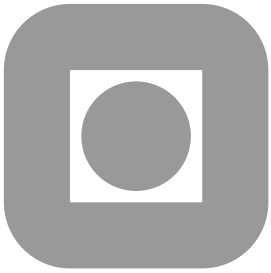
NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

On directional Metropolis–Hastings algorithms

by

Jo Eidsvik and Håkon Tjelmeland

PREPRINT
STATISTICS NO. 6/2003



NORWEGIAN UNIVERSITY OF SCIENCE AND
TECHNOLOGY
TRONDHEIM, NORWAY

This report has URL http://www.math.ntnu.no/preprint/statistics/2003/S6-2003.ps
Jo Eidsvik has homepage: http://www.math.ntnu.no/∼joeid
E-mail: joeid@stat.ntnu.no
Address: Department of Mathematical Sciences, Norwegian University of Science and Technology,
N-7491 Trondheim, Norway.

On directional Metropolis–Hastings algorithms

Jo Eidsvik and Håkon Tjelmeland
Department of Mathematical Sciences,
Norwegian University of Science and Technology,
Norway

Abstract

Metropolis–Hastings algorithms are used to simulate Markov chains with limiting distribution equal to a specified target distribution. The current paper studies target densities on $\mathcal{R}^n$. In directional Metropolis–Hastings algorithms each iteration consists of three steps i) generate a line by sampling an auxiliary variable, ii) propose a new state along the line, and iii) accept/reject according to the Metropolis–Hastings acceptance probability.

We consider two classes of algorithms. The first uses a point in $\mathcal{R}^n$ as auxiliary variable, the second uses an auxiliary direction vector. The directional Metropolis–Hastings algorithms considered here generalize previously proposed directional algorithms in that we allow the distribution of the auxiliary variable to depend on properties of the target at the current state. By letting the proposal distribution along the line depend on the density of the auxiliary variable, we then identify proposal mechanisms that give unit acceptance rate. Especially when we use direction vector as auxiliary variable, we get the advantageous effect of large moves in the Markov chain and the autocorrelation length of the chain is small. We illustrate the algorithms for a Gaussian example and in a Bayesian spatial model for seismic data.

Keywords: adaptive direction sampling, angular Gaussian distribution, auxilary variables, Markov chain Monte Carlo, Metropolis–Hastings, reversible jump MCMC, seismic inversion.

1 Introduction

One often comes across analytically intractable probability distributions. Stochastic simulation algorithms can be used to study such target distributions. A large number of simulation methods have been presented in the last years. In Metropolis–Hastings (MH) algorithms (Hastings, 1970) we simulate a Markov chain with limiting distribution equal to the target. At each iteration of the MH scheme a new state is proposed to replace the current state, and with a certain probability the proposed state is accepted, otherwise the old one is retained. It is important that convergence to the target happens in reasonable time. It is equally important to move efficiently within the target, i.e. produce small autocorrelation in subsequent samples. Although many problems can be solved satisfactorily with existing MH methods, see e.g. Gilks et al. (1996), there is a need for new algorithms with improved convergence and mixing properties. MH algorithms are often case specific and an algorithm that performs well in one case, might not work in another. The algorithms typically involve one or more tuning parameters. By trial and error a reasonable value of these parameters can be set. Tuning is sometimes beneficial, but ideally one wants to reduce the

amount of manual tuning, and hence obtain more generally applicable algorithms.

In this paper we focus on directional MH algorithms. At each iteration, we i) draw a line going through the current point, ii) propose a new state along this line, and iii) accept or reject the proposed value. Directional algorithms generate moves along directions different from the coordinate axes, and this might improve mixing in the Markov chain. Our algorithms define the line either by an auxiliary direction vector or by a point in the sample space. We sample the auxiliary variables using properties of the target at the current state of the Markov chain. The directional MH algorithms can be designed to produce unit acceptance rate, whatever distribution we choose for the auxiliary variable. However, it is not possible to sample directly from this proposal distribution. Instead we use a parametric approximation in the proposal step. Some of the directional MH algorithms discussed in this paper move efficiently within the target. However, many evaluations of the target are required per iteration. We take this into consideration when we evaluate the algorithms.

Several directional MH algorithms are presented in the literature: Chen and Schmeiser (1993) study a scheme moving along a uniform direction at each step; Roberts and Rosenthal (1998) present an algorithm that moves along a direction centered at the gradient direction evaluated in the current point; Gilks et al. (1994), Roberts and Gilks (1994) and Liu et al. (2000) use parallel chains to construct directional jumps. The methods outlined in this paper are different from these because we propose the auxiliary variables from some distribution that is allowed do depend on the current state. This choice has influence on the one dimensional proposal density along the line.

Generalized Gibbs formulations presented in Goodman and Sokal (1989) and Liu and Sabatti (2000) also include proposal distributions along the line that produce unit acceptance rate. However, they do not discuss the effect of letting the distribution of the auxiliary variable depend on the current state. This is the focus of this paper.

The paper is organized as follows. Section 2 describes MH for parametric forms on the proposal and defines a framework for directional MH algorithms. Our MH algorithms are formalized further in Section 3. In Section 4 we present two examples, including a Bayesian spatial model for a dataset from a North Sea petroleum reservoir. Finally, Section 5 contains discussion and concluding remarks.

2 Metropolis–Hastings algorithms

MH algorithms simulate a reversible Markov chain with limiting distribution identical to a specified target. The Markov transition kernel is defined by a proposal and an acceptance step. In this section we define MH algorithms using a parametric form for the proposal. We next put directional MH algorithms into this framework. Throughout the paper we restrict attention to continuous target distributions on $\mathcal{R}^n$, and let $\pi(\cdot)$ denote its density with respect to the Lebesgue measure on $\mathcal{R}^n$. We let $x \in \mathcal{R}^n$ denote the current state of the Markov chain.

2.1 Parametric Metropolis–Hastings updates

In this section we describe MH algorithms generating the proposal by a parametrized deterministic transformation. Such algorithms, see e.g. Green (1995) and Waagepetersen and Sørensen (2001), are most commonly used for Markov chains with varying dimensions and are referred to

as reversible jump algorithms. We regard only the fixed dimension case.

In the proposal step we first sample $t \in \mathcal{R}^m$ from some density $q(t|x)$, and then define the proposed value $y \in \mathcal{R}^n$ by a one-to-one transformation,

$$\left.\begin{array}{l} y = w_1(x,t) \\ s = w_2(x,t) \end{array}\right\} \Longleftrightarrow \left\{\begin{array}{l} x = w_1(y,s) \\ t = w_2(y,s), \end{array}\right. \tag{1}$$

where $w_1 : \mathcal{R}^{n+m} \to \mathcal{R}^n$ and $w_2 : \mathcal{R}^{n+m} \to \mathcal{R}^m$. The acceptance probability for y becomes

$$\alpha(y|x) = \min\left(1, \frac{\pi(y)q(s|y)}{\pi(x)q(t|x)} \cdot |J|\right), \tag{2}$$

where J is the Jacobian determinant of transformation (1), i.e.

$$J = \det\left\{\frac{\partial(y,s)}{\partial(x,t)}\right\} = \left|\begin{array}{cc} \frac{\partial y(x,t)}{\partial x} & \frac{\partial y(x,t)}{\partial t} \\ \frac{\partial s(x,t)}{\partial x} & \frac{\partial s(x,t)}{\partial t} \end{array}\right|. \tag{3}$$

In some cases it is useful to introduce auxiliary variables in the MH proposal step, and we discuss this possibility briefly here, returning to special cases below. For some sample space Φ, let $\phi \in \Phi$ be an auxiliary variable sampled from a density $f(\phi|x)$. The density for t may then depend on ϕ, and we write $q(t|\phi, x)$. The proposal $y \in \mathcal{R}^n$ is still given by (1), where ϕ may now appear as a parameter in the functions w_1 and w_2. Hence, ϕ remains the same in the forward and backward transformations. The acceptance probability for y becomes

$$\alpha(y|\phi, x) = \min\left(1, \frac{\pi(y)f(\phi|y)q(s|\phi, y)}{\pi(x)f(\phi|x)q(t|\phi, x)} \cdot |J_\phi|\right), \tag{4}$$

where J_ϕ is the determinant of the deterministic transformation with fixed ϕ.

2.2 Directional Metropolis–Hastings updates

Directional MH algorithms propose new values along a line defined by the current state x and an auxiliary variable ϕ. The auxiliary variable is introduced to encourage moves in promising directions. In our setting the auxiliary variable ϕ takes the form of either a point $z \in \mathcal{R}^n$ or a direction vector $u \in \mathcal{S}^n$, where $\mathcal{S}^n$ is half the unit sphere

$$\mathcal{S}^n = \{u \in \mathcal{R}^n \setminus \{0\} : ||u|| = 1 \quad \text{and} \quad u_k > 0 \quad \text{for} \quad k = \min(i; u_i \neq 0)\}. \tag{5}$$

The direction vector may be derived from an auxiliary point z by

$$u = \left\{\begin{array}{ll} \frac{z-x}{||z-x||} & \text{if} \quad \frac{z-x}{||z-x||} \in \mathcal{S}^n, \\ -\frac{z-x}{||z-x||} & \text{otherwise.} \end{array}\right. \tag{6}$$

However, note that different z's may produce the same u. We next derive directional MH algorithms using z and u.

Suppose first that we generate a direction vector $u \in \mathcal{S}^n$ from a density $g(u|x)$ with respect to the uniform measure on $\mathcal{S}^n$. The auxiliary variable u defines a line together with the current

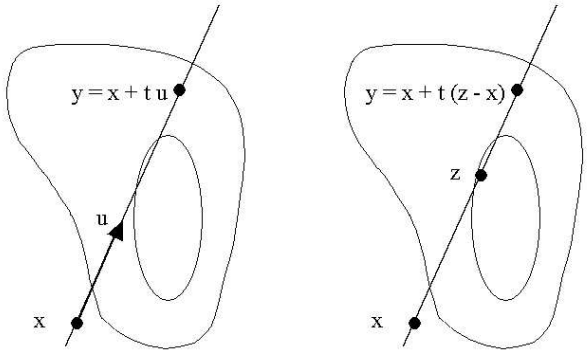


Figure 1: Schematic density contours illustrating direction sampling. Left: The current state x and auxiliary variable u. Right: The current state x and auxiliary variable z. The scalar value t parameterizes the line differently in the two cases.

state $x \in \mathcal{R}^n$, see Figure 1. The line can be parametrized by a scalar t, and we generate $t \in \mathcal{R}$ from some density $q(t|u,x)$. The proposed value $y \in \mathcal{R}^n$ is then deterministically defined by the one-to-one transformation

$$\left.\begin{array}{l} y = x + tu \\ s = -t \end{array}\right\} \Longleftrightarrow \left\{\begin{array}{l} x = y + su \\ t = -s, \end{array}\right. \tag{7}$$

where u is a parameter in this transformation. The Jacobian determinant becomes

$$J_u = \left| \begin{array}{cc} \frac{\partial y(x,t)}{\partial x} & \frac{\partial y(x,t)}{\partial t} \\ \frac{\partial s(x,t)}{\partial x} & \frac{\partial s(x,t)}{\partial t} \end{array} \right| = \left| \begin{array}{cc} \mathbf{I_n} & u \\ \mathbf{0} & -1 \end{array} \right| = -1,$$

where $\mathbf{0}$ is a length n vector of zeros. The MH acceptance rate from (4) becomes

$$\alpha(y|u,x) = \min\left(1, \frac{\pi(y)g(u|y)q(s|u,y)}{\pi(x)g(u|x)q(t|u,x)}\right). \tag{8}$$

Particular choices of $g(u|x)$ and $q(t|u,x)$ are discussed below.

Suppose alternatively that we generate an auxiliary point z from a density $h(z|x)$. The auxiliary variable z and the current point define a line, see Figure 1. This line is parameterized by a one dimensional value t. We next generate t from some density $q(t|z,x)$ and the proposal $y \in \mathcal{R}^n$ is deterministically defined by the one-to-one transformation

$$\left.\begin{array}{l} y = x + t[z - x] \\ s = -\frac{t}{1-t} \end{array}\right\} \Longleftrightarrow \left\{\begin{array}{l} x = y + s[z - y] \\ t = -\frac{s}{1-s}, \end{array}\right. \tag{9}$$

where z is a parameter in this transformation. The Jacobian determinant becomes

$$J_z = \left| \begin{array}{cc} \frac{\partial y(x,t)}{\partial x} & \frac{\partial y(x,t)}{\partial t} \\ \frac{\partial s(x,t)}{\partial x} & \frac{\partial s(x,t)}{\partial t} \end{array} \right| = \left| \begin{array}{cc} (1-t)\cdot \mathbf{I_n} & z - x \\ \mathbf{0} & -(1-t)^{-2} \end{array} \right| = -(1-t)^{n-2}.$$

Thus, the MH acceptance rate from (4) becomes

$$\alpha(y|z,x) = \min\left(1, \frac{\pi(y)h(z|y)q(s|z,y)}{\pi(x)h(z|x)q(t|z,x)} \cdot |1-t|^{n-2}\right). \tag{10}$$

Particular choices of $h(z|x)$ and $q(t|z,x)$ are discussed below.

3 Building blocks

In Section 2.2 we derived two MH algorithms generating directional moves, one with an auxiliary direction u, another with an auxiliary point z. We next discuss i) distributions for the auxiliary variables, i.e. $g(u|x)$ and $h(z|x)$, and ii) proposal densities for the scalar t along the line, i.e. $q(t|u,x)$ and $q(t|z,x)$.

3.1 Choices for $h(z|x)$ and $g(u|x)$

A key part of directional MH algorithms is to draw auxiliary variables corresponding to advantageous directions, and hence induce good mixing properties of the Markov chain. We aim to construct moves towards the central parts of the target density. In terms of the auxiliary point $z \in \mathcal{R}^n$ it seems natural to draw z from a density $h(z|x)$ that resembles the target density $\pi(\cdot)$. This entails that moves towards the central region of the density are likely when the current state x is in the tail, whereas if x is in the central parts of the distribution, z is likely to end up on either side of x. Similar ideas apply to densities for direction vector u. We use only Gaussian densities h in the following. For the u variable we consider $g(u|x)$'s originating from (6). This is linked to $h(z|x)$ by

$$g(u|x) = \int_{-\infty}^{\infty} |r|^{n-1} h(x+ru|x)dr. \tag{11}$$

The integral in (11) is analytically available when h is Gaussian. The resulting $g(u|x)$ is referred to as the Angular Gaussian Distribution, see e.g. Watson (1983) and Pukkila and Rao (1988).

The simplest way to draw the auxiliary variable is to use a fixed density for z, independent of x. We denote this choice by a fixed strategy, i.e.

$$\textbf{Fixed}, \quad h(z) = N(z; \mu_f, \Sigma_f). \tag{12}$$

where $N(z;\mu,\Sigma)$ denotes the Gaussian density with mean μ and covariance matrix Σ evaluated in z. For a fixed density, both the mean and variance are assessed beforehand, i.e. μ_f and Σ_f are fixed. This choice might work well if a Gaussian approximation to the target density is available.

We also discuss three options for $h(z|x)$ using local properties of the target to different degrees. Suppose first that we sample z from a density with mean value in the current state x, and identity covariance matrix, i.e.:

$$\textbf{Order } \mathbf{0}, \quad h(z|x) = N(z; x, \mathbf{I}). \tag{13}$$

In this density we use no information about the target density, and we refer to this as the zero order approximation.

Suppose next that we use local properties of the target density in terms of first derivatives at the current state, and let $V(x) = -ln[\pi(x)]$ denote the potential function for x. We set

$$\textbf{Order 1}, \quad h(z|x) = N(z; \hat{\mu}(x, \nabla V(x)), \mathbf{I}), \tag{14}$$

where we typically use $\hat{\mu}(x, b) = x - b$, i.e. matching first derivatives of h and the target density at x. We refer to this method as a first order approximation since it benefits from first order derivatives of the target density at the current state x.

Suppose next that we use both first and second order derivatives at the current state x. Thus, we set

$$\textbf{Order 2}, \quad h(z|x) = N(z; \hat{\mu}(x, \nabla V(x), \nabla^2 V(x)), \hat{\Sigma}(x, \nabla V(x), \nabla^2 V(x))), \tag{15}$$

where it is natural to choose

$$\hat{\mu}(x, b, A) = x - bA^{-1} \quad and \quad \hat{\Sigma}(x, b, A) = A^{-1}, \tag{16}$$

matching first and second order derivatives of h and π. However, if the matrix of second derivatives is non-positive definite, one has to alter (16). For example, one could add positive elements on the diagonal of the fitted covariance matrix.

Recall that we obtain a corresponding density $g(u|x)$ for each density $h(z|x)$ using (11). This gives eight possibilities: (z, f), (u, f), $(z, 0)$, $(u, 0)$, $(z, 1)$, $(u, 1)$, $(z, 2)$, $(u, 2)$, where the first index refers to auxiliary variable u or z, and the second index to the order of approximation. Note that in the (u, f) case the density g still depends on x from (6).

In the literature various approaches for drawing an auxiliary variable have been presented. The simplest is the hit-and-run algorithm, see e.g. Chen and Schmeiser (1993) and Kaufman and Smith (1998), that generates a uniform direction, independently of x, i.e. $g(u|x) \propto 1$. This is our $(u, 0)$ case. In Adaptive Direction Sampling (ADS) they get the auxiliary variable z from another Markov chain running in parallel, see Gilks and Roberts (1994) and Roberts and Gilks (1994). This corresponds to our (z, f) with $h = \pi$. In Liu et al. (2000), z is sampled from a proposal density that is more concentrated than the target π since they combine ADS with an optimization step. This is also a variant of (z, f).

3.2 A proposal q with unit acceptance rate

In this section we show how to obtain a density for the one dimensional value t that gives acceptance rate one. This is accomplished both for conditioning on u and for conditioning on z. Consider first the case with auxiliary variable u, and recall the acceptance probability in (8). One way to achieve unit acceptance rate is by choosing

$$q(t|u, x) = \frac{g(u|x + tu)\pi(x + tu)}{\int_{-\infty}^{\infty} g(u|x + ru)\pi(x + ru)dr}, \tag{17}$$

since then everything cancels in (8), including the normalizing constants. We note that $q(t|u, x)$ is a function of the density for u. Moreover, acceptance rate one is achieved regardless of our choice for g.

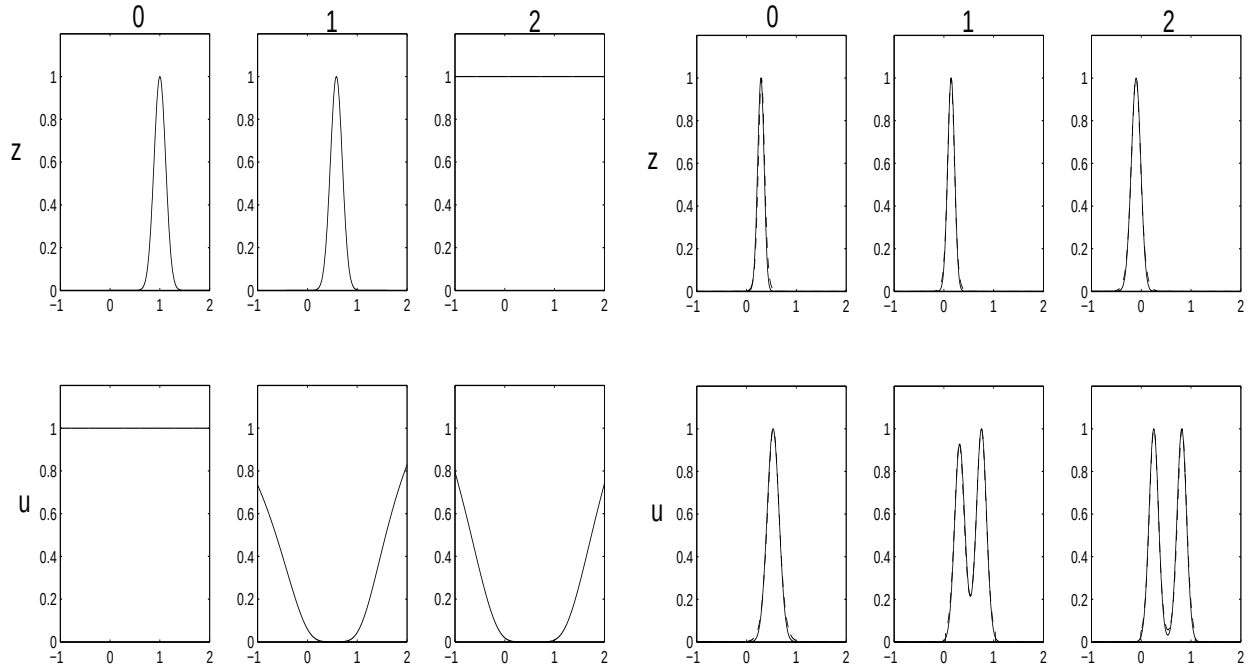


Figure 2: Gaussian example, auxiliary variables z (top) and u (bottom), Order 0, 1 and 2. Left: Functional dependencies on the densities for the auxiliary variable, as it enters in the proposal along the line. Right: Proposal densities along the line (solid) and corresponding approximations (dashed).

Consider next the case with conditioning on z. A density for the one dimensional value t that gives unit acceptance rate is

$$q(t|z,x) = \frac{|1-t|^{n-1}h(z|x+t[z-x])\pi(x+t[z-x])}{\int_{-\infty}^{\infty}|1-r|^{n-1}h(z|x+r[z-x])\pi(x+r[z-x])dr}. \tag{18}$$

Again everything in the acceptance rate in (10) cancels, including normalizing constants and the Jacobian term. The density along the line is again a function of the density we choose for z, and for all densities $h(z|x)$ we get acceptance rate one.

We illustrate the possibilities with a Gaussian example,

$$\pi(x) = N(x;\mathbf{0},\Sigma), \qquad \Sigma_{ij} = \gamma^{I(i\neq j)}, \quad i,j = 1,\ldots,n. \tag{19}$$

We set $n = 50$ and $\gamma = 0.25$ in the illustration and consider $(z,0)$, $(u,0)$, $(z,1)$, $(u,1)$, $(z,2)$, and $(u,2)$. In this Gaussian case we have $(z,f) = (z,2)$ and $(u,f) = (u,2)$ if we choose $h = \pi$ for the fixed density. In our illustration both x and z are drawn from the target and are kept fixed. In Figure 2(left) we plot densities for the auxiliary variables as they take part in $q(t|z,x)$ and $q(t|u,x)$, i.e. $h(z|x+t(z-x))$ and $g(u|x+tu)$ as functions of t. We also plot the corresponding $q(t|z,x)$ and $q(t|u,x)$ in Figure 2(right). It is interesting to compare the densities we get using u and z. Since both u and z define the same line, the difference in mixing properties is indicated by the step length along the line. For easier comparison we scaled the graphs in Figure 2 so that x refers to $t = 0$, and z to $t = 1$. The h function is centered at $t = 1$ for the zero order approximation, for the first order approximation it is largest between 0 and 1, and for the second order approximation

it is constant in this Gaussian case. The g function changes from being constant in the zero order approximation, to a bathtub shaped curve for first and second order approximations. The shape for g occurs because the u direction tends to get more likely as we go towards the tails of the distribution.

The resulting proposal densities are very different as can be seen in Figure 2. The form of the graphs can be tied to the densities defined in equations (17) and (18). For $q(t|z,x)$ the $|1-t|^{n-1}$ factor is involved when conditioning on z. The proposal $q(t|z,x)$ hence goes to zero at $t=1$. The density has at least one mode on each side of $t=1$. The $|1-t|^{n-1}$ term increases rapidly away from z, especially in high dimensions, and this results in a sharp shape on $q(t|z,x)$. Further, in high dimension x and z commonly lie in the tail with a high density region between the two points. Because of the $|1-t|^{n-1}$ shape, the q density typically has most of its mass in the mode near x, and negligible mass in the mode for $t>1$. The proposal density conditional on u is unimodal for small dimensions ($n<5$) and for the $(u,0)$ case (see Figure 2), but as the dimension increases the bathtub shaped g density for the first and second order cases results in a bimodal shape for $q(t|u,x)$. For high dimensions, the probability mass is almost always near $\frac{1}{2}$ in each mode, caused by the symmetry of the Gaussian distribution and the fact that samples typically are in the tail of the distribution in high dimensions. The second mode is usually far from x, and this should induce good mixing properties of the resulting Markov chain.

For the (u,f) case one can evaluate properties of the $g(u|x+tu)$ function that takes part in (17) using recursive formulas, see e.g. Pukkila and Rao (1988). The $g(u|x+tu)$ can be shown to be an even function around a symmetry point given by $t_{min}=(\mu_f-x)'\Sigma_f^{-1}u/u'\Sigma_f^{-1}u$. Hence, with the $g(u|x+tu)$ as part of the proposal we always try to throw mass away from this symmetry point.

3.3 Numeric approximation of q

The methods from Section 3.2 require draws from a one dimensional density $q(t|\phi,x)$, where ϕ is either an auxiliary point z or an auxiliary direction vector u. It is not possible to sample from these densities by fast methods such as inversion (Ripley, 1987) or adaptive rejection sampling (Gilks and Wild, 1992). We suggest numeric methods as part of this sampling step, fitting a density $\hat{q}(t|\phi,x)$ that resembles $q(t|\phi,x)$. Our approximation to the density is obtained by a numeric search for the modes of the distribution. We then fit a t-distribution in each mode.

Consider first the case with z as auxiliary variable. We know that q has one mode on each side of $t=1$. The first mode is obtained starting at the current point ($t=0$), while the other mode is located for $t>1$. When we condition on u we locate the first mode starting at $t=0$. From plots of densities (Figure 2) and the discussion above we recognize a bimodal shape for $q(t|u,x)$. In the numeric algorithm we step to both sides away from the first mode in order to locate a second mode. We typically find a significant mode on one side and no mode on the other.

Our numeric searches are based on stepping out to bracket a mode, and using a midpoint rule within the brackets. It seems robust, but faster searches are available, see e.g. Press et al. (1996). We fit the scale parameter of the t-distribution from the curvature (second derivatives) in the modes. The fitted densities that we obtain in this way are plotted in Figure 2 (right, dashed). One can hardly see the difference between q and $\hat{q}$ in this example.

The acceptance probability for proposed value $y\in\mathcal{R}^n$ using this numeric approximation be-

comes

$$\alpha(y|\phi,x) = \min\left(1, \frac{\pi(y)f(\phi|y)\hat{q}(s|\phi,y)}{\pi(x)f(\phi|x)\hat{q}(t|\phi,x)} \cdot |J_\phi|\right), \tag{20}$$

where f is either g or h, and ϕ is either u or z. In our experiments below the acceptance rate is close to one with this numeric approximimation. However, fitting a numeric approximation requires many evaluations of the target density, and each proposal takes much CPU time. Alternatively, we could use a simple, fixed proposal density with heavy tails, which is commonly done. This requires more tuning than in our numeric approach.

4 Examples

We first apply the directional MH algorithms to a Gaussian example. This provides a comparative study of the different directional algorithms outlined above. Random walk and Langevin algorithms are also implemented for this example. We next apply the (u, f) algorithm to a Bayesian spatial model for a seismic dataset from a North Sea petroleum reservoir. In this model we compare the algorithm with an independent proposal MH algorithm.

Integrated autocorrelation (IAC) is commonly used as an indicator of the performance of MCMC algorithms. The IAC for x can be estimated using a method from Geyer (1992), truncating the sum of sample autocorrelations at the point where noise dominates the estimated autocorrelation,

$$\text{IAC} = 1 + 2\sum_{t=1}^{2T+1}\rho_t, \quad \rho_t = \text{Corr}(x_s, x_{s+t}), \quad T = \max(\tau; \rho_{2t} + \rho_{2t+1} > 0 \text{ for all } t \leq \tau). \tag{21}$$

We use IAC as one attribute to compare our algorithms. The IAC makes it possible to calculate the number of iterations per independent sample. We also account for the number of evaluations (M) of the target density per iteration. A natural measure of algorithm cpu time is then the number of evaluations per independent sample. This becomes M $\times$ IAC.

4.1 Gaussian density

Consider the Gaussian target density (19). We first fix the parameters at $\gamma = 0.25$, and $n = 50$. We run the order 0, 1 and 2 approximations for both z and u. In addition, random walk and Langevin algorithms are used. We carefully tuned the parameters of random walk and Langevin to get close to the 'optimal' acceptance rates of 0.23 and 0.57, respectively, calculated in Roberts et al. (1997) and Roberts and Rosenthal (1998).

Figure 3 (top) shows autocorrelations for all algorithms plotted as a function of iteration number. Note that conditioning on u is always better than conditioning on z. We also note that the $(u, 2)$ algorithm has very short correlation length. The autocorrelation for the $(u, 1)$ algorithm also comes down quickly, but then levels out. We recognize this tendency for Langevin as well. This happens because the gradient does not point towards the center of the distribution when $\gamma \neq 0$. Figure 3 (top) is not entirely fair as a comparative plot, since the cpu time for one iteration differs dramatically for the various algorithms. The random walk algorithm requires only one evaluation of the target density per iteration, Langevin requires two evaluations, while the $(u, 2)$ algorithm,

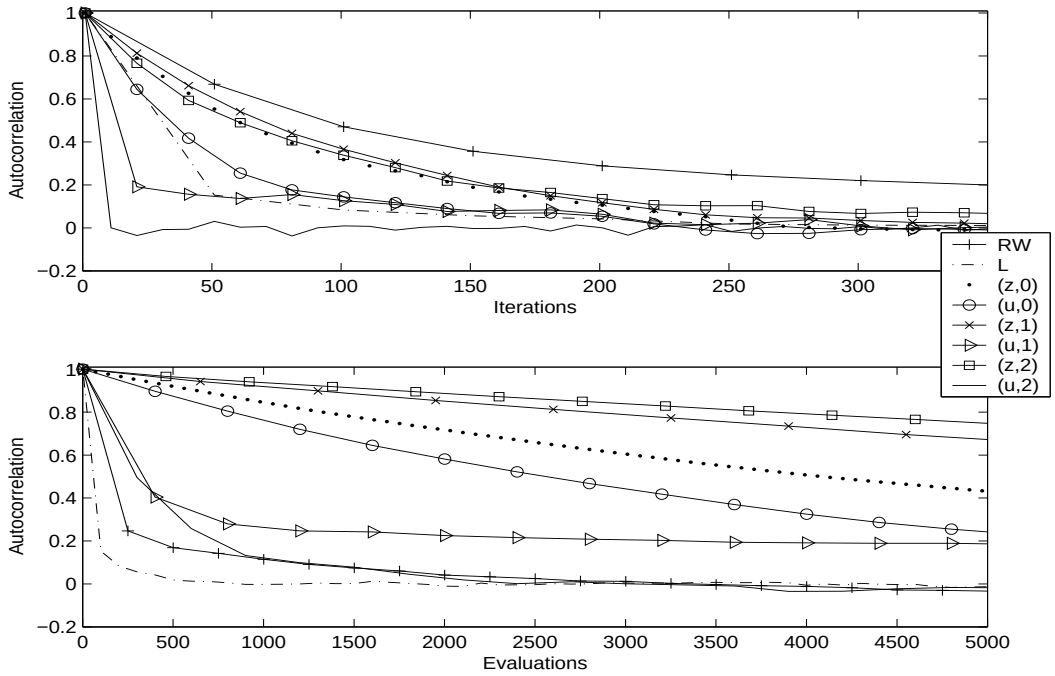


Figure 3: Gaussian example ($n = 50$, $\gamma = 0.25$): Top: Autocorrelation as a function of iterations for different algorithms. Bottom: Autocorrelation as a function of the number of evaluations.

with calculation of second order derivatives and a numeric search, uses approximately 300 evaluations per iteration. In Figure 3 (bottom) we correct for the mean number of evaluations required per iteration. The autocorrelation is then plotted as a function of the number of evaluations. Random walk and Langevin perform better than the directional algorithms in this case. The $(u, 2)$ algorithm is comparable to random walk and Langvin, while conditioning on z is not effective.

We now study the various algorithms for both $n = 50$ and $n = 100$, keeping $\gamma = 0.25$. Table 1 displays several aspects of the MH algorithms. The mean number of evaluations per iteration is presented for all cases. This is largest for the second order approximations because we require evaluations of both first and second derivatives of π at each step in the numerical approximation. It is slightly larger for u than for z, because of the design of our numeric approximation $\hat{q}$. Also shown in Table 1 are the acceptance rates for all algorithms. The acceptance rates for the directional MH algorithms are all close to one, showing that the fitted densities are good approximations of the intractable densities in (17) and (18). Some variability exists because the q density is more skewed for some of the algorithms, and $\hat{q}$ does not match q that well in these cases. Number of evaluations and acceptance rates do not seem to be affected by dimension. The mean step length between successive samples is larger for $(u, 2)$ than in the other directional algorithms. The step lengths for $(u, 1)$ and Langevin are also large, but the moves are along 'bad' directions. Langevin and random walk obtain smaller moves in larger dimensions ($n = 100$). Note also that the step length does not increase much with the approximation order for z. In Table 1 we also present estimated IAC from (21). IAC is smallest for the $(u, 2)$ algorithm, and much larger using z than u. It increases as dimension increases. From the estimated number of evaluations per independent sample displayed in Table 1 we note that this is relatively small for Langevin, random walk and

Table 1: Gaussian example: Attributes for the directional MH algorithms $(z, 0)$,. . .,$(u, 2)$, Langevin (L) and random walk (RW). Parameter is $\gamma = 0.25$.

n=50

Mean number of target evaluations per iteration

	0	1	2
z	70	130	230
u	80	200	300
L	2		
RW	1		

Mean acceptance rates:

	0	1	2
z	0.96	0.96	0.96
u	0.95	0.96	0.95
L	0.59		
RW	0.26		

Mean jump length:

	0	1	2
z	0.6	0.8	0.9
u	1.0	4.7	5.2
L	3.3		
RW	0.4		

Estimated Integrated Autocorrelation:

	0	1	2
z	170	190	105
u	100	60	5
L	625		
RW	1210		

Evaluations per independent sample ($\times 1000$):

	0	1	2
z	12	25	24
u	8	12	1.5
L	1.3		
RW	1.2		

n=100

	0	1	2
z	70	170	240
u	80	220	310
L	2		
RW	1		

	0	1	2
z	0.97	0.95	0.96
u	0.95	0.98	0.97
L	0.62		
RW	0.28		

	0	1	2
z	0.6	0.7	0.9
u	0.9	6.8	7.7
L	2.6		
RW	0.4		

	0	1	2
z	495	300	190
u	295	140	6
L	680		
RW	4500		

	0	1	2
z	34	50	45
u	24	30	1.8
L	1.4		
RW	4.5		

Table 2: Gaussian example: Estimated number of evaluations per independent sample ($M \times IAC$), presented in thousands, for directional MH algorithms $(z, 2)$ and $(u, 2)$, Langevin (L) and random walk (RW) algorithms for dimension n and correlation γ. The * means that we could not estimate this value in reasonable time.

n	50	50	100	100
γ	0.25	0.75	0.25	0.75
(z,2)	24	59	45	*
(u,2)	1.5	2.4	1.8	2.0
L	1.3	2.9	1.4	4.3
RW	1.2	15	4.5	32

$(u, 2)$. The numbers are smaller for auxiliary variable u than for z. For all algorithms the number of evalutions required per independent sample is larger for $n = 100$ than for $n = 50$.

We also simulate from the Gaussian example with parameter $\gamma = 0.75$ to study correlation effects. In Table 2 we summarize the estimated number of evaluations per independent sample for $\gamma = 0.25$, 0.75 and $n = 50$, 100. We present only the second order directional MH algorithms $(z, 2)$ and $(u, 2)$. From Table 2 the directional MH algorithm $(u, 2)$ is comparable to Langevin and random walk. Hence the cpu time needed for statistical inference should be about the same. Algorithm $(u, 2)$ appears to carry less dimension and correlation effects than the other algorithms. From Table 2 it is clear that conditioning on u gives a significant improvement compared to conditioning on z. Both Langevin and random walk require careful tuning to perform well. There is no tuning requirements for the directional algorithms, and this automatic nature of the directional MH algorithms is attractive.

4.2 Seismic inversion

Seismic data analysis enables petroleum companies to characterize the subsurface since seismic measurements are linked to rock and fluid properties. See e.g. Sheriff and Geldart (1995) for a general overview on the interpretation of seismic data. We analyze seismic data from a two dimensional domain in the Gullfaks petroleum reservoir in the North Sea. We use a Bayesian model and study the posterior of elastic reservoir parameters conditioned on seismic data.

Stochastic model

The seismic data are represented on a 128×8 grid and for three different angles. Figure 4 shows the data for each angle. This dataset is studied from a geophysical viewpoint in Landrø and Strønen (2003). Each gridnode covers $4ms$ in the vertical direction and $25m$ in the lateral direction. The grid hence covers approximately $500ms$ ($\approx 400m$) in depth and $200m$ laterally.

We represent the elastic reservoir parameters on the same 128×8 seismic grid. Let $x = \{x_{ij}; i = 1, \ldots, 128; j = 1, \ldots, 8\}$ denote the elastic parameters. Each x_{ij} consists of three variables and we set x_{ij}=(α_{ij}, β_{ij}, ρ_{ij}) for the variable at grid node (i, j), where α_{ij} and β_{ij} refers to logarithms of primary and secondary velocities, while ρ_{ij} is logarithm of mass density.

The seismic data $d = \{d_{ij}; i = 1, \ldots, 128; j = 1, \ldots, 8\}$, where $d_{ij} = (d^1_{ij}, d^2_{ij}, d^3_{ij})$ is the

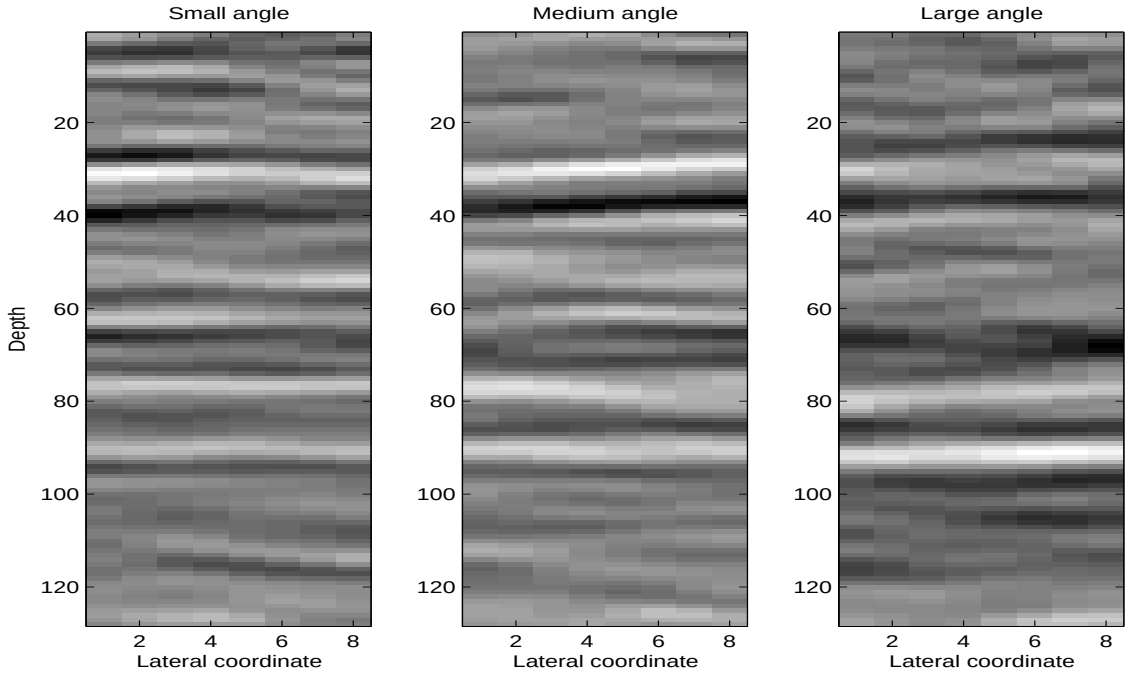


Figure 4: Seismic data from the Gullfaks reservoir. Data are collected for three angles.

collection of measurements for the three angles at node (i, j), is modeled by

$$d = WA(x) + \epsilon. \tag{22}$$

where $\epsilon = \{\epsilon_{ij}; i = 1, \ldots, 128; j = 1, \ldots, 8\}$ is a Gaussian error term. The matrix W represents a convolution model, while $A(x)$ is a non-linear link between x and the seismic reflections, see e.g. Sheriff and Geldart (1995) and Buland and Omre (2003).

Following Buland and Omre (2003), we model a Bayesian solution to the seismic inversion problem using a Gaussian prior for the elastic parameters,

$$\pi_0(x) = N(x; \mu_0, \Sigma_0), \tag{23}$$

where μ_0 and Σ_0 are prior mean and covariance matrix. We assign a multiplicative structure for Σ_0

$$\Sigma_0 = S \otimes V_0, \tag{24}$$

where S is a 3×3 covariance matrix for x_{ij}, V_0 is a spatial correlation matrix for each of $\alpha = \{\alpha_{ij}; i = 1, \ldots, 128; j = 1, \ldots, 8\}$, $\beta = \{\beta_{ij}; i = 1, \ldots, 128; j = 1, \ldots, 8\}$ and $\rho = \{\rho_{ij}; i = 1, \ldots, 128; j = 1, \ldots, 8\}$, and $\otimes$ denotes the Kronecker product. The likelihood is defined from (22) as

$$l(d|x) = N(d; WA(x), \Sigma_l), \tag{25}$$

where Σ_l is the covariance matrix for the observation error term. Also for this covariance matrix we assign a multiplicate covariance structure

$$\Sigma_l = T \otimes V_l, \tag{26}$$

where T is a 3×3 covariance matrix for ϵ_{ij}, and V_l contains the autocorrelations for $d^1 = \{d^1_{ij}; i = 1, \ldots, 128; j = 1, \ldots, 8\}$, $d^2 = \{d^2_{ij}; i = 1, \ldots, 128; j = 1, \ldots, 8\}$, and $d^3 = \{d^3_{ij}; i = 1, \ldots, 128; j = 1, \ldots, 8\}$. The posterior of interest is then

$$\pi(x|d) \propto l(d|x)\pi_0(x). \tag{27}$$

Because of the non-linear $A(x)$, this posterior is not analytically available.

Parameter values in prior and likelihood are copied from Buland et al. (2003). Briefly, this indicates that μ_x is a constant trivariate vector as a function of depth, set to $(8.0, 7.3, 7.7)$ for logarithms of primary and secondary velocities and mass density. The convolution window in the W matrix has length equal to 11 gridnodes.

Linearized model

Buland and Omre (2003) obtained an analytical Bayesian solution to the inversion problem. In short, they linearize the likelihood, getting

$$l_{lin}(d|x) = N(d; WAx, \Sigma_l), \tag{28}$$

where the A matrix is a linearized version of the $A(\cdot)$ function in (22). The matrix W remains the same. The posterior is then Gaussian with

$$\pi_{lin}(x|d) \propto l_{lin}(d|x)\pi_0(x) \propto N(x; \mu_{x|d}, \Sigma_{x|d}), \tag{29}$$

where $\mu_{x|d}$ and $\Sigma_{x|d}$ are available from standard Gaussian theory.

For our approach, the linearized posterior provides a fixed density for the auxiliary variable defined in (12). Moreover, the approximate posterior provides a proposal density for an independent proposal MH algorithm. The fixed density is a good approximation to the target if the linearization gives a good approximation to the non-linear model.

Fast Fourier transform

On the grid of size 128×8 it is very time consuming to draw a Gaussian variable x from the Gaussian model in (29) and also to evaluate the model in (27). Chan and Wood (1997) present an algorithm using the FFT (Fast Fourier Transform) to simulate Gaussian models. Buland et al. (2003) discuss the approach for their linearized model. If we let $\tilde{x} = Fx$ denote the two dimensional FFT of x and $\tilde{d} = Fd$ the two dimensional FFT of d, they show that $\pi_{lin}(\tilde{x}|\tilde{d})$ is Gaussian with a block diagonal covariance matrix. Computations can hence be done efficiently. Evaluation of the non-linear $\pi(\tilde{x}|\tilde{d})$ requires somewhat more computations.

The FFT calculations require that we wrap the two dimensional grid into a torus. In this way we also model circulant, positive definite correlation matrices for V_0 and V_l. When we wrap the grid into a torus we get the wrong correlation values at the boundaries of the grid. This entails bias and underestimation of the variance at the edges. Alternatively, we could simulate a larger domain, and wrap this onto a torus, but then we have no seismic data available on the additional part of the torus. To obtain a correct model we must then simulate the missing data as part of the Bayesian model. We do not consider this approach here.

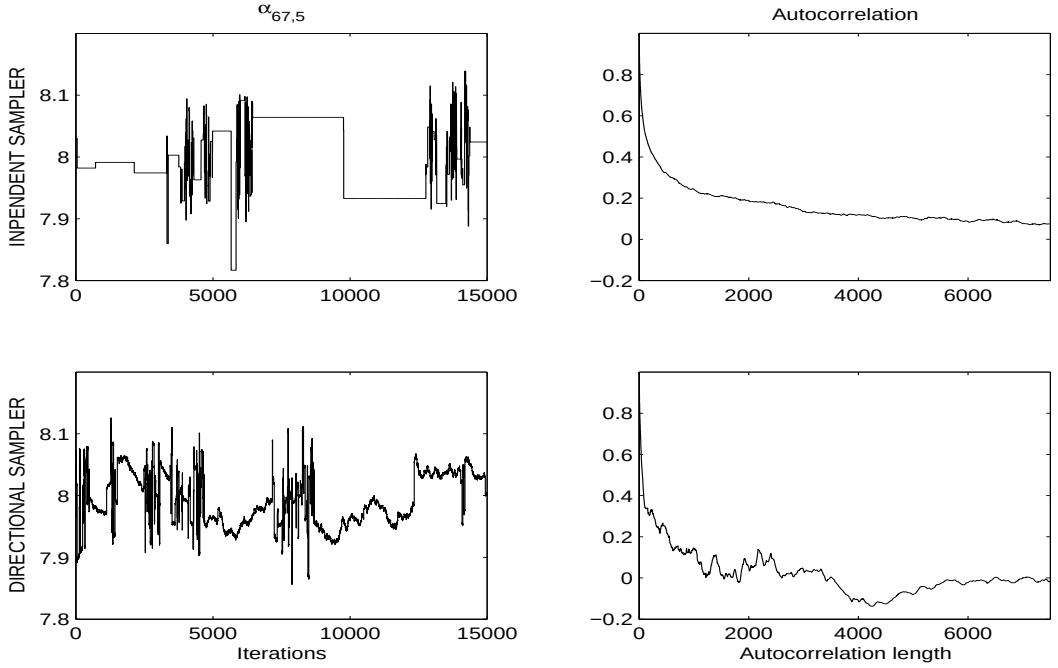


Figure 5: Left: Trace plot for primary velocity ($\alpha_{67,5}$) using independent proposal MH (top) and directional (u, f) MH (bottom). Right: Autocorrelation plots for primary velocities ($\alpha_{67,5}$) using independent proposal MH (top) and directional (u, f) MH (bottom).

Algorithms

We implement the directional (u, f) MH algorithm and an independent proposal MH algorithm for this application. We use the˜notation for $\tilde{x}$, $\tilde{u}$ and $\tilde{d}$ since the simulations and evaluations are done in the Fourier domain. The (u, f) updating scheme uses $\pi_{lin}(\tilde{z}|\tilde{d})$ as the fixed density h. From $\tilde{z}$ and $\tilde{x}$ we find the auxiliary variable $\tilde{u}$ by (6). This value of $\tilde{u}$ has density g as in (11). The $q(t|\tilde{u}, \tilde{x})$ density along the line is available in (17) with $\pi(\tilde{x}|\tilde{d})$ as the target density. The independent proposal MH algorithm uses the linearized density $\pi_{lin}(\tilde{y}|\tilde{d})$ as a proposal density for new state y. The acceptance rate is then

$$\alpha(\tilde{y}|\tilde{x}) = \min\left(1, \frac{\pi(\tilde{y}|\tilde{d})\pi_{lin}(\tilde{x}|\tilde{d})}{\pi(\tilde{x}|\tilde{d})\pi_{lin}(\tilde{y}|\tilde{d})}\right). \tag{30}$$

Results

In Figure 5 (left) we show trace plots of $\alpha_{67,5}$ for both independent proposal and directional (u, f) MH samplers. These trace plots show $15\,000$ iterations and illustrate the appearance of each sampler. The independence sampler (Figure 5, top left) either performs large moves or remains in the same state (no accept). The trace plot for (u, f) very rarely remain at the same state, see Figure 5, bottom left. It either moves long distances or only small distances. This apperance occurs because of the bimodal proposal distribution $q(t|\tilde{u}, \tilde{x})$. At some iterations this proposal density has two modes with significant mass (large moves), while at other times it has almost all mass in the mode closer to 0 (small moves). The mean acceptance rate for the (u, f) algorithm is 0.94, while

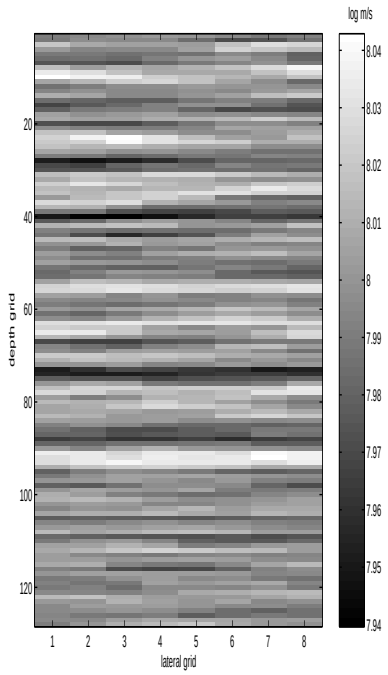


Figure 6: Estimated mean of logarithms of primary velocity α.

the mean acceptance rate for independent proposal is 0.02.

Figure 5, right, displays the estimated autocorrelation for both algorithms. The autocorrelations are presented as a function of iteration number. For the independent sampler this was estimated from $2\,00000$ iterations, for the (u, f) algorithm $25\,000$ iterations were used. The (u, f) algorithm has smaller correlation length than independent proposal. Estimated IAC from (21) is 550 for (u, f), compared to 3510 for independent proposal. The number of evaluations per iteration (M) is 70 for (u, f) compared to 1 for independent proposal. This means that the estimated number of evaluations per independent sample (M $\times$ IAC) is 38500 for (u, f), while it is 3510 for independent proposal MH.

Figure 6 shows marginal posterior mean of logarithms of primary velocity, α, estimated from $25\,000$ subsequent samples from (27) obtained by the (u, f) algorithm. Primary velocity is the best determined variable for this dataset. The vertical fluctuations in primary velocity captures some of the changes in reservoir properties. Since the prior density models no trend in velocity, the vertical trends that can be seen in Figure 6 are caused by the seismic data. Primary velocity is small around vertical depth $70 - 75$. This zone has been interpreted as the top of the reservoir (Landrø and Strønen, 2003). Another low velocity zone is recognized at vertical depth $85 - 90$, while primary velocity increases at $90 - 95$. The change at 90 was interpreted as a change in fluid saturation (Landrø and Strønen, 2003), a shift from oil to water saturated rocks. Primary velocity is larger in water-filled sand than in oil-filled sand, and we recognize this in the estimate.

The example shows that the directional algorithm performs adequately on a non-linear high dimensional example. However, since the independent proposal also performs quite well, we conclude that the linearized model from Buland et al. (2003) is a good approximation to the model

in this application. In this example we used the simplest form of approximation for the auxiliary variable from section 3, namely the (u, f) algorithm. A more sophisticated proposal for the auxiliary variable might have worked better. However, since we apply FFT we require stationarity of the Gaussian model. Moreover, the computation time increases if we use for example the current value of α, β, ρ in the linearization rather than the prior values.

We also tried to reduce the variance in the measurement noise ϵ. As this variance term decreases, the posterior distribution gets more non-linear, and both directional (u, f) and independent proposal MH algorithms run into problems. For the independent proposal we get acceptance rates very close to zero, whereas the (u, f) algorithm mix slowly because the $q(t|u, x)$ density has almost all mass in the mode near the current state and only small moves are established.

5 Closing remarks

This paper presents a framework for directional MH algorithms. We use an auxiliary variable to define the line, and next sample along this line. We apply our directional MH algorithms for a Gaussian density and in a non-linear Bayesian model. From our experiments it appears as if using an auxiliary direction vector is preferable to an auxiliary point. By using direction vector as an auxiliary variable, we are able to construct directional MH algorithms that mix well in high dimensions. Good mixing properties are achieved primarily because the proposal mechanism throws mass away from the current point. In our algorithms we obtain a bimodal proposal density with one mode far away from the current point, and hence large moves are encouraged.

The algorithms derived in this paper differ from other approaches since we draw the auxiliary variables conditional on properties of the target at the current state. We look for algorithms with unit acceptance rate and the proposal density along the line then depends on the density for auxiliary variable in a particular way. In our approach we recognize this proposal from the acceptance rate. Erland (2003) presents another directional MH algorithm with unit acceptance rate. His algorithm is constructed differently and does not belong to the class of algorithms that we present. This shows that there are more directional MH algorihtms with unit acceptance rate, but it is not clear to us how these are connected.

In the paper outline several algorithms using properties of the target densities to various degrees. The Order 2 approximation defined in Section 3.1 works better than the Order 1 approximation. However, our Order 2 approximation is not directly applicable when the matrix of second derivatives is non-positive definite. More sophisticated methods could be of interest in this step, such as using ideas from generalized Langevin diffusions MH studied by Stramer and Tweedie (1999).

In the MH setup we can use other proposal mechanisms than the particular ones above. For example we could choose $q^\star(t|u, x) = |t|q(t|u, x)$. Such proposals no longer give unit acceptance rate, but when the proposed value is accepted, it is far from x. This is related to antithetic ideas.

Directional MH algorithms transform the problem of a tricky target density to a tricky proposal density in one dimension. Our directional MH algorithms require few iterations to mix well, but use many evaluations per iteration. Random walk algorithms, on the other hand, require many iterations to mix well, but use only one evaluation per iteration. It is situation dependent which algorithm is better. In our examples the directional MH algorithms perform better than the ones we compare them with if we measure efficiency per iteration. Further research is required to derive fast

proposal mechanisms for the one dimensional proposal density q. This would reduce the number of evaluations per iteration. However, one advantage with the directional MH algorithms is that no tuning is required. A fast proposal mechanism should remain robust in this respect.

References

Buland,A., Kolbjørnsen,O. and Omre,H., 2003, Rapid spatially coupled AVO inversion in the Fourier domain, Geophysics, **68**, 824-836.

Buland,A. and Omre,H., 2003, Bayesian linearized AVO inversion, Geophysics, **68**, 251-272.

Chan,G. and Wood,A.T.A., 1997, An algorithm for simulating stationary Gaussian random fields, Applied Statistics, **46**, 171-181.

Chen,M. and Schmeiser,B., 1993, Performance of the Gibbs, Hit-and-Run and Metropolis Samplers, Journal of computational and graphical statistics, **2**, 251-272.

Erland,S., 2003, Metropolized hit-and-run, In preparation.

Geyer,C.J., 1992, Practical Markov chain Monte Carlo, Statistical Science, **7**, 473-483.

Gilks,W.R. and Wild,P., 1992, Adaptive rejection sampling for Gibbs sampling, Applied Statistics, **41**, 337-348.

Gilks,W.R., Roberts,G.O. and George,E.I., 1994, Adaptive direction sampling, The Statistician, **43**, 179-189.

Gilks,W.R., Richardson,S. and Spiegelhalter,D.J., 1996, Markov chain Monte Carlo in practice, Chapman and Hall.

Goodman,J. and Sokal,A.D., 1989, Multigrid Monte Carlo method. Conceptual foundations, Physical Review D, **40**, 2035-2072.

Green,P., 1995, Reversible jump Markov chain Monte Carlo computation and Bayesian model determination, Biometrika **82**, 711-732.

Hastings,W.K., 1970, Monte Carlo simulation methods using Markov chains and their applications, Biometrika, **57**, 97-109.

Kaufman,D.E. and Smith,R.L., 1994, Direction choice for accelerated convergence in hit-and-run sampling, Operations Research, **46**, 84-95.

Landrø,M. and Strønen,L.K., 2003, Fluid effects on seismic data - observations from the Gullfaks 4D study, Submitted.

Liu,J.S., Liang,F. and Wong,W.H., 2000, The multiple-try method and local optimization in Metropolis sampling, Journal of American Statistical Association, **95**, 121-134.

Liu,J.S. and Sabatti,C., 2000, Generalized Gibbs sampler and multigrid Monte Carlo for Bayesian computation, Biometrika, **87**, 353-369.

Press,W.H., Teukolsky,S.A., Vetterling,W.T., and Flannery,B.P., 1996, Numerical Recipes in C:The art of Scientific Computing, Cambridge University Press.

Pukkila,T.M. and Rao,C.R., 1988, Pattern recognition based on scale invariant discriminant functions, Information sciences, **45**, 379-389.

Ripley,B., 1987, Stochastic simulation, Wiley.

Roberts,G.O. and Gilks,W.R., 1994, Convergence of adaptive direction sampling, Journal of multivariate analysis, **49**, 287-298.

Roberts,G.O., Gelman,A. and Gilks,W.R., 1997, Weak convergence and optimal scaling of random walk Metropolis algorithms, Annals of Applied Probability, **7**, 110-120.

Roberts,G.O. and Rosenthal,J.S., 1998, Optimal scaling of discrete approximations to Langevin diffusions, Journal of Royal Statistical Society, Series B, **60**, 255-268.

Sheriff,R.E. and Geldart,L.P., 1995, Exploration seismology, Cambridge.

Stramer,O. and Tweedie,R.L., 2003, Langevin-Type Models II: Self-targeting candidates for MCMC algorithms, Methodology & Computing in Applied probability, **1**, 307-328.

Waagepetersen, R. and Sørensen, D., 2001, A tutorial on reversible jump MCMC with a view towards applications in QTL-mapping, International Statistical Review, **69**, 49-61.

Watson,G.S., 1983, Statistics on spheres, Wiley-Interscience.