Informed Proposal Monte Carlo

Khoshkholgh, Sarouyeh; Zunino, Andrea; Mosegaard, Klaus

Published in:
Geophysical Journal International

DOI:
10.1093/gji/ggab173

Publication date:
2021

Document version
Publisher's PDF, also known as Version of record

Document license:
CC BY

Citation for published version (APA):
Informed proposal Monte Carlo

Sarouyeh Khoshkholgh*, Andrea Zunino* and Klaus Mosegaard

Department of Physics of Ice, Climate and Earth, Tagensvej 16, 2200 Copenhagen N, Denmark. E-mail: sarouyeh@nbi.ku.dk

SUMMARY

Any search or sampling algorithm for solution of inverse problems needs guidance to be efficient. Many algorithms collect and apply information about the problem on the fly, and much improvement has been made in this way. However, as a consequence of the No-Free-Lunch Theorem, the only way we can ensure a significantly better performance of search and sampling algorithms is to build in as much external information about the problem as possible. In the special case of Markov Chain Monte Carlo (MCMC) sampling we review how this is done through the choice of proposal distribution, and we show how this way of adding more information about the problem can be made particularly efficient when based on an approximate physics model of the problem. A highly non-linear inverse scattering problem with a high-dimensional model space serves as an illustration of the gain of efficiency through this approach.

Key words: Inverse theory; Statistical methods; Waveform inversion; Computational seismology.

1 INTRODUCTION

Since its introduction in the late 1900s (Metropolis et al. 1953; Hastings 1970; Kirkpatrick et al. 1983; Duane et al. 1987; Marinari & Parisi 1992), Markov Chain Monte Carlo (MCMC) methods have been established as a main tool for providing solutions and uncertainty estimates for small- to intermediate-scale, highly non-linear inverse problems. This development is closely connected to the dramatic increase in computational speed over the last few decades. However, there has also been an increasing demand for solving inverse problems on a larger scale, with more time-consuming forward calculations (e.g. Dettmer et al. 2011; Fichtner et al. 2018), and more complex a priori information (e.g. Grana 1999; Khan et al. 2007; Lange et al. 2012; Cordua et al. 2015; Zunino et al. 2015; Hunziker et al. 2017; Laloy et al. 2018), some of which do not even rely on smoothness of the problem. In this connection it has become clear that straightforward use of standard MCMC algorithms is unfeasible, and recent years have seen a surge of improved samplers with more and more sophisticated sampling strategies (Tierney 1999; Liu 2002; MacKay 2003; Haario et al. 2006; Brooks et al. 2011; Vrugt 2016; Ying et al. 2020). Most of the methods developed for improved efficiency use gradients of the posterior density (or its logarithm) in their proposal mechanism (Roberts & Tweedie 1996; Dosso 2002; Girolami & Calderhead 2011; Neal 2011; Dosso et al. 2014) raising the problem that gradients are generally computationally expensive, except in cases where special, problem-specific properties of the posterior density can be utilized (Fichtner et al. 2018).

Despite all these improvements, there is a growing impression amongst applicants that MCMC strategies are fundamentally slow (Raj et al. 2014; Yogatama et al. 2014; Andersen et al. 2018), and that alternatives should be found. This experience has indeed led to improvements where quite efficient solutions, all tailored to the problem at hand through a priori constraints and/or well-chosen simplifying assumptions, have shown promising results. Notable examples are scalable Monte Carlo algorithms which aim at improved computational efficiency through parallelization or subsampling the data (Neiswanger et al. 2014; Rabinovich et al. 2015; Scott et al. 2016; Minsker et al. 2017; Nemeth & Sherlock 2018; Srivastava et al. 2018). They work for problems where samples can be drawn from a partial posterior that is conditioned on a limited subset of data. For this category of algorithms the challenge is to merge the many partial posterior samples to create a reasonable approximation to the full posterior. For the general non-linear problem, this may be difficult to accomplish with sufficient accuracy (Nemeth & Fearnhead 2020).

Another recent development is an attempt to perform often time-consuming likelihood calculations with neural networks, trained on a very large number of model-data pairs sampled from an a prior probability distribution (Andrieu et al. 2003; Hunziker et al. 2017; Laloy et al. 2017, 2018; Scheidt et al. 2018; Nawaz & Curtis 2019; Holm-Jensen & Hansen 2020).

The ‘race of Monte Carlo ideas’ has been accompanied by intense discussions in the research community about the efficiency of algorithms. Not only have intuitive ideas been held up against each other, but arguments for and against methodologies have also been...
accompanied by numerical experiments to support the conclusions. This approach rests apparently on a sound basis, but if we take a closer look at the way algorithm comparisons are typically carried out, we discover a common deficiency: In very few cases algorithms are compared by solving exactly the same problem. At the surface, test problems look similar, but a closer look reveals that the information available to algorithms in the same test differs significantly. All too often, the total number of numerical operations (forward calculations for an inverse problem) is not used as a measure of computational workload when evaluating the efficiencies. This can be a problem when presenting the results from machine learning, if the computational workload of ’training’ is not included when the efficiency is evaluated. Another example may be in presentation of gradient-based algorithms where efficiency is evaluated as ‘number of iterations to convergence’, without counting numerical operations needed to compute the gradients. As a result, comparisons often become meaningless, but there is one thing that seems clear from most comparative studies: The more additional information about the posterior probability density of the inverse problem we build into the code of an algorithm, the better is our chance to create an efficient the algorithm.

The purpose of this paper is to explore how additional information in MCMC sampling may significantly reduce the computational workload. We will first discuss the reasons for the often excessive time-consumption of simple MCMC strategies. We will then turn to the problem of finding and applying supplementary information available to algorithms in the same test differs significantly. The purpose of this paper is to explore how additional information in MCMC sampling may significantly reduce the computational workload. We will first discuss the reasons for the often excessive time-consumption of simple MCMC strategies. We will then turn to the problem of finding and applying supplementary information available to algorithms in the same test differs significantly.

2 MCMC AND THE PROPOSAL PROBLEM

2.1 Proposal distributions

The basic idea behind any implementation of MCMC is an interplay between proposals and rejections. In each iteration, sampling from a probability density \( f(x) \) over a space \( \mathcal{X} \) proceeds from a current value \( x \) by first randomly proposing a new value \( x' \) according to the so-called proposal distribution \( q(x'|x) \), followed by a random decision where \( x' \) is accepted, with probability (Metropolis et al. 1953; Hastings 1970):

\[
p_{\text{acc}}^{x \rightarrow x'} = \min \left( \frac{f(x')q(x'|x)}{f(x)q(x'|x)} , 1 \right). \tag{1}
\]

This acceptance probability ensures that, once an equilibrium sampling distribution is established, it will be maintained through detailed balance, because the probability \( p_{\text{acc}}^{x \rightarrow x'} q(x'|x)f(x) \) of a transition from \( x \) to \( x' \) equals the probability of the reverse transition, \( p_{\text{acc}}^{x' \rightarrow x} q(x|x')f(x') \) (Mosegaard & Sambridge 2002). At this point it is important to note that the proposal distribution has no influence on the distribution to which the sampling converges, it only influences the speed of convergence.

The two most common types of proposal distributions are:

(i) Local proposal distributions \( q \), where \( q(x'|x) \) depends on the starting point \( x \). A frequent assumption is translation invariance where \( q(x'|x) = q(x' - a|x - a) \) for any shift \( a \) in the parameter space. Another common assumption is symmetry: \( q(x'|x) = q(x|x') \), and in this case we get a simpler expression for the acceptance probability (1):

\[
p_{\text{acc}}^{x \rightarrow x'} = \min \left( \frac{f(x')}{f(x)} , 1 \right). \tag{2}
\]

(ii) Global proposal distributions \( q \) that are independent of the starting point \( x \). This means that \( q(x'|x) = h(x) \) where \( h(x) \) is fixed during the sampling process. If \( h(x) \) is in some sense close to the target distribution \( f(x) \), \( h \) is often called a ‘surrogate’ (for \( f \)).

An MCMC sampler is only efficient if large enough steps (connecting two areas of high values of \( f(x) \) in a few steps) are frequently proposed and accepted. This ability critically depends on \( q(x'|x) \), and requires that \( q(x'|x) \) is (at least locally) similar to \( f(x') \). This is revealed by a close look at the expression for the transition probability from \( x \) to \( x' \):

\[
P(x'|x) = q(x'|x) \cdot \min \left( \frac{f(x')q(x'|x)}{f(x)q(x'|x)} , 1 \right), \tag{3}
\]

showing that the probability of the transition \( x \rightarrow x' \) is high if

(i) \( q(x'|x) \) is large and

(ii) \( f(x') \approx C \cdot q(x'|x) \) where \( C \) is a constant.

We will now see how implementations of local and global proposals may address these questions.

2.2 Local proposals

The use of local proposals is an attempt to satisfy the above two conditions:

(i) This condition is obviously locally true (close to \( x \)) for a local proposal (per definition).

(ii) This condition is usually met by assuming that \( f \) is smooth and by choosing a smooth, sufficiently narrow \( q(x'|x) \). In this way the condition applies in most of \( q \)'s support.

Local proposals are widely used, but they have at least two serious drawbacks. First, if they are too narrow, the proposed steps will be so small that the algorithm needs many iterations to traverse the parameters space. As a result, many iterations are required to produce sufficiently many independent samples from the space. Secondly, condition (ii) may not be easy/possible to satisfy (or insufficient) in practice. Either because \( f \) is not smooth, or if \( f \) is smooth, but a ’sufficiently narrow’ \( q \) only allows unacceptably small steps.

To exemplify and quantify the possible problems with local proposals in high-dimensional spaces, let us consider the case where the target distribution of \( x \) is Gaussian with covariance matrix \( C \) and mean \( x_0: f(x) = N(x|\mu, C) \). Assume for illustration that our proposal distribution is an isotropic Gaussian \( q(x|x_0) = N(x|\mu, C_0) \) with mean \( x_0 \) and covariance matrix \( C_0 \), and that we, in the sampling process, have been fortunate to arrive at point with a high value
of \( f(x) \), say, for simplicity, at its maximum point \( x_0 \). We can now calculate the expected acceptance probability \( P^{x_0 \to x} \) proposed in the next step by the algorithm:

\[
E(P^{x_0 \to x}) = \int_{x_0} f(x) f(x_0) \frac{f(x | x_0)}{f(x_{\text{new}})} dx
\]

\[
= \int_{x_0} \frac{N_x(x_0, C)}{N_x(x_0, C)} \frac{N_x(x_0, C_{\text{new}}) dx}{N_x(x_0, C)}
\]

\[
= \frac{1}{N_x(x_0, C)} \int_{x_0} N_x(x, C_{\text{new}}) dx,
\]

where

\[
x_i = (C^{-1} + C_{\text{new}}^{-1})^{-1}(C^{-1} x_0 + C_{\text{new}}^{-1} x_0) = x_0
\]

and

\[
C_1 = (C^{-1} + C_{\text{new}}^{-1})^{-1}.
\]

Since the last integral in (4) is 1, we have the following expression for the expected acceptance probability:

\[
E(P^{x_0 \to x}) = \left( \frac{\det(2\pi C)}{\det(2\pi(C + C_{\text{new}}))} \right)^{1/2}.
\]

Both \( C_{\text{new}} = \sigma^2_j I \) (with \( \sigma^2_j > 0 \)) and \( C \) are diagonal in the frame spanned by \( C \)'s eigenvectors, and if we assume that the eigenvalues of \( C \) are \( \sigma^2_1 \geq \ldots \geq \sigma^2_N > 0 \), where \( N \) is the dimension of \( X \), the eigenvalues of \( C + C_{\text{new}} \) are \( \sigma^2_1 + \sigma^2_j \), \ldots, \( \sigma^2_N + \sigma^2_j \). From this we have

\[
E(P^{x_0 \to x}) = \prod_{n=1}^{N} \left( \frac{\sigma^2_n}{\sigma^2_n + \sigma^2_j} \right)^{1/2}.
\]

From (8) we see that for any non-zero values of \( \sigma_n \) and \( \sigma_j \) we have

\[
E(P^{x_0 \to x}) \to 0 \quad \text{for} \quad N \to \infty.
\]

If the proposed steps are kept very short (\( \sigma_j \) is small compared to all \( \sigma_n \)), the decrease of \( E(P^{x_0 \to x}) \) with \( N \) is slow. But this situation is of no practical value, because adequate sampling by the algorithm requires that it can traverse high-probability areas of \( f(x) \) within a reasonable amount of time. For non-negligible step lengths, the situation is radically different. Indeed, if there exists an integer \( K \) and a real constant \( c \) such that \( \sigma_j > c \sigma_n \) for all \( n > K \), then \( E(P^{x_0 \to x}) \) decreases more that exponentially with \( N \).

In the following we will take a closer look at methods for finding global proposals for inverse problems. However, before we proceed, we shall first understand the fundamental advantage of (ii) over (i). To this aim, we shall look into an important theorem, proven in the late 1990s, namely the No-Free-Lunch (NFL) Theorem (Wolpert & Macready 1997; Mosseegaard 2012).

### 3 NO-FREE-LUNCH THEOREMS AND THE IMPORTANCE OF INFORMATION

We will now make an important distinction between blind algorithms and informed algorithms. We use the following definitions:

(i) A blind algorithm is an algorithm whose search or sampling is performed only via an oracle. An oracle is a function that, when called by the algorithm, is able to evaluate the target distribution \( f \) at a given point \( x \). The oracle is used by the algorithm as a black box: No other properties of \( f \) than the corresponding inputs and outputs are used. In computer science, blind algorithms are often called heuristics. For inversion, there are many well-known examples of blind algorithms in use: Regular MCMC, Simulated Annealing, Genetic Algorithms, Neural Networks, etc.

(ii) An informed algorithm is an algorithm that, in addition to an oracle, uses known, external properties of \( f \) to guide/improve the search or sampling. By external properties we mean any information about \( f \) that is not given by samples from \( f \). Examples of informed algorithms used in geophysical inversion are HMC (Duane et al. 1987; Neal 2011), exploiting that for seismic wave fields adjacent...
methods can be used to efficiently compute misfit gradients (Fichtner et al. 2018) and Discriminative Variational Bayesian inversion exploiting knowledge about the statistics of the unknown model in case it is a Markov Random Field (Nawaz & Curtis 2019).

Based on the No-Free-Lunch Theorem (Wolpert & Macready 1997), Mosegaard (2012) considered limits for the performance of algorithms designed for solution of inverse problems. The conclusion was that all blind inversion algorithms in finite-dimensional spaces (optimization-based as well as sampling-based) have exactly the same performance when averaged over all conceivable inverse problems. Only an algorithm that take into account more characteristics of the ‘forward model’ than given by the oracle can ensure performance that is superior to blind inversion algorithms.

We can draw the conclusion that efficient inversion algorithms are the ones that operate in accordance with specific properties of the problem it is aiming to solve. If the problem is linear with known Gaussian noise statistics and a given Gaussian prior, it can be solved in ‘one iteration’ (applying a closed-form solution formula). If the problem is mildly non-linear with, for example Gaussian noise and Gaussian prior, our knowledge that the posterior probability distribution is unimodal will render the problem solvable in relatively few iterations. For a highly non-linear problem, the situation is, in principle, the same, except that the term ‘highly non-linear’ usually signals a lack of knowledge of the shape of the posterior. The posterior may be highly multimodal and possess other pathologies, but we may still have some sparse knowledge about it, for instance that it is smooth, so we can compute gradients. Irrespective of what we know about the target posterior distribution, we have the option of building this information into the algorithm. If we have plenty of information, we can create an efficient algorithm. If we have sparse information, our algorithm will need more computation time.

There are countless methods use interpolation methods to construct local or global approximations to the posterior and to use them as proposals in the sampling process (Christen 2005; Ginling et al. 2011; Jin 2011; Stuart et al. 2019; Ying et al. 2020). These methods are useful and may improve performance, but they still suffer from the limitations set by the NFL Theorem, because they do not bring in additional, external information.

In the following we will suggest an approach that allows us to design more efficient inversion algorithms through incorporation of additional, external information about the target distribution. The approach is general and can be used in deterministic as well as in sampling approaches. In this exposition we will focus on MCMC sampling, and our approach will be to replace a traditional, blind proposal mechanism with one built from external physical information, providing a reasonable global approximation to the posterior.

4 MCMC WITH PROBLEM-DEPENDENT PROPOSALS

The complete, probabilistic solution to an inverse problem—the posterior probability density—contains a certain, finite amount of information. In addition to the posterior, we may have additional, external information about properties of the posterior (e.g. smoothness, maximum value, principal axes at selected points). If the external information can be used to guide the sampling process, we gain efficiency, measured as number of numerical operations needed to collect a given number of independent samples. The more external information about the the ‘structure’ of the posterior we can build into the algorithm, the more efficient our inversion algorithm can be. Following the NFL Theorem, these statements must be true for any inversion algorithm, whether it is optimization-based (‘deterministic’) or sampling-based (Monte Carlo). Here we will focus on sampling-based algorithms, in particular MCMC.

As stated above, there are basically two ways of obtaining more information about the target (posterior) distribution:

(i) To collect information while sampling. This can for instance be done via calculation of gradients (e.g. used to derive principal axes proposals). These approaches are ‘blind’, in the sense that, to operate, they do not require the use of external information about the structure of the target distribution. They try to discover properties of the target distribution on the fly, and are therefore limited by the (finite) amount of information expressed therein. This category of algorithms is not the theme of this paper. ‘Blind’ algorithms are subject to the NFL Theorem, stating that the performance of all these algorithms is, when averaged over all conceivable problems, exactly the same. In other words, if an algorithm based on, for example principal axes proposals is good at solving certain problems, it must be poor in other cases (to maintain the general average).

(ii) The second way to obtain more information about the target distribution is to use external information about the structure of the posterior. Such information will make exploration of the posterior more easy, providing knowledge about the posterior that would otherwise have been difficult (time-consuming) to obtain from samples. This is the subject of our paper. By augmenting an existing blind sampler with a proposal, built from external information, we turn the sampler into a specialized algorithm directed towards sampling our concrete target distribution. In this way we obtain a ‘single purpose’ algorithm that will probably perform well in our case, but most likely will fail in other cases.

4.1 Constructing an MCMC algorithm with an informed proposal

Let us now try to create an MCMC algorithm augmented with external knowledge about the structure (properties) of the posterior to speed up the sampling process. Since our goal is only to improve efficiency of the algorithm (and not to modify the posterior), we want to leave the usual ‘prior’ and ‘likelihood’ untouched. To this aim, we want the external information (e.g. a physical model with simplified mathematics) to enter the Metropolis–Hastings algorithm via the proposal distribution, leaving the posterior asymptotically unbiased. Furthermore, we can allow even oversimplified external information without distorting the result. In the worst case it will only slow down the algorithm.

Let us assume that this information comes from a proposal in the form of an approximation to the posterior probability distribution \( \tilde{\sigma}(m) \approx \sigma(m) \). Then, our proposal will not only be close to \( \sigma \) in the neighbourhood of points already visited by the algorithm, it is also expected to work well far away from current samples. We will now see how approximate posteriors can be constructed for linear and non-linear problems.

4.2 Linear, Gaussian problems

Sampling of solutions to a linear Gaussian problem through MCMC sampling is straightforward. Since we have an explicit expression for the Gaussian posterior, the distribution itself can be used as an optimal proposal. Samples \( \varepsilon \) from an N-dimensional standard (isotropic) Gaussian (mean 0 and covariance \( I \)) can be generated with, for example the Box–Müller method, and the desired samples
\( \mathbf{m} \) from a \( N \)-dimensional multivariate Gaussian with mean \( \mathbf{m}_0 \) and covariance \( C \) can be calculated as \( \mathbf{m} = \mathbf{m}_0 + A \mathbf{e} \), where \( AA^T = C \). The matrix \( A \) can be found by, for instance, Cholesky decomposition.

### 4.3 Non-linear problems

For non-linear problems there are several ways to proceed, and the method will depend on the external information available about the true posterior. In the following, we will take a look at two important cases, namely (1) when external information about derivatives of the misfit is available and (2) when external information in the form of an approximate relation between model parameters and data is used. The latter case will investigated in a numerical example in Section (5).

#### 4.3.1 Using information about derivatives of the misfit

Any inverse problem with a smooth posterior can be sampled with fewer rejections if partial derivatives of the misfit function are available (Roberts & Tweedie 1996; Dosso 2002; Girolami & Calderhead 2011; Neal 2011; Dosso et al. 2014). However, if derivatives have to be calculated from internal information (that is, directly from samples of the posterior itself), the gained efficiency may be lost due to the many extra sample points required to estimate the derivatives. If, however, external information about the mathematical structure of the problem allows us to compute derivatives directly at each point in model space, as for instance when using the method of adjoints in seismic inversion, it allows us to devise efficient proposals. In general, the higher order derivatives we can get, the more accurate the Taylor approximation to the misfit or the posterior around any point in the model space will be (Girolami & Calderhead 2011). And using this approximation to build a global proposal will lead to high acceptance rates and large step lengths.

To see how gradient information can be used, consider an inverse problem with \( M \) model parameters and an everywhere smooth misfit function \( f(\mathbf{m}) = -\log(\sigma(\mathbf{m})) \), where \( \sigma(\mathbf{m}) \) is the posterior probability density. Assume that the gradient \( \nabla S(\mathbf{m}) \) is available for any given \( \mathbf{m} \), and that we have an approximate solution to the problem \( \mathbf{m} \). Assume further that in a subset of the parameter space containing \( \mathbf{m} \) we can define an invertible mapping \( \mathbf{x} = f(\mathbf{m}) \) between the original model parameters and new, orthogonal curvilinear coordinates \( \mathbf{x} \), such that, for any \( \mathbf{m} \), the first transformed coordinate is \( x_1 = S(\mathbf{m}) \), the first local base vector in the new \( x \)-coordinate system is \( \mathbf{b}_1 = \nabla S(\mathbf{m}) \), and the remaining base vectors \( \mathbf{b}_2 \ldots \mathbf{b}_M \) are orthonormal. \( \mathbf{b}_1 \ldots \mathbf{b}_M \) span a subspace in which \( S(\mathbf{m}) \) is locally constant around \( \mathbf{m} \), and this means that local sampling near \( \mathbf{m} \) along all the sampling directions \( \mathbf{b}_1 \ldots \mathbf{b}_M \) is simple: The step length in the \( \mathbf{b}_i \)-direction must be tuned according to the size of \( |\nabla S(\mathbf{m})| \), but local steps in the remaining \( M - 1 \) directions can be taken without rejections. This advantage of knowing the gradients is further increased when we observe that we can use the gradients (and indeed higher-order derivatives, if available) to iteratively move in directions of constant \( S(\mathbf{m}) \). This process will, in general, require updated base vectors \( \mathbf{b}_2 \ldots \mathbf{b}_M \), but it will allow large steps without rejections, and this is what we strive for when designing an efficient MCMC algorithm. When using this sampling strategy we should remember that we sample the posterior \( \sigma(\mathbf{x}) \) in transformed coordinates \( \mathbf{x} \), and that the desired posterior \( \sigma(\mathbf{m}) \) is computed from \( \sigma(\mathbf{x}) \) by multiplication with the absolute value of the Jacobian determinant of the transformation \( f \):

\[
\sigma(\mathbf{m}) = \sigma(\mathbf{x}) \left| \frac{\partial(x_1, \ldots, x_M)}{\partial(m_1, \ldots, m_M)} \right|.
\]

Since our coordinate transformation \( f \) locally can be viewed (approximately) as a pure rotation with Jacobian determinant 1, followed by a stretching/compression along the gradient with a factor \( |\nabla S(\mathbf{m})|^{-1} \), the Jacobian determinant for \( f \) becomes:

\[
\frac{\partial(x_1, \ldots, x_M)}{\partial(m_1, \ldots, m_M)} \propto |\nabla S(\mathbf{m})|.
\]

This procedure is an important example of how to apply knowledge about gradients everywhere in the model space to define a coordinate transformation \( \mathbf{m} \to \mathbf{x} \) where a good, partial approximation \( \sigma(\mathbf{x}) \) to the posterior can be defined in the new frame. The resulting proposal is approximately proportional to \( \sigma(\mathbf{x}) \) (in fact, constant) in all coordinate directions, except one (the gradient direction). The result is a significant gain in sampling efficiency.

A example of using the above method of derivatives is the application of Hamiltonian Monte Carlo (HMC) (Neal 2011) to full-waveform seismic inversion (Fichtner et al. 2018; Gebraad et al. 2020), in which waveform-misfit gradients are computed using the method of adjoints. Using an analogy from analytical mechanics, this method works in an augmented space where half of the parameters are the usual model parameters \( \mathbf{m} \) (viewed as generalized space coordinates), and the other half are ‘generalized momentum parameters’ \( \mathbf{p} \). The posterior distribution in the augmented space \( \tilde{\sigma} \) is the product of the posterior distributions over \( \mathbf{m} \) and \( \mathbf{p} \), respectively: \( \tilde{\sigma}(\mathbf{m}, \mathbf{p}) = \sigma(\mathbf{m})\sigma(\mathbf{p}) \) where \( \sigma(\mathbf{p}) \) is usually set to be Gaussian. Defining the joint misfit as \( S(\mathbf{m}, \mathbf{p}) = -\log(\tilde{\sigma}(\mathbf{m}, \mathbf{p})) \), the HMC sampler—knowing the derivatives—is now able to alternate between sampling along contours of constant \( \tilde{S} \), and jumping between contours of different \( \tilde{S} \). Stepping in directions of constant \( \tilde{S} \) is accomplished by identifying \( \tilde{S} \) with the ‘total energy’, and calculating the orbit of constant energy through integration of Hamilton’s equations. According to Liouville’s theorem in Hamiltonian mechanics (Tolman 2016), this approach will automatically include the Jacobian transformation (10). HMC gives an efficient sampling of the joint posterior, and the marginal posterior \( \sigma(\mathbf{m}) \) over the (real) model space is trivially obtained by simply discarding the artificial momentum variables.

The Hamiltonian dynamics used by this method is irrelevant for our discussion, but it is important to understand that the efficiency of this method originates from the knowledge of gradients in the joint space. These gradients carry important information about the (possibly strongly non-Gaussian) posterior, allowing the algorithm to take large steps without risking frequent rejections. The resulting proposal distribution is a partial approximation to the joint posterior in a large subset of the joint space, resulting in efficient sampling of the joint posterior, and hence its marginal in the \( \mathbf{m} \)-subspace.

#### 4.3.2 Using information derived from an approximate forward relation

Let us consider the general expression for the joint posterior probability in the formulation of Tarantola & Valette (1982):

\[
\sigma(d, \mathbf{m}) = \frac{\rho(d, \mathbf{m})\rho(d, \mathbf{m})}{\mu(d, \mathbf{m})}.
\]

where \( d \) is data, \( \mathbf{m} \) is the model parameters, \( \rho(d, \mathbf{m}) \) is the prior probability density, and \( \mu(d, \mathbf{m}) \) is the homogeneous probability density (assigning equal probabilities to equal volumes) in the joint
\((d, m)\)-space. The density \(\theta(d, m)\) expresses the ‘uncertainty of the forward relation’ between \(m\) and \(d\). For simplicity, let us assume that the homogeneous probability density \(\mu(d, m)\), as well as the marginal prior in the model space \(\rho_m(m)\), is constant, which leads us to the following expression for the joint posterior:

\[
\sigma(d, m) = k \cdot \rho(d) \theta(d, m).
\]

(13)

where \(k\) is a normalization constant. Under the further assumption that the observational data uncertainties are small, compared to the modelization errors (remembering that it is at small data uncertainties that MCMC algorithms show a critical slowing-down), we arrive at the following approximation to the posterior in the model space:

\[
\sigma_m(m) \propto \sigma(d_{\text{obs}}, m) \approx \theta(d_{\text{obs}}, m)
\]

(14)

This is a very rough approximation, but it should be remembered that we will not replace the accurate posterior by this expression. The approximation will only be used as a global proposal distribution to speed up the search/sampling from the correct posterior.

The basic idea will typically be to first solve the inverse problem with simplified physics, obtaining an approximate solution \(\hat{m}\). The deviation of this from the true solution is what we call the (true) modelization error \(\delta m_{\text{true}}\), and if we can estimate this error, we can use it to build a rough modelization error distribution \(\theta(d_{\text{obs}}, m)\).

Since we do not know the true solution for a real-data inverse problem, we cannot calculate the error \(\delta m_{\text{true}}\), after having solved the problem. \(\delta m_{\text{true}}\) would have been the difference between the true model vector \(m_{\text{true}}\) and the calculated model vector \(\hat{m}\), but we do not know \(m_{\text{true}}\). Our solution to this is to create an artificial inverse problem that is ‘close’ to the original problem, the difference being that the true model of the new problem is \(\hat{m}\). The data for this problem can be calculated as \(g(\hat{m})\). We can now solve this problem and compute the modelization error \(\delta m_{\text{approx}}\). Since the artificial inverse problem is ‘close’ to the original problem, we expect that \(\delta m_{\text{approx}}\) is close to \(\delta m_{\text{true}}\). In the following section (5) we will adopt the following simple procedure:

(i) Choose a simplified forward function \(g(m)\) expressing much of the essential physics, and at the same time allowing an efficient (but probably inaccurate) inversion. This step can be skipped if a direct way to the following step (without a formal inversion) is available.

(ii) Find a solution \(\hat{m} = h(d_{\text{obs}})\) to the simplified problem with an acceptable datafit. The ‘pseudo-inverse’ \(h\) must give a unique answer, for instance through application of a regularization procedure.

(iii) Estimate the modelization error introduced by using \(g(\hat{m})\) instead of the accurate forward function \(g(m)\). This error is quantified by the distribution \(\theta(d_{\text{obs}}, m)\), which is also our rough approximation to the posterior \(\sigma_m(m)\).

The procedure is:

(a) The ‘true’ modelization error is

\[
\delta m_{\text{true}} = \hat{m} - m_{\text{true}},
\]

but since \(m_{\text{true}}\) is unknown, we compute instead an approximate modelization error

\[
\delta m_{\text{approx}} = h(g(\hat{m})) - \hat{m}.
\]

The above formula estimates what the modelization would have been if \(\hat{m}\) had been the true model. In case \(\hat{m}\) is close to \(m_{\text{true}}\), we expect that \(\delta m_{\text{approx}}\) will be close to \(\delta m_{\text{true}}\).

(b) Use \(\delta m_{\text{approx}}\) to construct a reasonable approximation to the modelization error distribution \(\theta(d_{\text{obs}}, m)\), centred at \(\hat{m}\). This can be done by assuming a functional form for \(\theta(d_{\text{obs}}, m)\) and by using the components of \(\delta m_{\text{approx}}\) to obtain the parameters of \(\theta(d_{\text{obs}}, m)\).

An example of this can be found in the following section.

(iv) Use the approximate modelization error distribution as proposal distribution:

\[
q(m' | m) = \theta(d_{\text{obs}}, m')
\]

(15)

5 NUMERICAL EXAMPLE

To illustrate the gain of computational efficiency obtained by using even a rough approximation to a high-dimensional target posterior as proposal, we shall look at a 1-D inverse scattering problem (Fig. 1). The unknown model is a horizontally stratified medium with 1000 homogeneous layers. Fig. 1(b) shows the acoustic impedance as a function of distance from the surface. A plane-wave seismic pulse (modelled as a Ricker wavelet) is injected perpendicularly into the medium at the surface, and the data (backscattered waves from the medium) are recorded at the surface (Fig. 1a left-hand side). The data are synthetic 1-D full-waveform seismic signals generated by the propagator matrix method, containing all multiple reflections, transmission losses and damping effects, so the inverse problem of recovering the model from the data is highly non-linear. For comparison, an approximate seisimogram, computed by convolution of the reflectivity with the Ricker wavelet, is shown in Fig. 1a (middle), together with its error (deviation from the correct seisimogram) to the right. Fig. 1(c) shows an approximate solution to the inverse scattering problem in the absence of noise, computed by deconvolution, and converted to impedance through trace integration and addition of the slowly varying trend from Fig. 1(b). The approximate solution requires very little computation time (of the order of one forward calculation), but is clearly inaccurate (compare to the ‘true’ model in Fig. 1b). The purpose of the study is to show how the approximate result can be used to efficiently produce a more accurate solution with uncertainty estimates using Markov Chain Monte Carlo (MCMC).

Our aim is to produce enough samples from the posterior probability distribution in reasonable time, and this raises a well-known problem, namely that the traditional MCMC approach is unfeasible for problems with more than a couple of hundred parameters. Our way of speeding up the sampling is to construct a global proposal distribution for the MCMC sampling using the approximate solution \(\hat{m}\). First, we compute the estimated modelization error vector \(\delta m_{\text{approx}}\) using the method described in the previous section. Fig. 1(e) shows the envelope of the components of this vector, and for comparison, the true modelization error (known in this synthetic data case) is shown in Fig. 1(d). The modelization error distribution is then built as a Gaussian with mean \(\hat{m}\) and a diagonal covariance matrix \(C_\theta\), whose diagonal is the squared components of the envelope function, and used as a proposal distribution.

We now consider the solution of this 1000-parameter problem is four different ways.

5.1 Informed proposal Monte Carlo (IPMC)

Fig. 2 (lower curve) shows the convergence of the IPMC algorithm driven by our proposal derived above. A computational overhead of less that 3 forward calculations (deconvolution and calculation of error envelopes) is required to start the sampling process, and we see convergence to equilibrium after about 2000 iterations.
5.2 MCMC algorithm with an isotropic, local proposal

We solved the problem without an informed proposal, using a simple MCMC algorithm. We chose an isotropic, local proposal, changing one parameter at a time using a local Gaussian distribution. The step length (standard deviation of the Gaussian) was adjusted to obtain an acceptance rate of approximately 50 per cent. No additional forward calculations were needed to start (initialize) the sampling process. Fig. 2 (upper curve) shows the slow convergence for this algorithm, being far from equilibration after 2000 iterations. A rough estimate showed that the simple MCMC algorithm equilibrated between $10^3$ and $10^4$ times slower than the IPMC implementation.

5.3 MCMC using a principal axes proposal

An MCMC implementation with a principal axes proposal was considered for solution of the problem. This (blind) method requires an additional computational overhead connected to finding a reasonable initial model, and computation of the Jacobian in this model (and later recompute it after a few hundred iterations). In our test problem we have 4096 data and 1001 unknowns, so a Jacobian requires calculation of more than 4 million gradients (at least a few thousand forward calculations). Since the IPMC implementation converges only after about 2000 iterations, we did not further pursue a solution through the principal axes approach.

5.4 Hamiltonian Monte Carlo

We could also have solved our problem with a Hamiltonian Monte Carlo (HMC) approach. As explained above, HMC for seismic inversion is another example of the IPMC strategy using special properties of the posterior density that are not directly available from samples. The method of adjoints can be used to calculate gradients using only 1 forward and 1 adjoint calculation, at the computational cost of approximately 2 forward simulations, and the gradients allow us to define a proposal that is a partial approximation (similar in most directions) to the target distribution in the joint model/momentum space. However, HMC for seismic inversion is well-described elsewhere (Fichtner et al. 2018; Gebraad et al. 2020), so we have not included this type of solution here.

6 DISCUSSION

It is important to realize that the significantly improved efficiency provided by the physical proposal in this study is not resulting from prior constraints on the model parameters. Priors generally assign
Figure 2. The lower curve shows convergence towards equilibrium of our Informed Proposal Monte Carlo (IPMC) method, which was guided by linearized inversion. The upper curve shows the convergence of a simple MCMC algorithm with a Gaussian proposal, perturbing one parameter at a time, and tuned to an acceptance rate of around 50 per cent. Each iteration of IPMC and simple MCMC required one forward calculation, and involved about 3 and 0 forward calculations for initialization, respectively. The convergence of the simple MCMC was more than 10^3 times slower than that of the IPMC algorithm. A gradient-based algorithm was also considered, but would have required calculation of far more than 2000 derivatives (each with 1 forward calculation) to initialize the sampling procedure.

different probabilities to different solutions, but this is not the case with a proposal. A proposal only influences the frequency by which models are presented to the acceptance/rejection algorithm. The bias of the proposal will, asymptotically, be neutralized because it is compensated for in the acceptance probability. In this way it will only influence the efficiency of the sampler, not the asymptotic result. It should, however, be remembered that the most serious problem in non-linear inversion is that the number of models we can practically test is limited. And considering that highly non-linear problems are often so complex that they can only be safely solved with a high number of approximately independent samples from the posterior, it is clear that using an efficient proposal will not only be an improvement in speed, but also a potential improvement in quality of solutions. Simply speaking, we can expect to discover more significantly different solutions (peaks of the target distribution) within the allowed computer resources than with an arbitrary local proposal.

We have illustrated how important it is for the proposal to mimic the posterior in MCMC sampling of solutions to inverse problems. However, the idea of using the physics of the problem to build a posterior-like proposal is not restricted to Monte Carlo sampling. Any method depending on a search for sample solutions or good data fits can potentially benefit from this strategy. In an interesting recent paper on variational full-waveform inversion (Zhang & Curtis 2020), it is shown how variational methods may be used to modify samples from the prior into samples of the posterior in the solution of large-scale inverse problems. It is likely that this class of methods may, in the future, be further improved through application of informed proposal mechanisms.

In this paper, we have chosen to explain the basic ideas of the IPMC strategy and to support them with a strongly non-linear, synthetic test example in a high-dimensional model space. The test example was synthetic to ensure that we know the true solution and in this way have full control over the behavior of the algorithm. However, to give the reader an idea of how the algorithm could be applied to real data cases and real problems, we give two possible data inversion scenarios.

6.1 Elastic full-waveform inversion of pre-stack seismic reflection data

From seismic reflection data we wish to compute realizations of elastic properties of the upper few kilometres of the subsurface. The computationally demanding elastic wave simulations, combined with the strong non-linearity of the problem, is a serious challenge to MCMC methods. For this problem, an approximate solution can be found using classical data-processing and interpretation techniques. An interpreted, depth-migrated profile (volume), combined with optimized migration velocities, can be used to build a rough, approximate model of the subsurface under study. The approximate model and its (accurately computed) synthetic data will now be used in an artificial inverse problem to estimate modelization errors, and the proposal can be built. In a simple implementation, the approximate model will be the 'center' of the global proposal, and the modelization errors will provide the 'shape' (dispersion) of the proposal.

6.2 Inversion of flow data from a geothermal reservoir

In this problem we wish to compute realizations of the permeability field of a geothermal reservoir from water injection and water production in a large number of wells. Accurate reservoir simulations are computationally expensive, and the problem is highly non-linear.
Assume that we know that five discrete geological facies with approximately known permeabilities are present in the subsurface, and that we wish to build the model by assigning a facies, and a permeability value at each grid point. Each facies has its own prior probability, and its own prior probability distribution for the permeability. Due to the discrete facies parameters, the problem is not smooth. For this problem, an approximate flow simulation can be carried out by a so-called streamline simulator. This simulator estimates the pressure field in the reservoir, followed by flow-line tracing. The algorithm is extremely effective compared to accurate simulators, but its results are imprecise. We first fit a (smooth) least-squares permeability field to the flow data, using the streamline simulator in an iterative linearized inversion. Using our geological knowledge, we then assign facies to each grid point, using the computed permeability field. We now have a (very) rough approximate model. The approximate model and its (accurately computed) synthetic flow data together form an artificial inverse problem from which we can estimate modelization errors. The proposal can now be built, and in a simple implementation we will use the approximate model as the ‘centr’ of the global proposal, and the modelization errors as ‘dispersion’.

7 CONCLUSION
We have analysed the impact of proposal distributions on the performance of MCMC sampling methods when applied to the solution of inverse problems. We concluded that the ‘small step’ strategies used in traditional implementations are relatively efficient because they impose a local consistency between the proposal distribution and the target (posterior) distribution: the target probabilities tend to be large where the proposal probabilities are large. Nevertheless, we showed by a simple analytical example that even local consistency may be difficult to obtain when local ‘small-step’ proposals are arbitrary. Furthermore, a main problem with local proposals is the limited step length, which is strongly hampering the exploration of vast, high-dimensional spaces. The volumes of high-probability areas are negligible in such spaces, so burn-in times, and the times needed to pass from one maximum to another can be prohibitive for small-step algorithms.

Our solution to these problems is to use global proposals built from external information about the target distribution. We propose ways of using physical knowledge of the problem to ensure global consistency between the proposal and the target distribution. The efficiency of this ‘informed proposal’ approach is highly problem-dependent and strongly conditioned on the choice of the external proposal, but we successfully carried out a test on a 1000-parameter, highly non-linear inverse scattering problem. The performance of our informed proposal Monte Carlo algorithm, augmented with an informed proposal, compared favorably with the three other algorithms considered in this study.

ACKNOWLEDGEMENTS
This work was supported by Innovation Fund Denmark through the OPTION Project—Optimizing Oil Production by Novel Technology Integration. Joint industry project with Lloyd’s Register, Wettlec, Technical University of Denmark (DTU), and University of Copenhagen (5184-00025B). Klaus Mosegaard would like to thank Dr Amir Khan and colleagues at the Department of Earth Sciences, ETH, for their hospitality and inspiring discussions during the fall 2017 where this work was initiated.

DATA AVAILABILITY
No new data were generated or analysed in support of this research.

REFERENCES


