Math Geosci (2012) 44:783–803
DOI 10.1007/s11004-012-9417-2

A Frequency Matching Method: Solving Inverse Problems by Use of Geologically Realistic Prior Information

Katrine Lange · Jan Frydendall · Knud Skou Cordua · Thomas Mejer Hansen · Yulia Melnikova · Klaus Mosegaard

Received: 3 February 2012 / Accepted: 16 July 2012 / Published online: 22 September 2012

Abstract The frequency matching method defines a closed form expression for a complex prior that quantifies the higher order statistics of a proposed solution model to an inverse problem. While existing solution methods to inverse problems are capable of sampling the solution space while taking into account arbitrarily complex a priori information defined by sample algorithms, it is not possible to directly compute the maximum a posteriori model, as the prior probability of a solution model cannot be expressed. We demonstrate how the frequency matching method enables us to compute the maximum a posteriori solution model to an inverse problem by using a priori information based on multiple point statistics learned from training images. We demonstrate the applicability of the suggested method on a synthetic tomographic crosshole inverse problem.

Keywords Geostatistics · Multiple point statistics · Training image · Maximum a posteriori solution

1 Introduction

Inverse problems arising in the field of geoscience are typically ill-posed; the available data are scarce and the solution to the inverse problem is therefore not well-determined. In probabilistic inverse problem theory the solution to a problem is given as an a posteriori probability density function that combines states of information provided by observed data and the a priori information (Tarantola 2005). The ambiguities of the solution of the inverse problem due to the lack of restrictions on the solution is then reflected in the a posteriori probability.

K. Lange (✉) · J. Frydendall · K.S. Cordua · T.M. Hansen · Y. Melnikova · K. Mosegaard
Center for Energy Resources Engineering, Department of Informatics and Mathematical Modeling, Technical University of Denmark, Richard Petersens Plads, Building 321, 2800 Kongens Lyngby, Denmark
e-mail: katla@imm.dtu.dk

A priori information used in probabilistic inverse problem theory is often covariance-based a priori models. In these models the spatial correlation between the model parameters is defined by two-point statistics. In reality, two-point-based a priori models are too limited to capture curvilinear features such as channels or cross beddings. It is therefore often insufficient to rely only on the two-point statistics, and thus higher order statistics must also be taken into account in order to correctly produce geologically realistic descriptions of the subsurface. It is assumed that geological information is available in the form of a training image. This image could for instance have been artificially created to describe the expectations for the solution model or it could be information from a previous solution to a comparable inverse problem. The computed models should not be identical to the training image, but rather express a compromise between honoring observed data and comply with the information extracted from the training image. The latter can be achieved by ensuring that the models have the same multiple point statistics as the training image.

Guardiano and Srivastava (1993) proposed a sequential simulation algorithm that was capable of simulating spatial features inferred from a training image. Their approach was computationally infeasible until Strebelle (2002) developed the single normal equation simulation (*snesim*) algorithm. Multiple point statistics in general and the *snesim* algorithm in particular have been widely used for creating models based on training images and for solving inverse problems, see for instance Caers and Zhang (2004), Arpat (2005), Hansen et al. (2008), Peredo and Ortiz (2010), Suzuki and Caers (2008), Jafarpour and Khodabakhshi (2011). A method called the probability perturbation method (PPM) has been proposed by Caers and Hoffman (2006). It allows for gradual deformation of one realization of *snesim* to another realization of *snesim*. Caers and Hoffman propose to use the PPM method to find a solution to an inverse problem that is consistent with both a complex prior model, as defined by a training image, and data observations. PPM is used iteratively to perturb a realization from *snesim* while reducing the data misfit. However, as demonstrated by Hansen et al. (2012), as a result of the probability of the prior model not being evaluated, the model found using PPM is not the maximizer of the posterior density function, but simply the realization of the multiple point based prior with the highest likelihood value. There is no control of how reasonable the computed model is with respect to the prior model. It may be highly unrealistic.

The sequential Gibbs sampling method by Hansen et al. (2012) is used to sample the a posteriori probability density function given, for example a training image based prior. However, as with the PPM it cannot be used for optimization and locating the maximum a posteriori (MAP) model, as the prior probability is not quantified. The focus of our research is the development of the frequency matching (FM) method. The core of this method is the characterization of images by their multiple point statistics. An image is represented by the histogram of the multiple point-based spatial event in the image; this histogram is denoted the frequency distribution of the image. The most significant aspect of this method, compared to existing methods based on multiple point statistics for solving inverse problems, is the fact that it explicitly formulates an a priori probability density distribution, which enables it to efficiently quantify the probability of a realization from the a priori probability.

The classical approach when solving inverse problems by the least squares methods assumes a Gaussian prior distribution with a certain expectation. Solution models

to the inverse problem are penalized depending on their deviation from the expected model. In the FM method, the frequency distribution of the training image acts as the expected model and a solution image is penalized depending on how much its frequency distribution deviates from that of the training image. To perform this comparison we introduce a dissimilarity measure between a training image and a model image as the χ^2 distance between their frequency distributions. Using this dissimilarity measure for quantifying the a priori probability of a model the FM method allows us to directly compute the MAP model, which is not possible using known techniques such as the PPM and sequential Gibbs sampling methods.

Another class of methods are the Markov random fields (MRF) methods (Tjelmeland and Besag 1998). The prior probability density given by Markov methods involves a product of a large number of marginals. A disadvantage is therefore, despite having an expression for the normalization constant, that it can be computationally expensive to compute. Subclasses of the MRF methods such as Markov mesh models (Stien and Kolbjørnsen 2011) and partially ordered Markov models (Cressie and Davidson 1998) avoid the computation of the normalization constant, and this advantage over the MRF methods is shared by the FM method. Moreover, in contrast to methods such as PMM and MRF, the FM method is fully non-parametric, as it does not require probability distributions to be written in a closed form.

This paper is ordered as follows. In Sect. 2 we define how we characterize images by their frequency distributions, we introduce our choice of a priori distribution of the inverse problem and we elaborate on how it can be incorporated into traditional inverse problem theory. Our implementation of the FM method is discussed in Sect. 3. In Sect. 4 we present our test case and the results when solving an inverse problem using frequency matching-based a priori information. Section 5 summarizes our findings and conclusions.

2 Method

In geosciences, inverse problems involve a set of measurements or observations $\mathbf{d}^{\text{obs}}$ used to determine the spatial distribution of physical properties of the subsurface. These properties are typically described by a model with a discrete set of parameters, $\mathbf{m}$. For simplicity, we will assume that the physical property is modeled using a regular grid in space. The model parameters are said to form an image of the physical property.

Consider the general forward problem,

$$\mathbf{d} = g(\mathbf{m}), \tag{1}$$

of computing the observations $\mathbf{d}$ given the perhaps non-linear forward operator g and the model parameters $\mathbf{m}$. The values of the observation parameters are computed straightforwardly by applying the forward operator to the model parameters. The associated inverse problem consists of computing the model parameters $\mathbf{m}$ given the forward operator g and a set of observations $\mathbf{d}^{\text{obs}}$. As the inverse problem is usually severely under-determined, the model $\mathbf{m}$ that satisfies $\mathbf{d}^{\text{obs}} = g(\mathbf{m})$ is not uniquely determined. Furthermore, some of the models satisfying $\mathbf{d}^{\text{obs}} = g(\mathbf{m})$ within the required level of accuracy will be uninteresting for a geoscientist as the nature of the

forward operator g and the measurement noise in $\mathbf{d}^{\text{obs}}$ may yield a physically unrealistic description of the property. The inverse problem therefore consists of not just computing a set of model parameters satisfying Eq. 1, but computing a set of model parameters that gives a realistic description of the physical property while honoring the observed data. The FM method is used to express how geologically reasonable a model is by quantifying its a priori probability using multiple point statistics. Letting the a priori information be available in, for instance, a training image, the FM method solves an inverse problem by computing a model that satisfies not only the relation from Eq. 1 but a model that is also similar to the training image. The latter ensures that the model will be geologically reasonable.

2.1 The Maximum A Posteriori Model

Tarantola and Valette (1982) derived a probabilistic approach to solve inverse problems where the solution to the inverse problem is given by a probability density function, denoted the a posteriori distribution. This approach makes use of a prior distribution and a likelihood function to assign probabilities to all possible models. The a priori probability density function ρ describes the data independent prior knowledge of the model parameters; in the FM method we choose to define it as follows

$$\rho(\mathbf{m}) = \text{const.} \exp\big(-\alpha\ f(\mathbf{m})\big),$$

where α acts as a weighting parameter and f is a dissimilarity function presented in Sect. 2.4. Traditionally, f measures the distance between the model and an a priori model. The idea behind the FM method is the same, except we wish not to compare models directly but to compare the multiple point statistics of models. We therefore choose a traditional prior but replace the distance function such that instead of measuring the distance between models directly, we measure the dissimilarity between them. The dissimilarity is expressed as a distance between their multiple point statistics.

The likelihood function L is a probabilistic measure of how well data associated with a certain model matches the observed data, accounting for the uncertainties of the observed data,

$$L\big(\mathbf{m}, \mathbf{d}^{\text{obs}}\big) = \text{const.} \exp\left(-\frac{1}{2}\big\|\mathbf{d}^{\text{obs}} - g(\mathbf{m})\big\|^2_{\mathbf{C}_\text{d}}\right).$$

Here, $\mathbf{C}_\text{d}$ is the data covariance matrix and the measurement errors are assumed to be independent and Gaussian distributed with mean values 0. The a posteriori distribution is then proportional to the product of the prior distribution and the likelihood

$$\sigma(\mathbf{m}) = \text{const.}\rho(\mathbf{m})L\big(\mathbf{m}, \mathbf{d}^{\text{obs}}\big).$$

The set of model parameters that maximizes the a posteriori probability density is called the maximum a posteriori (MAP) model

$$\begin{aligned}\mathbf{m}^{\text{MAP}} &= \arg\max_{\mathbf{m}}\big\{\sigma(\mathbf{m})\big\} \\ &= \arg\min_{\mathbf{m}}\big\{-\log\sigma(\mathbf{m})\big\} \\ &= \arg\min_{\mathbf{m}}\left\{\frac{1}{2}\big\|\mathbf{d}^{\text{obs}} - g(\mathbf{m})\big\|^2_{\mathbf{C}_\text{d}} + \alpha\ f(\mathbf{m})\right\}.\end{aligned}$$

The dissimilarity function f is a measure of how well the model satisfies the a priori knowledge that is available, for example from a training image. The more similar, in some sense, the image from a set of model parameters $\mathbf{m}$ is to the training image the smaller the function value $f(\mathbf{m})$ is. Equivalently to the more traditional term $\|\mathbf{m} - \mathbf{m}^{\text{prior}}\|^2_{\mathbf{C}_\text{m}}$, stemming from a Gaussian a priori distribution of the model parameters with mean values $\mathbf{m}^{\text{prior}}$ and covariance matrix $\mathbf{C}_\text{m}$, $f(\mathbf{m})$ can be thought of as a distance. It is not a distance between $\mathbf{m}$ and the training image ($f(\mathbf{m})$ may be zero for other images than the training image), but a distance between the multiple point statistics of the image formed by the model parameters and the multiple point statistics of the training image.

2.2 The Multiple Point Statistics of an Image

Consider an image $Z = \{1, 2, \ldots, N\}$ with N voxels (or pixels if the image is only two dimensional) where the voxels can have the m different values $0, 1, \ldots, m-1$. We introduce the N variables, $z_1, z_2, \ldots, z_N$ and let z_k describe the value of the kth voxel of the image. It is assumed that the image is a realization of an unknown, random process satisfying:

1. The value of the kth voxel, z_k, is, given the values of voxels in a certain neighborhood $\mathcal{N}_k$ around voxel k, independent of voxel values not in the neighborhood. Voxel k itself is not contained in $\mathcal{N}_k$. Let $\mathbf{z}_k$ be a vector of the values of the ordered neighboring voxels in $\mathcal{N}_k$; we then have
$$f_Z(z_k|z_N, \ldots, z_{k+1}, z_{k-1}, \ldots, z_1) = f_Z(z_k|\mathbf{z}_k),$$
where f_Z denotes the conditional probability distribution of the voxel z_k given the values of the voxels within the neighborhood.
2. For an image of infinite size the geometrical shape of all neighborhoods $\mathcal{N}_k$ are identical. This implies that if voxel k has coordinates (k_x, k_y, k_z), and voxel l has coordinates (l_x, l_y, l_z), then
$$(n_x, n_y, n_z) \in \mathcal{N}_k \quad \Rightarrow \quad (n_x - k_x + l_x, n_y - k_y + l_y, n_z - k_z + l_z) \in \mathcal{N}_l.$$
3. If we assume ergodicity, that is, when two voxels, voxel k and voxel l, have the same values as their neighboring voxels, then the conditional probability distribution of voxel k and voxel l are identical
$$\mathbf{z}_k = \mathbf{z}_l \quad \Rightarrow \quad f_Z(z_k|\mathbf{z}_k) = f_Z(z_l|\mathbf{z}_l).$$

Knowing the conditionals $f_Z(z_k|\mathbf{z}_k)$ we know the multiple point statistics of the image, just as a variogram would describe the two-point statistics of an image. The basis of sequential simulation as proposed by Guardiano and Srivastava (1993) is to exploit the aforementioned assumptions to estimate the conditional probabilities $f_Z(z_k|\mathbf{z}_k)$ based on the marginals obtained from the training image, and then to use the conditional distributions to generate new realizations of the unknown random process from which the training image is a realization. The FM method, on the other hand, operates by characterizing images by their frequency distributions. As described in the following section, the frequency distribution of voxel values within the given neighborhood of an image is given by its marginal distributions. This means

that comparison of images is done by comparing their marginals. For now, the training image is assumed to be stationary. With the current formulation of the frequency distributions this is the only feasible approach. Discussion of how to avoid the assumption of stationarity exists in literature, see for instance the recent Honarkhah (2011). Some of these approaches mentioned here might also be useful for the FM method, but we will leave this to future research to determine.

2.3 Characterizing Images by their Frequency Distribution

Before presenting the FM method we define what we denote the frequency distribution. Given an image with the set of voxels $Z = \{1, \ldots, N\}$ and voxel values $z_1, \ldots, z_N$ we define the template function Ω as a function that takes as argument a voxel k and returns the set of voxels belonging to the neighborhood $\mathcal{N}_k$ of voxel k. In the FM method, the neighborhood of a voxel is indirectly given by the statistical properties of the image itself; however, the shape of a neighborhood satisfying the assumptions from Sect. 2.2 is unknown. For each training image one must therefore define a template function Ω that seeks to correctly describe the neighborhood. The choice of template function determines if a voxel is considered to be an inner voxel. An inner voxel is a voxel with the maximal neighborhood size, and the set of inner voxels, Z_{in}, of the image is therefore defined as

$$Z_{\text{in}} = \left\{ k \in Z \colon |\mathcal{N}_k| = \max_{l \in Z} |\mathcal{N}_l| \right\},$$

where $|\mathcal{N}_k|$ denotes the number of voxels in $\mathcal{N}_k$. Let n denote the number of voxels in the neighborhood of an inner voxel. Typically, voxels on the boundary or close to the boundary of an image will not be inner voxels. To each inner voxel z_k we assign a pattern value p_k; we say the inner voxel is the center voxel of a pattern. This pattern value is a unique identifier of the pattern and may be chosen arbitrarily. The most obvious choice is perhaps a vector value with the discrete variables in the pattern, or a scalar value calculated based on the values of the variables. The choice should be made in consideration of the implementation of the FM method. The pattern value is uniquely determined by the value of the voxel z_k and the values of the voxels in its neighborhood, $\mathbf{z}_k$. As the pattern value is determined by the values of $n + 1$ voxels, which can each have m different values, the maximum number of different patterns is m^{n+1}.

Let π_i, for $i = 1, \ldots, m^{n+1}$, count the number of patterns that have the ith pattern value. The frequency distribution is then defined as $\boldsymbol{\pi}$

$$\boldsymbol{\pi} = [\pi_1, \ldots, \pi_{m^{n+1}}].$$

Let p_Ω denote the mapping from voxel values of an image Z to its frequency distribution $\boldsymbol{\pi}$, that is, $p_\Omega(z_1, \ldots, z_N) = \boldsymbol{\pi}$.

Figure 1 shows an example of an image and the patterns it contains for the template function that defines neighborhoods as follows

$$\mathcal{N}_k = \left\{ l \in Z \setminus \{k\} \colon |l_x - k_x| \leq 1, |l_y - k_y| \leq 1 \right\}.$$

Recall from Sect. 2.2 that (l_x, l_y) are the coordinates of voxel l in this two-dimensional example image. We note that for a given template function the frequency

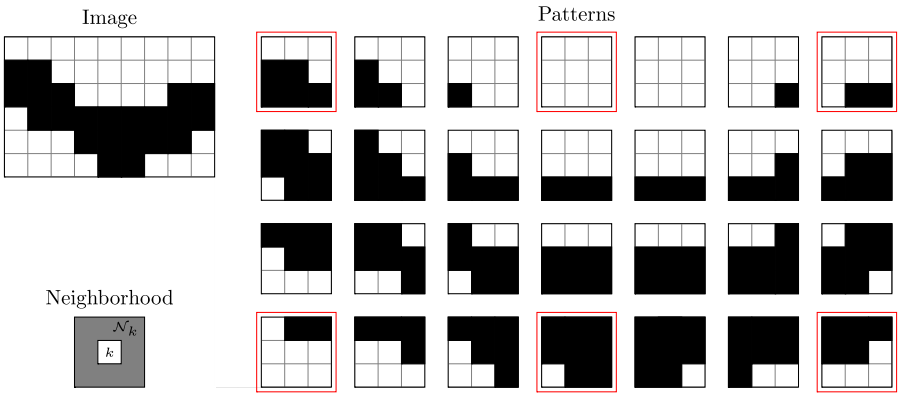


Fig. 1 Example of patterns found in an image. Notice how the image is completely described by the (ordered) patterns in every third row and column; the patterns are marked in red

distribution of an image is uniquely determined. The opposite, however, does not hold. Different images can, excluding symmetries, have the same frequency distribution. This is what the FM method seeks to exploit by using the frequency distribution to generate new images, at the same time similar to, and different from, our training image.

2.4 Computing the Similarity of Two Images

The FM method compares a solution image to a training image by comparing its frequency distribution to the frequency distribution of the training image. How dissimilar the solution image is to the training image is determined by a dissimilarity function, which assigns a distance between their frequency distributions. This distance reflects how likely the solution image is to be a realization of the same unknown process as the training image is a realization of. The bigger the distance, the more dissimilar are the frequency distributions and thereby also the images, and the less likely is the image to be a realization of the same random process as the training image. The dissimilarity function can therefore be used to determine which of two images is most likely to be a realization of the same random process as the training image is a realization of.

The dissimilarity function is not uniquely given but an obvious choice is the χ^2 distance also described in Sheskin (2004). It is used to measure the distance between two frequency distributions by measuring how similar the proportions of patterns in the frequency distributions are. Given two frequency distributions, the χ^2 distance estimates the underlying distribution. It then computes the distance between the two frequency distributions by computing each of their distances to the underlying distribution. Those distances are computed using a weighted Euclidean norm where the weights are the inverse of the counts of the underlying distribution, see Fig. 2. In our research, using the counts of the underlying distribution turns out to be a favorable weighting of small versus big differences instead of using a traditional p-norm as used by Peredo and Ortiz (2010).

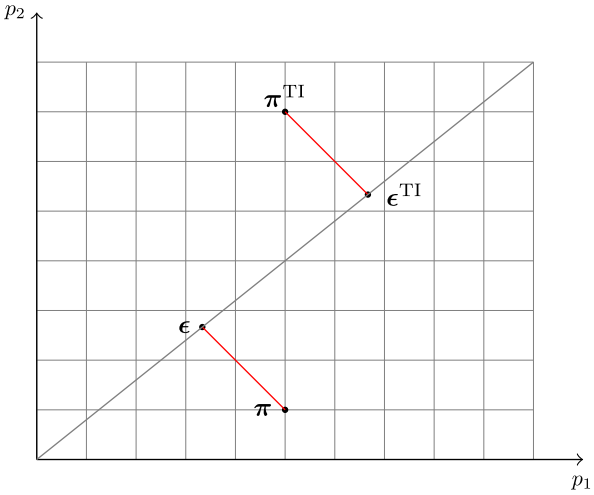


Fig. 2 Illustration of the χ^2 distance between two frequency distributions $\boldsymbol{\pi}$ and $\boldsymbol{\pi}^{\mathrm{TI}}$, each containing the counts of two different pattern values, p_1 and p_2. The difference between the frequency distributions is computed as the sum of the length of the two red line segments. The length of each line segment is computed using a weighted Euclidean norm. The counts of the underlying distribution are found as the orthogonal projection of the frequency distributions onto the a line going through the origin such that $\|\boldsymbol{\pi} - \boldsymbol{\epsilon}\|_2 = \|\boldsymbol{\pi}^{\mathrm{TI}} - \boldsymbol{\epsilon}^{\mathrm{TI}}\|_2$

Hence, given the frequency distributions of an image, $\boldsymbol{\pi}$, and of a training image, $\boldsymbol{\pi}^{\mathrm{TI}}$, and by letting

$$I = \left\{i \in \left\{1, \ldots, m^{n+1}\right\} \colon \pi_i^{\mathrm{TI}} > 0\right\} \cup \left\{i \in \left\{1, \ldots, m^{n+1}\right\} \colon \pi_i > 0\right\}, \tag{2}$$

we compute what we define as the dissimilarity function value of the image

$$c(\boldsymbol{\pi}) = \chi^2\left(\boldsymbol{\pi}, \boldsymbol{\pi}^{\mathrm{TI}}\right) = \sum_{i \in I} \frac{(\pi_i^{\mathrm{TI}} - \epsilon_i^{\mathrm{TI}})^2}{\epsilon_i^{\mathrm{TI}}} + \sum_{i \in I} \frac{(\pi_i - \epsilon_i)^2}{\epsilon_i}, \tag{3}$$

where ϵ_i denotes the counts of the underlying distribution of patterns with the ith pattern value for images of the same size as the image and ϵ_i^{TI} denotes the counts of the underlying distribution of patterns with the ith pattern value for images of the same size as the training image. These counts are computed as

$$\epsilon_i = \frac{\pi_i + \pi_i^{\mathrm{TI}}}{n_Z + n_{\mathrm{TI}}} n_Z, \tag{4}$$

$$\epsilon_i^{\mathrm{TI}} = \frac{\pi_i + \pi_i^{\mathrm{TI}}}{n_Z + n_{\mathrm{TI}}} n_{\mathrm{TI}}, \tag{5}$$

where n_Z and n_{TI} are the total number of counts of patterns in the frequency distributions of the image and the training image, that is, the number of inner voxels in the image and the training image, respectively.

2.5 Solving Inverse Problems

We define the frequency matching method for solving inverse problems formulated as least squares problems using geologically complex a priori information as the fol-

lowing optimization problem

$$\begin{aligned} &\min_{z_1,\ldots,z_N} \left\| \mathbf{d}^{\text{obs}} - g(z_1,\ldots,z_N) \right\|^2_{\mathbf{C}_\text{d}} + \alpha\, c(\boldsymbol{\pi}), \\ &\text{w.r.t. } \boldsymbol{\pi} = p_\Omega(z_1,\ldots,z_N), \\ &z_k \in \{0,\ldots,m-1\} \quad \text{for } k = 1,\ldots,N, \end{aligned} \tag{6}$$

where $c(\boldsymbol{\pi})$ is the dissimilarity function value of the solution image defined by Eq. 3 and α is a weighting parameter. The forward operator g, which traditionally is a mapping from model space to data space, also contains the mapping of the categorical values $z_k \in \{0,\ldots,m-1\}$ for $k = 1,\ldots,N$ of the image into the model parameters $\mathbf{m}$ that can take m different discrete values.

The value of α cannot be theoretically determined. It is expected to depend on the problem at hand; among other factors its resolution, the chosen neighborhood function and the dimension of the data space. It can be thought of as playing the same role for the dissimilarity function as the covariance matrix $\mathbf{C}_\text{d}$ does for the data misfit. So it should in some sense reflect the variance of the dissimilarity function and in that way determine how much trust we put in the dissimilarity value. Variance, or trust, in a training image is difficult to quantify, as the training image is typically given by a geologist to reflect certain expectations to model. Not having a theoretical expression for α therefore allows us to manipulate the α value to loosely quantify the trust we have in the training image. In the case where we have accurate data but only a vague idea of the structures of the subsurface the α can be chosen low, in order to emphasize the trust we have in the data and the uncertainty we have of the structure of the model. In the opposite case, where data are inaccurate but the training image is considered to be a very good description of the subsurface, the α value can be chosen high, to give the dissimilarity function more weight.

Due to the typically high number of model parameters, the combinatorial optimization problem should be solved by use of an iterative solution method; such a method will iterate through the model space and search for the optimal solution. While the choice of solution method is less interesting when formulating the FM method, it is of great importance when applying it. The choice of solution method and the definition of how it iterates through the solution space by perturbing images has a significant impact on the feasibility of the method in terms of its running time. As we are not sampling the solution space we do not need to ensure that the method captures the uncertainty of the model parameters, and the ideal would be a method that converges directly to the maximum a posteriori solution. While continuous optimization problems hold information about the gradient of the objective function that the solution method can use to converge to a stationary solution, this is not the case for our discrete problem. Instead we consider the multiple point statistics of the training image when perturbing a current image and in that way we seek to generate models which better match the multiple point statistics of the training image and thus guide the solution method to the maximum a posteriori model.

2.6 Properties of the Frequency Matching Method

The FM method is a general method and in theory it can be used to simulate any type of structure, as long as a valid training image is available and a feasible template

function is chosen appropriately. If neighborhoods are chosen too small, the method will still be able to match the frequency distributions. However, it will not reproduce the spatial structures simply because these are not correctly described by the chosen multiple point statistics and as a result the computed model will not be realistic. If neighborhoods are chosen too big, CPU cost and memory demand will increase, and as a result the running time per iteration of the chosen solution method will increase. Depending on the choice of iterative solution method, increasing the size n of the neighborhood is likely to also increase the number of iterations needed and thereby increase the convergence time. When the size of neighborhoods is increased, the maximum number of different patterns, m^{n+1}, is also increased. The number of different patterns present is, naturally, limited by the number of inner voxels, which is significantly smaller than m^{n+1}. In fact, the number of patterns present in an image is restricted further as training images are chosen such that they describe a certain structure. This structure is also sought to be described in the solutions. The structure is created by repetition of patterns, and the frequency distributions will reveal this repetition by having multiple counts of the same pattern. This means, the number of patterns with non-zero frequency is greatly smaller than m^{n+1} resulting in the frequency distributions becoming extremely sparse. For bigger test cases, with millions of parameters, patterns consisting of hundreds of voxels and multiple categories, this behavior needs to be investigated further.

The dimension of the images, if they are two or three dimensional, is not important to the FM method. The complexity of the method is given by the maximal size of neighborhoods, n. The increase in n as a result of going from two- to three-dimensional images is therefore more important than the actual increase in physical dimensions. In fact, when it comes to assigning pattern values a neighborhood is, regardless of its physical dimension, considered one dimensional where the ordering of the voxels is the important aspect. Additionally, the number of categories of voxel values m does not influence the running time per iteration. As with the number of neighbors, n, it only influences the number of different possible patterns m^{n+1} and thereby influences the sparsity of the frequency distribution of the training image. The higher m is, the sparser is the frequency distribution. It is expected that the sparsity of the frequency distribution affects the level of difficulty of the combinatorial optimization problem.

Strebelle (2002) recommends choosing a training image that is at least twice as large as the structures it describes; one must assume this advice also applies to the FM method. Like the *snesim* algorithm, the FM method can approximate continuous properties by discretizing them into a small number of categories. One of the advantages of the FM method is that by matching the frequency distributions it indirectly ensures that the proportion of voxels in each of the m categories is consistent between the training image and the solution image. It is therefore not necessary to explicitly account for this ratio. Unlike the *snesim* algorithm, the computed solution images therefore need very little post treatment—in the current implementation the solution receives no post treatment. However, the α parameter does allow for the user to specify how strictly the frequency distributions should be matched. In the case where the data are considered very informative or the training image is considered far from reality, decreasing the α allows for the data to be given more weight and the multiple point statistics will not be as strictly enforced.

Constraints on the model parameters can easily be dealt with by reducing the feasible set $\{0, \ldots, m-1\}$ for those values of k in the constraints of the problem stated in Eq. 6. The constrained voxels remain part of the image Z and when computing the frequency distribution of an image they are not distinguished from non-constrained voxels. However, when perturbing an image all constraints of the inverse problem should at all times be satisfied and conditioned to the hard data. The additional constraints on the model parameters will therefore be honored.

3 Implementation

This section describes the current implementation of the frequency matching method. Algorithm 1 gives a general outline of how to apply the FM method, that is, how to solve the optimization problem from Eq. 6 with an iterative optimization method. In the remainder of the section, the implementation of the different parts of the FM method will be discussed. It should be noted that the implementation of the FM method is not unique; for instance, there are many options for how the solution method iterates through the model space by perturbing models. The different choices should be made depending on the problem at hand and the current implementation might not be favorable for some given problems. The overall structure in Algorithm 1 will be valid regardless of what choices are made on a more detailed level.

Algorithm 1: The Frequency Matching Method

Input: Training image, Z^{TI}, Starting image Z
Output: Maximum a posteriori image Z^{FM}
Compute frequency distribution of training image $\boldsymbol{\pi}^{\mathrm{TI}}$ and pattern list $\mathbf{p}$ (Algorithm 2)
Compute partial frequency distribution of starting image $\boldsymbol{\pi}$ (Algorithm 3)
while *not converged* **do**
  Compute perturbed image $\overline{Z}$ based on Z (Algorithm 4)
  Compute partial frequency distribution of perturbed image $\overline{\boldsymbol{\pi}}$ (Algorithm 5)
  if *accept the perturbed image* **then**
    Set $Z \leftarrow \overline{Z}$ and $\boldsymbol{\pi} \leftarrow \overline{\boldsymbol{\pi}}$
  end
end

The current implementation is based on a Simulated Annealing scheme. Simulated Annealing is a well-known heuristic optimization method first presented by Kirkpatrick et al. (1983) as a solution method for combinatorial optimization problems. The acceptance of perturbed images is done using an exponential cooling rate and the parameters controlling the cooling are tuned to achieve an acceptance ratio of approximately 15 accepted perturbed models for each 100 suggested perturbed models. A perturbed model is generated by erasing the values of the voxels in a part of the image and then re-simulating the voxel values by use of sequential simulation.

3.1 Reformulation of the Dissimilarity Function

The definition of the dissimilarity function from Eq. 3 has one great advantage that we for computational reasons simply cannot afford to overlook. As discussed previously, the frequency distributions are expected to be sparse as the number of patterns present in an image is significantly smaller than m^{n+1}. This means that a lot of the terms in the dissimilarity function from Eq. 3 will be zero, yet the dissimilarity function can be simplified further. It will be shown that the dissimilarity function value of a frequency distribution, $c(\boldsymbol{\pi})$, given the frequency distribution of a training image, $\boldsymbol{\pi}$, can be computed using only entries of $\boldsymbol{\pi}$ where $\boldsymbol{\pi}^{\mathrm{TI}} > 0$. In other words, to compute the dissimilarity function value of an image we need only to know the count of patterns in the image that also appear in the training image. Computationally, this is a great advantage as we can disregard the patterns in our solution image that do not appear in the training image and we need not compute nor store the entire frequency distribution of our solution image, which is shown by inserting the expressions of the counts for the underlying distribution defined by Eqs. 4 and 5

$$\begin{aligned} c(\boldsymbol{\pi}) &= \sum_{i\in I} \frac{(\pi_i^{\mathrm{TI}} - \epsilon_i^{\mathrm{TI}})^2}{\epsilon_i^{\mathrm{TI}}} + \sum_{i\in I} \frac{(\pi_i - \epsilon_i)^2}{\epsilon_i} \\ &= \sum_{i\in I} \frac{(\sqrt{\frac{n_Z}{n_{\mathrm{TI}}}}\,\pi_i^{\mathrm{TI}} - \sqrt{\frac{n_{\mathrm{TI}}}{n_Z}}\pi_i)^2}{\pi_i^{\mathrm{TI}} + \pi_i}. \end{aligned} \tag{7}$$

This leads to the introduction of the following two subsets of I

$$\begin{aligned} I_1 &= \left\{ i \in I : \pi_i^{\mathrm{TI}} > 0 \right\}, \\ I_2 &= \left\{ i \in I : \pi_i^{\mathrm{TI}} = 0 \right\}. \end{aligned}$$

The two subsets form a partition of I as they satisfy $I_1 \cup I_2 = I$ and $I_1 \cap I_2 = \emptyset$. The dissimilarity function Eq. 7 can then be written as

$$\begin{aligned} c(\boldsymbol{\pi}) &= \sum_{i\in I_1} \frac{(\sqrt{\frac{n_Z}{n_{\mathrm{TI}}}}\pi_i^{\mathrm{TI}} - \sqrt{\frac{n_{\mathrm{TI}}}{n_Z}}\pi_i)^2}{\pi_i^{\mathrm{TI}} + \pi_i} + \frac{n_{\mathrm{TI}}}{n_Z} \sum_{i\in I_2} \pi_i \\ &= \sum_{i\in I_1} \frac{(\sqrt{\frac{n_Z}{n_{\mathrm{TI}}}}\pi_i^{\mathrm{TI}} - \sqrt{\frac{n_{\mathrm{TI}}}{n_Z}}\pi_i)^2}{\pi_i^{\mathrm{TI}} + \pi_i} + \frac{n_{\mathrm{TI}}}{n_Z}\left(n_Z - \sum_{i\in I_1} \pi_i \right) \end{aligned} \tag{8}$$

recalling that $\sum_{i\in I} \pi_i = n_Z$ and that $\pi_i = 0$ for $i \notin I$.

A clear advantage of this formulation of the dissimilarity function is that the entire frequency distribution $\boldsymbol{\pi}$ of the image does not need to be known; as previously stated, it only requires the counts π_i of the patterns also found in the training image, which is for $i \in I_1$.

3.2 Computing and Storing the Frequency Distributions

The formulation of the dissimilarity function from Eq. 3 and later Eq. 8 means that it is only necessary to store non-zero entries in a frequency distribution of a training

image $\boldsymbol{\pi}^{\mathrm{TI}}$. Algorithm 2 shows how the frequency distribution of a training image is computed such that zero entries are avoided. The algorithm also returns a list $\mathbf{p}$ with the same number of elements as the frequency distribution and it holds the pattern values corresponding to each entry of $\boldsymbol{\pi}^{\mathrm{TI}}$.

Algorithm 2: Frequency Distribution of a Training Image

Input: Training Image Z^{TI}
Output: Frequency distribution $\boldsymbol{\pi}^{\mathrm{TI}}$, list of pattern values $\mathbf{p}$
Initialization: empty list $\boldsymbol{\pi}^{\mathrm{TI}}$, empty list $\mathbf{p}$
for *each inner voxel, i.e.,* $k \in Z_{\mathrm{in}}^{\mathrm{TI}}$ **do**
  Extract pattern k
  Compute pattern value p_k
  if *the pattern was previously found* **then**
    Add 1 to the corresponding entry of $\boldsymbol{\pi}^{\mathrm{TI}}$
  else
    Add p_k to the list of pattern values $\mathbf{p}$
    Set the corresponding new entry of $\boldsymbol{\pi}^{\mathrm{TI}}$ equal to 1
  end
end

Algorithm 3 computes the partial frequency distribution $\boldsymbol{\pi}$ of an image that is needed to evaluate the dissimilarity function $c(\boldsymbol{\pi}) = \chi^2(\boldsymbol{\pi}, \boldsymbol{\pi}^{\mathrm{TI}})$ from Eq. 8. The partial frequency distribution only stores the frequencies of the patterns also found in the training image.

Algorithm 3: Partial Frequency Distribution of an Image

Input: Image Z, list of pattern values $\mathbf{p}$ from the training image
Output: Partial frequency distribution $\boldsymbol{\pi}$
Initialization: all zero list $\boldsymbol{\pi}$ (same length as $\mathbf{p}$)
for *each inner voxel, i.e.,* $k \in Z_{\mathrm{in}}$ **do**
  Extract pattern k
  Compute pattern value p_k
  if *the pattern is found in the training image* **then**
    Add 1 to the corresponding entry of $\boldsymbol{\pi}$
  end
end

3.3 Perturbation of an Image

The iterative solver moves through the model space by perturbing models and this is the part of the iterative solver that leaves the most choices to be made. An intuitive but naive approach would be to simply change the value of a random voxel. This will result in a perturbed model that is very close to the original model, and it will therefore require a lot of iterations to converge. The current implementation changes the values of a block of voxels in a random place of the image.

Before explaining in detail how the perturbation is done, let $Z^{\text{cond}} \subset Z$ be the set of voxels that we have hard data for, which means their value is known and should be conditioned to. First a voxel k is chosen randomly. Then the value of all voxels in a domain $\mathcal{D}_k \subset (Z \setminus Z^{\text{cond}})$ around voxel k are erased. Last, the values of the voxels in $\mathcal{D}_k$ are simulated using sequential simulation. The size of the domain should be chosen to reflect how different the perturbed image should be from the current image. The bigger the domain, the fewer iterations we will expect the solver will need to iterate through the model space to converge, but the more expensive an iteration will become. Choosing the size of the domain is therefore a trade-off between number of iterations and thereby forward calculations and the cost of computing a perturbed image.

Algorithm 4 shows how an image is perturbed to generate a new image.

Algorithm 4: Perturbation of an Image

Input: Image Z, partial frequency distribution $\boldsymbol{\pi}$ of Z
Output: Perturbed image $\overline{Z}$
Initialization: set $\overline{\boldsymbol{\pi}} = \boldsymbol{\pi}$
Pick random voxel k
for *each voxel l around voxel k, i.e., $l \in \mathcal{D}_k$* **do**
  Erase the value of voxel l, i.e., z_l is unassigned
end
for *each unassigned voxel l around voxel k, i.e., $l \in \mathcal{D}_k$* **do**
  Simulate z_l given all assigned voxels in $\mathcal{N}_l$.
end

3.4 Updating the Frequency Distribution

As a new image is created by changing the value of a minority of the voxels, it would be time consuming to compute the frequency distribution of all voxel values of the new image when the frequency distribution of the old image is known. Recall that n is the maximum number of neighbors a voxel can have; inner voxels have exactly n neighbors. Therefore, in addiction to changing its own pattern value, changing the value of a voxel will affect the pattern value of at most n other voxels. This means that we obtain the frequency distribution of the new image by performing at most $n+1$ subtractions and $n+1$ additions per changed voxel to the entries of the already known frequency distribution.

The total number of subtractions and additions can be lowered further by exploiting the block structure of the set of voxels perturbed. The pattern value of a voxel will be changed when any of its neighboring voxels are perturbed, but the frequency distribution need only be updated twice for each affected voxel. We introduce a set of voxels Z^{aff}, which is the set of voxels who are affected when perturbing image Z into $\overline{Z}$, that is, the set of voxels whose pattern values are changed when perturbing image Z into image $\overline{Z}$

$$Z^{\text{aff}} = \{k \in Z\colon\ p_k \neq \overline{p}_k\}. \tag{9}$$

How the partial frequency distribution is updated when an image is perturbed is illustrated in Algorithm 5.

Algorithm 5: Update Partial Frequency Distribution of an Image

Input: Image Z, partial frequency distribution $\boldsymbol{\pi}$ of Z, perturbed image $\overline{Z}$, set of affected voxels Z^{aff}, set of pattern values $\mathbf{p}$ from the training image
Output: Partial frequency distribution $\overline{\boldsymbol{\pi}}$ of $\overline{Z}$
Initialization: set $\overline{\boldsymbol{\pi}} = \boldsymbol{\pi}$
for *each affected voxel, i.e.,* $k \in Z^{\text{aff}}$ **do**
  Extract pattern k from both Z and $\overline{Z}$
  Compute both pattern values p_k and $\overline{p}_k$
  if *the pattern* p_k *is present in the training image* **then**
    Subtract 1 from the corresponding entry of $\overline{\boldsymbol{\pi}}$
  end
  if *the pattern* $\overline{p}_k$ *is present in the training image* **then**
    Add 1 to the corresponding entry of $\overline{\boldsymbol{\pi}}$
  end
end

As seen in Algorithm 1, the FM method requires in total two computations of a frequency distribution, one for the training image and one for the initial image. The FM method requires one update of the partial frequency distribution per iteration. As the set of affected voxels Z^{aff} is expected to be much smaller than the total image Z, updating the partial frequency distribution will typically be much faster than recomputing the entire partial frequency distribution even for iterations that involve changing the values of a large set of voxels.

3.5 Multigrids

The multigrid approach from Strebelle (2002) that is based on the concept initially proposed by Gómez-Hernández (1991) and further developed by Tran (1994) can also be applied in the FM method. Coarsening the images allows the capture of large-scale structures with relatively small templates. As in the *snesim* algorithm, the results from a coarse image can be used to condition upon for a higher resolution image.

The multigrid approach is applied by running the FM method from Algorithm 1 multiple times. First, the algorithm is run on the coarsest level. Then the resulting image, with increased resolution, is used as a starting image on the next finer level, and so on. The resolution of an image can be increased by nearest neighbor interpolation.

4 Example: Crosshole Tomography

Seismic borehole tomography involves the measurement of seismic travel times between two or more boreholes in order to determine an image of seismic velocities in the intervening subsurface. Seismic energy is released from sources located in one borehole and recorded at multiple receiver locations in another borehole. In this way a dense tomographic data set that covers the interborehole region is obtained.

Consider a setup with two boreholes. The horizontal distance between them is ΔX and they both have the depth ΔZ. In each borehole a series of receivers and sources

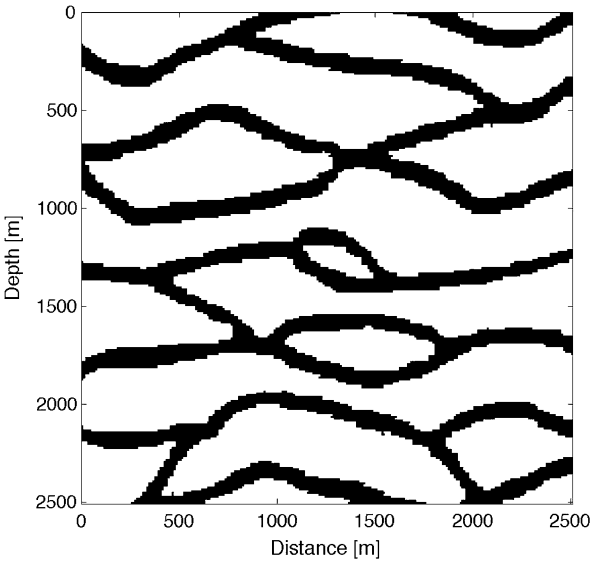


Fig. 3 Training image (resolution: 251 × 251 pixels)

Table 1 Parameter values for the test case

ΔX	500 m
ΔZ	1,200 m
Δx	10 m
Δz	10 m
d_{s}	250 m
d_{r}	100 m
v_{low}	1,600 m/s
v_{high}	2,000 m/s

is placed. The vertical domain between the two boreholes is divided into cells of dimensions Δx by Δz and it is assumed that the seismic velocity is constant within each cell. The model parameters of the problem are the propagation speeds of each cell. The observed data are the first arrival times of the seismic signals. For the series of sources and receivers in each borehole the distances between the sources are d_s and the distances between the receivers are d_r. We assume a linear relation between the data (first arrival times) and the model (propagation speed) from Eq. 1. The sensitivity of seismic signals is simulated as straight rays. However, any linear sensitivity kernel obtained using, for example, curvilinear rays or Fresnel zone-based sensitivity, can be used.

It is assumed that the domain consists of zones with two different propagation speeds, v_{low} and v_{high}. Furthermore a horizontal channel structure of the zones with high propagation speed is assumed. Figure 3 shows the chosen training image with resolution 251 cells by 251 cells where each cell is Δx by Δz. The training image is chosen to express the a priori information about the model parameters. The background (white pixels) represents a low velocity zone and the channel structures (black

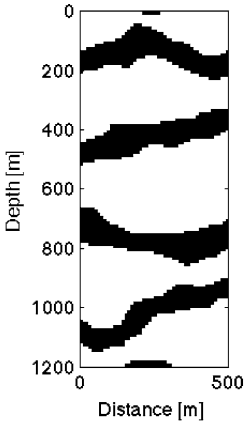


Fig. 4 Reference model (resolution: 50×120 pixels)

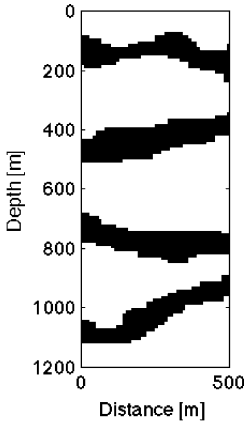


Fig. 5 Computed model for $\alpha = 1.8 \times 10^{-2}$ (resolution: 50×120 pixels)

pixels) are the high velocity zones. The problem is scalable and for the example we have chosen the parameters presented by Table 1.

The template function is chosen, such that the neighborhood of pixel k is the following set of pixels

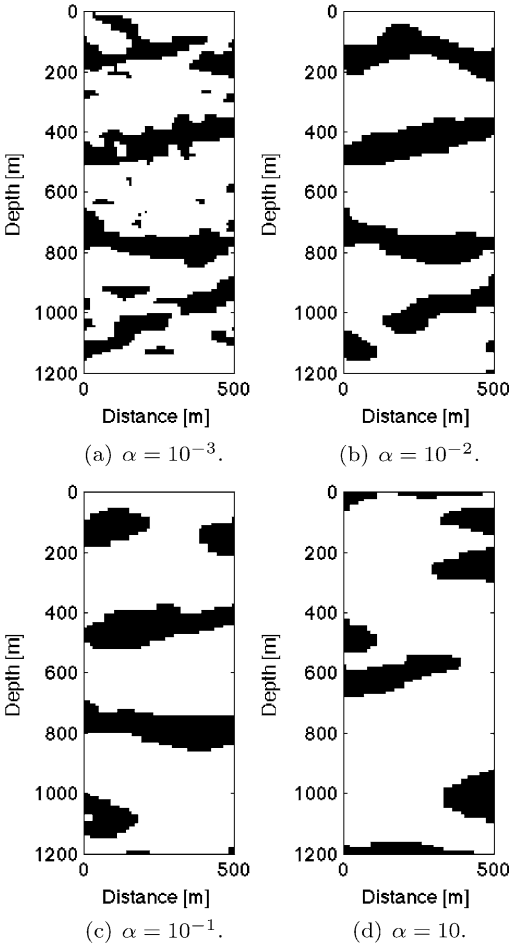
$$\mathcal{N}_k = \big\{ l \in Z \setminus \{k\} \colon |l_x - k_x| \leq 4, |l_z - k_z| \leq 3 \big\}.$$

Recall that pixel l has the coordinates (l_x, l_z); the first coordinate being the horizontal distance from the left borehole and the second coordinate being the depth, both measured in pixels. To compute a perturbed image, the domain used in Algorithm 4 is defined as follows

$$\mathcal{D}_k = \big\{ l \in Z \setminus Z^{\text{cond}} \colon |l_x - k_x| \leq 7, |l_z - k_z| \leq 7 \big\}.$$

The values of all pixels $l \in \mathcal{D}_k$ will be re-simulated using Sequential Simulation conditioned to the remaining pixels $l \notin \mathcal{D}_k$. We are not using any hard data in the example, which means $Z^{\text{cond}} = \emptyset$.

This choice of template function yields $n = 34$ where the geometrical shape of the neighborhood of inner pixels is a 7 pixels by 5 pixels rectangle. This is chosen based



(a) $\alpha = 10^{-3}$. (b) $\alpha = 10^{-2}$.

(c) $\alpha = 10^{-1}$. (d) $\alpha = 10$.

Fig. 6 The computed models for increasing values of α: (**a**) $\alpha = 10^{-3}$, (**b**) $\alpha = 10^{-2}$, (**c**) $\alpha = 10^{-1}$, (**d**) $\alpha = 10$

on the trends in the training image, where the distance of continuity is larger horizontally than vertically. However, it should be noted that this choice of template function is not expected to meet the assumptions of conditional independence of Sect. 2.2. The distance of continuity in the training image appears much larger horizontally than only seven pixels, and vertically the width of the channels is approximately ten pixels. This implies that, despite matched frequency distributions, a computed solution will not necessarily be recognized to have the same visual structures as the training image. The goal is solve the inverse problem which involves fitting the data and therefore, as our example will show, neighborhoods of this size are sufficient. The data-fitting term of the objective function guides the solution method, such that the structures from the training image are correctly reproduced. The low number of neighbors constrains the small-scale variations, which are not well-determined by the travel time data. However, the travel time data successfully determine the large-scale structures. The template function does not need to describe structures of the largest scales of the training image as long as the observed data are of a certain quality.

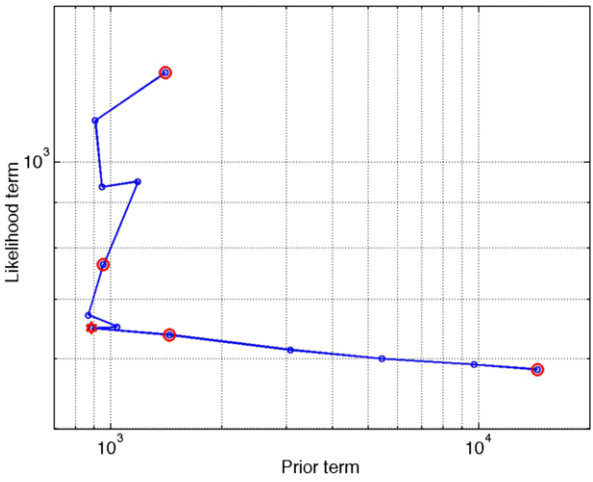


Fig. 7 L-curve used to determine the optimal α value. Models have been computed for 13 logarithmically distributed values of α ranging from 1 (*upper left corner*) to 10^{-3} (*lower right corner*). Each of the 13 models is marked with a blue circle. See the text for further explanation

Figure 4 shows the reference model that describes what is considered to be the true velocity profile between the two boreholes. The image has been generated by the *snesim* algorithm (Strebelle 2002) using the multiple point statistics of the training image. The arrival times $\mathbf{d}$ for the reference model $\mathbf{m}^{\text{ref}}$ are computed by a forward computation, $\mathbf{d} = G\mathbf{m}^{\text{ref}}$. We define the observed arrival times $\mathbf{d}^{\text{obs}}$ as the computed arrival times $\mathbf{d}$ added 5 % Gaussian noise. Figure 5 shows the solution computed using 15,000 iterations for $\alpha = 1.8 \times 10^{-2}$. The solution resembles the reference model to a high degree. The FM method detected the four channels; their location, width and curvature correspond to the reference model. The computations took approximately 33 minutes on a Macbook Pro 2.66 GHz Intel Core 2 Duo with 4 GB RAM.

Before elaborating on how the α value was determined, we present some of the models computed for different values of α. Figure 6 shows the computed models for four logarithmically distributed values of α between 10^{-3} and 10^{1}. It is seen how the model for lowest value of α is geologically unrealistic and does not reproduce the a priori expected structures from the training image as it primarily is a solution to the ill-posed, under-determined, data-fitting problem. As α increases, the channel structures of the training image are recognized in the computed models. However, for too large α values the solutions are dominated by the χ^2 term as the data have been deprioritized, and the solutions are not geologically reasonable either. As discussed, the chosen template is too small to satisfy the conditions from Sect. 2.2, yielding models that do in fact minimize the χ^2 distance, but do not reproduce the structures form the training image. The data misfit is now assigned too little weight to help compensate for the small neighborhoods, and the compromise between minimizing the data misfit and minimizing the dissimilarity that before worked out well is no longer present.

We propose to use the L-curve method (Hansen and O'Leary 1993) to determine an appropriate value of α. Figure 7 shows the value of $\chi^2(\mathbf{m}^{\text{FM}})$ versus the value of $\frac{1}{2}\|g(\mathbf{m}^{\text{FM}}) - \mathbf{d}^{\text{obs}}\|^2_{\mathbf{C}_{\text{d}}}$ for 13 models. The models have been computed for logarithmically distributed values of α ranging from 1 (upper left corner) to 10^{-3} (lower right corner). Each of the 13 models is marked with a blue circle. The models from Fig. 6

are furthermore marked with a red circle. The model from Fig. 5 is marked with a red star. We recognize the characteristic L-shaped behavior in the figure and the model from Fig. 5 is the model located in the corner of the L-curve. The corresponding value $\alpha = 1.8 \times 10^{-2}$ is therefore considered an appropriate value of α.

5 Conclusions

We have proposed the frequency matching method which enables us to quantify a probability density function that describes the multiple point statistics of an image. In this way, the maximum a posteriori solution to an inverse problem using training image-based complex prior information can be computed. The frequency matching method formulates a closed form expression for the a priori probability of a given model. This is obtained by comparing the multiple point statistics of the model to the multiple point statistics from a training image using a χ^2 dissimilarity distance.

Through a synthetic test case from crosshole tomography, we have demonstrated how the frequency matching method can be used to determine the maximum a posteriori solution. When the a priori distribution is used in inversion, a parameter α is required. We have shown how we are able to recreate the reference model by choosing this weighing parameter appropriately. Future work could focus on determining the theoretically optimal value of α as an alternative to using the L-curve method.

Acknowledgements The present work was sponsored by the Danish Council for Independent Research —Technology and Production Sciences (FTP grant no. 274-09-0332) and DONG Energy.

References

Arpat GB (2005) Sequential simulation with patterns. PhD thesis, Stanford University

Caers J, Hoffman T (2006) The probability perturbation method: a new look at Bayesian inverse modeling. Math Geol 38:81–100

Caers J, Zhang T (2004) Multiple-point geostatistics: a quantitative vehicle for integrating geologic analogs into multiple reservoir models. In: Grammer M, Harris PM, Eberli GP (eds) Integration of outcrop and modern analogs in reservoir modeling, AAPG Memoir 80, AAPG, Tulsa, pp 383–394

Cressie N, Davidson J (1998) Image analysis with partially ordered Markov models. Comput Stat Data Anal 29(1):1–26

Gómez-Hernández JJ (1991) A stochastic approach to the simulation of block conductivity fields conditioned upon data measured at a smaller scale. PhD thesis, Stanford University

Guardiano F, Srivastava RM (1993) Multivariate geostatistics: beyond bivariate moments. In: Geostatics-Troia, vol 1. Kluwer Academic, Dordrecht, pp 133–144

Hansen PC, O'Leary DP (1993) The use of the L-curve in the regularization of discrete ill-posed problems. SIAM J Sci Comput 14:1487–1503

Hansen TM, Cordua KS, Mosegaard K (2008) Using geostatistics to describe complex a priori information for inverse problems. In: Proceedings Geostats 2008. pp 329–338

Hansen TM, Cordua KS, Mosegaard K (2012) Inverse problems with non-trivial priors: efficient solution through sequential Gibbs sampling. Comput Geosci 16:593–611

Honarkhah M (2011) Stochastic simulation of patterns using distance-based pattern modeling. PhD dissertation, Stanford University

Jafarpour B, Khodabakhshi M (2011) A probability conditioning method (PCM) for nonlinear flow data integration into multipoint statistical facies simulation. Math Geosci 43:133–146
Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220:671–680
Peredo O, Ortiz JM (2010) Parallel implementation of simulated annealing to reproduce multiple-point statistics. Comput Geosci 37:1110–1121
Sheskin D (2004) Handbook of parametric and nonparametric statistical procedures. Chapman & Hal/CRC, London, pp 493–500
Stien M, Kolbjørnsen O (2011) Facies modeling using a Markov mesh model specification. Math Geosci 43:611–624
Strebelle S (2002) Conditional simulation of complex geological structures using multiple-point statistics. Math Geol 34:1–21
Suzuki S, Caers J (2008) A distance-based prior model parameterization for constraining solutions of spatial inverse problems. Math Geosci 40:445–469
Tarantola A (2005) Inverse problem theory and methods for model parameter estimation. Society for Industrial and Applied Mathematics, Philadelphia
Tarantola A, Valette B (1982) Inverse problems = quest for information. J Geophys 50:159–170
Tjelmeland H, Besag J (1998) Markov random fields with higher-order interactions. Scand J Stat 25:415–433
Tran TT (1994) Improving variogram reproduction on dense simulation grids. Comput Geosci 7:1161–1168