Monte Carlo sampling of solutions to inverse problems

Klaus Mosegaard
Niels Bohr Institute for Astronomy, Physics and Geophysics, Copenhagen
Albert Tarantola
Institut de Physique du Globe, Paris

This is a typeset LaTeX version of the paper originally published in
Journal of Geophysical Research, Vol. 100, No., B7, p 12,431–12,447, 1995.

Abstract

Probabilistic formulation of inverse problems leads to the definition of a probability distribution in the model space. This probability distribution combines a priori information with new information obtained by measuring some observable parameters (data). As, in the general case, the theory linking data with model parameters is nonlinear, the a posteriori probability in the model space may not be easy to describe (it may be multimodal, some moments may not be defined, etc.). When analyzing an inverse problem, obtaining a maximum likelihood model is usually not sufficient, as we normally also wish to have information on the resolution power of the data. In the general case we may have a large number of model parameters, and an inspection of the marginal probability densities of interest may be impractical, or even useless. But it is possible to pseudorandomly generate a large collection of models according to the posterior probability distribution and to analyze and display the models in such a way that information on the relative likelihoods of model properties is conveyed to the spectator. This can be accomplished by means of an efficient Monte Carlo method, even in cases where no explicit formula for the a priori distribution is available. The most well known importance sampling method, the Metropolis algorithm, can be generalized, and this gives a method that allows analysis of (possibly highly nonlinear) inverse problems with complex a priori information and data with an arbitrary noise distribution.

Introduction

Inverse problem theory is the mathematical theory describing how information about a parameterized physical system can be derived from observational data, theoretical relationships between model parameters and data, and prior information. Inverse problem theory is largely developed in geophysics, where the inquiry is how to infer information about the Earth's interior from physical measurements at the surface. Examples are estimation of subsurface rock density, magnetization, and conductivity from surface measurements of gravity or electromagnetic fields. An important class of complex inverse problems is found in seismology, where recorded seismic waves at the Earth's surface or in boreholes are used to compute estimates of mechanical subsurface parameters.

In what follows, any given set of values representing a physical system, we call a model. Every model $\mathbf{m}$ can be considered as a point in the model space $\mathcal{M}$. We will define different probability densities over $\mathcal{M}$. For instance, a probability density $\rho(\mathbf{m})$ will represent our a priori information on models, and another probability density, $\sigma(\mathbf{m})$ will represent our a posteriori information, deduced from $\rho(\mathbf{m})$ and from the degree of fit between data predicted from models and actually observed data. In fact, we will use the expression $\sigma(\mathbf{m}) = k\rho(\mathbf{m})L(\mathbf{m})$ [see *Tarantola*, 1987], where $L(\mathbf{m})$, the likelihood function, is a measure of the degree of fit between data predicted from the model $\mathbf{m}$ and the observed data (k is an appropriate normalization constant). Typically, this is done through the introduction of a misfit function $S(\mathbf{m})$, connected to $L(\mathbf{m})$ through an expression like $L(\mathbf{m}) = k\exp(-S(\mathbf{m}))$.

In seismology, the misfit function usually measures the degree of misfit between observed and computed seismograms as a function of the subsurface model parameters. It usually has many secondary minima. In terms of the probability density in the model space, we deal typically with a (possibly degenerate) global maximum, representing the most likely solution, and a large number of secondary maxima, representing other possible solutions. In such cases, a local search for the maximum likelihood solution using, for instance, a gradient method, is very likely to get trapped in secondary maxima. This problem is avoided when using a global search method. A global search is not confined to uphill (or downhill) moves in the model space and is therefore less influenced by the presence of local optima. Some global methods are not

influenced at all.

The simplest of the global search methods is the exhaustive search. A systematic exploration of the (discretized) model space is performed, and all models within the considered model subspace are visited. Although this method may be ideal for problems with low dimensionality (i.e., with few parameters), the task is computationally unfeasible when problems with many model parameters are considered.

When analyzing highly nonlinear inverse problems of high dimensionality, it is therefore necessary to severely restrict the number of misfit calculations, as compared to the exhaustive search. One way to do this is to use a Monte Carlo search, which consists of a (possibly guided) random walk in the model space. A Monte Carlo search extensively samples the model space, avoids entrapment in local likelihood maxima, and therefore provides a useful way to attack such highly nonlinear inverse problems.

In resolution studies, the advantages of Monte Carlo methods become even more significant. Resolution analysis carried out by means of local methods gives erroneous results due to the inherent assumption that only one minimum for the misfit function exists. However, a Monte Carlo method can take advantage of the fact that all local likelihood maxima will be sampled, provided a sufficient number of iterations are performed.

Early geophysical examples of solution of inverse problems by means of Monte Carlo methods, are given by *Keilis-Borok and Yanovskaya* [1967] and *Press* [1968, 1971]. Press made the first attempts at randomly exploring the space of possible Earth models consistent with seismological data. More recent examples are given by *Rothman* [1985, 1986], who nicely solved a strongly nonlinear optimization problem arising in seismic reflection surveys, and *Landa et al.* [1989], *Mosegaard and Vestergaard* [1991], *Koren et al.*, [1991], and *Cary and Chapman* [1988], who all used Monte Carlo methods within the difficult context of seismic waveform fitting. Cary and Chapman and Koren et al. described the potential of Monte Carlo methods, not only for solving a model optimization problem but also for performing an analysis of resolution in the inverse problem.

The idea behind the Monte Carlo method is old, but its actual application to the solution of scientific problems is closely connected to the advent of modern electronic computers. J. von Neumann, S. Ulam and E. Fermi used the method in nuclear reaction studies, and the name "the Monte Carlo method" (an allusion to the famous casino) was first used by *Metropolis and Ulam* [1949]. Four years later, Metropolis et al. [1953] introduced an algorithm, now known as the Metropolis algorithm, that was able to (asymptotically) sample a space according to a Gibbs-Boltzmann distribution. This algorithm was a biased random walk whose individual steps (iterations) were based on very simple probabilistic rules.

It is not difficult to design random walks that sample the posterior probability density $\sigma(\mathbf{m})$. However, in cases where $\sigma(\mathbf{m})$ has narrow maxima. these maxima (which are the most interesting features of $\sigma(\mathbf{m})$) will be very sparsely sampled (if sampled at all). In such cases, sampling of the model space can be improved by importance sampling, that is, by sampling the model space with a probability density as close to $\sigma(\mathbf{m})$ as possible. *Cary and Chapman* [1988] used the Monte Carlo method to determine $\sigma(\mathbf{m})$ for the refraction seismic waveform inversion problem, where the travel times were used as data, as well as waveforms, and the model parameters were the depths as a function of velocity. They improved the sampling of the model space by using a method described by *Wiggins* [1969, 1972] in which the model space was sampled according to the prior distribution $\rho(\mathbf{m})$. This approach is superior to a uniform sampling by crude Monte Carlo. However, the peaks of the prior distribution are typically much less pronounced than the peaks of the posterior distribution. Moreover, the peaks of the two distributions may not even coincide. It would therefore be preferable to draw sample models from the model space according to a probability distribution which is close to the posterior distribution $\sigma(\mathbf{m})$, the idea being to use a probability distribution that tends to $\sigma(\mathbf{m})$ as iterations proceed.

Geman and Geman [1984] discussed an application of simulated annealing to Bayesian image restoration. For their particular inverse problem, a two-dimensional deconvolution problem, they derived an expression for the posterior distribution from (1) the prior distribution, (2) a model of the convolutional two-dimensional image blurring mechanism, and (3) the parameters of the Gaussian noise model. By identifying this posterior distribution with a Gibbs-Boltzmann distribution, they performed a maximum a posteriori estimation in the model space, using a simulated annealing algorithm. In their paper, they mention the possibility of using the simulated annealing algorithm, not only for maximum a posteriori estimation but also to sample the model space according to the posterior distribution. However, they did not pursue this possibility further, nor did they describe how to extend this idea to inverse problems in general.

Marroquin et al. [1987] adopted an approach similar to that of Geman and Geman. However, they used the Metropolis algorithm to generate the posterior distribution, from which they computed model estimates. One of the problems raised by these authors was that their Bayesian approach requires an explicit formula for the a priori distribution.

Recent examples of using Bayes theorem and the Metropolis algorithm for generating a posteriori probabilities for an inverse problem are given by *Pedersen and Knudsen* [1990] and *Koren et al.* [1991].

In the present paper we will describe a method for random sampling of solutions to an inverse problem. The solutions are sampled at a rate proportional to their a posteriori probabilities, that is, models consistent with a priori information as well as observations are picked most often, whereas models that are in incompatible with either a priori information or observations (or both) are rarely sampled.

In brief our sampling algorithm can be described as consisting of two components. The first component generates a priori models, that is, models sampled with a frequency distribution equal to the a priori probability distribution in the model space. This is accomplished by means of a random walk, a kind of "Brownian motion" in the model space. The second component accepts or rejects attempted moves of the a priori random walk with probabilities that depend on the models ability to reproduce observations. Output from the combined algorithm consists of a collection of models that passed the test performed in the second component. This collection of models is shown to have a frequency distribution that is (asymptotically) proportional to the a posteriori probability distribution in the model space.

It is an important property of our method that in contrast to usual Bayesian inverse calculations, the a priori distribution need not be given by an explicit formula. In fact, the first component of our algorithm may consist of a large number of mutually dependent sub-processes, each of which generates part of the a priori models.

The definition of which models are accessible from a given model is an essential ingredient of the method, from a practical point of view. We will "jump" from a model to a neighboring model. But, what is a neighbor? The theory to be developed below is independent of the particular choice of model perturbations to be considered, but, as illustrated below, a bad definition of model neighborhood may lead to extremely inefficient algorithms.

Probabilistic Formulation of Inverse Problems

Parameters Taking Continuous Values

The "forward problem" is the problem of predicting (calculating) the "data values" $\mathbf{d}_{\rm cal} = \{d^1_{\rm cal}, d^2_{\rm cal}, \ldots\}$ that we should observe when making measurements on a certain system. Let the system be described (parameterized) by a parameter set $\mathbf{m} = \{m^1, m^2, \ldots\}$. One generally writes as

$$\mathbf{d}_{\rm cal} = g(\mathbf{m}) \tag{1}$$

the generally nonlinear, mapping from the model space $\mathcal{M}$ into the data space $\mathcal{D}$ that solves the forward problem.

In its crudest formulation, the "inverse problem" consists of the following question: An actual measurement of the data vector $\mathbf{d}$ gave the value $\mathbf{d}_{\rm obs} = \{d^1_{\rm obs}, d^2_{\rm obs}, \ldots\}$. Which is the actual value of the model parameter vector $\mathbf{m}$?

This problem may well be underdetermined, due to lack of significant data or due to experimental uncertainties. It can also be overdetermined, if we repeat similar measurements. Usually, it is both. A better question would have been: What information can we infer on the actual value of the model parameter vector $\mathbf{m}$?

The "Bayesian approach" to inverse problems, describes the "a priori information" we may have on the model vector, by a probability density $\rho(\mathbf{m})$. Then, it combines this information with the information provided by the measurement of the data vector and with the information provided by the physical theory, as described for instance by equation (2), in order to define a probability density $\sigma(\mathbf{m})$ representing the "a posteriori information". This a posteriori probability density describes all the information we have. It may well be multimodal, not have a mathematical expectation, have infinite variances, or some other pathologies, but it constitutes the complete solution to the inverse problem.

Whatever the particular approach to the problem may be [e.g., *Backus*, 1970a,b,c; *Tarantola and Valette*, 1982a; *Tarantola*, 1987], we end up with a solution of the form

$$\sigma(\mathbf{m}) = k\,\rho(\mathbf{m})L(\mathbf{m}), \tag{2}$$

where k is an appropriate normalization constant. The a posteriori probability density $\sigma(\mathbf{m})$ equals the a priori probability density $\rho(\mathbf{m})$ times a "likelihood function" $L(\mathbf{m})$ which, crudely speaking, measures the fit between observed data and data predicted from the model $\mathbf{m}$ (see an example below).

As an example, when we describe experimental results by a vector of observed values $\mathbf{d}_{\rm obs}$ with Gaussian experimental uncertainties described by a covariance matrix $\mathbf{C}$, then

$$L(\mathbf{m}) = k \exp\left[\frac{1}{2}\big(\mathbf{g}(\mathbf{m}) - \mathbf{d}_{\rm obs}\big)^t \mathbf{C}^{-1}\big(\mathbf{g}(\mathbf{m}) - \mathbf{d}_{\rm obs}\big)\right]. \tag{3}$$

If, instead, we describe experimental uncertainties using a Laplacian function, where $d^i_{\rm obs}$ are the "observed values" and σ^i are the estimated uncertainties, then

$$L(\mathbf{m}) = k \exp\left[-\sum_i \frac{|g^i(\mathbf{m}) - d^i_{\rm obs}|}{\sigma^i}\right]. \tag{4}$$

As a last example (to be used below), if the measured data values $d^i_{\rm obs}$ are contaminated by statistically independent, random errors ε_i given by a double Gaussian probability density function,

$$f(\varepsilon) = k\left[a \exp\left(-\frac{\varepsilon^2}{2\sigma_1^2}\right) + b \exp\left(-\frac{\varepsilon^2}{2\sigma_2^2}\right)\right], \tag{5}$$

then

$$L(\mathbf{m}) = k \prod_i \left[a \exp\left(-\frac{(g^i(\mathbf{m}) - d^i_{\text{obs}})^2}{2\sigma_1^2} \right) + b \exp\left(-\frac{(g^i(\mathbf{m}) - d^i_{\text{obs}})^2}{2\sigma_2^2} \right) \right] . \quad (6)$$

These three examples are very simplistic. While in this paper we show the way to introduce realistic a priori information in the model space, we do not attempt to advance in the difficult topic of realistically describing data uncertainties.

Discretization of Parameters

So far, the theory has been developed for parameters that although finite in number may take continuous values. Then, at any point $\mathbf{m}_i$ we can define a probability density $f(\mathbf{m}_i)$, but not a probability, which can only be defined for a region of the space:

$$P(\mathbf{m} \in \mathcal{A}) = \underbrace{\int dm^1 \int dm^2 \cdots}_{\mathcal{A}} f(\mathbf{m}). \quad (7)$$

Here, $m^1, m^2 \ldots$ denote the different components of the vector $\mathbf{m}$.

For numerical computations, we discretize the space by defining a grid of points, where each point represents a surrounding region $\Delta m^1 \Delta m^2 \ldots$, small enough for the probability densities under consideration to be almost constant inside it. Then, when we say "the probability of the point $\mathbf{m}_i$" we mean "the probability of the region $\Delta m^1 \Delta m^2 \ldots$ surrounding the point $\mathbf{m}_i$". In the limit of an infinitely dense grid and assuming a continuous $f(\mathbf{m})$, "the probability of the point $\mathbf{m}_i$ " tends to

$$f_i = f(\mathbf{m}_i) \Delta m^1 \Delta m^2 \ldots . \quad (8)$$

The discrete version of equation (2) is then

$$\sigma_i = \frac{\rho_i L(\mathbf{m}_i)}{\sum_j \rho_j L(\mathbf{m}_j)} , \quad (9)$$

where

$$\sigma_i = \sigma(\mathbf{m}_i) \Delta m^1 \, \Delta m^2 \ldots , \quad (10)$$

and

$$\rho_i = \rho(\mathbf{m}_i) \Delta m^1 \, \Delta m^2 \ldots . \quad (11)$$

For simplicity, we will rather write

$$\sigma_i = \frac{\rho_i \, L_i}{\sum_j \rho_j \, L_j} , \quad (12)$$

where we use the notation

$$L_i = L(\mathbf{m}_i) \quad (13)$$

(note that $\Delta m^1 \, \Delta m^2 \ldots$ does not enter into the definition of L_i).

Once the probability (12) has been defined, we could design a method to sample directly the posterior probability σ_i (and, in fact, the methods below could be used that way). But any efficient method will proceed by first sampling the prior probability ρ_i . It will then modify this sampling procedure in such a way that the probability σ_i is eventually sampled. This, after all, only corresponds to the Bayesian viewpoint on probabilities: one never creates a probability ex nihilo but rather modifies some prior into a posterior.

Monte Carlo Sampling of Probabilities

Essentially, the sampling problem can be stated as follows: given a set of points in a space, with a probability p_i attached to every point i , how can we define random rules to select points such that the probability of selecting point i is p_i ?

Terminology

Consider a random process that selects points in the model space. If the probability of selecting point i is p_i , then the points selected by the process are called "samples" of the probability distribution $\{p_i\}$. Depending on the random process, successive samples $i, j, k, \ldots$ may be dependent or independent, in the sense that the probability of sampling k may or may not depend on the fact that i and j have just been sampled.

An important class of efficient Monte Carlo (i.e., random) sampling methods is the random walks. The possible paths of a random walk define a graph in the model space (see Figure 1). All models in the discrete model space are nodes of the graph, and the edges of the graph define the possible steps of the random walk. The graph defines the "neighborhood" of a model as the set of all models directly connected to it. Sampling is then made by defining a random walk on the graph: one defines the probability P_{ij} for the random walker to go to point i if it currently is at the neighboring point j . P_{ij} is called the "transition probability". (As, at each step, the random walker must go somewhere, including the possibility of staying at the same point, P_{ij} satisfies $\sum_i P_{ij} = 1$.) For the sake of mathematical simplicity, we shall always assume that a graph connects any point with itself: staying at the point is considered as a "transition" (a"step"), and the current point, having been reselected, contributes with one more sample.

Consider a random walk, defined by the transition probabilities $\{P_{ij}\}$, and assume that the model where it is initiated is only known probabilistically: there is a probability q_i that the random walk is initiated at point i . Then, when the number of steps tends to infinity, the

probability that the random walker is at point i will converge to some other probability p_i [*Feller*, 1970]. We say that $\{p_i\}$ is an "equilibrium probability distribution" of

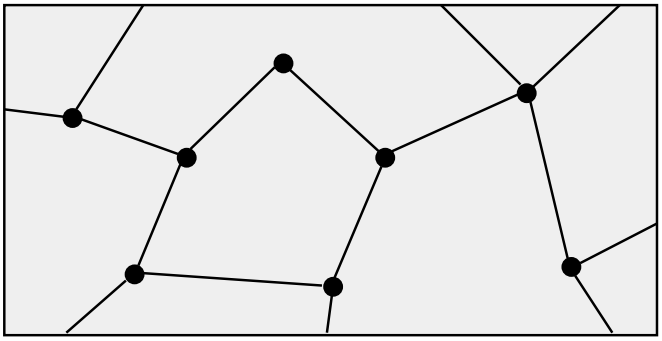

Figure 1: Part of a graph in the model space. The graph defines the possible steps of a random walk in the space. The random walk obeys some probabilistic rules that allow it to jump from one model to a connected model in each step. The random walker will, asymptotically, have some probability, say p_i , to be at point i at a given step. The neighborhood of a given model is defined as the models to which a random walker can go in one step, if it starts at the given model. Thus a neighborhood is defined solely through the graph and does not need to be a metric concept.

$\{P_{ij}\}$. (Then, $\{p_i\}$ is an eigenvector with eigenvalue 1 of $\{P_{ij}\}$: $\sum_j P_{ij} p_j = p_i$.) If the random walk always equilibrates at the same probability $\{p_i\}$, independent of the initial probability $\{q_i\}$, then there is only one equilibrium probability $\{p_i\}$. (Then, $\{p_i\}$ is a unique eigenvector of $\{P_{ij}\}$.) This is the case if the graph is "connected", that is, if it is possible to go from any point to any other point in the graph (in a sufficient number of steps) [*Feller*, 1970].

Many random walks can be defined that have a given probability distribution $\{p_i\}$ as their equilibrium probability. Some random walks converge more rapidly than others to their equilibrium probability. Successive models $i, j, k, \ldots$ obtained with a random walk will, of course, not be independent unless we only consider models separated by a sufficient number of steps. Instead of letting p_i represent the probability that a (single) random walker is at point i (in which case $\sum_i p_i = 1$), we can let p_i be the number of "particles" at point i . Then, $\sum_i p_i$ represents the total number of particles. None of the results presented below will depend on the way $\{p_i\}$ is normalized.

If, at some moment, the probability for the random walker to be at a point j is p_j and the transition probabilities are P_{ij} , then $f_{ij} = P_{ij} p_j$ represents the probability that the next transition will be from j to i while P_{ij} is the conditional probability of going to point i if the random walker is at j , f_{ij} is the unconditional probability that the next step will be a transition to i from j .

When p_i is interpreted as the number of particles at point i , f_{ij} is called the "flow", as it can be interpreted as the number of particles going to point i from point j in a single step. (The flow corresponding to an equilibrated random walk has the property that the number of particles p_i at point i is constant in time. Thus that a random walk has equilibrated at a distribution $\{p_i\}$ means that in each step, the total flow into a given point is equal to the total flow out from the point. Since each of the p_i particles at point i must move in each step (possibly to point i itself), the flow has the property that the total flow out from point i and hence the total flow into the point must equal p_i : $\sum_j f_{ij} = \sum_k f_{ki} = p_i$.) The concept of flow is important for designing rules that sample probabilities (see Appendix A).

Naïve Walks

Consider an arbitrary (connected) graph, as the one suggested in Figure 1, and denote by n_i the number of neighbors of point i (including the point i itself). Consider also a random walker that performs a "naïve random walk". That is, when he is at some point j , he moves to one of j's neighbors, say neighbor i , chosen uniformly at random (with equal probability). It is easy to prove (see Appendix B) that the random walk, so defined equilibrates at the probability distribution given by $p_i = n_i / \sum_j n_j$, i.e., with all points having a probability proportional to their number of neighbors.

Uniform Walks

Consider now a random walker that when he is at some point j , first chooses, uniformly at random, one of j's neighbors, say neighbor i , and then uses the following rule to decide if he moves to i or if he stays at j :

1. If $n_i \leq n_j$ (i.e., if the "new" point has less neighbors than the "old" point (or the same number), then always move to i .
2. If $n_i > n_j$ (i.e., if the "new" point has more neighbors than the "old" point), then make a random decision to move to i , or to stay at j , with the probability n_j / n_i of moving to i .

It is easy to prove (see Appendix B) that the random walk so defined equilibrates at the uniform probability, i.e., with all points having the same probability. This method of uniform sampling was first derived by *Wiggins* [1969].

The theory developed so far is valid for general, discrete (and finite) spaces, where the notion of metric is not necessarily introduced. In the special case of metric, Euclidean spaces, it is possible to choose Cartesian

coordinates, and to define the points in the space, where the random walk will be made, as a standard Cartesian grid of points. Let us, for instance, choose a graph as the one indicated in Figure 2. Then, away from the boundaries, the rule above degenerates into a (uniform) random choice of one of the $2N+1$ neighbors that any point has (including itself) in a space of dimension N . It can be shown (see Appendix B) that the walks so defined produce symmetric flows.

Modification of Random Walks

Assume that some random rules are given that define a random walk having $\{\rho_i\}$ as its equilibrium probability (uniform or not). How can the rules be modified so that the new random walk equilibrates at the probability.

$$\sigma_i = \frac{\rho_i L_i}{\sum_j \rho_j L_j} \ ? \tag{14}$$

Consider the following situation. Some random rules define a random walk that samples the prior probability $\{\rho_i\}$. At each step, the random walker is at point j , and

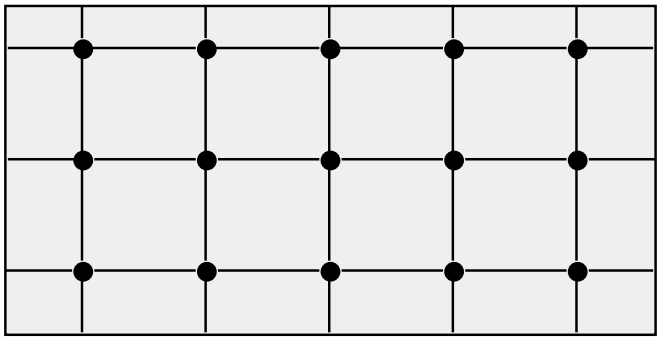

Figure 2: Part of a Cartesian graph in an Euclidean space. In this case, the definition of rules that sample points with the uniform probability is trivial.

an application of the rules would lead to a transition to point i . If that "proposed transition" $i \leftarrow j$ was always accepted, then the random walker would sample the prior probability $\{\rho_i\}$. Let us, however, instead of always accepting the proposed transition $i \leftarrow j$, sometimes thwart it by using the following rule to decide if he is allowed to move to i or if he is forced to stay at j :

1. If $L_i \geq L_j$ (i.e., if the "new" point has higher (or equal) likelihood than the "old" point), then accept the proposed transition to i .
2. If $L_i < L_j$ (i.e., if the "new" point has lower likelihood than the "old" point), then make a random decision to move to i , or to stay at j , with the probability L_i/L_j of moving to i .

Then it can be proved (see Appendix C) that the random walker will sample the posterior probability defined by equation (14). This modification rule, reminiscent of the Metropolis algorithm, is not the only one possible (see Appendix C).

To see that our algorithm degenerates into the Metropolis algorithm [*Metropolis et al.*, 1953] when used to sample the Gibbs-Boltzmann distribution, put $q_j = \exp(-E_j/T)/\sum_i \exp(-E_i/T)$, where E_j is an "energy" associated to the j-th point in the space and T is a "temperature". The summation in the denominator is over the entire space. In this way, our acceptance rule becomes the classical Metropolis rule: point i is always accepted if $E_i \leq E_j$, but if $E_i > E_j$, it is only accepted with probability $p_{ij}^{\text{acc}} = \exp(-(E_i - E_j)/T)$. Accordingly, we will refer to the above acceptance rule as the "Metropolis rule".

As an example, let us consider the case of independent, identically distributed Gaussian uncertainties. Then the likelihood function describing the experimental uncertainties (equation (3)) degenerates into

$$L(\mathbf{m}) = k \exp\left(-\frac{S(\mathbf{m})}{s^2}\right) \ , \tag{15}$$

where

$$S(\mathbf{m}) = \frac{1}{2}\sum_{i=1}^{N} \left(g^i(\mathbf{m}) - d^i_{\text{obs}}\right)^2 \tag{16}$$

is the misfit function, $\mathbf{m}$ is a model vector, $\mathbf{d}$ is a data vector, $\mathbf{g}(\mathbf{m})$ is the forward modeling function, and s^2 is the total "noise" variance. In this example, s^2 is the same for all N data values. The acceptance probability for a perturbed model becomes in this case

$$P_{\text{accept}} = \begin{cases} 1 & \text{if } S(\mathbf{m}_{\text{new}}) \leq S(\mathbf{m}_{\text{old}}) \\ \exp(-\frac{\Delta S}{s^2}) & \text{if } S(\mathbf{m}_{\text{new}}) > S(\mathbf{m}_{\text{old}}) \end{cases} , \tag{17}$$

where

$$\Delta S = S\big(\mathbf{m}_{\text{new}}\big) - S\big(\mathbf{m}_{\text{old}}\big) \ . \tag{18}$$

This means that the perturbation is accepted if the perturbed model improves the data fit, and has a probability of being accepted of $P_{\text{accept}} = \exp(-\Delta S/s^2)$ if it degrades the data fit. From (17) we see that in the case of uniform a priori distribution, our algorithm becomes identical to the traditional Metropolis algorithm by identifying the misfit function S with the thermodynamic energy E and by identifying the noise variance s^2 with (k times) the thermodynamic temperature T .

Starting a Random Walk

We have just shown how a random walk sampling some prior probability $\{\rho_i\}$ can be modified by the Metropolis rule to sample the posterior probability $\{\sigma_i\}$. This

procedure is very suitable for solution of inverse problems. Usually, we will define some probabilistic rules that, when applied directly, would generate models $\mathbf{m}_1, \mathbf{m}_2, \ldots$ that, by definition, would be samples of the prior probability $\{\rho_i\}$. The application of the Metropolis rule defined above will modify this random walk in the model space so that it produces samples of the posterior probability $\{\sigma_i\}$ instead.

The fact that we have a random walk that samples the prior does not imply that we have an expression that allows us to calculate the value of the prior probability ρ_i of any model $\mathbf{m}_i$. The numerical example below gives an example of this. Of course, using the random walk that samples the prior and making the histograms of the models selected would be a numerical way of obtaining the value of the prior probability ρ_i for every model $\mathbf{m}_i$, but this is not a question that normally arises.

Using random rules that, if unmodified, generate samples of the prior and using the Metropolis rule to modify this random walk in order to sample the posterior corresponds to the Bayesian way of modifying a prior probability into a posterior. This approach will usually lead to efficient random walks, since the algorithm only explores the (usually) very limited subset of models that are consistent with our a priori information.

It often happens that we have data of different nature, as for instance in geophysics, when we have gravity, magnetic, or seismic data. Then, typically, data uncertainties are independent, and the total likelihood of a model, $L(\mathbf{m})$, can be expressed as a product of partial likelihoods: $L(\mathbf{m}) = L_1(\mathbf{m})L_2(\mathbf{m})\ldots$, one for each data type. Using the Metropolis rule directly to the total likelihood $L(\mathbf{m})$ would force us to solve the full forward problem (usually the most time-consuming part of the algorithm) to every model proposed by the prior random walk. Instead, we can use the Metropolis rule in cascade: If the random walk sampling the prior is modified first by considering the partial likelihood; $L_1(\mathbf{m})$, then we define a random walk that samples the product of the prior probability density $\rho(\mathbf{m})$ and $L_1(\mathbf{m})$. In turn, this random walk can be modified by considering the partial likelihood $L_2(\mathbf{m})$, and so on, until the posterior probability density that takes into account the total data set is sampled. Practically this means that, once a model is proposed by the rules sampling the prior, the forward problem is solved for the first data subset. The proposed model may then be accepted or rejected. If it is rejected by the Metropolis rule (typically when there is a large misfit between the synthetic data and the observed data for this first data subset), then there is no need to solve the forward problem for the other data subsets, and the rules sampling the prior have to propose a new model. More generally: Each time the Metropolis rule rejects a model at some stage of the algorithm, we go back to the lower level and propose a new model. When the solution of the forward modeling is inexpensive for certain data subsets, using this "cascade rule" may render the algorithm much more efficient than using the Metropolis rule to the total data set.

If, for some reason, we are not able to directly design a random walk that samples the prior, but we have an expression that gives the value of the prior probability ρ_i for any model $\mathbf{m}_i$ (an example is given by expression (19) below), we can, for instance, start a random walk that samples the model space with uniform probability (see the section on uniform walks). Using the Metropolis rules given above but replacing the likelihood values L_i by the prior probabilities ρ_i , we will obviously produce a random walk that samples the prior (the product of a constant times ρ_i equals ρ_i). Then, in cascade, we can use the Metropolis rule, with the likelihood values L_i , to modify this random walk into a random walk that samples the posterior probability $\sigma_i = \text{const}\,\rho_i L_i$.

A second option is to modify directly a uniform random walk (using the Metropolis rule above but with the product $\rho_i L_i$ instead of L_i) into a walk that directly samples the posterior, but this results, generally, in an inefficient random walk.

Multistep Iterations

An algorithm will converge to a unique equilibrium distribution if the graph that describes the move of a random walker in a single iteration is connected [*Feller*, 1970]. Often, it is convenient to split up an iteration in a number of steps, having its own graph and its own transition probabilities. A typical example is a random walk on a set of discrete points in an N-dimensional Euclidean space, as the one suggested in Figure 2. In this case the points are located in a regular grid having N mutually perpendicular axes, and one is typically interested in dividing an iteration of the random walk into N steps, where the nth move of the random walker is in a direction parallel to the nth axis.

The question is now: if we want to form an iteration consisting of a series of steps, can we give a sufficient condition to be satisfied by each step such that the complete iteration has the desired convergence properties? It is easy to see that if the individual steps in an iteration all have the same distribution $\{p_i\}$ as an equilibrium distribution (not necessarily unique), then the complete iteration also has $\{p_i\}$ as an equilibrium distribution. (The transition probability matrix for a complete iteration is equal to the product of the transition probability matrices for the individual steps. Since the vector of equilibrium probabilities is an eigenvector with eigenvalue 1 for each of the step transition probability matrices, it is also an eigenvector with eigenvalue 1, and hence the equilibrium distribution, for the transition probability matrix

for the complete iteration.) If this distribution is to be the unique equilibrium distribution for the complete iteration, then the graph of the complete iteration must be connected. That is, it must be possible to go from any point to any other point by performing iterations consisting of the specified steps.

If the steps of an iteration satisfy these sufficient conditions, there is also another way of defining an iteration with the desired, unique equilibrium distribution. Instead of performing an iteration as a series of steps, it is possible to define the iteration as consisting of one of the steps, chosen randomly (with any distribution having nonzero probabilities) among the possible steps (see Appendix D). Of course, a step of an iteration can, in the same way, be built from substeps and in this way acquire the same (not necessarily unique) equilibrium distribution as the substeps.

Sampling the a Priori Probability Density

We have previously assumed that we were able to sample the a priori probability density $\rho(\mathbf{m})$. Let us see how this can be achieved.

There are two ways of defining the a priori probability distribution:

1. By defining a (pseudo) random process (i.e., a set, of pseudo random rules) whose output is models assumed to represent pseudo random realizations of $\rho(\mathbf{m})$
2. By explicitly giving a formula for the a priori probability density $\rho(\mathbf{m})$.

Let us see an example of each.

First Example

From nearby wells we may have found that in a certain area of locally horizontal stratification, the distribution of layer thicknesses is approximately an exponential distribution, and the mass densities in the layers follow a log-normal distribution. Hence we can decide to generate one dimensional Earth models for mass density by the following random walk in the model space:

In each iteration:

1. Select a layer uniformly at random.
2. Choose a new value for the layer thickness according to the exponential distribution.
3. Choose a value for the mass density inside the layer, according to the log-normal distribution.

If we decide to discretize the model at constant Δz intervals, $\mathbf{m} = \{\rho(z_1), \rho(z_2), \ldots\}$ will have some probability distribution (representing our a priori knowledge) for the parameters $\{\rho(z_1), \rho(z_2), \ldots\}$ which we may not need to characterize explicitly.

In this example, the pseudo random procedure produces, by its very definition, samples $\mathbf{m}_1, \mathbf{m}_2, \ldots$ of the a priori probability density $\rho(\mathbf{m})$. These samples will be the input to the Metropolis decision rule. We recommend in particular this way of handling the a priori information, as it allows arbitrarily complex a priori information to enter the solution to an inverse problem. For an example of this procedure, see the section on numerical example.

Second Example

We may choose the probability density

$$\rho(\mathbf{m}) = k \exp\left(- \sum_{\alpha} \frac{|m^{\alpha} - m^{\alpha}_{\text{prior}}|}{\sigma^{\alpha}} \right) , \qquad (19)$$

where m^{α} represent components of the vector $\mathbf{m}$.

In this example, where we only have an expression for $\rho(\mathbf{m})$, we have to generate samples from this distribution. This can be done in many different ways. One way is to start with a naïve walk, as described above, and then use the Metropolis rule to modify it, in order to sample $\rho(\mathbf{m})$.

Sampling the a Posteriori Probability Density

In the previous section we described how to perform a random walk in the model space producing samples $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \ldots$ of the a priori probability $\rho(\mathbf{m})$. In order to obtain samples of the a posteriori probability $\sigma(\mathbf{m}) = k\rho(\mathbf{m})L(\mathbf{m})$ we simply need to use the results given in the section on modification of random walks: if $\mathbf{m}_j$ is the "current point" and if the random walk sampling the prior would move from point $\mathbf{m}_j$ to point $\mathbf{m}_i$ (and whatever the used rules may be), accept the move if $L(\mathbf{m}_i) \geq L(\mathbf{m}_j)$, and decide randomly to accept or reject the move if $L(\mathbf{m}_i) < L(\mathbf{m}_j)$, with a probability $P = L(\mathbf{m}_i)/L(\mathbf{m}_j)$ of accepting the move.

Numerical Example

We now illustrate the theory developed in this paper with the inversion of gravity data. This is a classical example for testing any theory of inversion, and similar examples are given by *Dorman* [1975], *Parker* [1977] and *Jackson* [1979].

As the relationship between mass density and gravity data is strictly linear, one may wonder why we should illustrate a Monte Carlo method, with its inherent ability

to solve nonlinear problems, with the gravity inversion example. The reason is that our major concern is not the possibility of solving nonlinear problems, but the possibility of using, in standard geophysical inverse problems, realistic a priori information in the model space and realistic description of data uncertainties. This is what forces us to leave the comfortable realm of least squares and related methods and to develop the notions described here. It should be noted that the complex a priori knowledge used in this example renders the a posteriori distribution non-Gaussian.

The Problem

We consider a subsurface with a vertical fault, extending from the surface to infinite depth, as depicted in Figure 3. At the left of the fault the medium is homogeneous, while

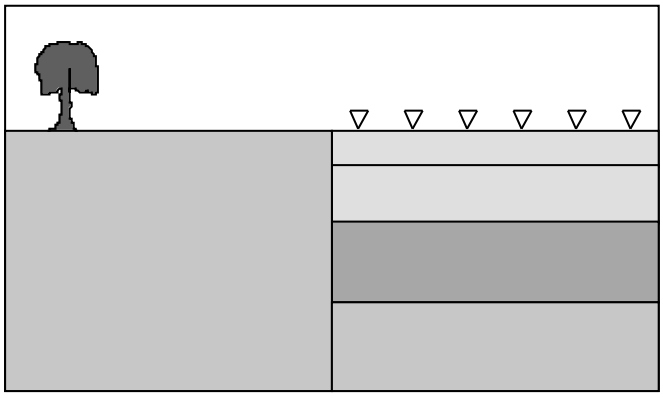

Figure 3: The geological model considered in our numerical example.

at the right of the fault the medium is depth dependent and characterized by a vertical profile of mass density $\rho(z)$.

The contrasts of mass density across the vertical fault produce a gravity anomaly at the surface. Let us assume that we have observed the horizontal gradient of the vertical component of the gravity at 20 equispaced points to the right of the fault, the first point being located 2 km from the fault, and the last point being located 40 km from the fault. The forward problem of computing the data values $d_i = d(x_i)$ from the density contrast function is solved by

$$d(x) = \frac{\partial g}{\partial x}(x) = 2G \int_0^\infty dz \frac{z \Delta\rho(z)}{z^2 + x^2} \ , \qquad (20)$$

where x is the horizontal distance from the fault, z is the depth, $g(x)$ is the vertical component of the gravity, $\Delta\rho(z)$ is the horizontal density contrast across the fault at depth z , and G is the gravitational constant.

The a Priori Information

Let us assume that in addition to the "hard" model constraints described above, we have the following a priori knowledge about the subsurface structure: The density of the rock to the left of the vertical fault is known to be 2570 kg/m^3. To the right of the fault is a stack of (half) layers, and we have the a priori information that the thicknesses ℓ_i of the layers are distributed according to the exponential probability density

$$f(\ell) = \frac{1}{\ell_0} \exp\left(-\frac{\ell}{\ell_0}\right) \ , \qquad (21)$$

where ℓ_0 , the mean layer thickness, has the value $\ell_0 = 4$ km.

Independently of the thickness of the layers, the mass density for each layer follows an empirical probability density, displayed in Figure 4. To simplify the calculation, the stack of layers is assumed to have a total thickness of 100 km, resting on a homogeneous basement having the same mass density as the half space at the left of the fault (2570 kg/m^3), and the top layer is truncated (eroded) at the surface.

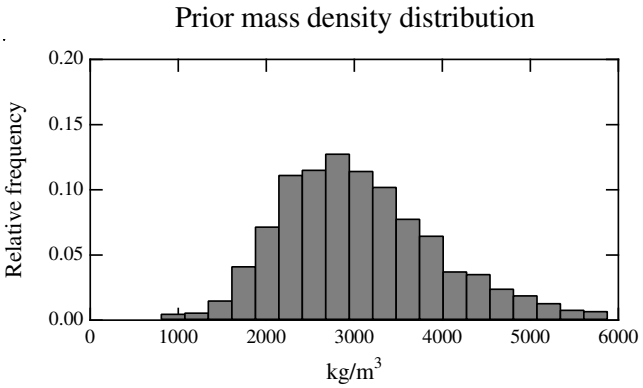

Figure 4: The a priori probability density function for the mass density inside each layer. The a priori probability density function for the thickness of each layer is an exponential function.

True Model, Experimental Uncertainties, and Observed Data Values

The measured data is assumed to be the response of a "true model" (Figure 5). The exact data corresponding to the true model are shown in Figure 6. The measured data values are assumed to be contaminated by statistically independent, random errors ε_i modeled by the sum of

two Gaussian probability density functions,

$$f(\varepsilon) = \frac{a}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{\varepsilon^2}{2\sigma_1^2}\right) + \frac{(1-a)}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{\varepsilon^2}{2\sigma_2^2}\right) , \quad (22)$$

where we have chosen the constants $\sigma_1 = 0.25\,10^{-9} s^{-2}$, $\sigma_2 = 1.25\,10^{-9} s^{-2}$, and $a = 0.25$ (see Figure 7).

The simulated observations, which are formed by summing the "true" data and the simulated noise, are displayed in Figure 6. Then, the likelihood function $L(\mathbf{m})$, measuring the degree of fit between synthetic and observed data is the one given by equation (6).

The Sampling Algorithm

The prior random walk. Let us now describe how our algorithm works. First, we define the graph in the model space that will guide our random walk. To ensure efficiency of the algorithm, it is important that very few of the possible steps in the model space lead to a radical change in the synthetic data generated from these models.

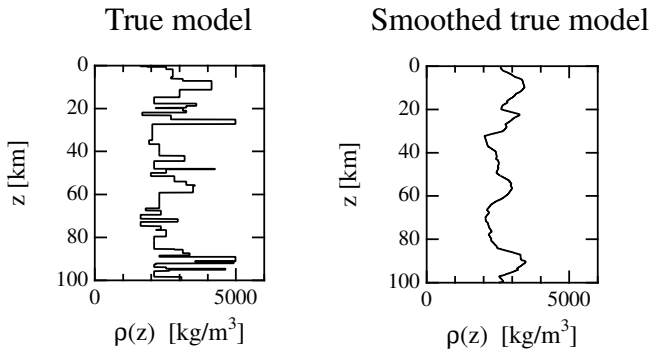

Figure 5: The true model used to generate synthetic data.

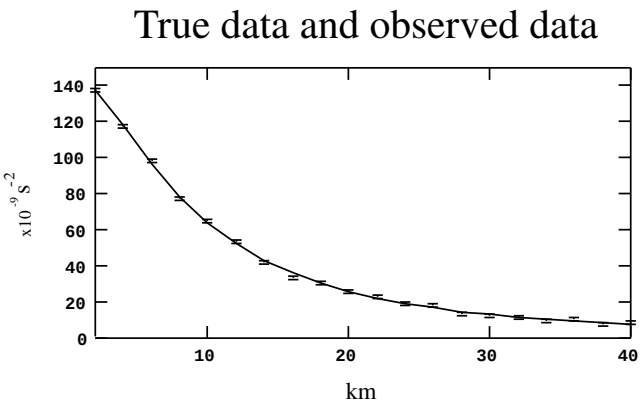

Figure 6: Synthetic data (solid line) used for the inversion, generated from the "true model" of Figure 5, and the "observed data" (points with error bars), equal to the "true data" plus some noise.

A simple way of sampling the a priori probability in the model space would be to use a random walk that generates successive models totally independently. To generate a new model, we could, for instance, pseudo- randomly generate layer thicknesses $\ell_1, \ell_2, \ldots$ from bottom to top, according to the exponential distribution given by equation (21), until they add up to the 100 km of total thickness ("eroding", if necessary, the top layer). Then we could pseudorandomly generate, inside each layer, the corresponding value for the mass density, according to the empirical distribution displayed in Figure 4. However this would produce a radical change in the synthetic data in each step of the random walk, and therefore it would be a very inefficient algorithm. The reason is that if the current model is one having a high posterior probability, a radical change would most likely lead to one of the very abundant models having a low posterior probability and would therefore be rejected by the algorithm.

Another way to produce samples of the a priori probability in the model space could be the following: Given a sample of the prior (i.e., given a model), we could produce another sample by, for instance, randomly choosing a layer and replacing its thickness by a new thickness

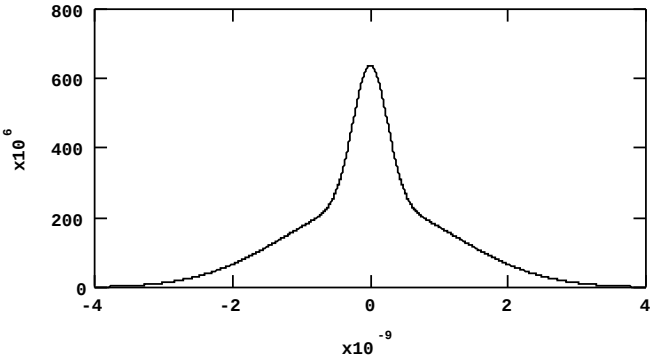

Figure 7: The arbitrary function used to model data uncertainties, as a sum of two Gaussians.

drawn from the exponential distribution given by equation (21) or by replacing its mass density by a new mass density drawn from the empirical distribution displayed in Figure 4.

It is obvious that iterating this procedure, we would always produce models whose layer thicknesses and mass densities are distributed properly; i.e., we would produce samples of the prior probability in the model space. Successive models will be "close" in some sense, but our numerical experimentation has shown that they are still too far apart: when testing models produced by this prior random walk by the likelihood function $L(\mathbf{m})$ (see be-

low), the probability of being accepted as samples of the a posteriori probability is extremely low. The reason is that when perturbing one layer thickness, all the layers above are shifted (remember that we go from bottom to top), and this strongly changes the synthetic data.

Therefore we decided to define the neighbors of a model as the models we can get, not by changing the thickness of a layer but by creating or destroying a new interface in the model (in a way described below). Then, all the other layers remain intact, and we only make a small perturbation in the synthetic data.

More precisely, the neighbors of a model are the models we can get by performing one of the following three perturbations:

(1) changing the mass density in one layer,

(2) adding a new layer boundary and assigning mass densities to the layers above and below it, or

(3) removing one layer boundary and assigning a mass density to the new compound layer.

To complete the description of our algorithm, we will now specify the random rules used by the random walk on the graph.

In each iteration it is first decided which kind of model perturbation step should be performed next. Performing a "pure" layer density perturbation has the same probability (0.5) as performing a layer boundary perturbation (removing or adding a boundary).

In case of a step involving a pure layer mass density perturbation, a layer is selected uniformly at random and a (new) density is chosen for that layer according to the density histogram of Figure 4.

In case of a layer boundary perturbation step we face the problem of adding or removing layer boundaries in such a way that if the step was iterated alone, it would leave the (a priori) distribution of models unchanged. In particular, the exponential layer thickness distribution $(1/\ell_0)\exp(-\ell/\ell_0)$ should be maintained. There is a simple solution to this problem: we exploit the fact that (approximately) exponentially distributed layer thicknesses can be obtained by assuming that the probability that a layer interface is present at a given depth (sample point) is equal to $(40\mathrm{m}/\ell_0) = 0.01$ and independent of the presence of other layer interfaces.

A layer boundary perturbation step therefore works as follows. First, we select one of the 2500 discrete points of the current mass density function, uniformly at random. We then randomly decide if there should exist a layer boundary at that point or not. The probability for the point to be a layer boundary is 0.01.

In case this operation creates a new layer boundary, we generate a mass density for the layers above and below the new layer boundary according to the a priori probability distribution shown in Figure 4.

In case this operation removes a layer boundary, we generate a mass density for the new compound layer (consisting of the layers above and below the removed layer boundary) according to the a priori probability distribution.

This exactly corresponds to the a priori information we wanted to input to our problem: the random walk in the model space so defined is sampling the probability density describing our a priori information.

The posterior random walk. Let us now describe how the above prior random walk is modified into a new random walk, sampling the posterior distribution.

Every time a model perturbation is attempted by the prior random walk, the gravity response is computed from the perturbed layer sequence $\mathbf{m}_{\,\mathrm{pert}}$ by summing up the contributions from the layers in the interval between 0 km depth and 100 km depth. The contribution from a homogeneous half layer is given by

$$G\Delta\rho\log\left(\frac{D^2+x^2}{d^2+x^2}\right) \tag{23}$$

where d is the depth to the top of the homogeneous half layer, D is the depth to the bottom of the half layer, $\Delta\rho$ is the layer density, and x is the horizontal distance to the edge of the half layer.

From the computed gravity response $\mathbf{g}(\mathbf{m}_{\,\mathrm{pert}})$ and the observed gravity response $\mathbf{d}_{\mathrm{obs}}$ the value of the likelihood function $L(\mathbf{m}_{\,\mathrm{pert}})$ is computed using equation (6). The attempted perturbation is now accepted or rejected according to the Metropolis rule, using the likelihoods $L(\mathbf{m}_{\mathrm{cur}})$ and $L(\mathbf{m}_{\,\mathrm{pert}})$ of the current and perturbed models, respectively (see the section on sampling the a posteriori probability density).

This completes the description of the algorithm used in our numerical example. There are, however, a few remaining issues concerning the use of its output models. Most importantly, we want independent samples from the a posteriori distribution.

If independent sample models are required, one has to wait some time between saving the samples. In practice, a single test run of, say, 1000 iterations is performed, and the value of the likelihood function is recorded for the current model of each iteration. After some iterations the likelihood has risen from the usually very low value of the initial model to a rather stable "equilibrium level", around which it fluctuates during the remaining iterations. By calculating the autocorrelation function for the equilibrium part of this series of likelihood values, it is possible to estimate the waiting time (in iterations) between statistically independent likelihood values. This waiting time a very rough measure of the minimum waiting time between statistically independent model samples from $\sigma(\mathbf{m})$. The waiting time between saving model samples in our computations is 100 iterations. A discussion

of the validity of the above measure is beyond the scope of this paper. It shall, however, be noted that the described method is only approximate and that the crucial problem of estimating how many iterations are needed to yield a sufficient number of samples (to characterize a given inverse problem) is still unsolved.

Making of a Movie

First, the comparison between computed and observed data is "turned off", so as to generate a sample of models representing the a priori probability. This has two purposes. First, it allows us to make statistics and to verify that the algorithm is working correctly. More importantly, it allows us to really understand which sort of a priori information we are inputting to the problem. Figure 8, for instance, shows 30 of the models representing the a priori probability distribution, of the many tens of thousands generated. We call this figure a "movie", as this is the way the whole set of generated models is displayed on a computer screen. These 30 models give an approximate idea of the sort of a priori information used. Of course, more models are needed if we want a more accurate representation of the a priori probability.

We may not be interested in the models per se but only in smooth Earth models (for instance, if we know that only smooth properties are resolved by the data). The movie of Figure 8 then easily becomes the smooth movie displayed in Figure 9 (where the density at each point is arbitrarily chosen to be a simple average over 250 points surrounding it).

"Turning on" the comparison between computed and observed data, i.e., using the Metropolis rule, the random walk sampling the prior distribution is modified and starts sampling the posterior distribution. Figure 10 shows a movie with some samples of the posterior distribution, and Figure 11 shows the smoothed samples.

Let us first concentrate on the a posteriori movie of Figure 10. It is obvious that many different models are possible. This is no surprise, as gravity data do not constrain strongly the Earth model. But it is important to look at Figure 12. We display the a priori and the a posteriori data movie, i.e., the synthetic data corresponding to models of the a priori random walk in the model space and the synthetic data corresponding to models of the a posteriori random walk in the model space, when the Metropolis rule is biasing the prior random walk towards the posterior. Even though the models in the posterior movie of Figure 10 are quite different, all of them predict data that, within experimental uncertainties, are models with high likelihood: gravity data alone can not have a preferred model.

Let us now analyze the smoothed models of Figure 11. They do not look as "random" as the models without smoothing: they all have a zone of high-density contrast centered around 10 km depth, which is a "structure" resolved by the data.

Answering Questions

From the viewpoint defended here, there are no well-posed questions or ill-posed questions, but just questions that have a probabilistic answer.

Making histograms. We may be interested in the value of the mass density at some depth, say z_0. Each of

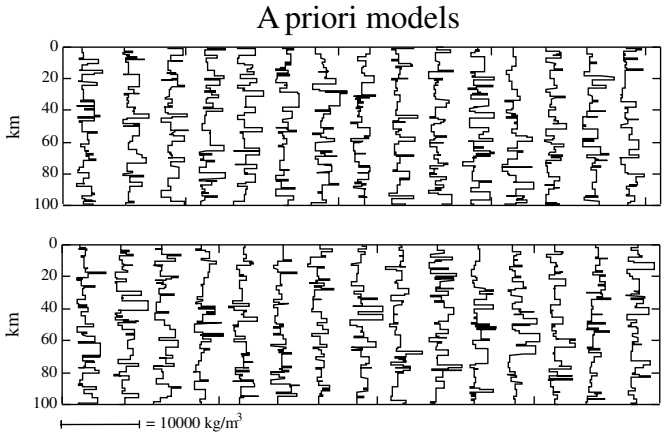

Figure 8: Some images of a movie representing the a priori probability density.

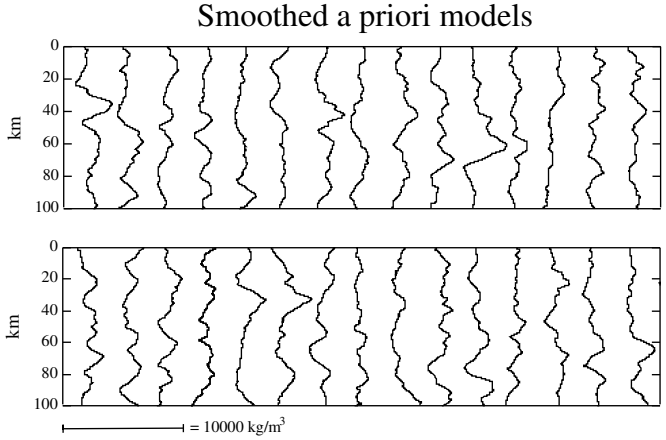

Figure 9: Same as Figure 8 but with the models smoothed.

our many samples (of both the a priori and the a posteriori probability in the model space) has a particular value of the mass density at z_0. The histogram of these values clearly represents the marginal probability distribution for the mass density at that point.

Figures 13 and 14 show both the prior and posterior histograms for the mass density at 2 km, 10 km and 80

km depth, respectively. In particular, we see, when comparing the prior and posterior histograms at 2 km depth, that the mass density to some extent has been resolved there: the histogram has been slightly "narrowed". This is not the case at 80 km depth. Instead of the value of the mass density at some particular depth, we may be interested in the average mass density between, say, z_1 and z_2 . Taking this average for all our samples gives the histogram shown at the bottom of Figure 14.

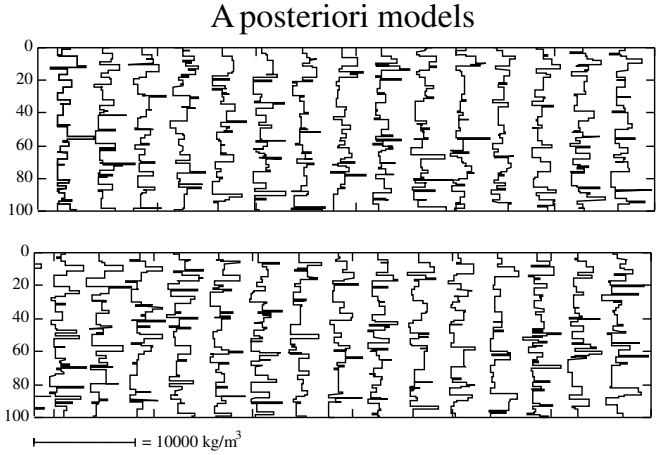

Figure 10: Some images of a movie representing the a posteriori probability density.

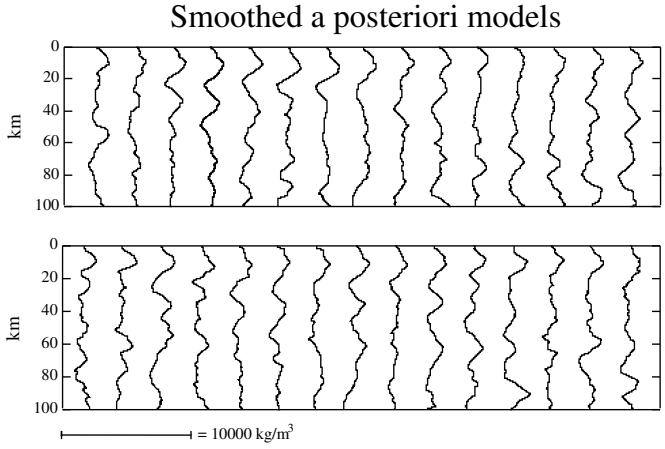

Figure 11: Same as Figure 10, smoothed. The smoothed models do not look as "random" as the models without smoothing (Figure 10): they all have a "bump" at about 10 km depth, which is a "structure" resolved by the data.

Computing central estimators, or estimators of dispersion. Central estimators and estimators of dispersion are traditional parameters used to characterize simple probability distributions. It is well known that while mean values and standard deviations are good measures

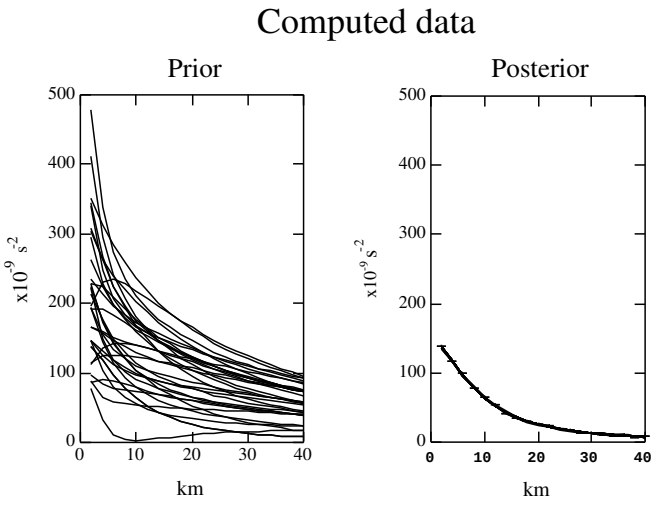

Figure 12: The a priori and a posteriori data movie.

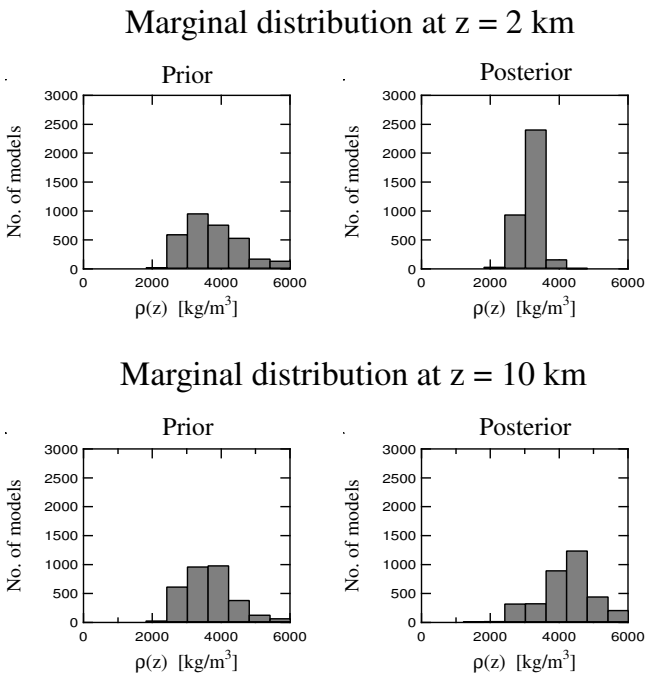

Figure 13: Prior and posterior histograms for the mass density respectively at 2 km and 10 km. When comparing the prior and posterior histograms at 2 km depth, we see that the mass density has been quite well resolved there. the histogram has been considerably "narrowed".

for Gaussian functions, median values and mean deviations are better adapted to Laplacian (double exponential) functions. We can compute both estimators (or any other), as we are not dependent on any particular assumption.

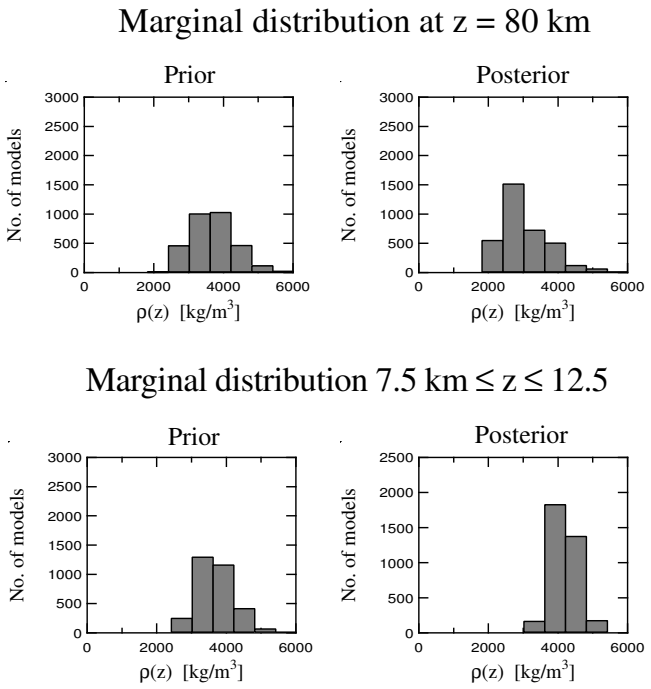

Figure 14: Prior and posterior histograms for the mass density at 80 km depth, and average mass density between 7.5 km and 12.5 km. The mass density at 80 km depth has been less well "resolved" that at 2 km depth (see Figure 13).

Figure 15 shows the mean value for the mass density, plus and minus the standard deviation, and the median, plus and minus the mean deviation for both the a priori and the a posteriori movie. Again, these plots represent the mean and median (and corresponding deviations) of the a priori and a posteriori probability distributions in the model space. Notice that the mean and the median a posteriori models both show the zone of high density contrast centered around 10 km depth, characteristic of the true model of Figure 5, a feature well resolved by our data.

Computing correlations. We may also ask how correlated are the mass density values at different depth locations. From our movies, we can, for instance, easily compute the covariance function $C(z, z')$. The correlation function is given by $c(z, z') = C(z, z')/(\sigma(z)\sigma(z'))$, where $\sigma(z)$ is the standard deviation at point z (just estimated). The correlation function, taking its values in the interval $(-1, +1)$, has a simpler interpretation than the covariance function.

We have chosen to compute the correlation between a point arbitrarily chosen at $z_0 = 10$ km and all other points, i.e., the function $c(z_0, z)$. The result is displayed in Figure 16.

Notice that correlations in the a priori probability distribution decay approximately exponentially, and that they are all positive. In the a posteriori probability distribution, anticorrelations appear. This means, roughly speaking, that if the mass density of any particular realization is in error at 10 km depth, it is likely that it will also be in error, but with opposite sign, in the layers just above and below 10 km.

The approximate exponential decay of the correlation in the prior probability results from the exponential prior probability chosen for the layer thicknesses. The anticorrelations appearing in the posterior probability describe the uncertainty in our posterior models due to the type of information brought by the gravity data.

Discussion

All the results presented in Figures 8 and 9, and the left parts of Figures 13 to 16 concern the a priori movie (i.e., they correspond to the sampling of the model space according to the a priori probability density). Should we at this point decide that we are not representing well enough our a priori information or that we are inputting a priori information that we do not actually have, it would be time to change the way we generate pseudorandom models. If the a priori movie is acceptable, we can "switch on" the synthetic data calculation, and the filter described above, to generate samples of the a posteriori probability distribution, i.e., to produce the a posteriori movie.

It should be properly understood in which way the feature at 10 km depth is "resolved" by the data. None of the models of the a posteriori movie shows a clear density bump at 10 km depth, as the considered inverse problem has a highly nonunique solution (i.e., many different models fit the data and are in accordance with the a priori information). From the a posteriori movie we can not conclude that the true model does have the bump, as many models without it are acceptable. Simply, models with the bump, and arbitrary "high frequencies" superimposed, have a greater chance of being accepted.

General Considerations

There are two major differences between our Metropolis rule (for solving inverse problems) and the original Metropolis algorithm. First, it allows an introduction of non-uniform a priori probabilities. Moreover, an explicit expression for the a priori probabilities is unnecessary: an algorithm that samples the model space according to the prior is sufficient. Second, our Metropolis rule is valid

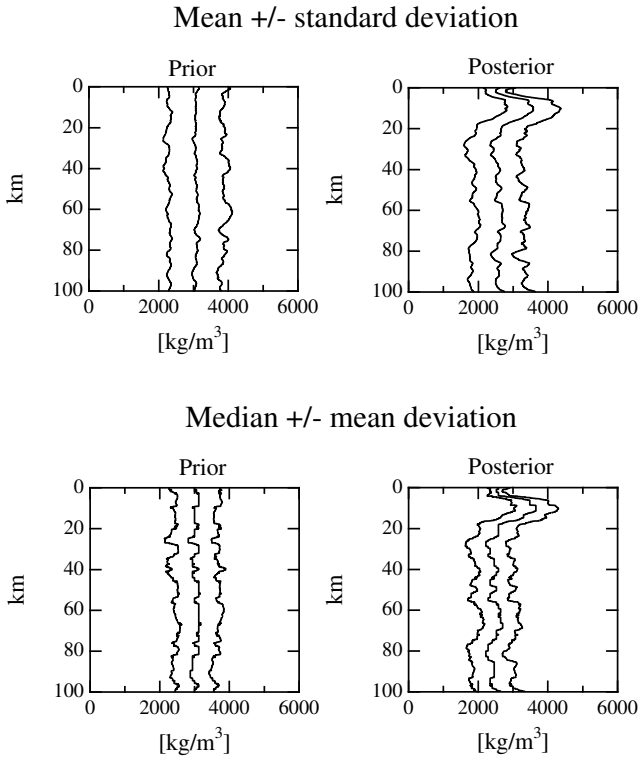

Figure 15: Mean value for the mass density, plus and minus the standard deviation, and the median, plus and minus the mean deviation for both, the a priori and the a posteriori movie. These represent the mean and median (and corresponding deviations) of the a priori and a posteriori probability distributions in the model space.

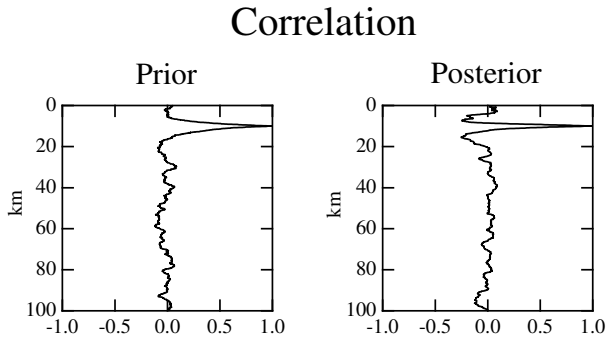

Figure 16: The (left) a priori and (right) a posteriori correlation functions $c(z_0, z)$ for $z_0 = 10$ km. Notice the anticorrelations appearing in the posterior correlation function.

for an arbitrary probability (i.e., it is not linked to the Gibbs-Boltzmann distribution).

Our algorithm has been developed for sampling of discrete spaces according to given probabilities. However, it can be used for optimization. The Metropolis algorithm is already used in simulated annealing [*Kirkpatrick et al.*, 1983], where the desired distribution is changed during the process, starting with a uniform distribution and ending with a near-delta distribution, centered at the optimal solution. We could also find the "best model" by artificially using in the equations values for the experimental uncertainties that tend to zero. However, we do not recommend paying any interest to this concept of "best model".

The method developed above is independent of the way probabilities have been normalized. This is important, as many interesting properties of a probability distribution can be inferred from a random walk, even before the walk has been so extensive that it allows an effective estimation of the denominator of equation (14).

Although we have designed a sampling algorithm (and given proof of its convergence to the desired distribution), we have only addressed heuristically the difficult problem of designing efficient algorithms. It can be shown that the Metropolis rule is the most efficient acceptance rule of the kind we consider (see Appendix C), but the acceptance rule is only part of the efficiency problem: defining the graph (i.e., how the models can be perturbed) is a nontrivial task, and we have only shown an example of it, having no general theory to propose.

Conclusion

We have described a near-neighbor sampling algorithm (random walk) that combines prior information with information from measurements and from the theoretical relationship between data and model parameters. The input to the algorithm consists of random models generated according to the prior distribution $\rho(\mathbf{m})$ and the corresponding values of the likelihood function that carries information from measurements and the theoretical data/model relationship. Output from the algorithm are pseudo-random realizations of the posterior distribution $\sigma(\mathbf{m})$. We applied the algorithm to a highly nonunique, linear inverse problem, to show the method's ability to extract information from noisy data.

The a posteriori distribution contains all the information about the parameterized physical system that can be derived from the available sources. Unfortunately, this distribution is multidimensional and is therefore impossible to display directly.

It is important to direct future efforts toward the development of methods for analyzing and displaying key properties of a posteriori distributions of highly nonlinear inverse problems. For this class of inverse problems, the a posteriori distributions are typically multimodal, and traditional techniques for analyzing error and resolution properties of unimodal a posteriori distributions

break down. There is no known way of understanding uncertainties in the result of a highly nonlinear inverse problem. Here, we have defined the a posteriori probability density $\sigma(\mathbf{m})$, which contains all the information, but how to extract it? Clearly, computing standard deviations or covariances may be meaningless if the posterior probability density is far from Gaussian, which is always the case for highly nonlinear problems. Also, an extensive exploration of the model space can not be made if the space is of high dimension, as, for instance, in the problem of interpretation of seismic reflection data.

In that problem, each model is usually represented by an image. Using the methods described above, we should start by generating pseudo random models with the prior distribution $\rho(\mathbf{m})$. The movie should show models that, on the grounds of our prior information, are more or less likely. In geophysics, this is the right time for a geologist to tell us if he agrees with the movie or if, on the contrary, he sees too many unlikely or too few likely models. When the geologist is satisfied, we now can turn to look at the data, and to run the Metropolis rule, using data misfits, to converge to the posterior probability distribution $\sigma(\mathbf{m})$. The movie is now showing only models which are likely after examination of prior evidence and of geophysical data.

It must be understood that this point of view is much more general than the usual one. For instance, imagine a problem where certain parameters can be resolved deterministically and other parameters can only be resolved statistically. This is the case, for instance, when inverting seismograms to obtain earth models. The major impedance contrasts, for instance, can be deterministically resolved from reflected energy. However, imagine that our space of admissible models contains models with very fine layering, much finer than the seismic wavelength. The position of these very fine layers can not be resolved deterministically, but, as some properties of the seismograms (coda amplitude decay, etc.) do contain information on the average density of fine layers, models (with fine layering) compatible with this information should be generated. Those fine layers could of course not be located individually, but if the data, say, perfectly resolve the average density of a series of layers, all the selected models should display the same average density of these layers. A simple illustration of this possibility has been made here with the "bump" in our mass density models.

From the final collection of models we can start posing questions. Ask for instance for any particular property of the model, for instance, the depth of a particular layer, the smoothed matter density distribution, etc. We have now many examples of that property. It may happen that all the models give the same value for it: the property is well constrained by the data. Some, using old terminology, would say that asking for that property is a "well-posed question". On the contrary it may happen that all the models give absolutely different answers to the question.

In general, we are able to estimate statistics on that property and give answers with a clear probabilistic meaning. In almost all the interesting cases, those statistics will not follow the nice bell-shaped Gaussian distribution, but this should not be an obstacle to a proper analysis of uncertainties. We are well aware of the often tremendous computational task imposed by this approach to inversion. However, the alternative may be an uncertain estimation of uncertainties.

Appendix A: Design of Random Walk With a Desired Equilibrium Distribution

The design of a random walk that equilibrates at a desired distribution $\{p_i\}$ can be formulated as the design of an equilibrium flow having a throughput of p_i particles at point i . The simplest equilibrium flows are symmetric, that is, they satisfy $f_{ij} = f_{ji}$: the transition $i \leftarrow j$ is as likely as the transition $i \rightarrow j$. It is easy to define a symmetric flow on any graph, but it will in general not have the required throughput of p_j particles at point j . This requirement can be satisfied if the following adjustment of the flow is made: first, multiply all the flows f_{ij} with the same positive constant c . This constant must be small enough to assure that the throughput of the resulting flows cf_{ij} at every point j is smaller than its desired probability p_j . Finally, at every point j , add a flow f_{jj} , going from the point to itself, such that the throughput at j gets the right size p_j . Neither the flow scaling nor the addition of f_{jj} will destroy the equilibrium property of the flow. In practice, it is unnecessary to add a flow f_{jj} explicitly, since it is implicit in our algorithms that if no move away from the current point takes place, the move goes from the current point to itself. This rule automatically adjusts the throughput at j to the right size p_j .

Appendix B: Naïve and Uniform Random Walks

Naïve Walks

Consider two arbitrary neighbors, i and j , having n_i and n_j neighbors, respectively, and a random walk with the simple transition probabilities $p_{ji} = 1/n_i$ and $p_{ij} = 1/n_j$ (choosing one of the neighbors, as the next point, uniformly at random). If we want the equilibrium flow to be symmetric, $p_{ji}q_i = p_{ij}q_j$, which is satisfied if $q_i = n_i$.

Furthermore, the above probabilities make all the flows $f_{ji} = p_{ji}q_i$ equal to unity. So, the total throughput through point i is $\sum_k f_{ik} = \sum_j f_{ji} = n_i = q_i$. Hence $q_i = n_i$ must be the equilibrium distribution for the random walk.

Uniform Walks

The rules for the uniform walk follows now directly from applying the Metropolis rule (see later) to the above random walk. The Metropolis acceptance probabilities are $p_{ji}^{\rm acc} = \min(v_j, v_i)/v_i$, where $v_i = 1/q_i$ and $v_j = 1/q_j$ are the "modification probabilities".

Appendix C: Modifying a Random Walk by Introduction of an Acceptance Rule

Consider a random walk P_{ij} with equilibrium distribution ρ_i and equilibrium flow f_{ij} . We can multiply f_{ij} with any symmetric flow ψ_{ij}, where $\psi_{ij} \leq L_j$, for all i and j , and the resulting flow $\varphi_{ij} = f_{ij}\psi_{ij}$ will also be symmetric and hence an equilibrium flow. The transition probabilities of a "modified" algorithm with flow φ_{ij} and equilibrium probability σ_j is obtained by dividing φ_{ij} with the product probability $\sigma_j = \rho_j L_j$. This gives the transition probability: $P_{ij}^{\rm modified} = f_{ij}\psi_{ij}/\rho_j L_j = P_{ij}\psi_{ij}/L_j$, which is equal to the product of two factors: the initial transition probability, and a new probability: the acceptance probability $P_{ij}^{\rm acc} = \psi_{ij}/L_j$. If we choose to multiply f_{ij} with the symmetric flow $\psi_{ij} = \min(L_i, L_j)$, we obtain the Metropolis acceptance probability $P_{ij}^{\rm metrop} = \min(L_i, L_j)/L_j$, which is one for $L_i \geq L_j$, and equals L_i/L_j when $L_i < L_j$. Choosing, instead, $\psi_{ij} = L_i L_j/(L_i + L_j)$, we get the "logistic rule" with acceptance probability $P_{ij}^{\rm log} = L_i/(L_i + L_j)$. The simplest algorithm can be derived from $\psi_{ij} = \min_i(L_i)$, giving the acceptance probability $P_{ij}^{\rm evap} = \min_i(L_i)/L_j$. The acceptance rule for this constant flow we call the "evaporation rule", as the move by a random walker away from the current point depends only on the desired probability at that point and that this recalls the behavior of a water molecule trying to evaporate from a hot point. A last example appears by choosing $\psi_{ij} = L_i L_j$, which gives the acceptance probability $P_{ij}^{\rm cond} = L_i$. We refer to this acceptance rule as the "condensation rule", as it recalls the behavior of a water molecule trying to condensate at a cold point. The efficiency of an acceptance rule can be defined as the sum of acceptance probabilities for all possible transitions. The acceptance rule with maximum efficiency is obtained by simultaneously maximizing ψ_{ij} for all pairs of points j and i . Since the only constraint on ψ_{ij} (except for positivity) is that ψ_{ij} is symmetric and $\psi_{kl} \leq L_l$, for all k and l , we have $\psi_{ij} \leq L_j$ and $\psi_{ij} \leq L_i$. This means that the acceptance rule with maximum efficiency is the Metropolis rule, where $\psi_{ij} = \min(L_i, L_j)$.

Appendix D: An Iteration Consisting of a Randomly Chosen Step

In this case, the transition probability matrix for the iteration is equal to a linear combination of the transition probability matrices for the individual steps. The coefficient of the transition probability matrix for a given step is the probability that this step is selected. Since the vector of desired probabilities is an equilibrium distribution (eigenvector with eigenvalue 1) for each of the step transition probability matrices, and since the sum of all the coefficients in the linear combination is equal to 1, it is also an equilibrium distribution for the transition probability matrix for the complete iteration. This equilibrium distribution is unique, since it is possible, following the given steps, to go from any point to any other point in the space.

Acknowledgments. We thank Zvi Koren and Miguel Bosch for helpful discussions on different aspects of Monte Carlo optimization. This research has been supported in part by the Danish Energy Ministry, the Danish Natural Science Research Council (SNF), the French Ministry of National Education (MEN), the French National Research Center (CNRS,INSU) and the sponsors of the Groupe de Tomographic Gèophysique (Amoco, CGG, DIA, Elf, IFP, Schlumberger, Shell, Statoil).

References

[1] Backus, G., Inference from inadequate and inaccurate data, I, Proc. Natl. Acad. Sci. U.S.A., 65 (I), 1–105, 1970a.

[2] Backus, G., Inference from inadequate and inaccurate data, 11, Proc. Natl. Acad. Sci. U.S.A., 65 (2), 281–287, 1970b.

[3] Backus, G., Inference from inadequate and inaccurate data, 111, Proc. Natl. Acad. Sci. U.S.A., 67 (I), 282–289, 1970c.

[4] Cary, P.W., and C.H. Chapman, Automatic 1-D waveform inversion of marine seismic refraction data, Geophys. J. R. Astron. Soc., 93, 527–546, 1988.

[5] Dorman, L.M., The gravitational edge effect, J, Geophys. Res., 80, 2949–2950, 1975.

[6] Feller, W., An Introduction to Probability Theory and Its applications? New York 1970.

[7] Geman, S., and D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, IEEE Trans. Pattern Anal. Mach. Intel., PAMI-6, 721–741, 1984.

[8] Jackson, D.D., The use of a priori data to resolve non- uniqueness in linear inversion, Geophys. J. R. Astron. SOC., 57, 137–157, 1979.

[9] Keilis-Borok, V.J., and T.B. Yanovskaya, Inverse problems in seismology (structural review), Geophys. J. R. Astron. SOC., 13, 223–234, 1967.

[10] Kirkpatrick, S., C.D. Gelatt, Jr., and M.P. Vecchi, Optimization by simulated annealing, Science, 220, 671–680, 1983.

[11] Koren, Z., K. Mosegaard, E. Landa, P. Thore, and A. Tarantola, Monte Carlo estimation and resolution analysis of seismic background velocities, J. Geophys. Res., 96, 20289–20299, 1991.

[12] Landa, E., W. Beydoun, and A. Tarantola, Reference velocity model estimation from prestack waveforms: Coherency optimization by simulated annealing, Geophysics, 54, 984–990, 1989.

[13] Marroquin, J., S. Mitter, and T. Poggio, Probabilistic solution of ill-posed problems in computational vision, J. Am. Stat. Assoc., 82, 76–89, 1987.

[14] Metropolis, N., and S.M. Ulam, The Monte Carlo method, J. Am. Stat. Assoc., 44, 335–341, 1949.

[15] Metropolis, N., A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys., 1, (6), 1087–1092, 1953.

[16] Mosegaard, K., and P.D. Vestergaard, A simulated annealing approach to seismic model optimization with sparse prior information, Geophys. Prospect., 39, 599–611, 1991.

[17] Parker, R.L., Understanding Inverse Theory, Annu. Rev. Earth Planet. Sci., 5, 35–64, 1977.

[18] Pedersen, J.B., and 0. Knudsen, Variability of estimated binding parameters, Biophys. Chem., 36, 167–176, 1990.

[19] Press, F., Earth models obtained by Monte Carlo inversion, J. Geophys. Res., 73, 5223–5234, 1968.

[20] Press, F., An introduction to Earth structure and seismotectonics, Proc. of the Int. Sch. Phys. Enrico Fermi, 209–241, 1971.

[21] Rothman, R.H., Nonlinear inversion, statistical mechanics, and residual statics estimation, Geophysics, 50, 2797–2807, 1985.

[22] Rothman, D.H., Automatic estimation of large residual statics corrections, Geophysics, 51, 332–346, 1986.

[23] Tarantola, A., Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation, Elsevier, New York, 1987. Tarantola, A., and B. Valette, Inverse problems = Quest for information, J. Geophys., 50, 159–170, 1982a.

[24] Tarantola, A., and B. Valette, Generalized nonlinear inverse problems solved using the least squares criterion, Rev. Geophys., 20, 219–232, 1982b.

[25] Wiggins, R.A., Monte Carlo inversion of body wave observations, J. Geophys. Res., 74, 3171–3181, 1969.

[26] Wiggins, R.A., The general linear inverse problem: Implication of surface waves and free oscillations for Earth structure, Rev. Geophys., 10, 251–285, 1972.