NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

An MCMC algorithm for sampling from the posterior conditioned to production history

by

Håkon Tjelmeland

PREPRINT
STATISTICS NO. 2/2000

NORWEGIAN UNIVERSITY OF SCIENCE AND
TECHNOLOGY
TRONDHEIM, NORWAY

This report has URL http://www.math.ntnu.no/preprint/statistics/2000/S2-2000.ps
Håkon Tjelmeland has homepage: http://www.math.ntnu.no/~haakont
E-mail: haakont@stat.ntnu.no
Address: Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491
Trondheim, Norway.

An MCMC algorithm for sampling from the posterior conditioned to production history

Håkon Tjelmeland
Department of Mathematical Sciences
Norwegian University of Science and Technology
N-7491 Trondheim, Norway
e-mail: Haakon.Tjelmeland@stat.ntnu.no
URL: http://www.math.ntnu.no/~haakont

Abstract

In evaluation of petroleum reservoirs, observed production history contains valuable information. A natural approach to integrate this type of data with other sources of observations is to adopt a Bayesian model. To sample from the resulting posterior distribution, MCMC algorithms have to be applied. Even if MCMC provides a very flexible framework in which to construct simulation algorithms, so far only standard MCMC algorithms have been employed for sampling from this type of posterior. As a result, a large number of iterations has been necessary to obtain convergence. In this paper, we first discuss the characteristics of the type of distribution in question and thereafter define an MCMC algorithm specifically designed to cope with these features.

1 Introduction

In evaluation of petroleum reservoirs, the observed production history contains valuable information. Recently, much attention has focused on how to integrate this type of data with other sources of information to obtain an improved description of reservoir variables and thereby a better description of future production with associated uncertainty. The problem is addressed either within a Bayesian context (Oliver et al., 1997; Hegstad, 1997) or defined as an optimisation problem (Sahuquillo et al., 1992). Only the first alternative allow consistent treatment of different types of observations.

Within the Bayesian scheme, two quite different modelling strategies has been suggested, the main difference being how the fluid flow simulator is considered in the stochastic model. The first possibility, which is dominating in the geostatistical literature, is to include the fluid flow simulator as part of the likelihood function for production data. Then, to sample from the resulting posterior distribution, Markov chain Monte Carlo (MCMC) is the only viable alternative. As this can be extremely resource requiring also faster approximative algorithms have been suggested, see Oliver et al. (1997) and Floris and Bos (1998). However, the quality of the approximations are not necessarily good, see Tjelmeland (1997). The alternative Bayesian modelling option is to consider the fluid flow process as an unknown function for which a prior is defined, see O'Hagan et al. (1998) and references therein. Thus, results from runs of the fluid flow simulator is included in the model as observations.

In this paper, we consider the "history matching" problem within a Bayesian formulation and adopt the first modelling alternative discussed above. Within this framework, the focus here is on how to design efficient MCMC algorithms for sampling from the resulting posterior distribution. MCMC algorithms is iterative and a stochastic rule for proposing potential new states is the principal part of the procedure. Very large flexibility exists in the choice of the proposal rule and this choice is essential for the rate of convergence. As each iteration requires evaluation of a fluid flow simulator, a very fast convergence is essential for the procedure to be of any practical value in a history matching context.

2 Metropolis-Hastings algorithm

In this section we give a brief description of the general Metropolis-Hastings algorithm. For nice reviews of theory and different techniques within the Metropolis-Hastings class, see, for example, Chib and Greenberg (1995) and references therein.

The Metropolis-Hastings algorithm has in recent years become the basic general procedure to generate samples from a specified probability distribution, $f(x)$ say. The algorithm is iterative and each iteration consists of two steps; with x denoting the current state, a potential new state x' is first generated according to a probability distribution $\mathrm{Q}^{\varphi}(x'|x)$ and thereafter accepted with an acceptance probability $\alpha^{\varphi}(x'|x)$. It should be noted that very large flexibility exists in the choice of proposal distribution $\mathrm{Q}^{\varphi}(x'|x)$ and, as indicated in the notation, it may depend on a stochastic quantity φ, which should be independent of everything else. Let $f(\varphi)$ denote the distribution for φ. The choice of $f(\varphi)$ and $\mathrm{Q}^{\varphi}(x'|x)$ has no effect on the limiting distribution of the algorithm, but is crucial for the rate of convergence and thereby for the number of iterations necessary before convergence.

In the following, the adopt the setup of Tjelmeland and Hegstad (1999) and let the proposal distribution be given as a combination of two distributions;

$$\mathrm{Q}^{\varphi}(x'|x) = \frac{1}{2}\mathrm{Q}_0^{\varphi}(x'|x) + \frac{1}{2}\mathrm{Q}_1^{\varphi}(x'|x). \tag{1}$$

Thus, the x' is generated by 1) simulate a φ from $f(\varphi)$, 2) draw $i = 0$ or 1 with probability 1/2 for each and 3) simulate x' according to $\mathrm{Q}_i^{\varphi}(\cdot|x)$. Obviously, this requires $f(\varphi)$, $\mathrm{Q}_0^{\varphi}(x'|x)$ and $\mathrm{Q}_1^{\varphi}(x'|x)$ to be chosen to allow direct sampling from these distributions. The probability of accepting x' as the new state should be

$$\alpha^{\varphi}(x'|x) = \min\left\{1, \frac{f(x')}{f(x)}\frac{\mathrm{Q}_{1-i}^{\varphi}(x|x')}{\mathrm{Q}_i^{\varphi}(x'|x)}\right\}, \tag{2}$$

where φ and i are the values generated in 1) and 2) above. Thus, the acceptance step should be implemented as 1) draw u uniformly on the interval $[0,1]$ and 2) if $u \leq \alpha^{\varphi}(x'|x)$ let x' be the new current state, otherwise keep x as the current state.

An important special case of equation (1) is with $\mathrm{Q}_0^{\varphi}(x'|x)$ and $\mathrm{Q}_1^{\varphi}(x'|x)$ equal. This is by far the most popular choice, but the possibility of combining two different proposal densities becomes important if the target distribution, $f(x)$, is multi-modal, i.e. if $f(x)$ has several high density areas separated with large regions of very low probability.

3 Bayesian model formulation

In this section we define a stochastic model for the problem. We adopt a Gaussian based Bayesian model of the same type as previously defined in Hegstad and Omre (1997).

Let $r \in \Re^n$ be a vector of the reservoir characteristics of interest, possibly after a suitable transformation. Conditioned on a vector of model parameters, θ, we assume r to be Gaussian distributed with mean $F\theta$ and a given covariance matrix Σ_r, i.e.

$$r = F\theta + \varepsilon_r, \tag{3}$$

where F is a given matrix and ε_r is multiGaussian with vanishing mean and covariance Σ_r. For the model parameters θ, we adopt a Gaussian distribution with a given mean vector θ_0 and a covariance matrix Σ_θ. It should be noted that this prior model is the same as used for Bayesian kriging, see Omre and Halvorsen (1989) and Hjort and Omre (1994).

We consider two types of observations; well observations and production data, denoted d_w and d_p, respectively. For d_w we assume the Gaussian linear model

$$d_w = Ar + \varepsilon_w, \tag{4}$$

where A is a given matrix and ε_w is a Gaussian distributed error term with zero mean and a given covariance matrix Σ_w. Typically, d_w is low dimensional and the matrix A is sparse with non-zero elements corresponding to well locations. For a given production strategy, p, we model d_p as

$$d_p = \omega(r,p) + \varepsilon_p, \tag{5}$$

where $\omega(r,p)$ is the production as predicted by a fluid flow simulator and the error term ε_p is assumed Gaussian with zero mean and a given covariance matrix Σ_p.

The assumptions above specify the four densities $f(\theta)$, $f(r|\theta)$, $f(d_w|r)$ and $f(d_p|r)$ and thereby the posterior distribution of interest, $f(r|d_w,d_p)$. The non-linearity of $\omega(r,p)$ implies that $f(r|d_w,d_p)$ becomes non-Gaussian and analytically unavailable. However, the partial posterior $f(r|d_w)$ is Gaussian, a fact that can be used to reparameterise the model to a form more suitable for defining efficient MCMC algorithms.

3.1 Reparameterisation

Let v be a (low dimensional) vector defined from r by

$$v = Br \quad ; \quad v \in \Re^m, \tag{6}$$

where $B \in \Re^{m\times n}$ is a given matrix. For the validity of the reparameterisation defined below and the simulation algorithm defined in the next section, the choice of the matrix B is of no importance. However, the choice of B is crucial for the effectiveness of the simulation. Ideally, B should be chosen so that v includes the characteristics of r that is most important in the posterior distribution. We leave further discussion of this topic to Section 5 and for now let B be some fixed matrix.

Given a matrix B, define a one to one transformation from r to the pair (v,ε), where v is given by equation (6) and

$$\varepsilon = r - \mathrm{E}[r|d_w, Br] \quad ; \quad \varepsilon \in \{\varepsilon \in \Re^n | B\varepsilon = 0\}. \tag{7}$$

Thus, the inverse transformation, from (v,ε) to r is given by

$$r = \mathrm{E}[r|d_w, Br = v] + \varepsilon, \tag{8}$$

i.e. r is expressed as the sum of a smooth kriged field given by v and the residual field ε. Both v and ε are linear in r and thereby joint multiGaussian. Given d_w, the expectation vector and covariance matrix of v is easily found to be

$$\mathrm{E}[v|d_w] = BF\theta_0 \quad \text{and} \quad \mathrm{Cov}[v|d_w] = B(F\Sigma_\theta F^T + \Sigma_r)B^T, \tag{9}$$

respectively. Moreover, from the construction it readily follows that $\mathrm{Cov}[v,\varepsilon|d_w] = 0$, i.e. v and ε are conditionally independent given d_w. Thus, the posterior distribution of interest, $f(r|d_w,d_p)$, can be re-expressed by v and ε in the form

$$f(v,\varepsilon|d_w,d_p) \propto f(v|d_w)f(\varepsilon|d_w)f(d_p|v,\varepsilon). \tag{10}$$

It should be noted that the vector of model parameters is integrated out and thereby no longer explicitly represented in the reparameterised model. Obviously, this does not change the model.

We end this section by some remarks on computational aspects. First, as the dimension of v is small, computation of $f(v|d_w)$ for a given v is easily done by equation (9). Note, however, that care should be taken in computation of $\mathrm{Cov}[v|d_w]$ as the dimension of r may be too large to allow explicit representation of Σ_r. Second, as the dimension of ε is typically large, computation of $f(\varepsilon|d_w)$ would involve operations with huge matrices and is therefore infeasible. However, sampling from $f(\varepsilon|d_w)$ can be readily done, using the definition in equation (7), i.e. first sample r unconditionally from $f(r)$ and then subtract the required mean via a kriging procedure.

4 Simulation algorithm

In this section we discuss how the general Metropolis-Hastings setup described in Section 2 can be used to simulate from the posterior distribution $f(v, \varepsilon|d_w, d_p)$ given by equation (10). In doing this, three aspects need special attention:

Aspect 1: To evaluate the term $f(d_p|v, \varepsilon)$ is computer intensive as it includes a fluid flow simulation. This imply that each iteration of Metropolis-Hastings will be time consuming, which is in contrast to the situation in most other applications of MCMC. Thus, to obtain convergence within reasonable time, the proposal distributions should be designed to both propose large changes in the state vector and to avoid too small acceptance probabilities.

Aspect 2: To evaluate the term $f(\varepsilon|d_w)$ is in practice infeasible. Thus, the proposal distributions must be designed so that this term is not included in the acceptance probability. However, remember that sampling of an ε according to $f(\varepsilon|d_w)$ is easily done.

Aspect 3: For given observed production, d_p, and production strategy, p, the existence of several very different vectors r that "solves" equation (5) seems quite likely. Thus, we should expect the posterior distribution $f(v, \varepsilon|d_w, d_p)$ to be multi-modal.

In our description of Metropolis-Hastings in Section 2, the same $\mathrm{Q}^{\varphi}(x'|x)$ is used in all iterations. In practice, it is common to define several proposal distributions, each of which may be a combination of two densities as in equation (1), and in each iteration to chose one of them, deterministically of stochastically. To simulate from $f(v, \varepsilon|d_w, d_p)$ we define three types of updates, all according to the setup of Section 2. All three update types address aspects 1 and 2 above, whereas aspect 3 is only addressed in the third update type.

Update type 1: Here, only a change in the residual field is proposed, i.e. $v' = v$. Moreover, the two proposal distributions Q_0^{φ} and Q_1^{φ} are equal and no stochastic quantity φ is used. The potential new residual field ε' is generated as

$$\varepsilon' = c \cdot \varepsilon_0 + \sqrt{1 - c^2} \cdot \varepsilon, \tag{11}$$

where $c \in (0, 1]$ is a constant, ε_0 is a sample from $f(\varepsilon|d_w)$ and ε is the current residual field. Using $\mathrm{E}[\varepsilon|d_w] = 0$, straight forward calculations give that the corresponding acceptance probability reduces to

$$\alpha((v, \varepsilon')|(v, \varepsilon)) = \min\left\{1, \frac{f(d_p|v, \varepsilon')}{f(d_p|v, \varepsilon)}\right\}. \tag{12}$$

The constant c is a tuning parameter and should be chosen to get large changes in the residual field without letting the acceptance probabilities become too small.

Update type 2: This update type only propose a change in v, i.e. $\varepsilon' = \varepsilon$. Moreover, again the two proposal distributions Q_0^φ and Q_1^φ are equal and no stochastic quantity φ is in use. The potential new vector v' is generated from a multiT distribution with ν degrees of freedom, mean vector v and a covariance matrix $\Sigma(v,\varepsilon)$ that is adapted to the form of the posterior at the current location (v,ε). The precise definition of $\Sigma(v,\varepsilon)$ is somewhat technical and left for Appendix A. The acceptance probability becomes

$$\alpha((v',\varepsilon)|(v,\varepsilon)) = \min\left\{1, \frac{f(v'|d_w)f(d_p|v',\varepsilon)}{f(v|d_w)f(d_p|v,\varepsilon)} \frac{\mathrm{T}_\nu(v',\Sigma(v',\varepsilon))(v)}{\mathrm{T}_\nu(v,\Sigma(v,\varepsilon))(v')}\right\}, \tag{13}$$

where $\mathrm{T}_\nu(\mu,\Sigma)(v)$ denotes the density at location v of a multiT distribution with ν degrees of freedom and mean and covariance matrix μ and Σ, respectively. The ν here plays the role of a tuning parameter that should be chosen to get a high acceptance rate.

Update type 3: This update type is included to cope with the problem of multi-modality and for this we adopt ideas from Tjelmeland and Hegstad (1999). New values for both v and ε are proposed. Two different proposal distributions Q_0^φ and Q_1^φ are in use, where $\varphi \sim \mathrm{N}(0,\Sigma_\varphi)$ is an m-dimensional stochastic vector. A central component in the procedure is a local minimisation algorithm. In the simulation example presented in Section 5 we used a quasi-Newton algorithm with line search, but any optimisation algorithm used within the procedure specified below will define a correct MCMC algorithm.

For a background discussion on how this type of updates are designed to be able to cope with multi-modal distributions, see the reference given above. Here, we only briefly specify Q_0^φ and Q_1^φ by their corresponding simulation algorithm; 1) Draw φ from $\mathrm{N}(0,\Sigma_\varphi)$, 2) Draw a c uniformly distributed on the unit interval, 3) Draw ε' according to equation (11) where c is the value sampled in step 1), 4) If sampling according to Q_0^φ, set $v_0 = v + \varphi$, if sampling according to Q_1^φ, set $v_0 = v - \varphi$, i.e. set $v_0 = v + (1-2i)\varphi$ where $i = 0$ for Q_0^φ and $i = 1$ for Q_1^φ, 5) Run a local optimisation of $f(v,\varepsilon'|d_w,d_p)$ with respect to v, starting at $v = v_0$, let $\mu(v_0,\varepsilon')$ denote the local maximum found, and 5) Draw v' from a multiT distribution with ν degrees of freedom, expectation vector $\mu(v_0,\varepsilon')$ and covariance matrix $\Sigma(\mu(v_0,\varepsilon'),\varepsilon')$. For the definition of the function $\Sigma(\cdot,\cdot)$, again see Appendix A.

With this construction, as for update type 1, all factors involving $f(\varepsilon|d_w)$ cancels from the acceptance probability;

$$\alpha^\varphi((v',\varepsilon')|(v,\varepsilon)) = \tag{14}$$
$$\min\left\{1, \frac{f(v'|d_w)f(d_p|v',\varepsilon')}{f(v|d_w)f(d_p|v,\varepsilon)} \frac{\mathrm{T}_\nu(\mu(v'-(1-2i)\varphi,\varepsilon),\Sigma(\mu(v'-(1-2i)\varphi,\varepsilon),\varepsilon))(v)}{\mathrm{T}_\nu(\mu(v+(1-2i)\varphi,\varepsilon'),\Sigma(\mu(v+(1-2i)\varphi,\varepsilon'),\varepsilon'))(v')}\right\}.$$

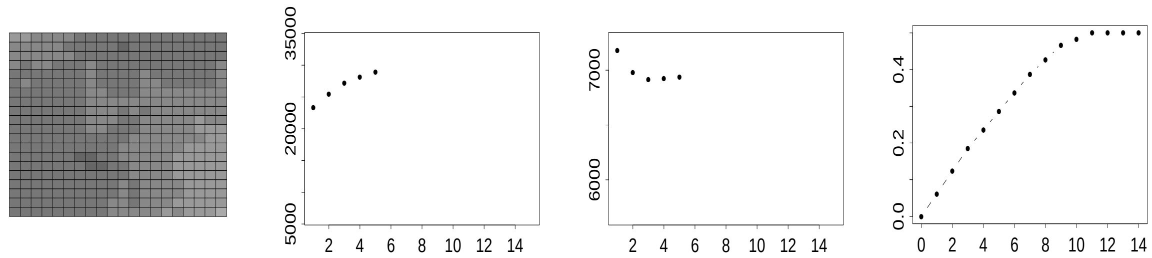
One should note that we, as proposed in Tjelmeland and Hegstad (1999), optimise with respect to the low dimensional vector v, thus avoiding a costly optimisation in the high dimensional r-space. This is the main motivation for the reparameterisation discussed in Section 3.1.

5 Simulation example

In this section we present a small simulation exercise. First, "true" reservoir characteristics and corresponding observations are generated. Thereafter, the Metropolis-Hastings algorithm presented in Section 4 is used to generate samples from the resulting posterior distribution. We start by presenting the stochastic model.

5.1 Model

We consider a system of $20 \times 20 \times 1$ grid blocks, see Figure 1(a). The size of each grid block is $800m \times 800m \times 50m$. The elements of $r \in \Re^{400}$ represent the logarithm of permeability values in the



(a) (b) (c) (d)

Figure 1: (a) The "true" permeability values, a log-linear scale is used and the values range from 125 to $2,500mD$. (b) Corresponding "observed" production rate in $(10,10)$ and (c) "observed" bottom hole pressure in $(1,1)$. The horizontal axes in (b) and (c) are in 100 days. (d) Variogram used to define the covariance matrix Σ_r. Units along the horizontal axes are in number of grid blocks.

grid blocks. The porosity values are assumed known and fixed to 0.11 for all grid blocks.

A hierarchical prior model as outlined in Section 3 is used for r. The vector of model parameters, θ, is six dimensional and the matrix F is chosen to give a parabolic mean function for $\mathrm{E}[r|\theta]$. More precisely, the conditional mean at grid block (i,j) is given as

$$\mathrm{E}[r(i,j)|\theta] = \theta_1 + \frac{\theta_2}{9.5}\cdot(i-10.5) + \frac{\theta_3}{9.5}\cdot(j-10.5) + \frac{\theta_4}{9.5^2}\cdot(i-10.5)^2 \qquad (15)$$

$$+\frac{\theta_5}{9.5^2}\cdot(j-10.5)^2 + \frac{\theta_6}{9.5^2}\cdot(i-10.5)\cdot(j-10.5) \quad \text{for} \quad i,j=1,2,\ldots,20.$$

Note that linear and second order terms are specified relative to the center of the grid, $(10.5,10.5)$, and the standardisation factors 9.5 and 9.5^2 allow simpler interpretations of the model parameters. A very vague prior distribution is specified for θ, with $\mu_\theta = (\ln(400),0,0,0,0,0)$ and $\Sigma_\theta = \mathrm{diag}(6^2,2^2,2^2,2^2,2^2,2^2)$. The covariance matrix Σ_r is defined via the isotropic variogram given in Figure 1(d).

Two wells are defined in the reservoir, one with gas injection and one with oil production, located at grid blocks $(1,1)$ and $(20,20)$, respectively. Two independent well observations are assumed, i.e. d_w is two dimensional and in matrix A of equation (4) all but two elements equal zero. In the upper row of A, the element corresponding to grid block $(1,1)$ is one, and in the lower row, the element corresponding to block $(20,20)$ is one. The corresponding covariance matrix is specified as $\Sigma_w = \mathrm{diag}(0.1^2, 0.1^2)$. The production strategy specified for the reservoir is constant gas injection rate at $(1,1)$ and oil production at constant bottom hole pressure at $(20,20)$. Thus, observed production history is bottom hole pressure in the injection well and rate of oil production in the producing well. These two variables are observed at 100 days interval from 100 days to 500 days, i.e. d_p is 10-dimensional. The error term for observed bottom hole pressure is assumed independent of the error term for oil rate. The variances of the error terms for bottom hole pressure and oil rate are set to 400^2 and 40^2, respectively. The error term for pressure at times t and t' are assumed correlated with correlation coefficient $0.8^{(t-t')/100}$ for $t,t' = 100,200,\ldots,500$. The same correlation is also assumed for oil rate observations.

For fluid flow simulation we use ATHOS, which is developed at Institut Francais du Petrole and Beicip-Franlab in France. This system is able to compute derivatives of production variables with respect to the elements of v, which is essential for an efficient optimisation in our update type 3.

5.2 Generation of "true" permeability field

For the simulation exercise, we first generate a "true" permeability field. For doing this, we fixed θ to $\theta = \theta_{true} = (5.5, 0.25, -0.25, 0.5, 0.5, -0.75)$ and then sampled an r_{true} from $f(r|\theta_{true})$, see Figure 1(a).

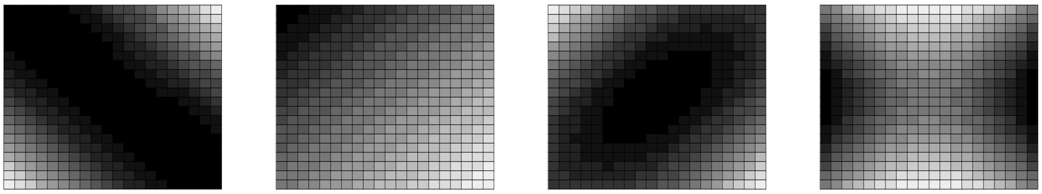


Figure 2: The last four rows in the matrix B defining the auxiliary vector $v = (v_1, v_2, \ldots, v_6)$ used in the reparameterisation of the posterior distribution. Darker colour indicates higher values.

The values of r_{true} range from 125 to $2,500 mD$. Thereafter, we sampled observations d_w and d_p from $f(d_w|r_{true})$ and $f(d_p|r_{true})$, respectively. The d_p is presented in Figures 1(b) and (c).

5.3 Choice of matrix B in the reparameterisation

To sample from the resulting posterior distribution, we first perform a reparameterisation. The procedure for this is given in Section 3.1, what remains to be specified is the choice of matrix B in equation (6). Ideally, as much of the variation of r in $f(r|d_w, d_p)$ should be included in v, i.e. one should let B consist of the first principal components of $\text{Cov}[r|d_w, d_p]$. However, the properties of the distribution $f(r|d_w, d_p)$ are unknown, that is why we want to sample from it in the first place. Thus, the best we can do is to use our quite limited understanding of the system to try to define a v that contains as much of the variability as possible.

It is well known that permeability values in and close to wells are crucially important for production fluid flow. Thus, we chose the first to components of v equal to the values of r in the two well positions. In addition, we include four more elements in v, chosen as the four first principal components of $\text{Cov}[r|d_w]$, see Figure 2.

5.4 Simulation results

We used the Metropolis-Hastings algorithm specified in Section 4 to sample from the resulting posterior distribution. In the simulation we alternate between the three different update types defined, doing update type 1 and 2 more frequently than type 3, as 1 and 2 are much faster. We used $c = 0.25$ in update type 1 and $\nu = 15$ in update types 2 and 3. A sequence of 21 updates of types 1, 2 and 3 in the following order, 112111121131121111211, we denote one iteration. The number of fluid flow iterations required for each iteration varied due to the optimisation for update type 3, but the average number was approximately 45.

To evaluate convergence of the MCMC algorithm, we adopt standard output analysis (Ripley, 1987, Ch. 6). We plot $v_i; i = 1, \ldots, 6$ and $-\ln f(v, \varepsilon|d_w, d_p)$ as function of iteration and use stability of the graphs as an indication for convergence, see Figure 3. Our run seems to have converged in less than 25 iterations. A very clear bimodality of the distribution for v_4 and v_6 can be seen from Figure 3. Moreover, all moves between the to modes are by update type 3. This demonstrates how important it is to include steps in the MCMC algorithm which is able to cope with multi-modal distributions. If a simulation with only update types 1 and 2 had been run, only one of the two modes would have been revealed. Another important indication for performance of the algorithm is the rate of acceptance of the proposed new states. For update types 1, 2 and 3, acceptance rates were 0.24, 0.32 and 0.12,

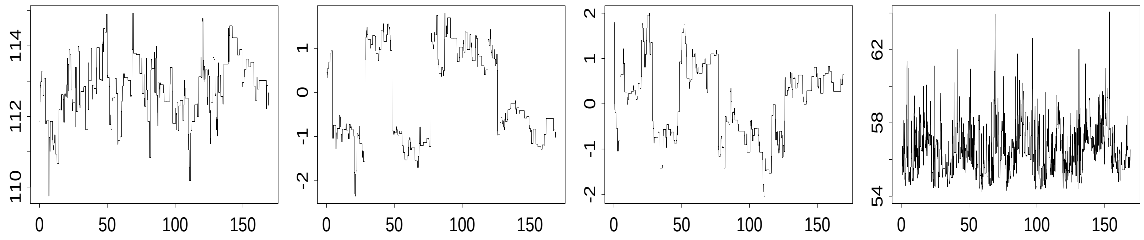


Figure 3: Plots of v_2, v_4, v_6 and $-\ln f(v, \varepsilon | d_w, d_p)$, respectively, as function of iteration number.

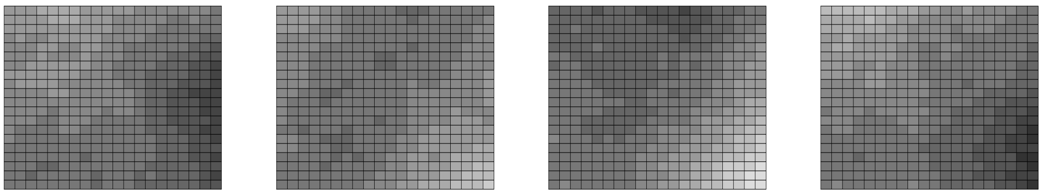


Figure 4: Permeability fields corresponding to r's sampled from $f(r|d_w, d_p)$ by the Metropolis-Hastings algorithm. The scale is as in Figure 1(a).

respectively, which are quite reasonable figures for a Metropolis-Hastings algorithm. Figure 4 shows the permeability fields after 50, 85, 120 and 155 iterations. It should be noted that no clear correlation between successive permeability fields is visible, giving an indication of reasonable fast mixing of the algorithm. For samples of r from $f(r|d_w, d_p)$, corresponding production curves are shown in Figure 5. The variability of the curves shows the uncertainty in the prediction. In evaluation of the curves, it should be remembered that the porosity is fixed in this simulation exercise, all uncertainty comes from unknown permeability fields.

6 Closing remarks

In this paper we have discussed how to design Metropolis-Hastings algorithms to sample from posterior distributions conditioned to production history. To obtain this, we have combined ideas previously discussed in the literature, namely the mode jumping proposals of Tjelmeland and Hegstad (1999) and

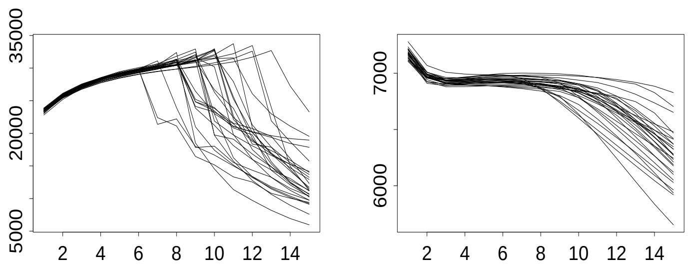


Figure 5: Oil production rate (left) and corresponding pressure (right), for reservoir characteristics sampled from the posterior distribution.

the optimisation in a lower dimensional space of Sahuquillo et al. (1992) and Oliver et al. (1997). The results seems promising as the convergence rate obtained is much faster than with traditional MCMC algorithms. Combined with fast fluid flow simulators this should make it possible to tackle even quite large problems within a consistent Bayesian framework. Still, however, the required number of fluid flow simulations is quite high and prohibit direct application of the methodology for large problems when standard fluid flow simulators are used. Thus, algorithms with even faster convergence (measured in computer time) should be developed. We believe that important components of this will include adaptive MCMC algorithms (Gilks et al., 1998; Holden, 1998) and the use of fast simulators to design good proposal distributions.

Acknowledgements

We are grateful to Institut Francais du Petrole and Beicip-Franlab, France who kindly let us use their code ATHOS for fluid flow simulation.

References

Chib, S. and Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm, *The American statistician* **49**: 327–335.

Floris, F. J. T. and Bos, C. F. M. (1998). Quantification of uncertainty reduction by conditioning to dynamic production data, *Proceedings of European Conference on Mathematics of oil recovery VI.* Peebles, Scotland, 1998.

Gilks, W. R., Roberts, G. O. and Sahu, S. K. (1998). Adaptive Markov chain Monte Carlo through regeneration, *Journal of the American Statistical Association* **93**: 1045–1054.

Hegstad, B. K. (1997). *Sampling from stochastic reservoir models constrained by production data*, PhD thesis, Norwegian University of Science and Technology.

Hegstad, B. K. and Omre, H. (1997). Uncertainty assessment in history matching and forecasting, *in* E. Y. Baafi and N. A. Schofield (eds), *Geostatistics Wollongong '96*, Vol. 1, Kluwer Academic Publishers, London, pp. 585–596. Procedings from the 5th International Geostatistical Congress, Wollongong, Australia, 1996.

Hjort, N. L. and Omre, H. (1994). Topics in spatial statistics (with discussion), *Scandinavian Journal of Statistics* **21**: 289–357.

Holden, L. (1998). Adaptive chains, *SAND/11/98*, Norwegian Computing Center, Oslo, Norway.

O'Hagan, A., Kennedy, M. C. and Oakly, J. E. (1998). Uncertainty analysis and other inference tools for complex computer codes (with discussion), *in* J. M. Bernardo, J. O. Berger, A. P. David and A. F. M. Smith (eds), *Bayesian statistics 6*, Oxford University Press, 1998, pp. 503–524.

Oliver, D. S., Cunha, L. B. and Reynolds, A. C. (1997). Markov chain Monte Carlo methods for conditioning a permeability field to pressure data, *Mathematical Geology* **29**: 61–91.

Omre, H. and Halvorsen, K. B. (1989). The Bayesian bridge between simple and universal kriging, *Mathematical Geology* **21**: 767–786.

Ripley, B. D. (1987). *Stochastic simulation*, Wiley, New York.

Sahuquillo, A., Capilla, J. E., Gómez-Hernández, J. J. and Andreu, J. (1992). Conditional simulation of transmissivity fields honoring piezometric data, *in* W. R. Blain and E. Cabrera (eds), *Hydraulic Engineering Software IV*, Vol. 2, Elsevier Applied Science, pp. 201–214. Procedings from Fourth International Conference on Hydraulic Engineering Software, HYDROSOFT/92, Valencia, Spain, 1992.

Tjelmeland, H. (1997). A note on the Bayesian approach to history matching of reservoir characteristics, *in* V. Pawlowsky-Glahn (ed.), *IAMG'97*, Vol. 2, CIMNE, Barcelona, Spain, pp. 772–777. Procedings from the third annual conference of the international association for mathematical geology, Barcelona, Spain, 1997.

Tjelmeland, H. and Hegstad, B. K. (1999). Mode jumping proposals in MCMC, *Statistics no. 1*, Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway. Available from http://www.math.ntnu.no/~haakont/.

A The covariance matrix $\Sigma(v, \varepsilon)$

The covariance matrix $\Sigma(v, \varepsilon)$ is used to propose potential new values for v in update types 2 and 3 of Section 4 and should fit the curvature of $f(v, \varepsilon | d_w, d_p)$ at (v, ε). To obtain this, we define the potential function $U(v, \varepsilon | d_w, d_p) = -\ln[f(v, \varepsilon | d_w, d_p)]$. Equation (10) then reads

$$U(v, \varepsilon | d_w, d_p) = \frac{1}{2}(v - BF\theta_0)^T \Sigma_{v|d_w}^{-1} (v - BF\theta_0) \tag{16}$$

$$+\frac{1}{2}(d_p - \omega(v, \varepsilon, p))^T \Sigma_p^{-1} (d_p - \omega(v, \varepsilon, p)) + \mathrm{const}(\varepsilon, d_w, d_p),$$

where $\Sigma_{v|d_w} = \mathrm{Cov}[v|d_w] = B(F\Sigma_\theta F^T + \Sigma_r)B^T$, the fluid flow simulator, ω, is expressed as a function of v and ε instead of as a function of r, and all terms which are not functions of v have been collected in $\mathrm{const}(\varepsilon, d_w, d_p)$. Let $\nabla_v^2 U$ denote the $m \times m$ matrix of second order partial derivatives of $U(v, \varepsilon | d_w, d_p)$ with respect v, the elements of which become

$$(\nabla_v^2 U)_{kl} = (\Sigma_{v|d_w}^{-1})_{kl} \tag{17}$$

$$+\sum_{i=1}^{n_p} \sum_{j=1}^{n_p} (\Sigma_p^{-1})_{ij} \left[\frac{\partial \omega_i(v, \varepsilon, p)}{\partial v_k} \cdot \frac{\partial \omega_j(v, \varepsilon, p)}{\partial v_l} - ((d_p)_i - \omega_i(v, \varepsilon, p)) \cdot \frac{\partial^2 \omega_j(v, \varepsilon, p)}{\partial v_k \partial v_l} \right],$$

where n_p is the dimension of d_p and $(d_p)_i$ and $\omega_i(v, \varepsilon, p)$ denote element i of the vector d_p and of the vector function $\omega(v, \varepsilon, p)$, respectively. One should then have chosen $\Sigma(v, \varepsilon)$ as the inverse of $\nabla_v^2 U$, to obtain identical curvature for $f(v, \varepsilon | d_w, d_p)$ and the proposal distribution. However, the partial second derivatives of $\omega(v, \varepsilon, p)$ are not available from the fluid flow simulator. As our simulation algorithm in Section 4 is valid for any choice of $\Sigma(v, \varepsilon)$, we therefore just ignore second order derivate terms in equation (17) and define $\Sigma(v, \varepsilon)$ as being the inverse of what remains. Thus, $\Sigma(v, \varepsilon)$ can then be expressed as

$$\Sigma(v, \varepsilon) = \left[\Sigma_{v|d_w}^{-1} + (\nabla_v \omega)^T \Sigma_p^{-1} (\nabla_v \omega) \right]^{-1}, \tag{18}$$

where $\nabla_v \omega$ is a $n_p \times m$ matrix with element (k, l) given by

$$(\nabla_v \omega)_{kl} = \frac{\partial \omega(v, \varepsilon, p)}{\partial v_l}. \tag{19}$$