

# Comparing the Gradual Deformation with the Probability Perturbation Method for Solving Inverse Problems<sup>1</sup>

Jef Caers<sup>2</sup>

---

*Inverse problems are ubiquitous in the Earth Sciences. Many such problems are ill-posed in the sense that multiple solutions can be found that match the data to be inverted. To impose restrictions on these solutions, a prior distribution of the model parameters is required. In a spatial context this prior model can be as simple as a Multi-Gaussian law with prior covariance matrix, or could come in the form of a complex training image describing the prior statistics of the model parameters. In this paper, two methods for generating inverse solutions constrained to such prior model are compared. The gradual deformation method treats the problem of finding inverse solution as an optimization problem. Using a perturbation mechanism, the gradual deformation method searches (optimizes) in the prior model space for those solutions that match the data to be inverted. The perturbation mechanism guarantees that the prior model statistics are honored. However, it is shown with a simple example that this perturbation method does not necessarily draw accurately samples from a given posterior distribution when the inverse problem is framed within a Bayesian context. On the other hand, the probability perturbation method approaches the inverse problem as a data integration problem. This method explicitly deals with the problem of combining prior probabilities with pre-posterior probabilities derived from the data. It is shown that the sampling properties of the probability perturbation method approach the accuracy of well-known Markov chain Monte Carlo samplers such as the rejection sampler. The paper uses simple examples to illustrate the clear differences between these two methods.*

---

**KEY WORDS:** inverse problems, prior model, geostatistics.

## INTRODUCTION

This paper is intended as a comparison paper between two methods for solving inverse problems under a prior model constraint. One example of such inverse problems is that related to the inversion of flow and pressure data, in the petroleum industry better known as “history matching,” in hydrology also known as “dynamic data” matching. Another prominent example is that of the inversion

---

<sup>1</sup>Received: 12 July 2005; accepted: 21 April 2006; Published online: 21 February 2007.

<sup>2</sup>Energy Resources Engineering Department, Stanford University, Stanford, California 94305, USA; e-mail: jcaers@stanford.edu

of various types of geophysical data, be it of electro-magnetic, gravity or seismic nature.

Inverse problems are generally framed and solved in a Bayesian context, in particular spatial ones, as recognized early by Tarantola (1987) in his book on (mostly linear) inverse theory. The probabilistic framework offered by Bayes' rule allows the modeler to recognize that many inverse problems do not have a unique solution. Therefore a set of solutions are determined, in fact possibly an infinite set, which can be regarded as samples from a probability distribution, termed posterior distribution. In mathematical notation: sets of measurements  $\mathbf{d}$  are used to determine the spatial distribution of a physical attribute; this is described mathematically by a probability model with a set of parameters  $\mathbf{m}$ , to use the same notation as Tarantola. For example, in a reservoir characterization context,  $\mathbf{d}$  could consist of the combined set of seismic and production data and  $\mathbf{m}$  the set of unknown permeability and porosity on a grid that discretizes the reservoir model. In a Bayesian context the posterior distribution (the pdf of inverse solutions) is further decomposed as follows

$$f(\mathbf{m}|\mathbf{d}) = \frac{f(\mathbf{d}|\mathbf{m})f(\mathbf{m})}{f(\mathbf{d})} \quad (1)$$

Solving the inverse problem is then equivalent to drawing samples from this posterior distribution (Omre and Tjelmeland, 1996). The posterior distribution can be sampled once its two constitutive components, the prior  $f(\mathbf{m})$  and likelihood  $f(\mathbf{d}|\mathbf{m})$  are known. The denominator need not be explicitly known for most samplers. While this framework appears straightforward, the method of sampling usually is not and most research consists of designing samplers that are both accurate and efficient in drawing samples from  $f(\mathbf{d}|\mathbf{m})$ . The two methods compared in this paper can be interpreted as samplers in this Bayesian context.

The prior plays the important role of constraining the inverse solutions. Prior to sampling from the posterior a large set of candidate solutions exist: the prior model space. In a Bayesian interpretation, sampling is nothing more than selecting from the prior candidates those solutions that fit the data "up to a certain extent," that extent being determined in a probability model called the likelihood. If the likelihood distribution is a spike at the data  $\mathbf{d}$ , then the data need to be matched exactly. Note that this assumes that the prior contains the inverse solutions; hence the construction of a suitable prior is often an important problem on its own, but not addressed in this paper. Also, in this paper, we will not deal with model or data uncertainty/errors, hence the likelihood function will always be a spike.

In a spatial context that prior set consists of all realizations that follow a certain spatial dependency, as characterized by the spatial law (or random function) that is deemed representative for  $\mathbf{m}$ . The word "deemed" may be important as typically very few data may be available to determine that prior spatial dependency. A prior

model can be analytically defined such as in the case of a multi-Gaussian prior with given covariance matrix (variogram). It can be defined through an algorithm (algorithmically defined priors, Deutsch and Journel, 1998), where an algorithm is used to generate any number of realizations, without the need for an explicit analytical model for  $f(\mathbf{m})$ . The (possibly infinite) set of realizations generated from the algorithm defines the prior model space. Alternatively, the prior can be defined using a Boolean (object-based) model.

The two methods discussed in this paper explicitly deal with the issue of generating inverse solution that attempt to be consistent with a given prior. These two methods are gradual deformation (Hu and Rogerro, 1998; Hu, Blanc, and Noetinger, 2001; Le Ravalec-Dupin et al., 2002) and probability perturbation (Caers, 2003; Hoffman and Caers, 2005; Caers and Hoffman, 2005). Their seemingly similar set-up warrants an in depth comparison paper which will reveal some fundamental theoretical as well as practical differences. The examples shown are simple, but illustrative in nature and designed to show the differences between the two methods. It is not the intent to compare these methods with real data, as both methods have been extensively applied and tested on real cases in other references (see for example, Reis and others, 2000; Hoffman and Caers, 2004; Caers and others, 2006).

## GRADUAL DEFORMATION

### Basic Principle

Several variations exist of what is termed “the gradual deformation method,” but all rely on the same core idea.

The gradual deformation aims at generating gradual, i.e. continuous perturbations of an initial prior model realization (or prior model sample), such that the perturbed realization matches better the data  $\mathbf{d}$ . At the same time any perturbations generated, are still samples from that same prior model. Although the perturbations are continuous or gradual, the variable being perturbed can be discrete/categorical. To achieve this, a well-known property of Gaussian random variables is exploited. Namely, a standard Gaussian random variable  $Y$  can be written as the following linear combination of two other independent standard Gaussian variables  $Y_1$  and  $Y_2$ ,

$$Y = Y_1 \cos r + Y_2 \sin r \quad (2)$$

This is true for any value of  $r$ . Similarly, a standard Gaussian random vector  $\mathbf{Y}$  can be written as a linear combination of two other independent standard Gaussian vectors, sharing the same covariance matrix

$$\mathbf{Y} = \mathbf{Y}_1 \cos r + \mathbf{Y}_2 \sin r \quad (3)$$

The components of  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$  of this vector can be correlated or uncorrelated. If the correlation of the components of  $\mathbf{Y}_1$  is expressed in a variance-covariance matrix  $C$ , and the components of  $\mathbf{Y}_2$  share the same covariance, then the components of  $\mathbf{Y}$  also have the same variance-covariance matrix  $C$ . The parameter  $r$  in the expressions (2) and (3) can be interpreted as a perturbation parameter of the variable  $\mathbf{Y}_1$ . When  $r = 0$ , then  $\mathbf{Y} = \mathbf{Y}_1$ , hence any outcome of  $\mathbf{Y}$  is equal to an outcome of  $\mathbf{Y}_1$ , if  $r$  increases, then any outcome of  $\mathbf{Y}$  will become “gradually” different from the corresponding outcomes of  $\mathbf{Y}_1$ , in the limit case, when  $r = \pi/2$  any outcome of  $\mathbf{Y}$  equals an outcome of  $\mathbf{Y}_2$ . Note that  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$  must be independent random vectors. This basic principle is now used in the construct various, but similar gradual deformation methods.

### **GDLC: Gradual Deformation by Linear Combination of Multi-Gaussian Realizations**

#### *Unconditional Method*

If  $\mathbf{y}_1$  and  $\mathbf{y}_2$  are outcomes of correlated random vectors  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$  and each random vector represents a set of spatially correlated random variables with a positive definite spatial covariance table, then the linear combination  $\mathbf{Y}$  represents a perturbation of  $\mathbf{Y}_1$ , the magnitude of that perturbation being given by  $r$ .  $\mathbf{Y}$  may for example describe the outcomes of the parameters  $\mathbf{m}$  earlier mentioned in the inverse modeling context.

Consider now the problem where one would like to generate an outcome  $\mathbf{y}$  of  $\mathbf{Y}$  that matches a certain data set  $\mathbf{d}$ . The dataset itself can be regarded as an outcome of a random vector  $\mathbf{D}$ . A forward model  $g$  is defined that relates  $\mathbf{Y}$  to  $\mathbf{D}$ ,

$$\mathbf{D} = g(\mathbf{Y})$$

The forward model could represent a physical law linking model parameters and data. That physical law may have a numerical implementation, for example finite difference or finite element model.

An objective function is formulated that expresses the difference between the observed data  $\mathbf{d}$  and the forward modeled or “simulated” data,  $g(\mathbf{y})$ ,

$$O = \|\mathbf{d} - g(\mathbf{y})\|$$

If  $\mathbf{y} = \mathbf{y}(r)$  is parameterized with a perturbation of the type (3) then this objective is a function of  $r$

$$O(r) = \|\mathbf{d} - g(\mathbf{y}(r))\| \tag{4}$$

A one-dimensional optimization problem is solved to find the value of  $r$  that minimizes (4), hence to find the model  $\mathbf{y}(r_{\text{opt}})$  that best matches the data  $\mathbf{d}$ .

A single optimization of  $r$  would not likely result in a model that matches sufficiently the data  $\mathbf{d}$ .

Indeed,  $r$  defines a gradual deformation between any two realizations  $\mathbf{y}_1$  and  $\mathbf{y}_2$ , each one generated with a unique random seed. One can regard the optimization of (4) as a line minimization in a large dimensional space of all possible realizations. A single line minimization is not likely to produce a realization  $\mathbf{y}(r_{\text{opt}})$  that matched the data. Hence, the line minimization is repeated, by selecting a new realization and repeating the optimization of (4). This process is repeated until the data is matched to an acceptable level. The following algorithm describes the entire method:

1. Generate two Gaussian realization  $\mathbf{y}_1$  and  $\mathbf{y}_2$
2. Until the data  $\mathbf{d}$  is matched
  - a. Solve an optimization problem in  $r$

$$r_{\text{opt}} = \min_r \{O(r) = \|\mathbf{d} - g(\mathbf{y}_1 \cos r + \mathbf{y}_2 \sin r)\|\}$$

b. Set

- i.  $\mathbf{y}_1 \leftarrow \mathbf{y}(r_{\text{opt}})$
- ii. Draw a new realization  $\mathbf{y}$  (new seed)
- iii.  $\mathbf{y}_2 \leftarrow \mathbf{y}$

Note that a histogram transformation is needed to transform standard Gaussian realization into realizations that honor the marginal distribution of the prior model.

### *Conditional Method*

A method for conditioning to hard data is proposed in Ying and Gomez (2000) by linearly combining three realizations

$$\mathbf{y} = \alpha_1 \mathbf{y}_1 + \alpha_2 \mathbf{y}_2 + \alpha_3 \mathbf{y}_3$$

Under the joint conditions

$$\alpha_1 + \alpha_2 + \alpha_3 = 1$$

$$\alpha_1^2 + \alpha_2^2 + \alpha_3^2 = 1$$

This guarantees that the resulting model  $\mathbf{y}$  honors the hard data as well as the variogram. The problem can easily be re-parameterized to a single parameter  $r$

recognizing the following

$$\begin{aligned}\alpha_1 &= \frac{1}{3} + \frac{2}{3} \cos r \\ \alpha_2 &= \frac{1}{3} + \frac{2}{3} \sin \left( -\frac{\pi}{6} + r \right) \\ \alpha_3 &= \frac{1}{3} + \frac{2}{3} \sin \left( -\frac{\pi}{6} - r \right)\end{aligned}$$

Suzuki (2004) shows that this method is also applicable to realizations  $\mathbf{y}$  generated using direct sequential simulation, hence non-Gaussian models.

### **FFT-MA: Gradual Deformation on the FFT Moving Average Generator**

#### *Method*

Instead of directly combining multi-Gaussian realizations, one can also combine uncorrelated Gaussian realization (“Gaussian Noise”), then linearly transform this uncorrelated noise into a correlated Gaussian realization. A simple way to achieve this is the LU decomposition (Alabert, 1987). Consider the decomposition of a covariance matrix  $C$  into an upper and lower triangular matrix

$$C = LU$$

Then an uncorrelated Gaussian realization  $\mathbf{y}$  (Gaussian noise) can be linearly transformed into a correlated one using

$$\mathbf{z} = \mathbf{m} + L \mathbf{y}$$

Hence, by perturbing the noise  $\mathbf{y}$  using (3), one obtains a perturbation of  $\mathbf{z}$ . This method has several advantages over the above method

1. Conditioning with kriging: conditioning to hard data can be performed by generating unconditional realizations  $\mathbf{z}$ , then perform the conditioning with kriging (Journel and Huijbregts, 1978).
2. Multi-parameter perturbation: the perturbation can be parameterized with more parameters, as explained further on.

The LU-decomposition approach is however not applicable to large dimensions of  $\mathbf{Y}$  or  $\mathbf{Z}$ . In that case the LU decomposition of the covariance matrix  $C$  becomes CPU prohibitive. One solution, via a sequential decomposition of the multi-variate distribution (sequential simulation) will be discussed next. Another

solution using Fourier transform is proposed in Le Ravalec and others (2002) using a moving average method of Oliver (1995). Instead of the LU decomposition of a covariance matrix, a covariance function  $C(\mathbf{h})$  is considered as a convolution product of a function  $g(\mathbf{h})$  and its transpose  $g(-\mathbf{h})$

$$C(\mathbf{h}) = g \otimes \tilde{g}$$

If this function  $g$  is available, then a Gaussian random vector can be written as

$$\mathbf{z} = \mathbf{m} + g \otimes \mathbf{y}$$

Based on Oliver (1995), Le Ravalec-Dupin and others (2000) show how in practice this function can be derived from the Fourier transform of the covariance function (the power spectrum), namely

$$G(f) \tilde{G}(f) = S(f) \quad \text{for example take } G(f) = \sqrt{S(f)}$$

with  $S$ , the power spectrum. In short the algorithm works as follows

1. Sample the covariance function  $C$
2. Generate a realization  $\mathbf{y}$
3. Calculate the FFT of  $C$ , termed  $S$  and of  $\mathbf{y}$ , termed  $\mathbf{y}_f$
4. Derive  $G$  from  $S$
5. Multiply  $G$  with  $\mathbf{y}_f$
6. Calculate the inverse FFT of this product
7. Add the mean  $\mathbf{m}$ , to get  $\mathbf{z}$

$\mathbf{y}$  can now be perturbed using Eq. (3), which leads to a perturbation of  $\mathbf{z}$ .

### *Multi-parameter Gradual Deformation*

The advantage of the FFT-MA over the GDLC is not immediately apparent, unless one considers a multi-parameter gradual deformation method. In many practical cases, the use of a single perturbation parameter  $r$  to define the perturbation will be insufficient to solve the inverse problem. Indeed, a single  $r$ , constant for all grid locations gives the same perturbation, at least in expected value, everywhere. This may work for simple inverse problems, with not too complex forward model and low dimensional  $\mathbf{d}$ . In more complicated cases, more flexibility in the parameterization of the perturbation may be required, in the least, the perturbation should be spatially varying, with some locations being perturbed more than others. In the FFT-MA, a multi-parameter perturbation can be constructed as follows. Consider the Gaussian noise  $\mathbf{Y} = \{Y(\mathbf{u}_1), Y(\mathbf{u}_2), \dots, Y(\mathbf{u}_N)\}$ , each component  $Y(\mathbf{u})$  being

attached to a location  $\mathbf{u}$  in space. Instead of using a single  $r$ , one can vary  $r$  by location

$$\mathbf{Y} = \mathbf{Y}_1 \mathbf{r}_{\cos}^T + \mathbf{Y}_2 \mathbf{r}_{\sin}^T$$

with  $\mathbf{r}_{\cos} = \{\cos r(\mathbf{u}_1), \cos r(\mathbf{u}_2), \dots, \cos r(\mathbf{u}_N)\}$  and  $\mathbf{r}_{\sin} = \{\sin r(\mathbf{u}_1), \sin r(\mathbf{u}_2), \dots, \sin r(\mathbf{u}_N)\}$ . Note that  $\mathbf{Y}$  is still Gaussian noise, hence any FFT-MA generated  $\mathbf{z}$  will still be Gaussian and the covariance structure will be preserved even in this multi-parameter case. Not all  $r$  values need be different for each location  $\mathbf{u}$ . Typically, one creates regions in space, with all locations  $\mathbf{u}$  in that region being assigned the same  $r$ -value. Alternatively, one can interpolate a set of “pilot-point”  $r$ -values using kriging (Hoffman, 2005). The problem with this extension is that one has to solve a multi-dimensional optimization problem. Gradients can be calculated (Hu and Le Ravalec-Dupin, 2004) to make this optimization somewhat efficient, but evidently, the efficiency and simplicity of the single parameter method is lost. A simple yet effective solution to this problem was proposed in Hoffman and Caers (2005), in the context of the probability perturbation method and will be discussed later on.

In the GDLC method a higher dimensional parameterization cannot be achieved. Linearly combining two realizations, with different  $r$ -values at different locations may lead to spatial discontinuities, moreover, the final combined realizations are no longer multi-Gaussian, and there is no guarantee that the variogram reproduction is maintained.

### GDSS: Gradual Deformation of Sequential Simulations

Sequential simulation is a popular approach for generating conditional realizations. Sequential simulation can accommodate various prior models, such as a Multi-Gaussian but also more flexible models such as sequential indicator simulation for simulating categorical phenomena. Recently (Strebelle, 2002), a new sequential simulation algorithm, *snesim*, was proposed to simulate spatial structures borrowed from training images.

Any sequential simulation algorithm consists of

1. A random path visiting each location to be simulated
2. A method for modeling the local conditioning distribution (ccdf) of the variable at the visited location given any hard data and previously simulated values
3. Random numbers for drawing from these local ccdfs.

In the Multi-Gaussian case, sequential simulation is equivalent to LU decomposition if at each location to be simulated a global kriging neighborhood is used.

However, this would render, sequential simulation impractical for large dimensions of  $\mathbf{Y}$ . Therefore, sequential simulation uses a limited search neighborhood when simulating each visited location. Hu and others (2001), proposed a perturbation of sequential perturbation by means of a perturbation of the random numbers used to draw from local conditional distributions. This is achieved through Eq. (3) as follows. Consider a vector of random numbers  $\mathbf{v}$ , this vector contains the random numbers that are used to draw from the local ccdfs. This vector can be transformed into Gaussian noise, using the inverse cumulative Gaussian distribution of  $G$ ,

$$G^{-1}(\mathbf{v}) = \mathbf{y}$$

Next  $\mathbf{y}$  is perturbed using Eq. (3), resulting in  $\mathbf{y}(r)$ . This perturbed Gaussian vector is back-transformed into a uniform vector of perturbed random numbers.

$$\mathbf{v}(r) = G(\mathbf{y}(r))$$

This vector  $\mathbf{v}(r)$  is a perturbation of  $\mathbf{v}$ , hence when  $\mathbf{v}(r)$  is used in sequential simulation instead of  $\mathbf{v}$ , a perturbation is achieved.

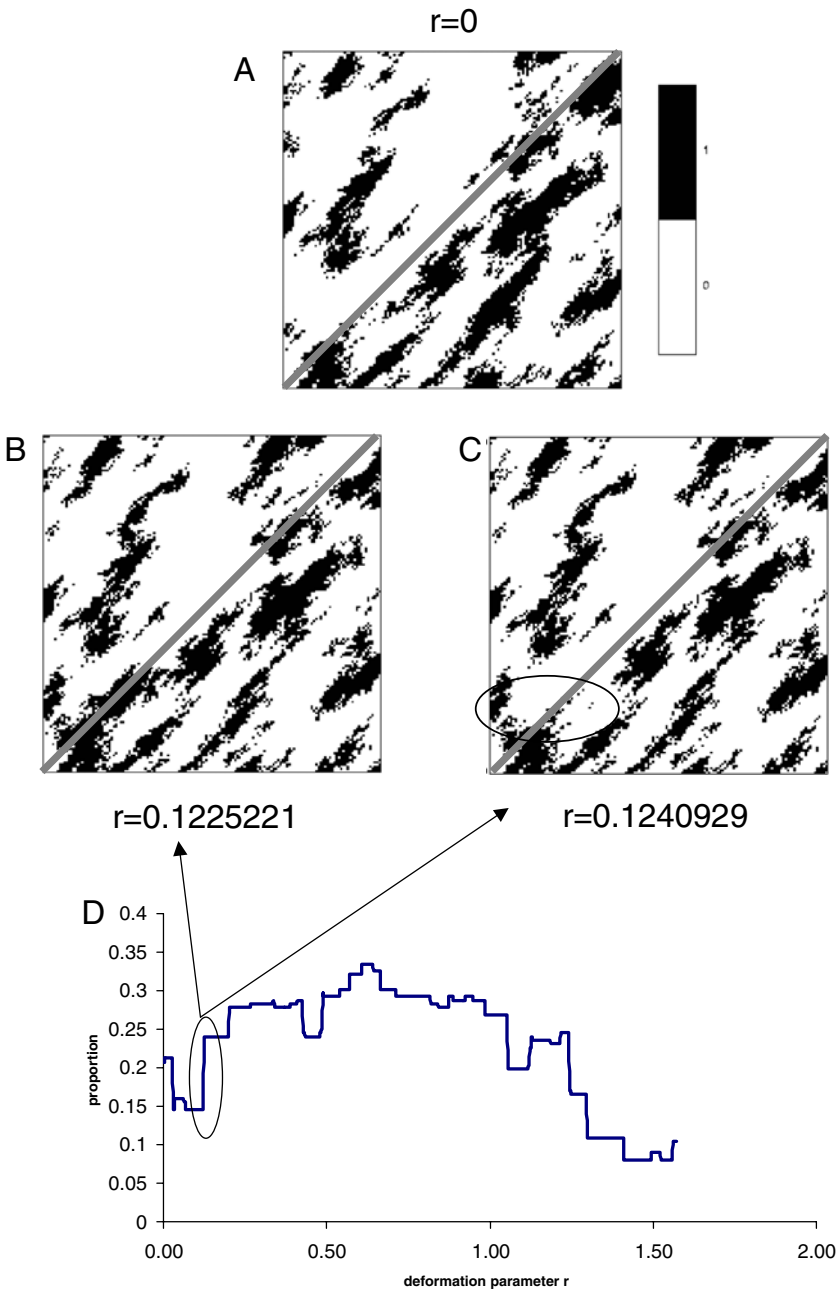
A multi-parameter method can be constructed similar to the FFT-MA method.

### How Gradual is Gradual Deformation?

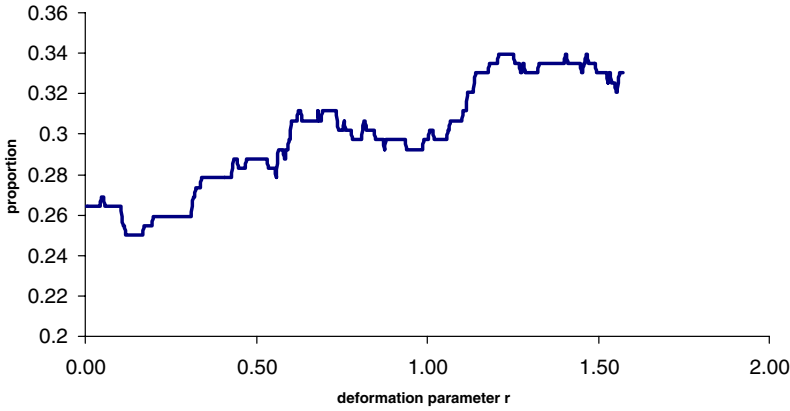
Consider the following simple example. A sequential indicator simulation generates a binary realization on a  $150 \times 150$  grid. The variogram of the binary variable has NE-SW anisotropy with major range equal to 30 and minor range equal to 10. There is 15% nugget effect in the variogram. The proportion of pixels with value '1' in Figure 1A, B, C is 60%. A single realization is generated shown in Figure 1A. The value of  $r$  is discretized between  $[0, \pi/2]$  into one thousand values. For each such  $r$ -value a new gradually deformed realization is generated, hence a total of one thousand realizations are generated. For each realization, the proportion of '1' pixels along the diagonal of each realization is calculated. We evaluate the behavior of this proportion as the realizations are perturbed using the GDSS method.

Figure 1D shows the proportion plotted against the value of  $r$ , clearly the change is not gradual, large jumps of 10% occur for very small changes in  $r$ . Upon further inspection of one such discontinuity shown in Figure 1B, C, we observe that the sequential simulation result changes dramatically when going from  $r = 0.1225221$  to  $r = 0.1240929$ .

Realization of a discrete variable, generated using sequential simulation algorithms cannot be perturbed in a gradual way, simply because of the sequential nature of the algorithm. A small change in the random numbers used to simulate



**Figure 1.** (A) Reference initial realization, (B) perturbation with  $r = 0.1225221$ , (C)  $r = 0.1240929$ , (D) proportion of black pixels along 45 degree line as function of  $r$ .



**Figure 2.** Gradual deformation using a truncated Gaussian method, notice how the scale is somewhat different from Figure 1.

the initial values along the random path could cascade into large changes along the way.

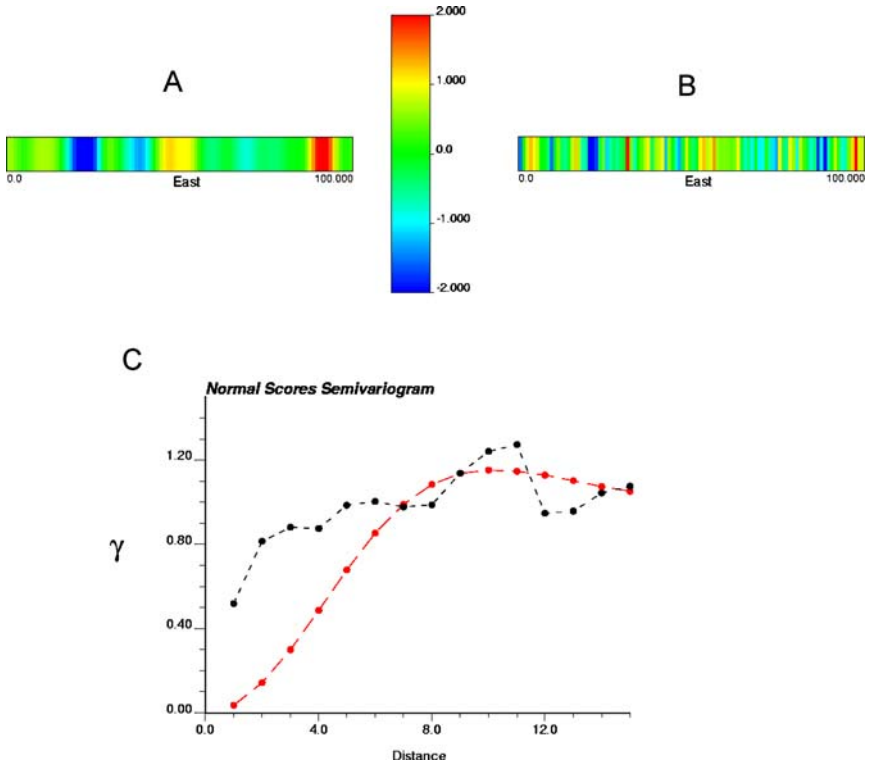
Instead, an alternative approach is advised for this particular example, using truncated Gaussian simulation. Truncated Gaussian simulation relies on the generation of a discrete variable by truncating/thresholding a continuous variable. The underlying Gaussian field need not be generated using sequential simulation. A similar setup as above is followed, except that truncated Gaussian, instead of sequential indicator simulation is used. A variation of the proportion as function of  $r$  is calculated exhaustively and shown in Figure 2, the deformation is more gradual, the large jumps of 10% have been eliminated, although there are still small scale irregularities, mainly attributed to the discrete nature of the proportion function.

This experiment shows that a gradual deformation is only truly gradual when the resulting realizations are derived from a correlated Gaussian realization, such as in sequential Gaussian simulation, or as in truncated and pluri-Gaussian simulation for the discrete case. In all other cases, such as in sequential indicator simulation, *snesim* or even in gradual deformation of Boolean models the deformation, in terms of the objective function is not gradual.

## Some Theoretical Issues

### *Preservation of Prior Model Statistics*

Hu and Le Ravalec-Dupin (2004) reported a problem with the gradual deformation method in terms of reproducing the prior model statistics. Consider the

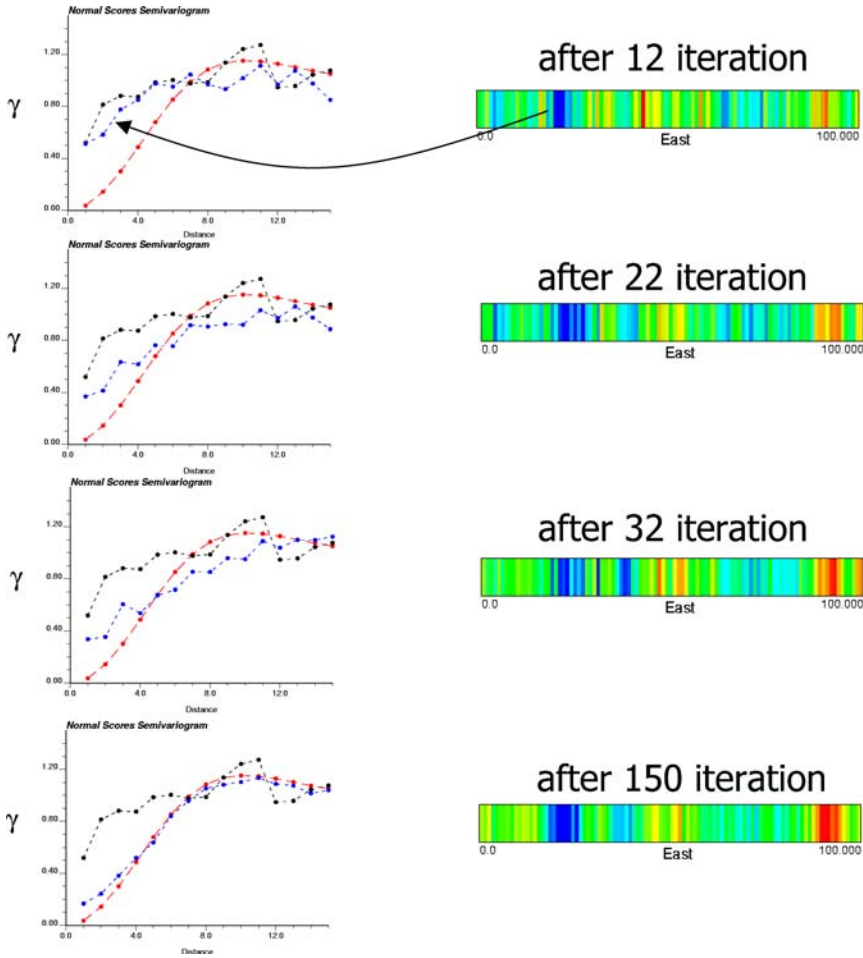


**Figure 3.** (A) Target model with Gaussian variogram, (B) initial realization of prior model, (C) variograms of realizations (A) and (B).

following simple inverse problem. In Figure 3A, a 1D realization of a multi-Gaussian distribution with Gaussian variogram is given. Denote this data as  $\mathbf{d} = \{d(\mathbf{u}_1), d(\mathbf{u}_2), \dots, d(\mathbf{u}_N)\}$ . The inverse problem consist of finding a realization  $\mathbf{y} = \{y(\mathbf{u}_1), y(\mathbf{u}_2), \dots, y(\mathbf{u}_N)\}$  that matches the data  $\mathbf{d}$ , i.e. minimizing the objective function

$$O = \sum_{i=1}^N (d(\mathbf{u}_i) - y(\mathbf{u}_i))^2$$

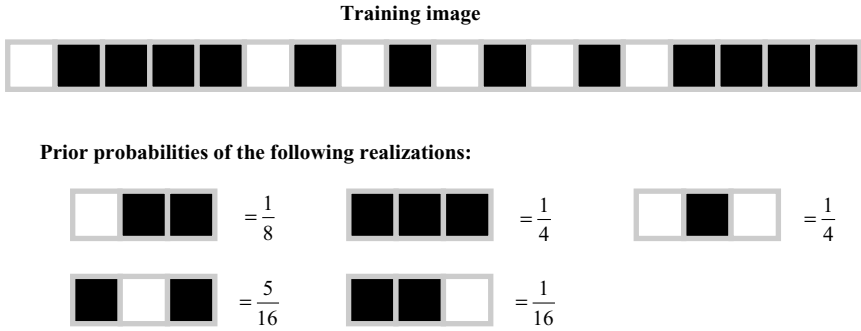
The prior model of  $\mathbf{Y}$  is also Multi-Gaussian but has an exponential variogram. In other words, one is looking for a realization with exponential variogram that matches best the given realization with Gaussian variogram, Figure 3B shows a single realization and Figure 3C shows the difference between the variogram of the data and the prior model. The GDLC is applied to find such realization. Figure 4



**Figure 4.** Evolution of the prior model statistics during iteration.

shows the progression of the gradual deformation. Clearly the exponential nature of the variogram is not preserved, the final realization has a Gaussian variogram.

The reason for this problem lies in Eq. (2). In an  $N$ -dimensional problem,  $N$  being the dimension of  $\mathbf{Y}$ , there can only be constructed  $N$  linearly independent realizations, in other words, any  $(N + 1)^{\text{th}}$  realization can be written as a linear combination of the  $N$  previous ones. This mean that the linear combination of Eq. (2) is no longer Gaussian, since Eq. (2) requires the combination of vectors are drawn independently from each other.



**Figure 5.** Training image and prior probabilities derived from it.

In most practical cases  $N$  is large and this problem may be avoided, although, it should be noted that the problem occurs much earlier than after  $N$  linear combinations. Figure 4 shows that the degradation in variogram reproduction occurs at 22 iteration, where in this case  $N = 100$ .

### *Accuracy of the Gradual Deformation Sampler*

The following problem is related to the sampling properties of the gradual deformation method. As stated in the introduction, samplers should be efficient and should draw samples from the posterior distribution accurately. We setup a simple example with a given posterior model and assess how well the gradual deformation samples approach that posterior.

Consider a simple inverse problem. The model consists of a three parameter each representing a binary variable of a  $3 \times 1$  one-dimensional field. The binary variable at each location can be either black,  $I(\mathbf{u}) = 1$  or white,  $I(\mathbf{u}) = 0$ . The model  $\mathbf{m}$  is therefore simply  $\{I(\mathbf{u}_1), I(\mathbf{u}_2), I(\mathbf{u}_3)\}$ .

The spatial dependency is described using a 1D training image shown in Figure 5. The prior distribution of the model parameters is extracted, by scanning the training image with a  $3 \times 1$  template, and shown in Figure 5. The data  $\mathbf{d}$  consists of two parts: the first part is a point measurement namely,  $i(\mathbf{u}_2) = 1$  (a black pixel in the middle), in other words, every solution needs to have a black cell in the middle. The second part of the data is that  $I(\mathbf{u}_1) + I(\mathbf{u}_2) + I(\mathbf{u}_3) = 2$ . The question is not so much what are possible solutions as those are obviously

Solution 1:  $\{I(\mathbf{u}_1) = 0, I(\mathbf{u}_2) = 1, I(\mathbf{u}_3) = 1\}$

Solution 2:  $\{I(\mathbf{u}_1) = 1, I(\mathbf{u}_2) = 1, I(\mathbf{u}_3) = 0\}$

Rather, the question is: what is the posterior probability of these two solutions. For this simple problem, the posterior is can be calculated by simple elimination

on the prior models:

$$\Pr\{I(\mathbf{u}_1) = 0, I(\mathbf{u}_2) = 1, I(\mathbf{u}_3) = 1|\mathbf{d}\} = 1 - \Pr\{I(\mathbf{u}_1) = 1, I(\mathbf{u}_2) = 1, I(\mathbf{u}_3) = 0|\mathbf{d}\} = \frac{2}{3}$$

To estimate the posterior using gradual deformation with sequential simulation, a Monte-Carlo experiment is performed. Multiple realization are generated, each obtained using iterative gradual deformation method; the probability is estimated simply by observing the frequency of the solution 1. The result is

$$\hat{\Pr}_{\text{GDSS}}\{I(\mathbf{u}_1) = 0, I(\mathbf{u}_2) = 1, I(\mathbf{u}_3) = 1|\mathbf{d}\} \simeq 0.73$$

The posterior probability of the gradual deformation method is significantly different from the posterior probability defined by the model. This is likely true for most samplers. In reality most samplers, including the MCMC (Markov chain Monte Carlo) samplers such as the Metropolis sampler, only approximately sample the space defined by the model. In real cases, sampling accurately the posterior model is often too prohibitive CPU-wise. The problem with the gradual deformation method is that is not clear where that approximation lies and how one can correct for it.

This result on the inaccuracy of the gradual deformation sampler contradicts the findings of Liu and Oliver (2004). These authors verified experimentally based on a single flow problem with a Gaussian permeability field and a Gaussian likelihood function, that the sampling of gradual deformation is indeed a reasonable approximation to the exact rejection sampling (MCMC). The example shown here is quite different (discrete, non-Gaussian), suggesting that the conclusions of Liu and Oliver cannot be extended to non-Gaussian cases.

## THE PROBABILITY PERTURBATION METHOD

The probability perturbation method (PPM) takes a somewhat different approach to solving the inverse problem and has currently been applied only in the context of prior model realizations generated using sequential simulation. We will provide first an intuitive understanding of the method, and then review how this method frames within the Bayesian inverse modeling context. Similarities and difference will be explained along the way.

### *Methodology for a Binary Variable*

Consider an initial realization of a binary variable  $I(\mathbf{u})$  generated using a sequential simulation algorithm (e.g. sisim, snesim). A random seed uniquely

determines the random path and set of random numbers used in drawing from the local ccdfs. In short this ccdf is written as  $\Pr(A|\mathbf{B} = \mathbf{b})$  where  $A = \{I(\mathbf{u}) = 1\}$  and  $\mathbf{b}$  is the set of hard data and previously simulated values.

To generate a perturbation of  $i^0(\mathbf{u})$ , one considers a new probability, defined based on the initial realization, the marginal and a free parameter  $r$ , whose role will be explained later

$$\Pr\{I(\mathbf{u}) = 1|\mathbf{D} = \mathbf{d}\} = \Pr\{A|\mathbf{D}\} = (1 - r)i^0(\mathbf{u}) + r P(A) \quad (5)$$

Now, the sequential simulation is repeated, but with different random seed. At each node to be simulated one draws from a new probability  $P(A|B, D)$  that depends on  $P(A|B)$ , the (recalculated) local ccdf and the probability  $P(A|D)$  in Eq. (5) through the relationship

$$\Pr(A|\mathbf{B}, \mathbf{D}) = \frac{1}{1 + x} \text{ with}$$

$$\frac{x}{a} = \left(\frac{b}{a}\right)^{\tau_1} \left(\frac{d}{a}\right)^{\tau_2}$$

where :

$$(6)$$

$$b = \frac{1 - \Pr(A|\mathbf{B})}{\Pr(A|\mathbf{B})}, d = \frac{1 - \Pr(A|\mathbf{D})}{\Pr(A|\mathbf{D})}, a = \frac{1 - \Pr(A)}{\Pr(A)}$$

This model for  $\Pr(A|\mathbf{B}, \mathbf{D})$  is also known as the tau-model (Journel, 2002). The tau-model models the redundancy, using the parameters  $\tau_1$  and  $\tau_2$ , between two data  $\mathbf{b}$  and  $\mathbf{d}$  whose individual dependence with  $A$  is expressed through the pre-posterior distributions  $\Pr(A|\mathbf{B})$  and  $\Pr(A|\mathbf{D})$ . The new realization generated,  $i(\mathbf{u}, r)$ , depends on the value of  $r$ . In the same sense as for the gradual deformation method,  $r$  represents a perturbation of the initial realization, where in the limit case

- when  $r = 0$ , then  $\Pr(A|\mathbf{D}) = i^0(\mathbf{u})$ , hence  $\Pr(A|\mathbf{B}, \mathbf{D}) = i^0(\mathbf{u})$ , per Eq. (6) the initial realization is regenerated: no perturbation;
- when  $r = 1$ , then  $\Pr(A|\mathbf{D}) = \Pr(A)$ , hence  $\Pr(A|\mathbf{B}, \mathbf{D}) = \Pr(A|\mathbf{B})$ , per Eq. (6), and since the seed has changed, another realization from the prior is generated.

$r$  defines a perturbation between two realizations, each with a different random seed. This seems at first sight similar to the gradual deformation method, but some important difference should be noted.

Compared to GDLC: GDLC defines a perturbation between two realizations generated with different seed, but, the realization need to be Gaussian nor does GDLC allow a multi-parameter perturbation. PPM works for non-Gaussian models, and allows a multi-parameter perturbation as explained further.

Compared to GDSS: GDSS perturbs realizations generated using sequential simulation, however the random path needs to be fixed in GDSS, otherwise the perturbations would be purely random. This has the disadvantage that the perturbations are only small, since changing the random path results in a significant change in the simulated realization. This may slow the convergence of the method.

The following algorithm describes fully the PPM for the binary case.

1. Generate an initial realization  $i^0(\mathbf{u})$  with seed  $s$
2. Until data is matched
  - a. Change random seed
  - b. Until an optimal value of  $r$  is found
    - i. Set  $r$
    - ii. Calculate  $P(A|\mathbf{D})$
    - iii. Generate a new realization  $i(\mathbf{u}, r)$ ,
    - iv. Evaluate objective function as function of  $i(\mathbf{u}, r)$
  - c. Set  $i(\mathbf{u}, r_{\text{opt}})$  equal to  $i^0(\mathbf{u})$

The algorithm is called a probability perturbation, because the perturbation is generated by perturbing the probability  $\Pr(A|\mathbf{B})$  using another probability,  $\Pr(A|\mathbf{D})$ , that probability dependent on the data  $\mathbf{d}$ .

### *Methodology for Multi-category Variables*

The multi-category method is simply an extension of the binary case. Consider a multi-category random variable  $S(\mathbf{u})$  at every location  $\mathbf{u}$  that can take  $K$  possible classes  $s_k$  as outcomes, then for each class one defines an indicator random variable  $I(\mathbf{u}, s_k) k = 1 \dots K$ . An initial realization of multiple indicators  $i^0(\mathbf{u}), k = 1, \dots, K$  is available. The perturbation is achieved using the probability

$$\Pr\{I(\mathbf{u}, s_k) = 1 | \mathbf{D} = \mathbf{d}\} = \Pr\{A_k | \mathbf{D}\} = (1 - r) i^0(\mathbf{u}, s_k) + r P(A_k)$$

Note that

$$0 \leq \Pr\{A_k | \mathbf{D}\} \leq 1 \text{ and } \sum_{k=1}^K \Pr\{A_k | \mathbf{D}\} = 1$$

### *Multi-parameter PPM Method*

Similar to the gradual deformation method is the design of a multi-parameter PPM. This is achieved by making  $r$  spatially dependent.

$$\Pr\{I(\mathbf{u}, s_k) = 1 | \mathbf{D} = \mathbf{d}\} = \Pr\{A_k | \mathbf{D}\} = (1 - r(\mathbf{u})) i^0(\mathbf{u}, s_k) + r(\mathbf{u}) P(A_k)$$

*Methodology for Continuous Variables*

In the case of a continuous variable, one may opt to bin the variable in a large amount of classes and apply the above multi-parameter PPM. A simpler, more efficient and equivalent alternative presents itself using a co-kriging method.

Consider a continuous vector  $\mathbf{Y} = Y(\mathbf{u}_1), Y(\mathbf{u}_2), \dots, Y(\mathbf{u}_N)$  and consider a prior model of  $\mathbf{Y}$  that is fully determined by the correlogram of  $\mathbf{Y}$ , for simplicity and without lack of generality, assume each  $Y(\mathbf{u})$  has zero mean and unit variance. One such model could be the multi-Gaussian. Another such model could be defined algorithmically using the direct sequential simulation method, which produces realizations that are non multi-Gaussian, but are ensured to honor a certain variogram (Journel, 1993). In either case, to obtain a perturbation of an initial realization  $y^0(\mathbf{u})$  one would perform a sequential simulation, where the local distribution is defined using the mean and simple kriging variance using the following estimator

$$y_{SK}^*(\mathbf{u}) = \sum_{\alpha=1}^N \lambda_{\alpha} y(\mathbf{u}_{\alpha}) + \lambda_{\alpha} y^0(\mathbf{u})$$

where  $y^0(\mathbf{u})$  is an initial realization of that prior model. To calculate the kriging weights, the correlogram between  $y^0(\mathbf{u})$  and  $y(\mathbf{u})$  is set equal to

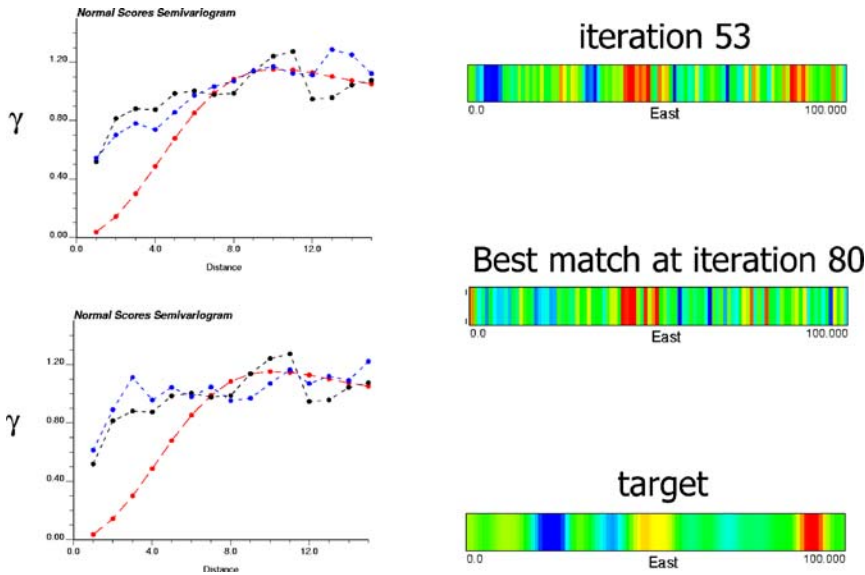
$$\rho_{YY^o} = (1 - r) \rho_Y$$

where  $r$  plays the role of the perturbation parameter, similar to all methods presented above. Discussing the limit case provides an understanding about this choice and on how the perturbation occurs:

When  $r = 0$ , then  $\rho_{YY^o} = \rho_Y$ , in other words the initial realization is considered as “hard data,” hence it will be exactly reproduced.

When  $r = 1$ , then  $\rho_{YY^o} = 0$ , in other words, the initial realization will be ignored in the kriging since  $\lambda_o = 0$  and another realization is generated.

In geostatistical jargon, the initial realization is used as soft data in the generation of a perturbed realization, the degree of “softness” given by  $r$ . The resulting perturbed realization will always reproduce the variogram. This is simply due to the fact that the soft data used has a correlogram equal to the variable being simulated and a cross-correlogram that is proportional to it (Journel, 1999). The perturbation of the local ccdf is now achieved through the kriging system by a soft data set whose correlation is made dependent on the data  $\mathbf{d}$  through the perturbation parameter  $r$ . At first glance, the continuous method appears different from the previously described categorical case which involved the tau-model (Eq. (6)). However, considering that the tau-parameters in Eq. (6) explicitly model the data-redundancy between the  $\mathbf{B}$  and  $\mathbf{D}$  data in estimating  $A$ , such redundancy in traditional geostatistics is modeled using a cross-variogram. In co-located



**Figure 6.** Results for the PPM.

co-kriging the cross-variogram becomes a simple function of the correlation-coefficient (Journel, 1999), hence the correlation-coefficient can be regarded as a way to express data redundancy.

The same example as Figure 1 is repeated using the probability perturbation method for continuous variables. To make the comparison complete, a rejection sampler as follows:

1. Generate a realization from the prior
2. Calculate the mismatch
3. Mark that realization as the current best
4. Start iteration:
  - a. Generate a realization from the prior
  - b. Calculate the mismatch
  - c. If better than the previous, keep it as the current best

Figure 6 shows a resulting realization generated as such, the best match was obtained after 80 iterations, no further improvement was found by iterating beyond this point. The variogram of the resulting realization is similar to the prior model.

The convergence behavior of the rejection sampler and PPM are similar, as shown in Figure 7. Unlike gradual deformation, the PPM does not suffer from any degradation of the prior model statistics as shown in Figure 7.

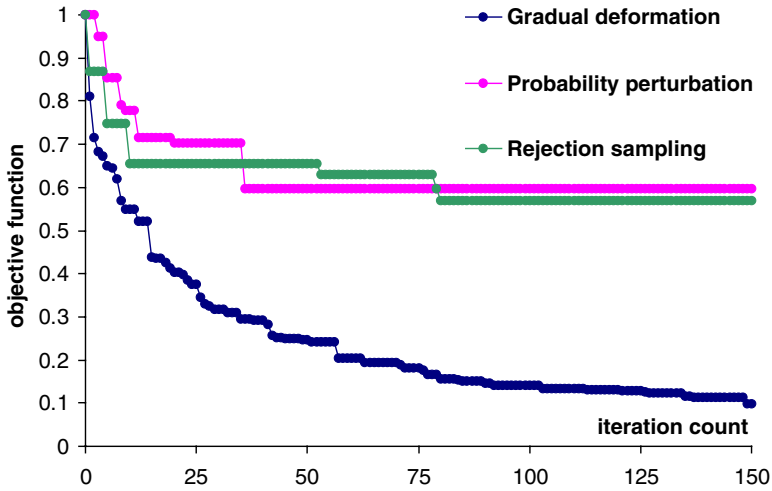


Figure 7. Convergence behavior of the objective function.

In many ways, the PPM works like a rejection sampler, but is more efficient, as will be tested further on. To understand this better, we first require framing the PPM with Bayesian inverse modeling.

### PPM within Bayesian Inversion

#### *Bayes' Framework*

Bayesian inversion relies on the decomposition of the posterior into prior and likelihood as given by Eq. (1). It was pointed out in Caers and Hoffman (2006) that the PPM differs from traditional iterative (MCMC) samplers in two main areas:

- The prior model is sampled using sequential simulation. This allows the definition of a rich variety of priors, both analytically and algorithmically defined. Moreover sequential simulation is non-iterative, robust and efficient
- The explicit specification of likelihood is avoided, instead the method works with pre-posterior distribution as explained next.

We review shortly the main points of the Caers and Hoffman paper and revisit the examples used in discussing the accuracy of the gradual deformation method.

*Sampling the Prior*

As discussed above, in sequential simulation, the prior  $f(\mathbf{m})$  is sampled using a sequential scheme of conditional distributions. Consider without lack of generality the sampling of a binary variable  $I(\mathbf{u})$ . The sequential decomposition is written as

$$\begin{aligned} f(\mathbf{m}) &= \Pr\{I(\mathbf{u}_1) = i(\mathbf{u}_1), I(\mathbf{u}_2) = i(\mathbf{u}_2), \dots, I(\mathbf{u}_N) = i(\mathbf{u}_N)\} \\ &= \Pr\{I(\mathbf{u}_1) = i(\mathbf{u}_1)\} \times \\ &\quad \Pr\{I(\mathbf{u}_2) = i(\mathbf{u}_2)|i(\mathbf{u}_1)\} \times \dots \times \\ &\quad \Pr\{I(\mathbf{u}_N) = i(\mathbf{u}_N)|i(\mathbf{u}_1), \dots, i(\mathbf{u}_{N-1})\} \end{aligned}$$

Sequential methods allow the additional conditioning of direct information (hard data) or indirect information (soft data) through some form of co-kriging or simple prior calibration.

*Sampling the Posterior*

The relation between data  $\mathbf{d}$  and model  $\mathbf{m}$  is given by the forward model  $g$

$$\mathbf{d} = g(\mathbf{m}) = g(I(\mathbf{u}_1), I(\mathbf{u}_2), \dots, I(\mathbf{u}_N))$$

In sampling the posterior distribution, we again rely on the sequential decomposition in conditional distributions

$$\begin{aligned} f(\mathbf{m}|\mathbf{d}) &= \Pr\{I(\mathbf{u}_1) = i(\mathbf{u}_1)|\mathbf{d}\} \times \\ &\quad \Pr\{I(\mathbf{u}_2) = i(\mathbf{u}_2)|i(\mathbf{u}_1), \mathbf{d}\} \times \dots \times \\ &\quad \Pr\{I(\mathbf{u}_N) = i(\mathbf{u}_N)|i(\mathbf{u}_1), \dots, i(\mathbf{u}_{N-1}), \mathbf{d}\} \end{aligned}$$

where each conditional distribution is of the general form:

$$\begin{aligned} \Pr\{I(\mathbf{u}_j) = 1|i(\mathbf{u}_1), \dots, i(\mathbf{u}_{j-1}), \mathbf{D} = \mathbf{d}\} &= \Pr(A_j|\mathbf{B}_j, \mathbf{D}) \\ \text{with } A_j &= \{I(\mathbf{u}_j) = i(\mathbf{u}_j)\} \\ \mathbf{B}_j &= \{i(\mathbf{u}_1), \dots, i(\mathbf{u}_{j-1})\} \\ \mathbf{D} &= \mathbf{d} \end{aligned}$$

The PPM relies on a further decomposition of  $\Pr(A_j|\mathbf{B}_j, \mathbf{C})$  into two pre-posterior probabilities,  $\Pr(A_j|\mathbf{B}_j)$  and  $\Pr(A_j|\mathbf{D})$ . The first term is modeled through sequential

simulation, the second one is a function of the data  $\mathbf{d}$  through the model (5) which contains the free parameter  $r$ . The two pre-posterior distributions are combined using the tau-model of Eq. (6).

The PPM is therefore a sampler of a posterior probability  $f(\mathbf{m}|\mathbf{d})$ . In fact the model definition of  $f(\mathbf{m}|\mathbf{d})$  and the sampler of  $f(\mathbf{m}|\mathbf{d})$  are intertwined in the PPM. This makes it different from traditional Bayesian methods, where one first states the model, then samples the model defined using MCMC. The PPM method recognizes that the full posterior is never measured and always needs to be assumed. The most one can know about the posterior is that the model realizations need to match some known (or assumed) prior model statistics as defined in the prior and it needs to match the data  $\mathbf{d}$ . Nevertheless, the question begs how well PPM performs if the posterior is assumed to be known.

### *Accuracy of the PPM Sampler*

We revisit using the PPM, the example in Figure 5 in the context of testing the accuracy of the sampler. We set both tau parameters equal to one and calculate the posterior probability using a Monte Carlo experiment, we find that

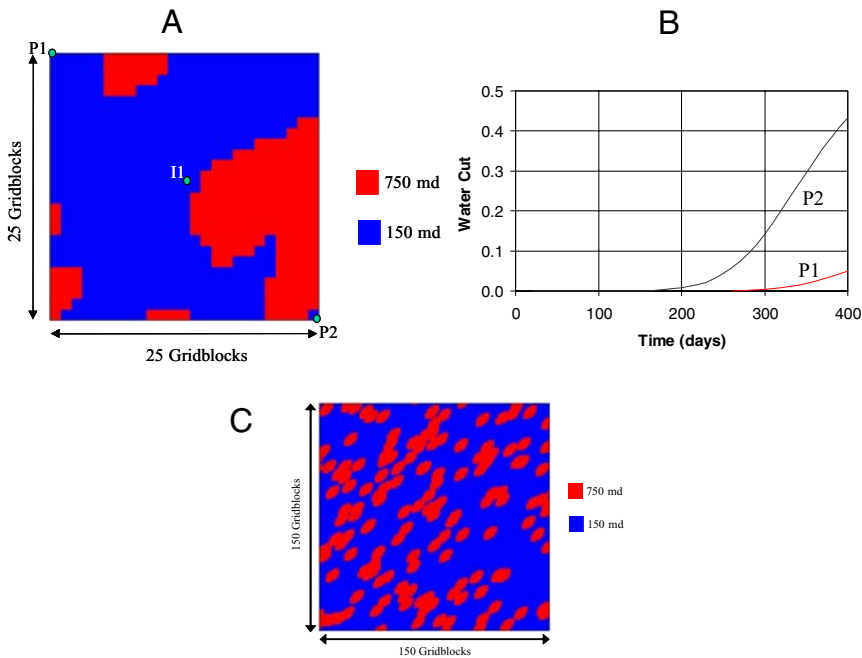
$$\hat{\text{Pr}}_{\text{PPM}}\{I(\mathbf{u}_1) = 0, I(\mathbf{u}_2) = 1, I(\mathbf{u}_3) = 1|\mathbf{d}\} \simeq 0.65$$

close to the true posterior of  $2/3$ . The point of this example is not so much to show that, for this specific case the PPM samples more accurately than GDSS, but that PPM has at least the modeling capability, in terms of the tau-model, to tune the sampler to make better approximations to the posterior. Indeed, with some trial and error one find that  $\tau_1 = 1$  and  $\tau_2 = 2.1$  would provide the correct answer ( $2/3$ ). The current problem with PPM is that is not yet a systematic way to determine the  $\tau$ -values in Eq. (6), this is still ongoing research (Krishnan, 2004).

### *Comparison of PPM with Rejection Sampling*

Setting  $r = 1$  at every change of the random seed is equivalent to rejection sampling. Instead, one optimizes  $r$ . The optimization of  $r$  costs several forward model evaluation, typically frozen to six as tested in Hoffman (2005). The question is if this “additional” cost in optimizing  $r$  is offset by lesser iterations on the random seed. In other words, is PPM more efficient than rejection sampling?

A simple example is used to show the difference in CPU. A simple flow problem is shown in Figure 8. The flow model example is a simple two facies model with 25 gridblocks in the x and y directions; the “true” reference model is



**Figure 8.** (A) Reference model, (B) water-cut data to be matched, (C) Training image.

shown in Figure 8A high permeability facies has a permeability of 750 md, and a lower permeability facies is 150 md. There are three wells, one injector in the center of the model (I1) and two producers at the corners of the model (P1 and P2). The majority of the high permeability facies is located between the injector and well P2; consequently, the water-cut profiles for the two production wells are significantly different, see Figure 8B.

For this example, the two hard data of facies measurements at the well locations are used. The prior model is generated by the *snesim* algorithm (Strebel, 2002) that uses as input the training image of Figure 8C. The dimensions of the training image are 150 by 150, while the model is only 25 by 25, so the high permeability bodies appear larger in the model than the training image, but in fact the bodies have the same dimensions.

Rejection sampling is performed by accepting all realization that honor the three hard data and that have a mismatch of less than 1% with the watercut data in Figure 8. The single probability perturbation method is also compared with a two-parameter perturbation. In the latter the grid nodes are divided in two regions, each region is assigned a parameter  $r$ . The regions are obtained by splitting the model by a diagonal line running from the lower left to the upper right corner.

The average number of flow simulations required for each method to generate one realization that matches the watercut data equals:

Rejection sampling: 3022  
 Single parameter PPM: 456  
 Two-parameter PPM: 216

which shows that the PPM method is about one order of magnitude faster than rejection sampling in this case. The gain is probably larger with higher dimensional and more complex data **d**.

## CONCLUSIONS

The probability perturbation method and gradual deformation method are two methods for solving spatial inverse problems. Both attempt to honor prior information when obtaining inverse solutions. The gradual deformation treats the problem of finding such solutions as an optimization problem, i.e. it tries to search (optimize) in the space of all possible prior model realizations, those realizations that match the data in question. Apart from the minor issue of combining dependent realizations illustrated in Figure 4, the method reproduces the spatial statistics of the prior model well. However, it appears that the particular method employed in gradual deformation does not always lead to a “correct” sampling of the posterior distribution as illustrated with one example. This warrants further investigation. On the other hand, one may argue that such “correct” sampling is not desirable in the first place since the posterior distribution is never known from data, it always needs to be assumed, just like a prior distribution needs to be assumed.

The probability perturbation treats the inverse problem as a “data integration problem,” in the sense that it integrates prior information with other available data. The method relies on the tau-model of Journel (Eq. 6) to integrate various data types, each of them being coded into a probability  $P(A|D_i)$ .

The prior information is coded as a series of conditional probabilities  $P(A|B_i)$  that are used to generate prior model realizations using sequential simulation. Sequential simulation allows modeling a large variety of priors, either variogram-based or training-image based. Unlike gradual deformation, the PPM does not find realizations by direct optimization; instead it treats the pre-posterior  $P(A|D_i)$  as unknown and estimates it by using a simple parameterization that depends on the prior  $P(A)$ , a current realization and a single parameter  $r$  (varying  $r$  spatially leads to a multi-parameter model). This estimation is done by means of optimization on  $r$ , i.e. one finds the  $r$ -value that provides a prior model realization with a better match to the data **d**. As such, the probability perturbation method seems to share many of the properties of Markov chain Monte Carlo samplers such as the rejection sampler.

These initial experimental findings will eventually lead to a better understanding of the properties of both methods and potential theoretical understanding of how to improve each one.

## ACKNOWLEDGMENTS

I would like to thank Lin-Ying Hu of the IFP for his helpful comments on an earlier version of this manuscript.

## REFERENCES

- Alabert, F., 1987, The practice of fast conditional simulations through LU decomposition of the covariance matrix: *Math. Geol.*, v. 19, no. 5, p. 369–386.
- Caers, J., 2003, History matching under a training image-based geological model constraint: *Soc. Petrol. Eng.*, SPE paper no. 74716, p. 218–226.
- Caers, J., and Hoffman, B. T., 2006, The probability perturbation method: an alternative Bayesian approach to history matching: *Math. Geol.*, v. 38, no. 1, p. 81–100.
- Caers, J., Hoffman, B. T., Strebelle, S., and Wen, X.-H., 2006, Probabilistic integration of geological information, seismic and production data: *The Leading Edge*, v. 25, p. 240–244.
- Deutsch, C. V., and Journel, A. G., 1998., *GSLIB: the geostatistical software library*: Oxford University Press, New York, 369 p.
- Hoffman B. T., and Caers J., 2004, History matching with the regional probability perturbation method-applications to a North Sea reservoir, In *Proceeding to the ECMOR IX, Cannes*, Aug 29–Sept. 2, 2004.
- Hoffman, B. T., and Caers, J., 2005, Regional probability perturbations for history matching: *Jour. Pet. Sci. Eng.*, v. 46, p. 53–71.
- Hoffman, B. T., 2005, *Geologically consistent history matching of facies models*: Doctoral Dissertation, Stanford University, 212 p.
- Hu, L.-Y., Blanc, G., and Noetinger, B., 2001, Gradual deformation and iterative calibration of sequential simulations: *Math. Geol.*, v. 33, p. 475–489.
- Hu, L.-Y., and Le Ravalec-Dupin, M., 2004, An improved gradual deformation method for reconciling random and gradient searches in stochastic optimizations: *Math. Geol.*, v. 36, no. 6, p. 703–720.
- Journel, A. G., and Huijbregts, 1978, *Mining Geostatistics*: The Blackburn Press, 600 p.
- Journel, A. G., 1993, Modeling Uncertainty: some conceptual thoughts, *in* Dimitrakopoulos, R., ed., *Geostatistics for the next century*: Kluwer, Dordrecht, Holland, p. 30–43.
- Journel, A. G., 1999, Markov models for cross-covariances: *Math. Geol.*, v. 31, no. 8, p. 955–964.
- Journel, A. G., 2002, Combining knowledge from diverse data sources: an alternative to traditional data independence hypothesis: *Math. Geol.*, v. 34, no. 1, p. 573–596.
- Krishnan, S., 2004, *The tau model to integrate prior probabilities*: Doctoral Dissertation, Stanford University, 202 p.
- Le Ravalec-Dupin, M., Noetinger, B., and Hu, L.-Y., 2002, The FFT moving average (FFT-MA) generator: an efficient numerical method for generating and conditioning Gaussian simulations: *Math. Geol.*, v. 32, no. 6, p. 701–723.
- Liu, N., and Oliver, D. S., 2004, Experimental assessment of the gradual deformation method: *Math. Geol.*, v. 36, no. 1, p. 65–78.
- Oliver, D. S., 1995, Moving averages for Gaussian simulation in two and three dimensions: *Math. Geol.*, v. 27, no. 8, p. 939–960.

- Omre, H., and Tjelmeland, H., 1997, Petroleum geostatistics, *in* Baafi, E.Y. and Schofield, N.A., eds., *Geostatistics Wolongong '96*, Kluwer, Dordrecht, Wollongong Australia, v. 1, p. 41–52.
- Reis, L. C., Hu, L. Y., de Marsily, G., and Eschard, R., 2000, Production data integration using a gradual deformation approach: application to an oil field (Offshore Brazil). Soc. Petrol. Eng., paper no. 63064. SPE Annual Technical Conference and Exhibition, 1–4 October, Dallas, Texas.
- Roggero, F., and Hu, L.-Y., 1998, Gradual deformation of continuous geostatistical models for history matching. Soc. Petrol. Eng., SPE paper no. 49004 presented at the 1998 SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, Sept. 27–30.
- Suzuki S., 2004, Effect of structure geometry on history matching faulted reservoirs: Stanford Center for Reservoir Forecasting, Annual Report 14, Stanford University, Stanford.
- Tarantola, A., 1987, *Inverse problem theory*: Elsevier, 342 p.
- Ying, Z., and Gomez, J. H., 2000, An improved deformation algorithm for automatic history matching: Stanford Center for Reservoir Forecasting, Annual Report 13, Stanford University, Stanford.