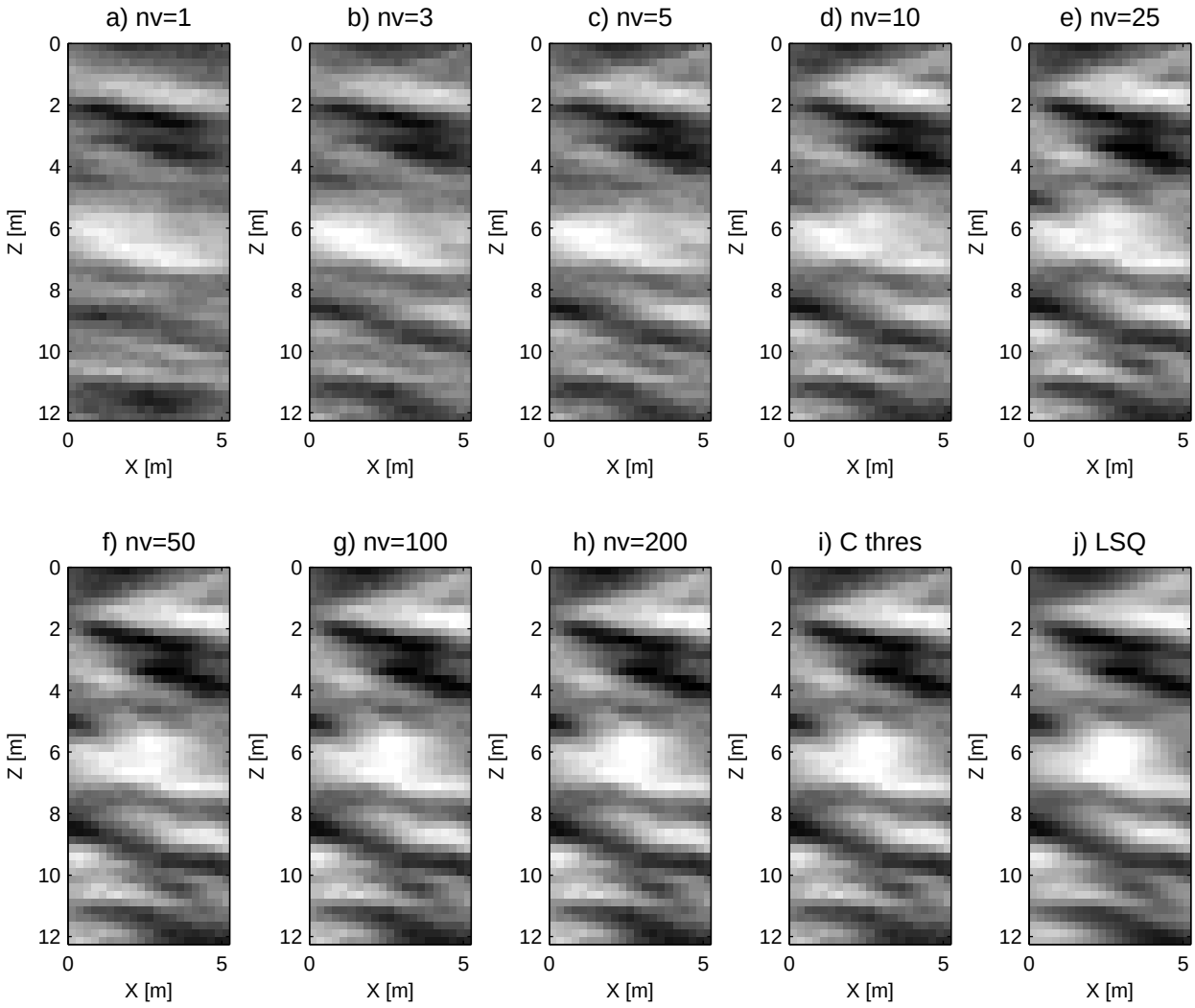

a) nv=1
b) nv=3
c) nv=5
d) nv=10
e) nv=25
f) nv=50
g) nv=100
h) nv=200
i) C thres
j) LSQ
Z [m]
X [m]
0
2
4
6
8
10
12
0
5

Fig. 10.

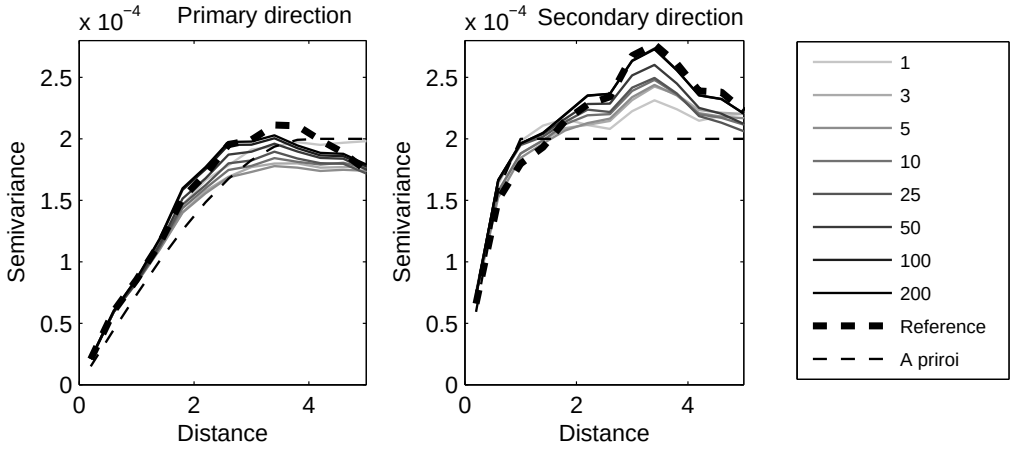

x 10^-4
Primary direction
Semivariance
Distance
x 10^-4
Secondary direction
Semivariance
Distance
1
3
5
10
25
50
100
200
Reference
A priroi

Fig. 11.

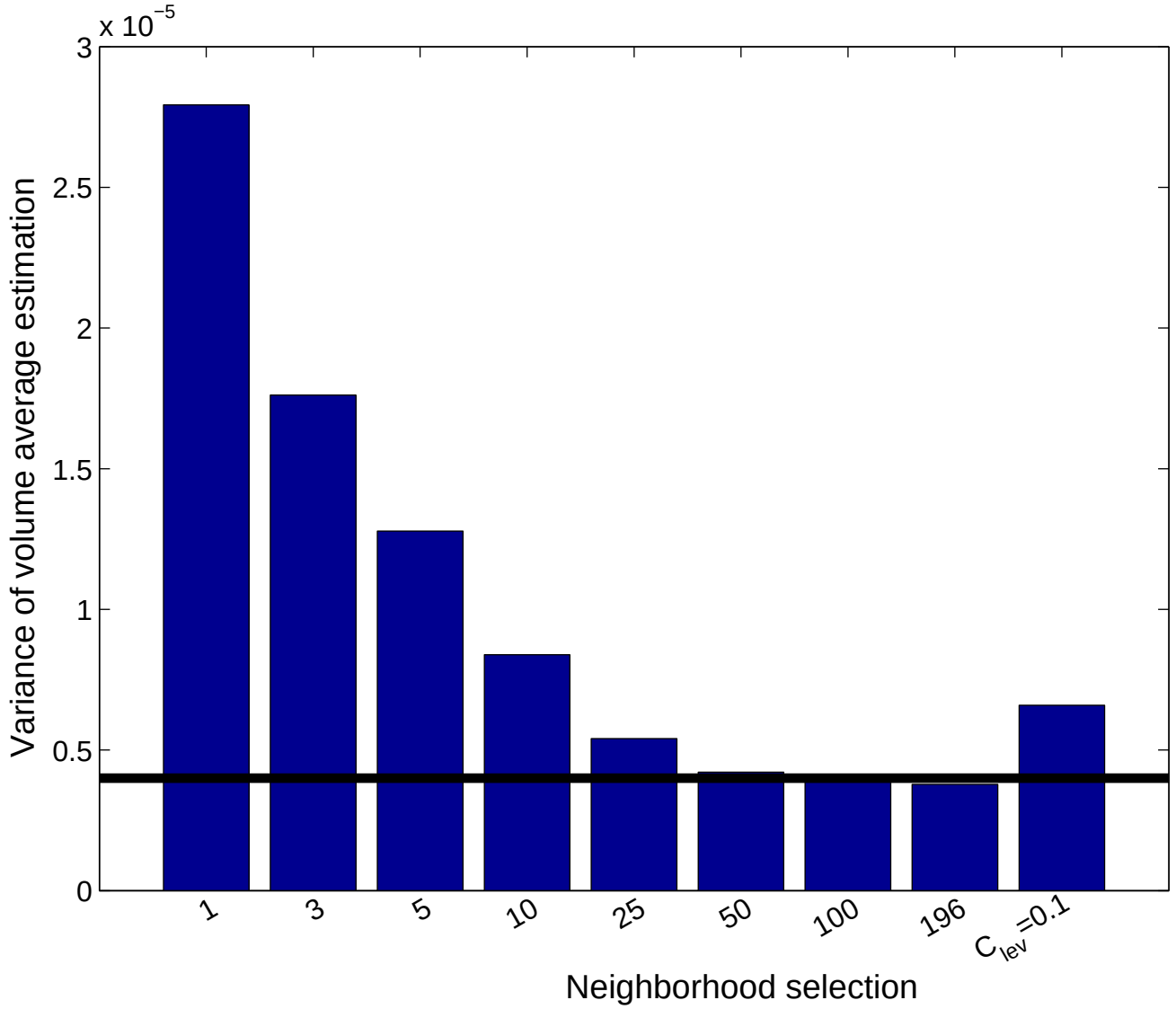

x 10^−5
3
2.5
2
1.5
1
0.5
0
Variance of volume average estimation
1
3
5
10
25
50
100
196
C_lev=0.1
Neighborhood selection

Fig. 12.

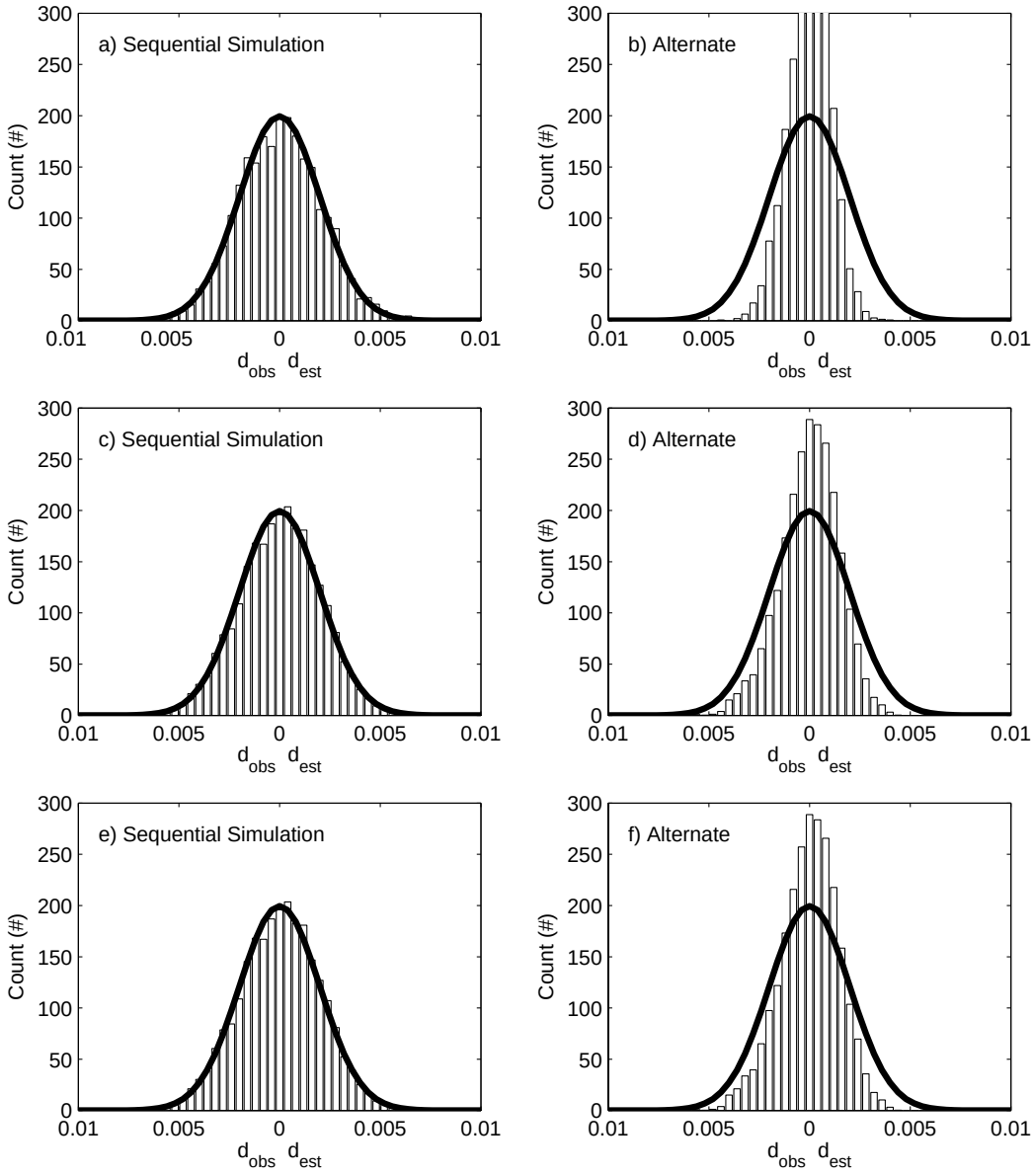

a) Sequential Simulation
b) Alternate
c) Sequential Simulation
d) Alternate
e) Sequential Simulation
f) Alternate
Count (#)
0
50
100
150
200
250
300
0.01
0.005
0
0.005
0.01
$d_{obs}$ $d_{est}$

Fig. 13.

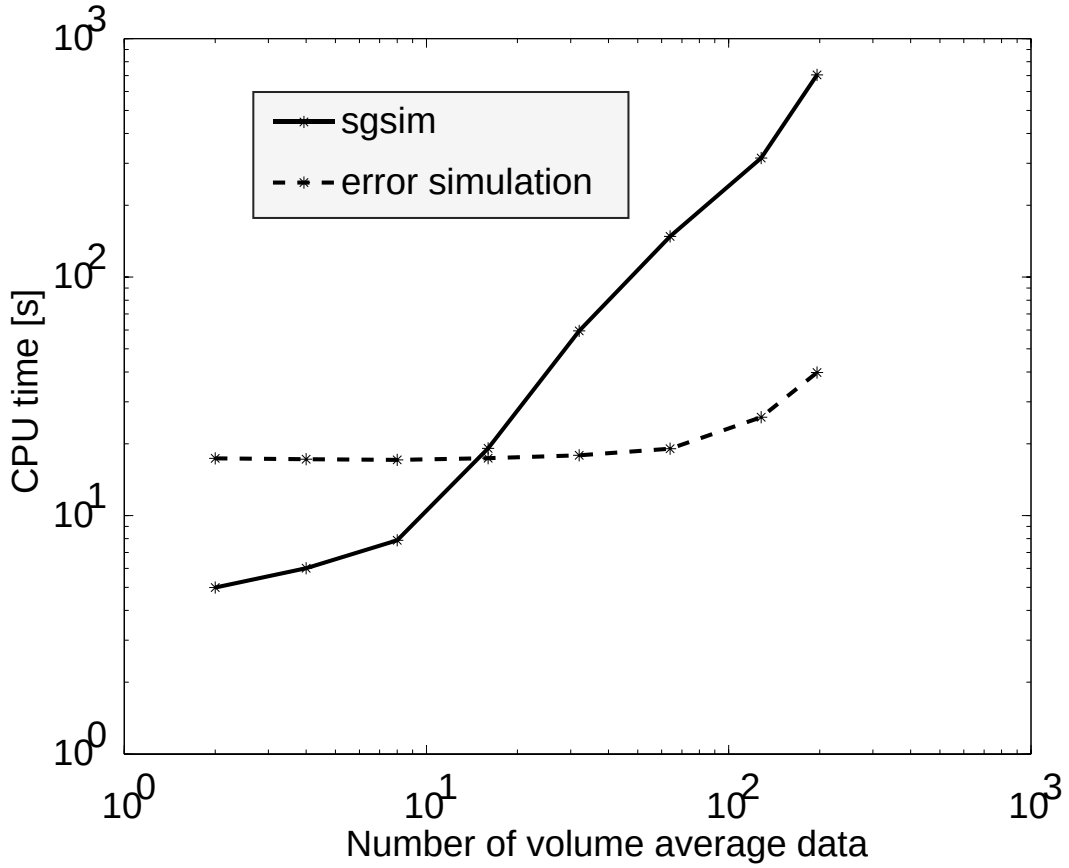

sgsim
error simulation
CPU time [s]
Number of volume average data
10^0
10^1
10^2
10^3

Fig. 14.