Approximate forward-backward algorithm for a switching linear Gaussian model – with application to seismic inversion

Hugo Hammer and Håkon Tjelmeland
Department of Mathematical Sciences
Norwegian University of Science and Technology
Trondheim, Norway

Abstract

Motivated by the application of seismic inversion in the petroleum industry we consider a hidden Markov model with two hidden layers. The bottom layer is a Markov chain and given this the variables in the second hidden layer are assumed conditionally independent and Gaussian distributed. The observation process is assumed Gaussian with mean values that are linear functions of the second hidden layer. This model class, which we call switching linear Gaussian models, has clear similarities with what is known as switching linear dynamical systems and conditionally Gaussian state space models. The important difference is that we allow the observations to depend on both past and future values of the hidden Gaussian process. The forward-backward algorithms is not directly feasible for switching linear Gaussian models as the recursions then involve a mixture of Gaussian densities where the number of terms grows exponentially with the length of the Markov chain. We propose an approximate forward-backward algorithm by dropping the less important terms and thereby obtain a computationally feasible algorithm that generates samples from an approximation to the conditional distribution of the unobserved layers given the data. We also use this approximation as a proposal distribution in a Metropolis–Hastings setting and obtain high acceptance rates and good mixing properties. We demonstrate the effectiveness and quality of the approximation in simulation examples.

1 Introduction

Hidden Markov models combined with recursive algorithms have successfully been used in many areas, see the discussion and references in MacDonald and Zucchini (1997), Künsch (2000), Scott (2002) and Cappé et al. (2005). In a hidden Markov model the observations are assumed to be an incomplete and noisy function of an underlying unobserved process, where the latent unobserved process has a discrete state space and is assumed to be Markov. The goal is typically to restore the underlying Markov chain from the noisy observations and perhaps also to estimate unknown parameters in both the latent and observation processes. With reasonable assumptions for the observation process these goals are achieved via efficient recursive computations known as the forward-backward algorithms, corresponding to the famous Kalman filter equations when the latent Markov process is Gaussian.

Hidden Markov models with two hidden layers have been considered by several authors. Switching linear dynamical systems (Zoeter and Heskes, 2006; Bar-Shalom and Li, 1998) and switching state space models (Barber, 2006) are hidden Markov models with two latent layers. The bottom hidden layer is a discrete state space Markov chain and conditioned on this bottom layer the second hidden layer is a Gaussian Markov process. Finally, given the two hidden layers the observations are assumed independent and Gaussian. The mean vector and covariance matrix for the observed value at any time index are functions of the two unobserved states at the same time index. Larsen et al. (2006) consider a similar model, but allow the

observations to be functions of both past and future values of the hidden Gaussian process. The goal is again to restore the unobserved Markov chain. The forward-backward recursions can be formulated also for these models, but they are not computationally feasible as the forward recursion involves a mixture of Gaussian distributions where the number of terms grows exponentially with the length of the Markov chain. In the two first references given above, approximate forward recursions are defined by substituting the mixture of Gaussian distributions by a single Gaussian term. Larsen et al. (2006) define approximate recursions by approximating the marginal distribution for the unobserved continuous process by a product of Gaussian densities.

Motivated by the application of seismic inversion in the petroleum industry, we consider the same model as discussed in Larsen et al. (2006). Here the Markov chain represents lithology-fluid classes along a vertical trace through the underground, the intermediate Gaussian layer represents elastic parameters of the rock along the same trace, and the observations are the seismic data. The focus is again to restore the unobserved layers and particularly the Markov chain. Parameter estimation is clearly also of interest, but not considered here. To get a computationally feasible algorithm we must again consider approximate forward-backward algorithms. In the forward part of the forward-backward algorithm we propose to drop terms associated with small weights in the Gaussian mixture. Thus, our approximation is less dramatic than what is used before, but to a somewhat higher computational cost. The quality of the approximation depends on the number and importance of the terms that are dropped. Using the output of the resulting approximate forward-integration-backward-simulation algorithm as proposals in a Metropolis–Hastings setting (Hastings, 1970) we are able to correct for the induced approximation and the associated Metropolis–Hastings acceptance rate is a natural measure of the quality of the approximation.

The paper is organised as follows. Section 2 defines and introduces necessary notation for what we call the switching linear Gaussian model. In Section 3 we give a brief introduction to the seismic inversion application and explain how the switching linear Gaussian model constitutes the core part of this model. In Section 4 we first develop the exact forward recursions for the switching linear Gaussian model, then discuss the dropping Gaussian terms approximation and develop the backward simulation algorithm. In Section 5 we evaluate the approximate forward–backward algorithm in simulation examples. Finally, in Section 6 we provide conclusions.

2 The switching linear Gaussian model

In this paper we represent (multivariate) Gaussian distributions in its canonical form, as this simplify the forward-backward recursions. A Gaussian distribution with mean vector μ and covariance matrix Σ is in its canonical form parametrised by the precision matrix $Q = \Sigma^{-1}$ and the vector $q = Q\mu$, and we use $\mathrm{N}(q, Q)$ to denote this distribution. The corresponding density we denote by $\mathrm{N}(u|q, Q)$ and this reads

$$\mathrm{N}(u|q,Q) = \frac{\sqrt{|Q|}}{(2\pi)^{\frac{r}{2}}} \exp\left\{-\frac{1}{2} q^T Q^{-1} q\right\} \exp\left\{-\frac{1}{2}\left[u^T Q u - 2q^T u\right]\right\}. \tag{1}$$

Consider a three layer hidden Markov model $\{(x_i, y_i, z_i)\}_{i=1}^n$ as visualised in Figure 1, where $x_i \in \{1, \ldots, L\}$, $y_i = (y_{i1}, \ldots, y_{ir})^T \in \mathbb{R}^r$ and $z_i = (z_{i1}, \ldots, z_{is})^T \in \mathbb{R}^s$ for $i = 1, \ldots, n$. We require the number of possible values for the x_i's, L, to be small. In the seismic data

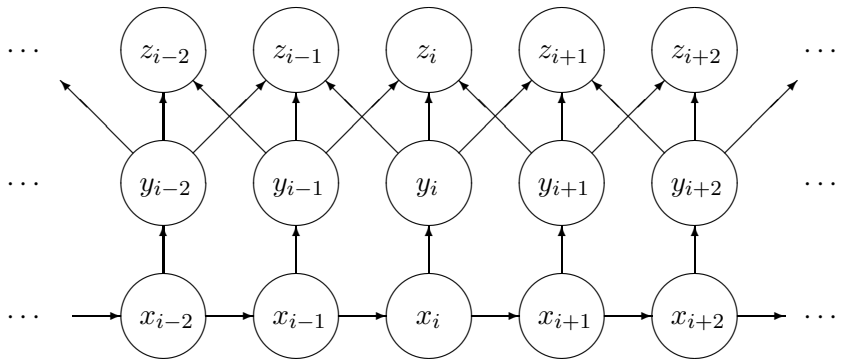


Figure 1: Directed acyclic graph representation of the hidden Markov model discussed in Sections 2 and 4.

example in Section 5 we have $L = 4$. We assume $x = (x_1, \ldots, x_n)$ to be a stationary, aperiodic and ergodic Markov chain with transition matrix

$$\mathrm{P} = [p(x_i|x_{i-1})]_{x_{i-1},x_i=1}^{L}. \tag{2}$$

Thus, the marginal distribution of x_1 equals the limiting distribution induced by P, which we denote by $p(x_1)$. Conditioned on x we assume the elements of $y = (y_1, \ldots, y_n)$ to be independent and Gaussian distributed, where the (conditional) mean vector and precision matrix of y_i are (known) functions of x_i, which we denote by $\mu(x_i)$ and $Q(x_i)$, respectively. Thus,

$$y_i|x \sim \mathrm{N}(q(x_i), Q(x_i)), \tag{3}$$

where $q(x_i) = Q(x_i)\mu(x_i)$. Given y we assume the elements of the third layer, $z = (z_1, \ldots, z_n)$, to be independent of x and independent of each other. The conditional distribution for z_i is Gaussian with mean vector $a_i^T y_{i-1} + b_i^T y_i + c_i^T y_{i+1}$ and precision matrix R_i, i.e.

$$z_i|y \sim \mathrm{N}\left(A_i^T y_{i-1} + B_i^T y_i + C_i^T y_{i+1}, R_i\right), \tag{4}$$

where $A_i = a_i R_i$, $B_i = b_i R_i$ and $C_i = c_i R_i$. Note that we allow the coefficient matrices a_i, b_i and c_i to vary with i, and in particular we of course require $a_1 = c_n = 0$.

3 The seismic inversion application

Seismic inversion is the discipline of predicting lithology-fluid characteristics in a reservoir from seismic data. Numerous introductory books to seismic terminology and inversion exist, see for example Sheriff and Geldart (1995).

Seismic data is created by an explosion which sends sound waves into the ground. Parts of the waves are reflected, returned upwards and observed by microphones (geo- or hydrophones). These observations are the basis for the seismic data. A forward model, describing what we observe for given lithology–fluid characteristics, is known from physics theory. In seismic inversion we are interested in the corresponding inverse problem, predicting the lithology–fluid characteristics given observed seismic data.

Seismic inversion is typically done based on either pre-stack or post-stack data. The pre-stack data are also referred to as amplitude versus offset (AVO) data or common midpoint gather (CMP). In a CMP gather, seismic data with different offsets (reflection angles) related to the same vertical profile are gathered together. Before the data is used for inversion, noise effects like moveout, multiples and the effect of geometrical spreading and absorption are removed. The data is also pre-stack migrated such that any dip related effects are removed. Post-stack data are generated from pre-stack data using the method of "stacking" (Sheriff and Geldart, 1995). Here we consider how to do seismic inversion from one vertical profile of pre-stack seismic data.

Our forward model is similar to the ones in Buland et al. (2003) and Larsen et al. (2006). When dealing with seismic data the depth is typically not referenced by distance from the surface, but by the time used by the sound wave from the surface to a location in the underground and back, called travel time and denoted by t. A problem not considered here is how to convert travel times to depths. Let the discrete variable $x(t)$ be the lithology–fluid class in the reservoir at travel time t. Assuming an isotropic and elastic medium, the material properties are uniquely defined by the continuous variables P-wave velocity, S-wave velocity and density. At time t we let

$$\begin{aligned} y(t,1) &= \ln(\alpha(t)) \\ y(t,2) &= \ln(\beta(t)) \\ y(t,3) &= \ln(\rho(t)) \end{aligned} \tag{5}$$

denote the logarithms of each of these three quantities, respectively. The forward relation from x to y is based on the so-called Gassman equations (Berryman, 1995), we return to this below. For each depth t and offset (reflection angle) θ a reflection coefficient, $r(t,\theta)$, results from y. For this we use what is known as a weak contrast approximation (Aki and Richards, 1980; Buland and Omre, 2003), which reads

$$r(t,\theta) = \gamma_\alpha(\theta)\frac{\partial}{\partial t}y(t,1) + \gamma_\beta(\theta)\frac{\partial}{\partial t}y(t,2) + \gamma_\rho(\theta)\frac{\partial}{\partial t}y(t,3), \tag{6}$$

where

$$\gamma_\alpha(\theta) = \frac{1}{2}(1+\tan^2(\theta)) \tag{7}$$

$$\gamma_\beta(\theta) = -4\overline{\beta/\alpha}^2\sin^2(\theta) \tag{8}$$

$$\gamma_\rho(\theta) = -4\left(\overline{\beta/\alpha}^2\sin^2(\theta)\right), \tag{9}$$

and one has assumed the ratio $\beta(t)/\alpha(t)$ to have an approximately constant value, $\overline{\beta/\alpha}$, in the reservoir. Seismic data depends on the reflection coefficients and is observed for each travel time t and offset θ. Still following Buland et al. (2003) and Larsen et al. (2006), we use a convolution model with additive noise for this,

$$d(t,\theta) = \int w(u,\theta)\cdot r(t-u,\theta)\mathrm{d}u + \varepsilon(t,\theta), \tag{10}$$

where $w(u,\theta)$ is a wavelet function and $\varepsilon(t,\theta)$ is Gaussian noise. Similar to Buland and Omre (2003) we assume that the main part of the Gaussian noise has a correlation structure

corresponding to the wavelet. The argument for this is that both the signal and noise parts are the results of sound waves going through the (same) underground.

For doing the inversion we consider a discrete version of the forward model discussed above. We use $i = 1, \ldots, n$ to denote n travel times along the vertical profile and let x_i and $y_i \in \mathbb{R}^3$ denote corresponding values for $x(t)$ and $y(t)$. We consider s offset values $\theta_1, \ldots, \theta_s$ and let $r_i = (r_{i1}, \ldots, r_{is})^T \in \mathbb{R}^s$ and $d_i = (d_{i1}, \ldots, d_{is})^T \in \mathbb{R}^s$ denote reflection coefficients and seismic data at travel time i, respectively. As prior for $x = (x_1, \ldots, x_n)^T$ we use a Markov chain, as specified in (2). The distribution for $y = (y_1^T, \ldots, y_n^T)^T$ given x is, as mention above, based on the Gassman equations and we assume Gaussian distributions as specified in (3). Using a central difference approximation for the partial derivatives in (6) we get an expression for $r = (r_1^T, \ldots, r_n^T)^T$ as a function of y,

$$r_i = \Gamma^T \cdot \frac{y_{i+1} - y_{i-1}}{2} \text{ for } i = 1, \ldots, n, \tag{11}$$

where

$$\Gamma = \begin{bmatrix} \gamma_\alpha(\theta_1) & \gamma_\alpha(\theta_2) & \cdots & \gamma_\alpha(\theta_s) \\ \gamma_\beta(\theta_1) & \gamma_\beta(\theta_2) & \cdots & \gamma_\beta(\theta_s) \\ \gamma_\rho(\theta_1) & \gamma_\rho(\theta_2) & \cdots & \gamma_\rho(\theta_s) \end{bmatrix}. \tag{12}$$

To avoid boundary problems for $i = 1$ and $i = n$, we use forward and backward difference approximations, respectively, for the derivatives there.

Finally we discretise (10) to get the distribution for $d = (d_1^T, \ldots, d_n^T)^T$ given r,

$$d_{ij} = \sum_{u=-k}^{k} w(u, \theta_j) \cdot r_{i-u,j} + \varepsilon_{ij}, \tag{13}$$

where the zero mean Gaussian noise ε_{ij} is given as

$$\varepsilon_{ij} = \sum_{u=-k}^{k} w(u, \theta_j) \varepsilon^1_{i-u,j} + \varepsilon^2_{i,j} \tag{14}$$

and $\varepsilon^1_{i,j}$ and $\varepsilon^2_{i,j}$ are independent Gaussian white noise and we assume the variance of ε^1_{ij} to be much larger than the variance in ε^2_{ij} so that most of the noise are on the same form as the wavelet. Clearly, we also assume the wavelet to be nonzero only in a finite interval so that the sum in (13) has a finite number of terms.

3.1 Simulating from the seismic model

We are now interested in simulating the variables x and y conditioned on the data d from the model above. To define an effective MCMC algorithm for this, we introduce $z = (z_1, \ldots, z_n)^T$, where $z_i = (z_{i1}, \ldots, z_{is})^T \in \mathbb{R}^s$ for $i = 1, \ldots, n$,

$$z_i = r_i + \varepsilon^1_i \tag{15}$$

and $\varepsilon^1_i = (\varepsilon^1_{i1}, \ldots, \varepsilon^1_{is})^T$. The distribution for x, y and z is then on the form specified in Section 2. In (4) we have $A_i = -\Gamma\, R_i/2$, $B_i = \mathbf{0}$ and $C_i = \Gamma\, R_i/2$ for $i = 2, \ldots, n-1$, and using forward and backward difference at the boundaries $B_1 = -\Gamma\, R_1$, $C_1 = \Gamma\, R_1$, $A_n = -\Gamma\, R_n$ and

$B_n = \Gamma R_n$. Finally we use $R_i = \sigma_1^{-2} I$ for $i = 1, \ldots, n$, where I denote the identity matrix. Combining (13), (14) and (15) we get the relation between z and d,

$$d_{ij}|z \sim \mathrm{N}\left(\sigma_2^{-2} \sum_{u=-k}^{k} w(u, \theta_j) \cdot z_{i-u,j}, \sigma_2^{-2} I\right). \tag{16}$$

To simulate effectively from the specified model we use a Metropolis–Hastings algorithm (Han and Green, 1992; Hastings, 1970) consisting of two steps in each iteration. In the first step we use a joint Gibbs update for y and z. The joint full conditional for y and z is Gaussian and thereby easy to sample from. The full conditional for x and y is a model of the type specified in Section 2. In the next section we discuss how to generate samples from an approximation to this distribution and we adopt this as proposal distribution.

Remark 1. A perhaps more direct Metropolis–Hastings algorithm would be a three block Gibbs algorithm, updating each of x, y and z separately. The full conditionals for y and z are Gaussian and to sample the full conditional for x the (standard and exact) forward-backward algorithm can be used. However, this algorithm has an extremely slow mixing because x and y are highly correlated. This is the reason we in stead propose the two step algorithm specified above.

Remark 2. A variant of the two block Metropolis–Hastings algorithm defined above is to propose new values only for parts of x and y in each iteration. In simulations we have found that one has to adopt this alternative if n is large, as otherwise either the approximate proposal distribution is not computational feasible or the resulting Metropolis–Hastings acceptance rate becomes very low.

4 Simulation algorithm

In this section we define the approximate forward-backward algorithm for the model defined in Section 2 and discuss how this can be used as a proposal distribution in a Metropolis–Hastings algorithm. We start by deriving the exact forward-integration recursion.

4.1 Forward integration

The conditional distribution of interest is

$$\pi(x, y|z) \propto \pi(x)\pi(y|x)\pi(z|y) = p(x_1) \cdot \prod_{i=2}^{n} p(x_i|x_{i-1}) \cdot \prod_{i=1}^{n} \mathrm{N}\left(y_i|q(x_i), Q(x_i)\right) \tag{17}$$

$$\cdot \prod_{i=1}^{n} \mathrm{N}\left(z_i|A_i^T y_{i-1} + B_i^T y_i + C_i^T y_{i+1}, R_i\right).$$

To make the forward integration easier we need to introduce more notation. Let $T_1(x_1, y_1, x_2, y_2, y_3)$ be a product of all the factors in $\pi(x, y|z)$ containing x_1 or y_1, let $T_2(x_2, y_2, x_3, y_3, y_4)$ be a product of all factors in $\pi(x, y|z)$ containing x_2 or y_2 and that is not already included in $T_1(x_1, y_1, x_2, y_2, y_3)$, and so on. Thus, by adopting the conventions $p(x_{n+1}|x_n) = 1$ and $B_{n+1} = C_{n+1} = 0$ we have

$$\pi(x, y|z) \propto \prod_{i=1}^{n} T_i(x_i, y_i, x_{i+1}, y_{i+1}, y_{i+2}), \tag{18}$$

with

$$T_1(x_1, y_1, x_2, y_2, y_3) = p(x_1)p(x_2|x_1)$$
$$\mathrm{N}(y_1|q(x_1), Q(x_1))\mathrm{N}(z_1|B_1 y_1 + C_1 y_2, R_1)\mathrm{N}(z_2|A_2^T y_1 + B_2^T y_2 + C_2^T y_3, R_2), \quad (19)$$
$$T_i(x_i, y_i, x_{i+1}, y_{i+1}, y_{i+2}) =$$
$$p(x_{i+1}|x_i)\mathrm{N}(y_i|q(x_i), Q(x_i))\mathrm{N}(z_{i+1}|A_{i+1}^T y_i + B_{i+1}^T y_{i+1} + C_{i+1}^T y_{i+2}, R_{i+1}) \quad (20)$$

for $i = 2, \ldots, n$. Recalling $C_n = 0$ and the notational conventions just made we see that none of the above functions really depends on x_{n+1}, y_{n+1} or y_{n+2}. We include these in the argument lists just to simplify the notation. Now define $U_i(x_i, y_i, x_{i+1}, y_{i+1}, y_{i+2})$ and $V_i(x_i, x_{i+1}, y_{i+1}, y_{i+2})$ for $i = 1, \ldots, n-1$ so that

$$\begin{aligned} \pi(x_i, \ldots, x_n, y_i, \ldots, y_n|z) &\propto U_i(x_i, y_i, x_{i+1}, y_{i+1}, y_{i+2}) \\ &\cdot \prod_{j=i+1}^{n} T_j(x_j, y_j, x_{j+1}, y_{j+1}, y_{j+2}), \quad (21) \\ \pi(x_i, x_{i+1}, \ldots, x_n, y_{i+1}, \ldots, y_n|z) &\propto V_i(x_i, x_{i+1}, y_{i+1}, y_{i+2}) \\ &\cdot \prod_{j=i+1}^{n} T_j(x_j, y_j, x_{j+1}, y_{j+1}, y_{j+2}). \quad (22) \end{aligned}$$

This gives $U_1(x_1, y_1, x_2, y_2, y_3) = T_1(x_1, y_1, x_2, y_2, y_3)$,

$$U_i(x_i, y_i, x_{i+1}, y_{i+1}, y_{i+2}) = T_i(x_i, y_i, x_{i+1}, y_{i+1}, y_{i+2}) \sum_{x_{i-1}=1}^{L} V_{i-1}(x_{i-1}, x_i, y_i, y_{i+1}) \quad (23)$$

for $i = 2, \ldots, n-1$, and

$$V_i(x_i, x_{i+1}, y_{i+1}, y_{i+2}) = \int U_i(x_i, y_i, x_{i+1}, y_{i+1}, y_{i+2}) \mathrm{d}y_i \quad (24)$$

for $i = 1, \ldots, n$. Note that none of the U_i or V_i functions are really functions of x_{n+1}, y_{n+1} or y_{n+2}, but again we include them as arguments to simplify the notation. In the following we use $\mathbf{0}$ and $\mathbf{I}$ to denote an $r \times r$ matrix with all elements equal to zero and the r-dimensional identity matrix, respectively. Moreover, let D_i and F_i be $3r \times 3r$ and $2r \times 2r$ identity matrices, respectively, for $i = 1, \ldots, n-2$, and define

$$D_{n-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix}, \; D_n = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} \text{ and } F_{n-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}. \quad (25)$$

The following Theorem presents the forward recursions for $i = 1, \ldots, n$.

Theorem 1. *Consider the hidden Markov model defined in Section 2 and the notation intro-*

duced above. We then have

$$U_i(x_i, y_i, x_{i+1}, y_{i+1}, y_{i+2}) \propto p(x_{i+1}|x_i) \cdot \sum_{j=1}^{N_i} f_{ij}(x_i) N\left(D_i \begin{bmatrix} y_i \\ y_{i+1} \\ y_{i+2} \end{bmatrix} \middle| g_{ij}(x_i), G_{ij}(x_i) \right) \quad \textit{for } i = 1, \ldots, n \tag{26}$$

$$V_i(x_i, x_{i+1}, y_{i+1}, y_{i+2}) \propto p(x_{i+1}|x_i) \cdot \sum_{j=1}^{N_i} f_{ij}(x_i) N\left(F_i \begin{bmatrix} y_{i+1} \\ y_{i+2} \end{bmatrix} \middle| k_{ij}(x_i), K_{ij}(x_i) \right) \quad \textit{for } i = 1, \ldots, n-1 \tag{27}$$

$$V_n(x_n, x_{n+1}, y_{n+1}, y_{n+2}) \propto \sum_{j=1}^{N_n} f_{nj}(x_n), \tag{28}$$

where $N_i = L^{i-1}$ *and* $f_{ij}(x_i)$, $g_{ij}(x_i)$, $G_{ij}(x_i)$, $k_{ij}(x_i)$, *and* $K_{ij}(x_i)$ *can be computed recursively. Initial values for* $g_{11}(x_1)$, $G_{11}(x_1)$ *and* $f_{11}(x_1)$ *are given by*

$$G_{11}(x_1) = \begin{bmatrix} Q(x_1)+B_1R_1^{-1}B_1^T+A_2R_2^{-1}A_2^T & B_1R_1^{-1}C_1^T+A_2R_2^{-1}B_2^T & A_2R_2^{-1}C_2^T \\ C_1R_1^{-1}B_1^T+B_2R_2^{-1}A_2^T & C_1R_1^{-1}C_1^T+B_2R_2^{-1}B_2^T & B_2R_2^{-1}C_2^T \\ C_2R_2^{-1}A_2^T & C_2R_2^{-1}B_2^T & C_2R_2^{-1}C_2^T \end{bmatrix} \tag{29}$$

$$g_{11}(x_1) = \begin{bmatrix} q(x_1)+B_1z_1+A_2z_2 \\ C_1z_1+B_2z_2 \\ C_2z_2 \end{bmatrix} \tag{30}$$

$$f_{11}(x_1) = p(x_1)\sqrt{|Q(x_1)|}\exp\left\{ -\frac{1}{2}\left(q(x_1)^T Q(x_1)^{-1} q(x_1) \right) \right\}. \tag{31}$$

For $i = 2, \ldots, n-2$, $j = 1, \ldots, N_i$ *and* $l = 1, \ldots, L$, *we have the following recursions*

$$G_{i,(j-1)L+l}(x_i) = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} Q(x_i) \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} K_{i-1,j}(l) \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} A_{i+1} \\ B_{i+1} \\ C_{i+1} \end{bmatrix} R_{i+1}^{-1} \begin{bmatrix} A_{i+1}^T & B_{i+1}^T & C_{i+1}^T \end{bmatrix}, \tag{32}$$

$$g_{i,(j-1)L+l}(x_i) = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} q(x_i) + \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} k_{i-1,j}(l) + \begin{bmatrix} A_{i+1} \\ B_{i+1} \\ C_{i+1} \end{bmatrix} z_{i+1}, \tag{33}$$

$$K_{i,(j-1)L+l}(x_i) = \left(\begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} G_{i,(j-1)L+l}(x_i)^{-1} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \right)^{-1}, \tag{34}$$

$$k_{i,(j-1)L+l}(x_i) = K_{i,(j-1)L+l}(x_i) \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} G_{i,(j-1)L+l}(x_i)^{-1} g_{i,(j-1)L+l}(x_i). \tag{35}$$

For the final two iterations, $i = n-1$ *and* n, *the recursions are slightly different. For* $j =$

$1, \ldots, N_{n-1}$ *and* $l = 1, \ldots, L$ *we have*

$$G_{n-1,(j-1)L+l}(x_{n-1}) = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{0} \end{bmatrix} Q(x_{n-1}) \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} + K_{n-2,j}(l) + \begin{bmatrix} A_n \\ B_n \end{bmatrix} R_n^{-1} \begin{bmatrix} A_n^T & B_n^T \end{bmatrix} \quad (36)$$

$$g_{n-1,(j-1)L+l}(x_{n-1}) = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{0} \end{bmatrix} q(x_{n-1}) + k_{n-2,j}(l) + \begin{bmatrix} A_n \\ B_n \end{bmatrix} z_n \quad (37)$$

$$K_{n-1,(j-1)L+l}(x_{n-1}) = \left(\begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} G_{n-1,(j-1)L+l}(x_{n-1}) \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix} \right)^{-1} \quad (38)$$

$$k_{n-1,(j-1)L+l} = K_{n-1,(j-1)L+l}(x_{n-1}) \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} G_{n-1,(j-1)L+l}(x_{n-1})^{-1} g_{n-1,(j-1)L+l}(x_{n-1}), \quad (39)$$

and for $j = 1, \ldots, N_n$ *and* $l = 1, \ldots, L$

$$G_{n,(j-1)L+l}(x_n) = Q(x_n) + K_{n-1,j}(l) \quad (40)$$

$$g_{n,(j-1)L+l}(x_n) = q(x_n) + k_{n-1,j}(l). \quad (41)$$

Finally, the recursion for $f_{ij}(x)$ *is, for* $i = 2, \ldots, n$, $j = 1, \ldots, N_{n-1}$ *and* $l = 1, \ldots, L$

$$\begin{aligned} f_{i,(j-1)L+l}(x_i) = f_{i-1,j}(l) p(x_i|l) \sqrt{\frac{|Q(x_i) K_{i-1,j}(l)|}{|G_{i,(j-1)L+l}(x_i)|}} \exp\Big\{ -\frac{1}{2} \big[q(x_i)^T Q(x_i)^{-1} q(x_i) + \\ k_{i-1,j}(l)^T K_{i-1,j}(l) k_{i-1,j}(l) - g_{i,(j-1)L+l}(x_i)^T G_{i,(j-1)L+l}(x_i)^{-1} g_{i,(j-1)L+l}(x_i) \big] \Big\}. \end{aligned} \quad (42)$$

The theorem is proved by induction. Reordering terms in $T_1(x_1, y_1, x_2, y_2, y_3)$ straightforwardly gives (26) for $i = 1$ and initial values (29), (30) and (31). Integrating out y_i in (26) and using well known properties of the multivariate Gaussian distribution gives (27) and (28), and the recursion formulas (34), (35), (38) and (39). We represent Gaussian distributions in the canonical form and not by the mean vector and covariance matrix and this is the reason for the somewhat unfamiliar expressions in these four equations. Summing over x_i in (27) and renumbering terms give after straightforward but tedious calculations (26) and the recursions (32), (33), (36), (37), (40), (41) and (42).

The number of Gaussian terms in (26) and (27) grows exponentially with i and the recursive algorithm defined by Theorem 1 is thereby computationally feasible only for very small values of n. In the next section we discuss how to approximate $U_i(x_i, y_i, x_{i+1}, y_{i+1}, y_{i+2})$ and $V_i(x_i, x_{i+1}, y_{i+1}, y_{i+2})$ by ignoring less important terms in (26) and (27).

4.2 Approximate forward integration algorithm

In this section we propose an approximate and computationally feasible version of the recursions defined in Theorem 1. We first compute the (exact) representation of $U_1(x_1, y_1, x_2, y_2, y_3)$ and $V_1(x_1, x_2, y_2, y_3)$ as given in Theorem 1. When starting iteration $i - 1 \leq n - 2$ of the sequential algorithm we have available an approximation of $V_{i-1}(x_{i-1}, x_i, y_i, y_{i+1})$ on the form

$$\begin{aligned} \widetilde{V}_{i-1}(x_{i-1}, x_i, y_i, y_{i+1}) \propto p(x_i|x_{i-1}) \\ \cdot \sum_{j=1}^{\widetilde{N}_{i-1}(x_{i-1})} \widetilde{f}_{i-1,j}(x_{i-1}) \mathrm{N}\left(\begin{bmatrix} y_i \\ y_{i+1} \end{bmatrix} \middle| \widetilde{k}_{i-1,j}(x_{i-1}), \widetilde{K}_{i-1,j}(x_{i-1}) \right), \end{aligned} \quad (43)$$

where we use tilde to distinguish quantities in the approximation from the corresponding exact ones. Note that the approximate representation is on the same form as in (27), except that in the approximation the number of Gaussian terms may depend on the value of x_{i-1}. Of course, for $i = 2$ we use $\widetilde{V}_{i-1}(x_{i-1}, x_i, y_i, y_{i+1}) = V_{i-1}(x_{i-1}, x_i, y_i, y_{i+1})$. Starting with $\widetilde{V}_{i-1}(x_{i-1}, x_i, y_i, y_{i+1})$ we perform iteration $i-1$ in two steps. First we compute approximate representations of $U_i(x_i, y_i, x_{i+1}, y_{i+1}, y_{i+2})$ and $V_i(x_i, x_{i+1}, y_{i+1}, y_{i+2})$ using the same type of recursion formulas as in Theorem 1. More precisely, letting $U_i^\star(x_i, y_i, x_{i+1}, y_{i+1}, y_{i+2})$ and $V_i^\star(x_i, x_{i+1}, y_{i+1}, y_{i+2})$ denote the approximations, we have

$$U_i^\star(x_i, y_i, x_{i+1}, y_{i+1}, y_{i+2}) \propto p(x_{i+1}|x_i) \cdot \sum_{j=1}^{N_i^\star(x_i)} f_{ij}^\star(x_i) \mathrm{N}\left(\left[\begin{array}{c} y_i \\ y_{i+1} \\ y_{i+2} \end{array} \right] \middle| g_{ij}^\star(x_i) G_{ij}^\star(x_i) \right), \quad (44)$$

$$V_i^\star(x_i, x_{i+1}, y_{i+1}, y_{i+2}) \propto p(x_{i+1}|x_i) \cdot \sum_{j=1}^{N_i^\star(x_i)} f_{ij}^\star(x_i) \mathrm{N}\left(\left[\begin{array}{c} y_{i+1} \\ y_{i+2} \end{array} \right] \middle| k_{ij}^\star(x_i), K_{ij}^\star(x_i) \right), \quad (45)$$

where $N_i^\star(x_i) = L\widetilde{N}_{i-1}(x_i)$, and $G_{ij}^\star(x_i)$, $g_{ij}^\star(x_i)$, $f_{ij}^\star(x_i)$, $K_{ij}^\star(x_i)$ and $k_{ij}^\star(x_i)$ are given from expressions (32) to (35) and (42) after substituting $G_{i,(j-1)L+l}(x_i)$, $g_{i,(j-1)L+l}(x_i)$, $f_{i,(j-1)L+l}(x_i)$, $K_{i,(j-1)L+l}(x_i)$, $k_{i,(j-1)L+l}(x_i)$, $K_{i-1,j}(l)$, $k_{i-1,j}(l)$ and $f_{i-1,j}(l)$ by $G_{i,(j-1)L+l}^\star(x_i)$, $g_{i,(j-1)L+l}^\star(x_i)$, $f_{i,(j-1)L+l}^\star(x_i)$, $K_{i,(j-1)L+l}^\star(x_i)$, $k_{i,(j-1)L+l}^\star(x_i)$, $\widetilde{K}_{i-1,j}(l)$, $\widetilde{k}_{i-1,j}(l)$ and $\widetilde{f}_{i-1,j}(l)$, respectively. In the second step of the iteration we start from $V_i^\star(x_i, x_{i+1}, y_{i+1}, y_{i+2})$ and generate a second (and final) approximation for $V_i(x_i, x_{i+1}, y_{i+1}, y_{i+2})$, denoted $\widetilde{V}_i(x_i, x_{i+1}, y_{i+1}, y_{i+2})$, by dropping the less important Gaussian terms in the first approximation (45) and thereafter renumbering the remaining terms. We use the same approximation procedure also for the final two iterations, $i = n-1$ and n. The expressions are again slightly modified for this iterations.

What terms in (45) that are of less importance is not obvious as they are functions of y_{i+1} and y_{i+2}, which are still unknown when this decision has to be taken. Natural strategies for handling this are either to maximise over or to integrate out y_{i+1} and y_{i+2} before comparing the size of the different terms. To maximise over y_{i+1} and y_{i+2} is obtained by evaluating the Gaussian density in (45) at its mean value. Thus, for a threshold value ε this strategy gives that we should drop terms in (45) that have

$$\frac{f_{ij}^\star(x_i) \mathrm{N}\left(\mu^\star(x_i) | \, k_{ij}^\star(x_i), K_{ij}^\star(x_i) \right)}{\max_{k=1,\ldots,N_i^\star(x_i)} \left\{ f_{ik}^\star(x_i) \mathrm{N}\left(\mu^\star(x_i) | \, k_{ik}^\star(x_i), K_{ik}^\star(x_i) \right) \right\}} < \varepsilon, \quad (46)$$

where $\mu^\star(x_i) = \left(K_{ij}^\star \right)^{-1} k_{ij}^\star(x_i)$. With the second strategy, to integrate out y_{i+1} and y_{i+2}, only $f_{ij}^\star(x_i)$ remains to compare. Thus, again for a given threshold ε, we drop all terms that corresponds to a $f_{ij}^\star(x_i)$ that have

$$\frac{f_{ij}^\star(x_i)}{\max_{k=1,\ldots,N_i^\star(x_i)} \left\{ f_{ik}^\star(x_i) \right\}} < \varepsilon. \quad (47)$$

In the simulation examples in Section 5 we adopt the first strategy, but we do not expect the second strategy to behave much differently. One should note that we decide what terms to drop separately for each possible value of x_i, and as a result the number of remaining terms, $\widetilde{N}_i(x_i)$, may differ for the different values of x_i.

Clearly, alternative term dropping strategies may be defined. First, one may use the term dropping step for $U_i^\star(x_i, y_i, x_{i+1}, y_{i+1}, y_{i+2})$ instead of for $V_i^\star(x_i, x_{i+1}, y_{i+1}, y_{i+2})$, but we do not expect this to make much difference. Second, instead of choosing a specific threshold value ε, one may fix the number of terms we want to keep in $\widetilde{V}_i(x_i, x_{i+1}, y_{i+1}, y_{i+2})$ and drop the necessary number of small terms in $V_i^\star(x_i, x_{i+1}, y_{i+1}, y_{i+2})$. Thereby the memory requirements for running the algorithm will be known in advance, but the quality of the approximation may be more variable than with the strategy we have chosen.

4.3 Backward simulation

When the (exact or approximate) forward integration has been done and the necessary quantities stored in memory, backward simulation is straight forward. Here we give the necessary equations for the approximate, computational feasible algorithm.

Sequentially for $i = n, \ldots, 1$ we first simulate x_i from

$$\pi^\star(x_i | x_{i+1}, \ldots, x_n, y_{i+1}, \ldots, y_n, z) \propto V_i^\star(x_i, x_{i+1}, y_{i+1}, y_{i+2}) \tag{48}$$

and then draw y_i from

$$\pi^\star(y_i | x_i, \ldots, x_n, y_{i+1}, \ldots, y_n) \propto U_i^\star(x_i, y_i, x_{i+1}, y_{i+1}, y_{i+2}). \tag{49}$$

The first is a discrete distribution and the second a mixture of r-variate Gaussian densities, so both are simple to generate realizations from. The resulting sample is thereby simulated from an approximation to the conditional distribution $\pi(x, y|z)$,

$$\pi^\star(x, y|z) = \prod_{i=1}^{n} \left[\pi^\star(x_i | x_{i+1}, \ldots, x_n, y_{i+1}, \ldots, y_n, z) \pi^\star(y_i | x_i, \ldots, x_n, y_{i+1}, \ldots, y_n) \right]. \tag{50}$$

One should also note that it is straight forward to evaluate $\pi^\star(x, y|z)$ for the generated sample (x, y), but to do this correctly one must of course remember to include the normalizing constants in the two conditional distributions $\pi^\star(x_i | x_{i+1}, \ldots, x_n, y_{i+1}, \ldots, y_n, z)$ and $\pi^\star(y_i | x_i, \ldots, x_n, y_{i+1}, \ldots, y_n)$.

4.4 Simulation from the hidden Markov model

To correct for the error introduced by the approximation discussed above one may use $\pi^\star(x, y|z)$ as a proposal distribution in a independent proposal Metropolis–Hastings algorithm. The resulting acceptance rate is also a natural measure for the quality of the approximation.

5 Simulation examples

We study the approximate forward-backward algorithm introduced above in a number of simulation exercises. We consider five sets of parameter values, which we denote by P1 to P5. Parameter set P1, which is our base case, are chosen to reflect realistic values for the seismic inversion application. P2 and P3 are identical to P1 except that we decrease and increase, respectively, the noise variance. The parameter values in P4 is the same as in P1 except for the rock physics model, the variances in y given x are larger in P4 than in P1. In all these first simulation studies we use a seismic profile of length $n = 100$, which is sufficiently small

to allow us to use the approximate forward-backward algorithm for all of x and y. In a last simulation example we increase the resolution of the lattice and consider $n = 400$. For this we use parameters corresponding to P1, but where some parameter values are changed to compensate for the increased resolution. We denote the resulting parameter values by P5. With $n = 400$ we are no longer able to update the whole profile, but must adopt the block Metropolis–Hastings algorithm discussed in Remark 2 in Section 3.1 to obtain satisfactory convergence and mixing results.

In the following we first, in Section 5.1, define the precise parameter values used for P1 to P5. Thereafter, in Section 5.2, we quantify the noise level in the five models by defining and reporting values for the signal-to-noise-ratios. In Section 5.3 we present simulation results and use these to discuss the quality of the approximate algorithms and convergence and mixing properties of the corresponding Metropolis–Hastings algorithms. We consider both simulation of x and y given z and the seismic inversion problem of sampling x, y and z given the seismic data d. In Section 5.4 we present results for the seismic inversion problem for parameter sets P1 to P4. Finally, in Section 5.5 we give results for a small sensitivity study, using a different parameter set for inversion than what is used to generate the data.

5.1 Parameter values

Our base case parameter values, P1, are chosen to be realistic for the seismic inversion application and is based on the values used in Larsen et al. (2006). We consider a model with $L = 4$ possible classes for x_i, where $x_i = 1, 2$ and 3 represent gas-, oil- and brine (water) saturated sandstone, respectively, and $x_i = 4$ represents shale. Sandstone is porous and allows flow of gas, oil, and water, whereas the porosity in shale is negligible and therefore acts as a barrier to fluid flow. Our choice of transition matrix for x, P, is based on the values used in Larsen et al. (2006), but we consider a coarser seismic resolution than done there. Numbering the nodes from the bottom of the trace, we use

$$\mathrm{P} = \begin{bmatrix} 0.9441 & 0 & 0 & 0.0559 \\ 0.0431 & 0.9146 & 0 & 0.0424 \\ 0.0063 & 0.0230 & 0.9422 & 0.0284 \\ 0.0201 & 0.0202 & 0.1006 & 0.8591 \end{bmatrix}. \tag{51}$$

This is obtained by first cubing the P used in Larsen et al. (2006) and thereafter moving the resulting values at the three positions with zeros above to the diagonal. The three zero elements are important in the seismic application as these represent known physical properties. As water has a higher density than oil, which again has a higher density than gas, water can not be above gas or oil and oil can not be above gas, unless separated by a non-porous shale layer. The resulting marginal probabilities for x_i are $[0.24, 0.16, 0.38, 0.22]$.

As discussed in Section 3 we consider a situation where $y_i \in \mathbb{R}^3$, where the three elements represent the logarithms of P-wave and S-wave velocities and density, respectively. In Larsen et al. (2006) the distribution of $y_i|x_i$ is represented as empirical distributions given by a set of corresponding x_i and y_i values. We use the same set of (x_i, y_i) values to estimate mean vectors and covariance matrices for the four assumed Gaussian distributions. The resulting mean vectors are

$$\mu(1) = \begin{bmatrix} 8.052 \\ 7.492 \\ 7.688 \end{bmatrix}, \quad \mu(2) = \begin{bmatrix} 8.071 \\ 7.472 \\ 7.730 \end{bmatrix}, \quad \mu(3) = \begin{bmatrix} 8.121 \\ 7.467 \\ 7.746 \end{bmatrix}, \quad \mu(4) = \begin{bmatrix} 8.166 \\ 7.546 \\ 7.846 \end{bmatrix} \tag{52}$$

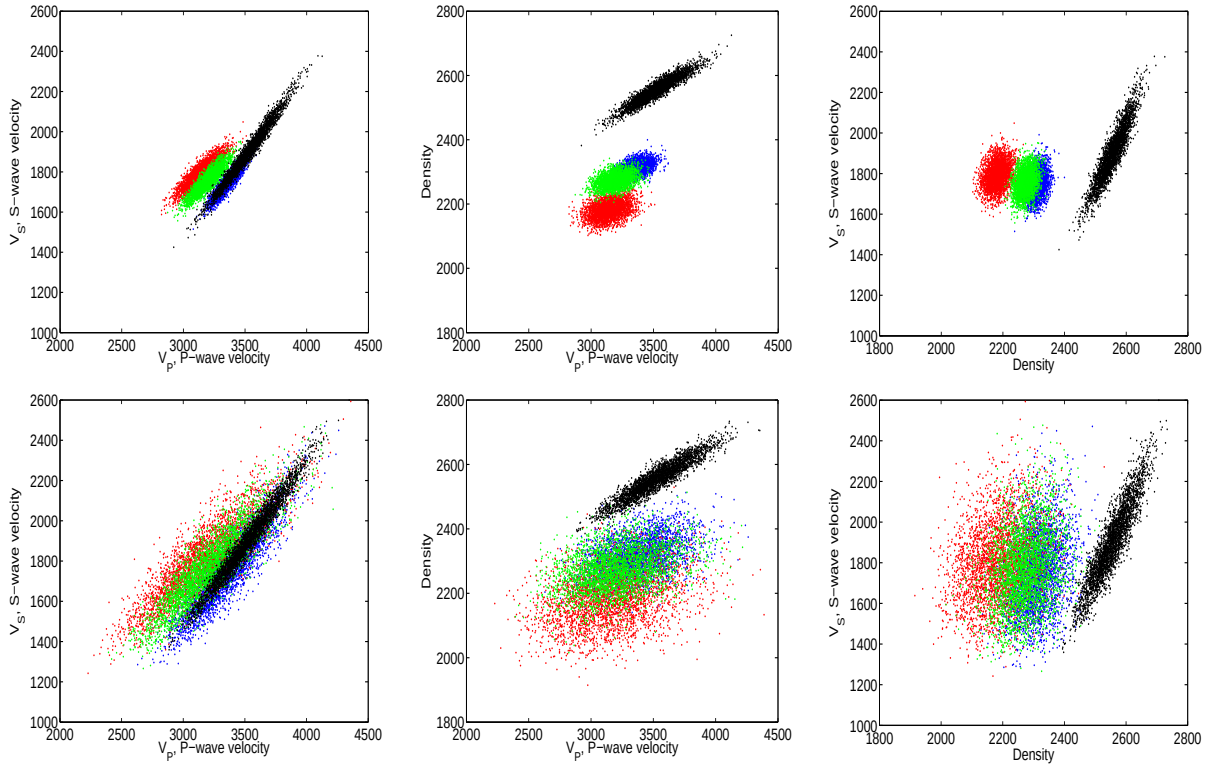


Figure 2: Scatter plots of P- and S-wave velocities and density of samples from the distribution adopted for $y_i|x_i$ in P1, P2, P3 and P5 (upper row) and P4 (lower row). Red, green, blue and black is used for gas-, oil- and brine-saturated sandstone and shale, respectively.

and the covariance matrices are given by the following figures, where the diagonal and off-diagonal entries give standard deviations and correlations, respectively, for each of the four possible values of x_i,

$$\begin{bmatrix} 0.031 & 0.876 & 0.322 \\ 0.876 & 0.033 & 0.271 \\ 0.322 & 0.271 & 0.012 \end{bmatrix}, \quad \begin{bmatrix} 0.027 & 0.891 & 0.384 \\ 0.891 & 0.032 & 0.295 \\ 0.384 & 0.295 & 0.009 \end{bmatrix} \tag{53}$$

$$\begin{bmatrix} 0.022 & 0.912 & 0.453 \\ 0.912 & 0.032 & 0.317 \\ 0.453 & 0.315 & 0.008 \end{bmatrix}, \quad \begin{bmatrix} 0.044 & 0.982 & 0.935 \\ 0.982 & 0.068 & 0.917 \\ 0.935 & 0.917 & 0.015 \end{bmatrix}. \tag{54}$$

The upper row in Figure 2 shows scatterplots of simulated P- and S-wave velocities and density according to the specified distributions. Here and in all the following figures we use the colours red, green, blue and black for gas-, oil- and brine-saturated sandstone and shale, respectively. We observe that shale is well separated from the other three classes and that gas-saturated sandstone is reasonably well separated from oil- and brine-saturated sandstone, whereas there is large overlap between the densities of oil- and brine-saturated sandstone.

To specify the model of d given y it remains to specify the number of and what angles that are used, the wavelet for each angle, and variances for the two error terms ε_{ij}^1 and ε_{ij}^2. Analogously to Larsen et al. (2006) we use the $s = 5$ angles $\theta = 0°, 10°, 20°, 30°$ and $40°$. We

also use the same angle independent Ricker wavelet, except that we scale it to compensate for coarser resolution. We use

$$w(u,\theta) = (1 - 2(\pi\,\phi\,u)^2)\exp\left(-(\pi\,\phi\,u)^2\right),\ u = -k,\ldots,k, \tag{55}$$

where $\phi = 0.11$ and $k = 10$. Also for the error variances we use corresponding values as in Larsen et al. (2006), except for an appropriate scaling to compensate for the difference in seismic resolution. We use $\mathrm{Var}(\varepsilon_{ij}^1) = 0.015^2$ and $\mathrm{Var}(\varepsilon_{ij}^2) = \mathrm{Var}(\varepsilon_{ij}^1)/10^4$.

The above completely defines the parameter values in P1. P2 to P4 are small modifications of P1. In P2 and P3 the variances for the noise terms ε_{ij}^1 and ε_{ij}^2 are scaled to give higher and lower signal-to-noise ratios, respectively. We use $\mathrm{Var}(\varepsilon_{ij}^1)$ equal to 0.0085^2 and 0.026^2 in P2 and P3, respectively, still with $\mathrm{Var}(\varepsilon_{ij}^2) = \mathrm{Var}(\varepsilon_{ij}^1)/10^4$. The parameter values in P4 are identical to P1 except for a scaling of the covariance matrices in the Gaussian densities for $y_i|x_i$ so that the four classes are less separated. For classes 1 to 3, gas-, oil- and brine-saturated sandstone, the covariance matrices used in P1 are multiplied with 10, whereas for class four, shale, the covariance matrix is multiplied with 2. These scaling values are also used in Larsen et al. (2006). Scatter plots of simulated P-wave and S-wave velocities and density are shown in the lower row of Figure 2.

Parameter set P5 corresponds to P1, but some of the parameter values are changed to compensate for an increased lattice resolution. Thus, P5 corresponds to the base case P1 on a finer lattice. The transition matrix P for P5 is the same as in Larsen et al. (2006)

$$\mathrm{P} = \begin{bmatrix} 0.980 & 0 & 0 & 0.020 \\ 0.015 & 0.970 & 0 & 0.015 \\ 0.002 & 0.008 & 0.980 & 0.010 \\ 0.007 & 0.007 & 0.036 & 0.950 \end{bmatrix} \tag{56}$$

with resulting marginal probabilities for x_i equal to $[0.23, 0.16, 0.39, 0.22]$. The distribution for y_i given x_i is the same as for P1. The wavelet has the same parametric form as for P1, but the parameter values are changed to compensate for the difference in resolution. We use (55) with $\phi = 0.03$ and $k = 30$, which is identical to the choice in Larsen et al. (2006). To compensate for the change in resolution we also need to scale the error variances to get approximately the same noise level as in P1. We set $\mathrm{Var}(\varepsilon_{ij}^1) = 0.015^2/3$ and again $\mathrm{Var}(\varepsilon_{ij}^2) = \mathrm{Var}(\varepsilon_{ij}^1)/10^4$.

5.2 Signal-to-noise-ratio

It is common to quantify the amount of noise in an inversion problem by a signal-to-noise (SN) ratio. In the setting we consider here it is not obvious how to split d into signal and noise parts. The x and y represent physical quantities, and so it may be reasonable to classify variability in d that has its origin in x or y as the signal part. However, our main interest is to regain x from d, and this makes it natural to include in the signal only the variability in d that originates from x. We define one signal-to-noise ratio for each of these two views, starting with the first.

Let $\widetilde{d}$ denote the data in the absence of observation noise so that (13) becomes

$$\widetilde{d}_{ij} = \sum_{u=-k}^{k} w(u,\theta_j)\cdot r_{i-u,j}. \tag{57}$$

Parameter set	P1	P2	P3	P4	P5
SN	2.82	8.35	0.92	6.15	3.10
SN*	1.33	2.47	0.64	0.45	1.10

Table 1: Signal-to-noise ratios for the various parameter sets.

It is natural to define the strength of the signal by $\mathrm{Var}(\widetilde{d}_{ij})$. However, this quantity is a function of angle j and, due to border effects, node number i. To get one number we average over i and j,

$$\mathrm{S} = \frac{1}{ns}\sum_{i=1}^{n}\sum_{j=1}^{s}\mathrm{Var}(\widetilde{d}_{ij}). \tag{58}$$

This can not be calculated analytically, but is easily estimated via simulation. The strength of the noise is correspondingly defined by

$$\mathrm{N} = \frac{1}{ns}\sum_{i=1}^{n}\sum_{j=1}^{s}\mathrm{Var}(\varepsilon_{ij}). \tag{59}$$

The resulting signal-to-noise ratio we denote by $\mathrm{SN} = \mathrm{S}/\mathrm{N}$.

Let $\mathrm{S}^\star$ and $\mathrm{N}^\star$ denote the strength of the signal and noise, respectively, when only the variation that results from x is considered as signal. $\mathrm{S}^\star$ is then again given by (58) if $\widetilde{d}_{ij}$ is replaced by

$$d^\star_{ij} = \sum_{u=-k}^{k} w(u,\theta_j)\cdot r^\star_{i-u,j}, \tag{60}$$

where $r^\star_{ij}$ is given from (11) by substituting y_{i-1} and y_{i+1} by their mean values,

$$r^\star_i = (r^\star_{i1},\ldots,r^\star_{is})^T = \Gamma^T\cdot\frac{\mu(x_{i+1})-\mu(x_{i-1})}{2}. \tag{61}$$

$\mathrm{N}^\star$ can then be expressed as

$$\mathrm{N}^\star = \frac{1}{ns}\sum_{i=1}^{n}\sum_{j=1}^{s}\mathrm{Var}(d_{ij}-d^\star_{ij}) \tag{62}$$

and the corresponding signal-to-noise ratio is $\mathrm{SN}^\star = \mathrm{S}^\star/\mathrm{N}^\star$. Again the quantities that are not analytically available can easily be found via simulation.

The resulting signal-to-noise ratios for parameter sets P1 to P5 are given in Table 1. We see that P4 has higher SN than P1, even if retrieving x from d is clearly harder in P4 than in P1. We find $\mathrm{SN}^\star$ to be a better measure for the difficulty of the inverse problem, but signal-to-noise ratio definitions similar to SN are most common in the seismic inversion literature, probably because the objective often is inversion to y only.

5.3 Evaluation of approximate forward-backward algorithm

In this section we evaluate the performance of the proposed approximate forward-backward algorithm in a simulation exercise for each of the parameter sets P1 to P5. The length of the

Parameter set	P1	P2	P3	P4	P5
ε	10^{-4}	10^{-4}	10^{-3}	10^{-3}	10^{-4}
acceptance rate	0.83	0.91	0.55	0.51	0.63

Table 2: Threshold values ε used in (46) and resulting Metropolis–Hastings acceptance rates for each of the parameter sets P1 to P5.

lattice is $n = 100$ for P1 to P4 and $n = 400$ for P5. In each case we first simulate x, y, z and d according to the model specified in Sections 2 and 3 and thereafter use the proposed algorithms to simulate from the resulting posterior distributions. We consider both simulating x and y given z by the algorithm discussed in Section 4.4 and the seismic inversion problem of simulating x, y and z given d by the algorithm defined in Section 3.1. We evaluate the quality of the approximate algorithm by the acceptance probability and the convergence and mixing properties of the Metropolis–Hastings algorithms.

The approximate forward-backward algorithm, and thereby the corresponding Metropolis–Hastings algorithms, contains one tuning parameter, ε in (46). The quality of the approximation and effectiveness of the algorithm clearly depends on this value. If ε is too small the number of terms that needs to be computed and stored becomes large and the algorithm becomes infeasible. If ε is too large the quality of the approximation deteriorates with very low Metropolis–Hastings acceptance probabilities as a result. For each of the parameter sets we found a reasonable value for ε by trial and error, the values are reported in Table 2. As expected ε must be higher when the signal-to-noise-ratios $\mathrm{SN}^\star$ is low, and for the same value of ε the acceptance rate decreases when $\mathrm{SN}^\star$ decreases. The acceptance rates for P3 and P4 can easily be increased by using a smaller ε, but then the number of Gaussian terms to store quickly becomes too large for the memory of typical computers. Our experience is that it is mainly the available memory that limits the possible value of ε and thereby the quality of the approximation, not the computation time. The acceptance rates reported in Table 2 is when simulating x, y and z given d, but simulating x and y given z produced essentially the same rates.

Figures 3 to 6 present the simulation results for parameter sets P1 to P4, respectively. To upper rows show the simulated "true" values. Note that we use the same realisation of x in all four cases to make comparison easier. The middle and lower rows contain posterior simulation results when conditioning on z and d, respectively. The two lower rows consist of three parts. To the left the "true" x is replotted for easier comparison, in the middle each state of the Metropolis–Hastings algorithm is plotted side by side, and the plots to the right show resulting marginal probabilities for each x_i. The Markov runs shown in Figures 3 to 6 are all initiated by setting all $x_i = 1$ and drawing y (and z when conditioning on d) values from the corresponding full conditional. In all the runs the initial state is left within very few iterations, so the burn-in phases of the Markov chains are not visible in the figures. We have also tried starting with all $x_i = 4$, and other initial values, without experiencing burn-in problems. The results clearly show that the approximate forward-backward algorithm gives a very good approximation to the distribution of interest and produces thereby very good mixing properties when used as a proposal distribution in a Metropolis–Hastings setting.

Figure 7 presents simulation results for parameter set P5. As mentioned above, this is for $n = 400$ and then the memory requirements for the approximate algorithm made it impossible to update all of x and y jointly. In stead we adopted the block Metropolis–Hastings algorithm

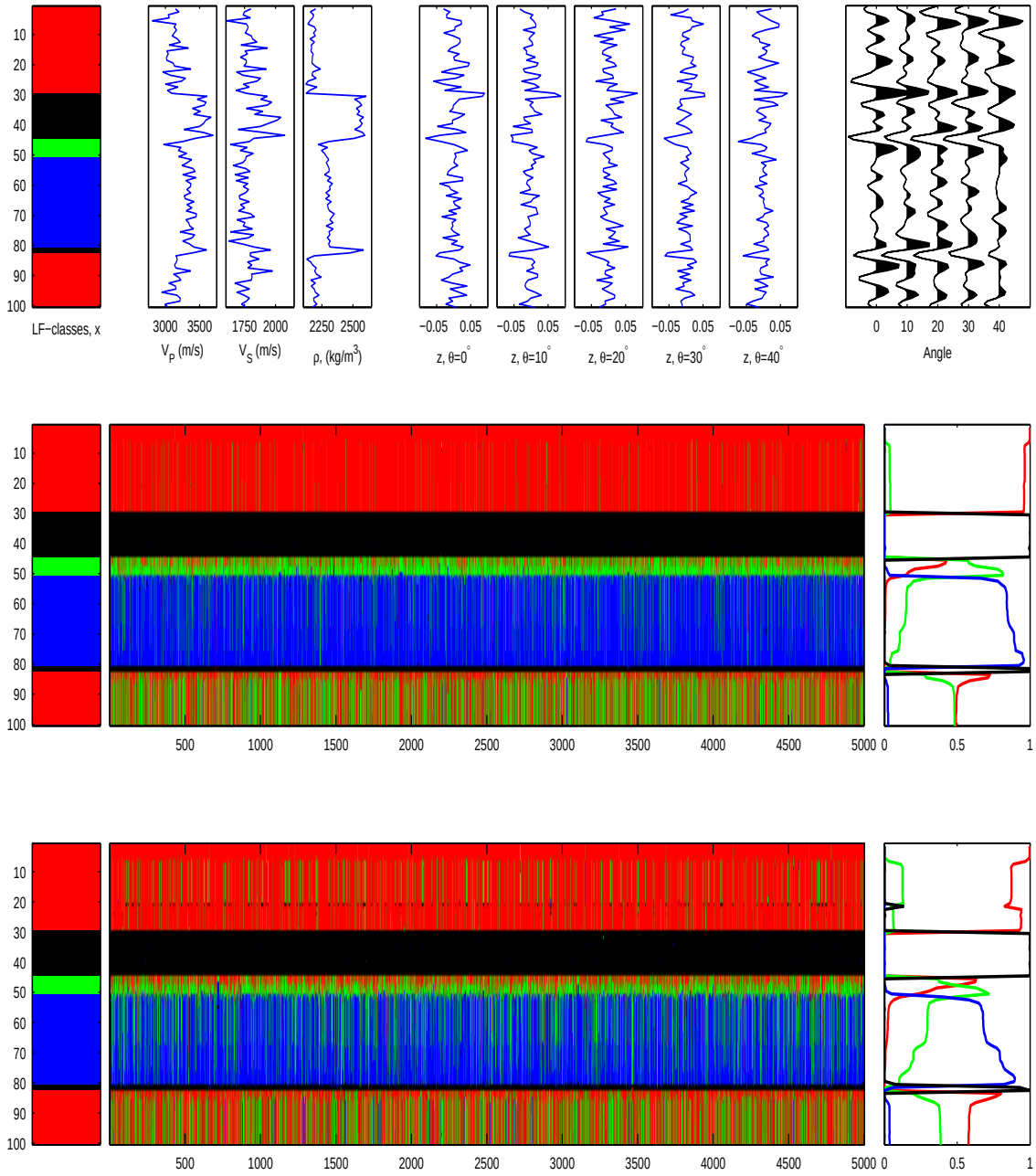


Figure 3: Simulation results for parameter set P1: The upper row gives, from left to right, the simulated "true" x, the simulated "true" elastic parameters, z and d. The middle and lower rows contain posterior simulation results when condition on z and d, respectively. From left to right, the two rows give the true x (for easier comparison), each state of the Metropolis–Hastings algorithm plotted side by side, and resulting marginal posterior probabilities for each x_i. Red, green, blue and black represent gas-, oil- and brine-saturated sandstone and shale, respectively.

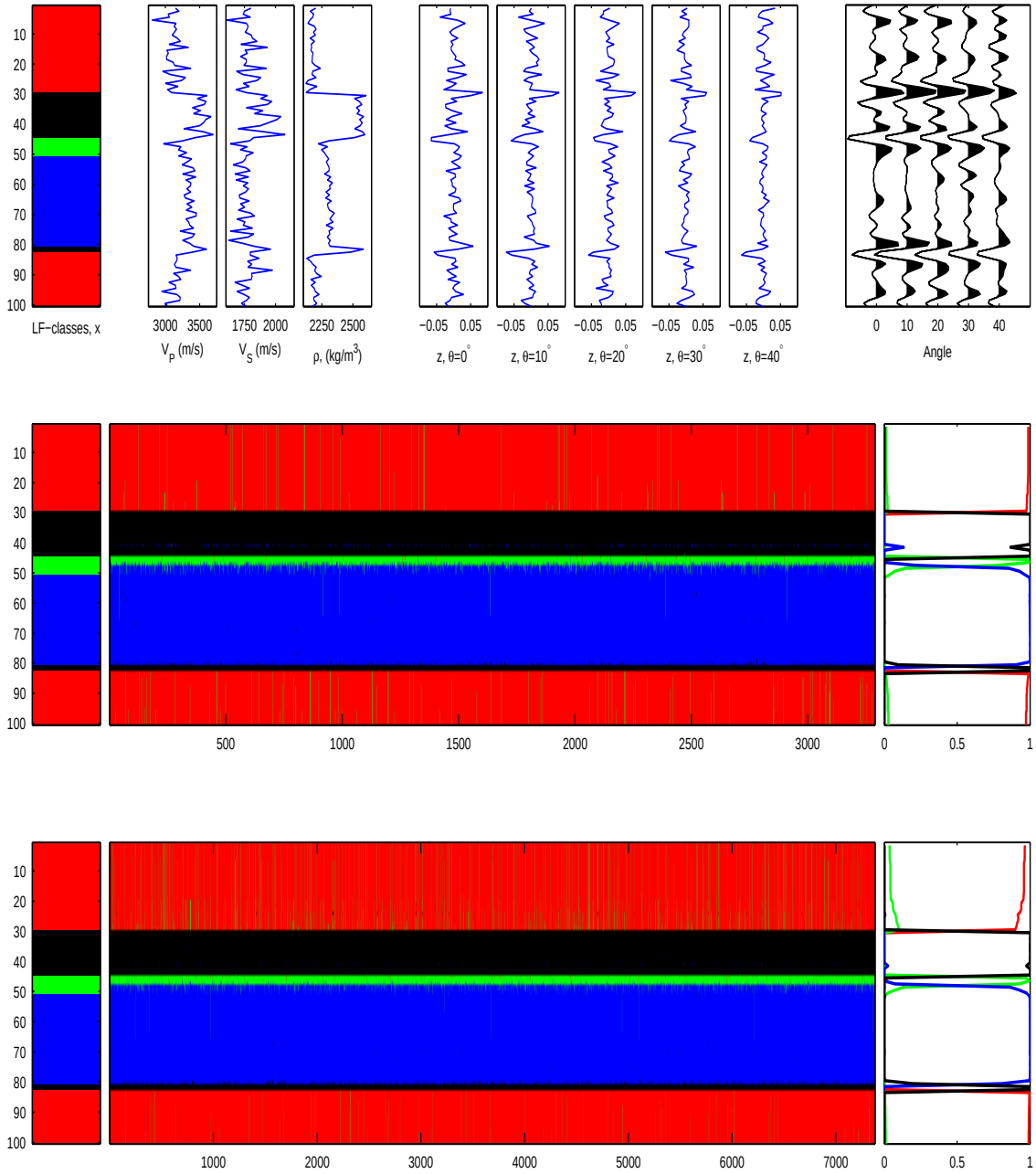


Figure 4: Simulation results for parameter set P2: See Figure 3 for an explanation of the different parts of the figure.

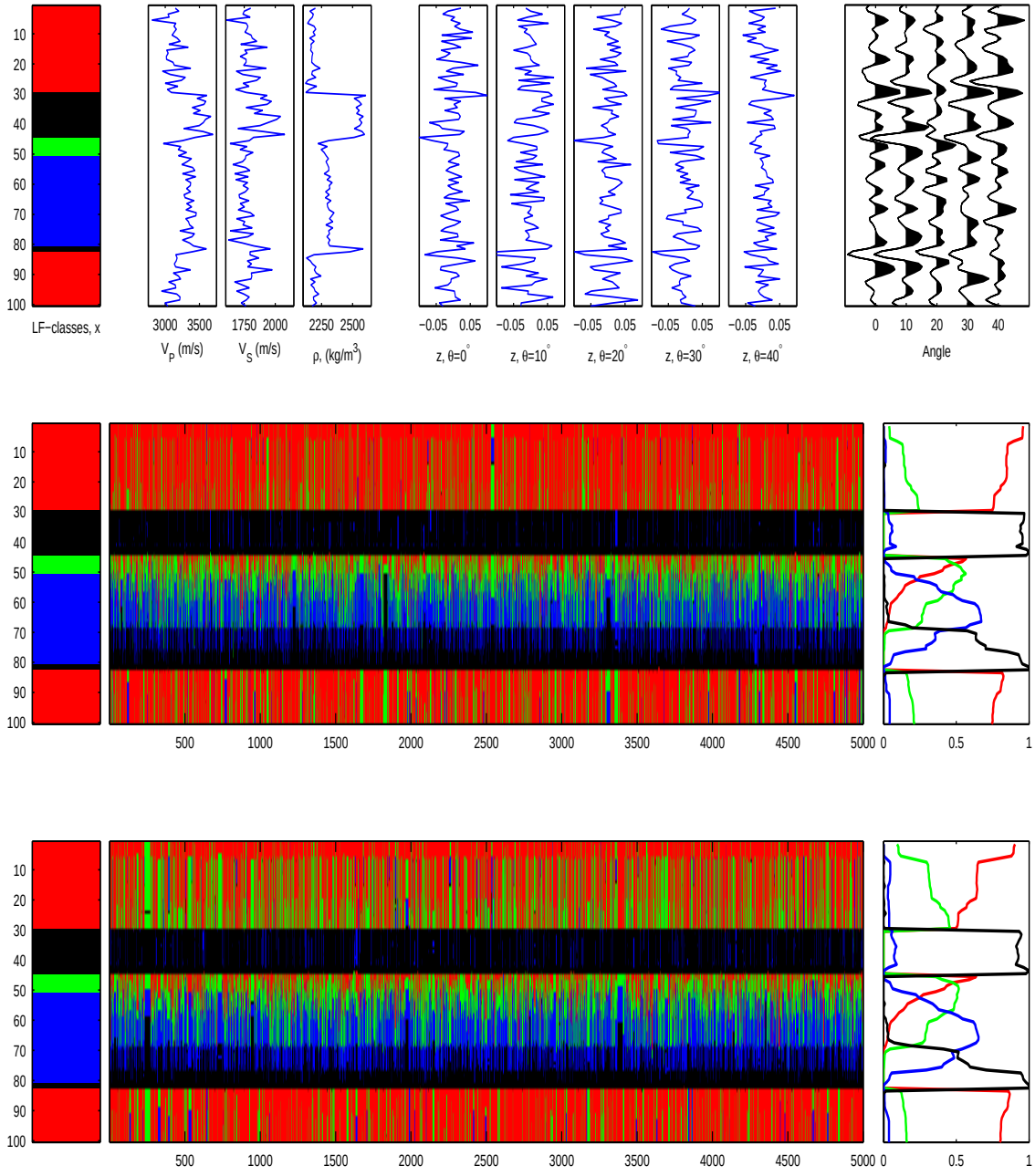


Figure 5: Simulation results for parameter set P3: See Figure 3 for an explanation of the different parts of the figure.

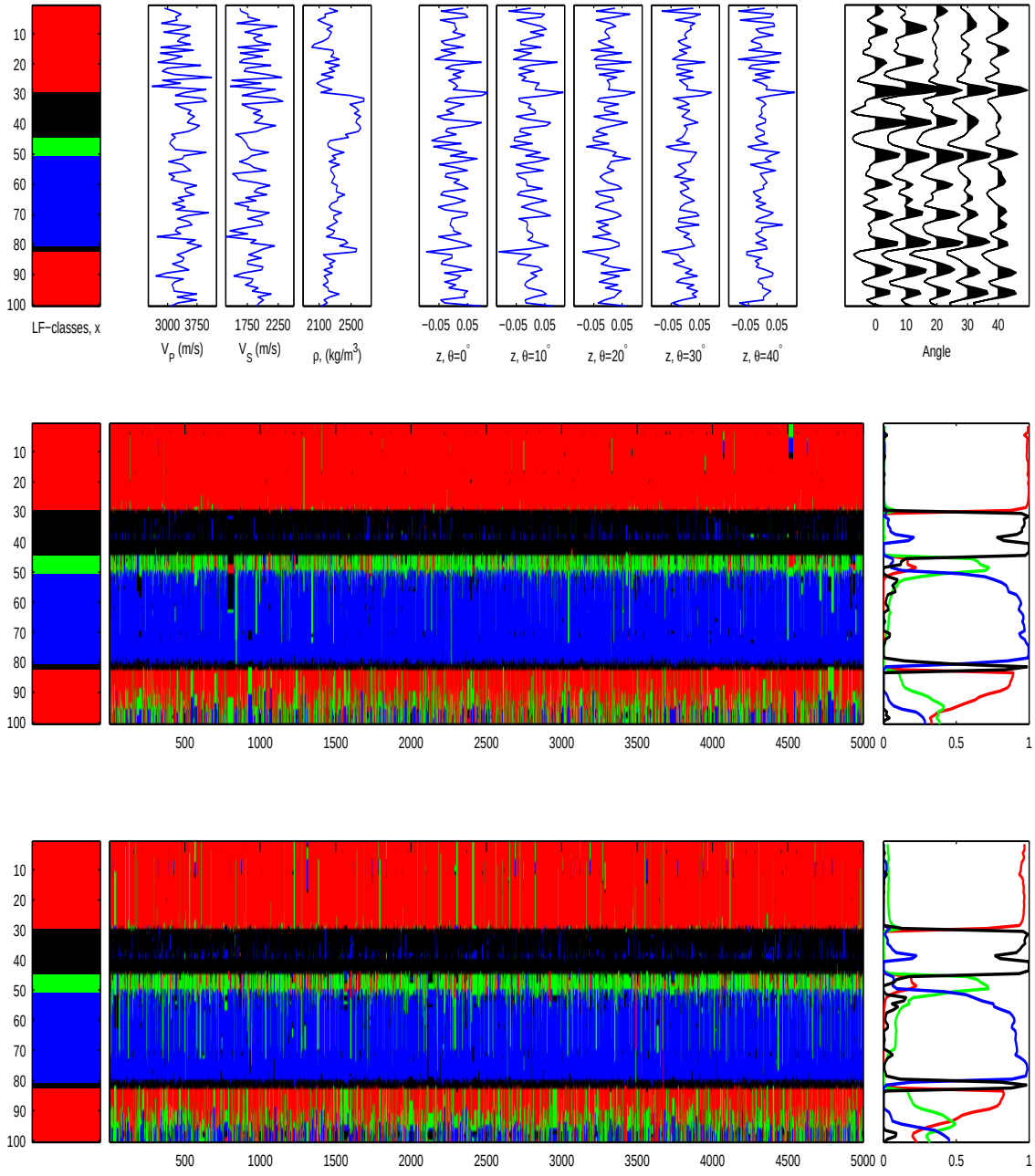


Figure 6: Simulation results for parameter set P4: See Figure 3 for an explanation of the different parts of the figure.

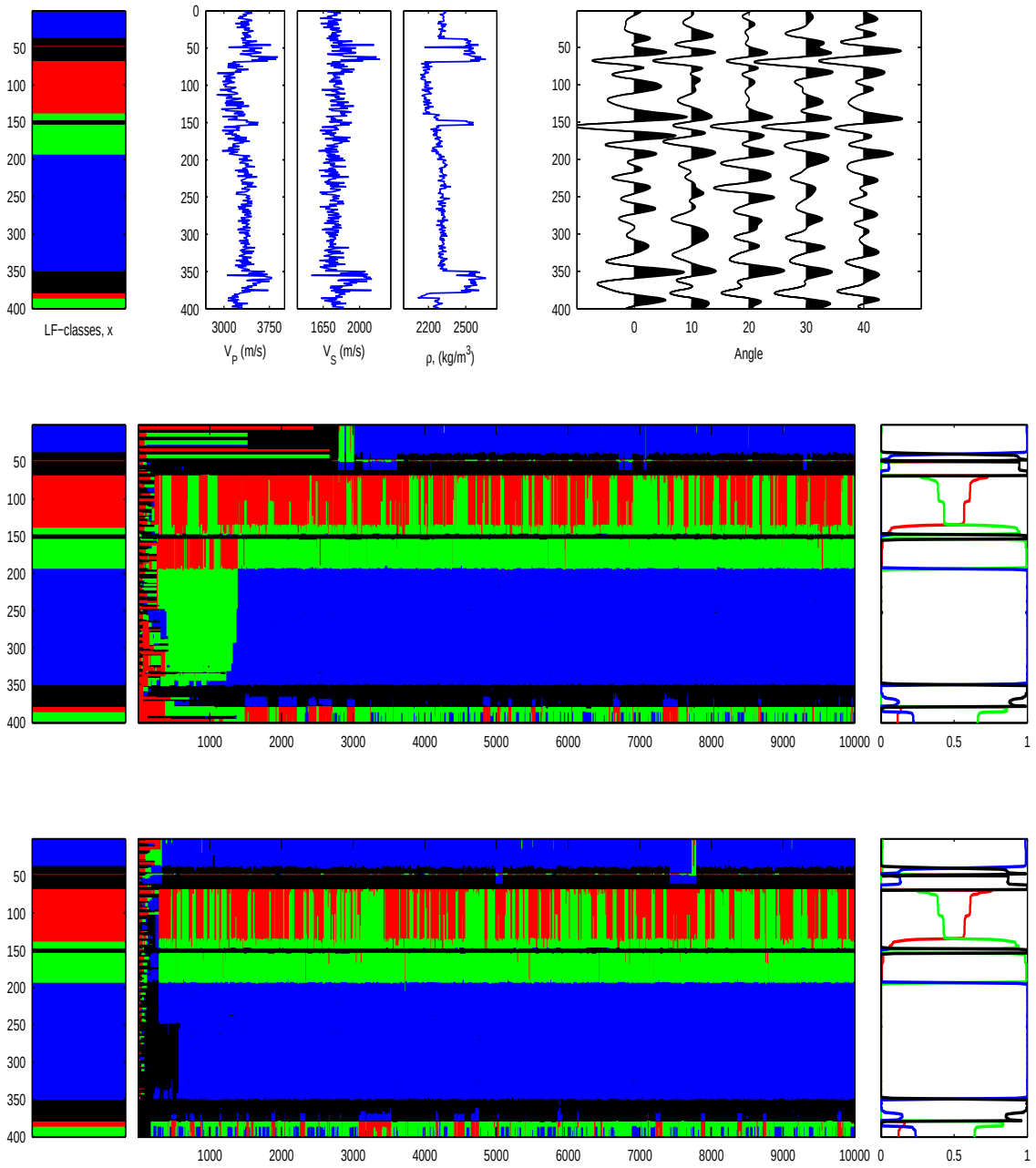


Figure 7: Simulation results for parameter set P5: The upper row gives, from left to right, the simulated "true" x, the simulated "true" elastic parameters and d. The two lower rows contain posterior simulation results when condition on d. The middle and lower rows show runs initiated with all $x_i = 1$ and $x_i = 4$, respectively. From left to right, the two rows give the true x (for easier comparison), each state of the Metropolis–Hastings algorithm plotted side by side, and resulting marginal posterior probabilities for each x_i. Red, green, blue and black represent gas-, oil- and brine-saturated sandstone and shale, respectively.

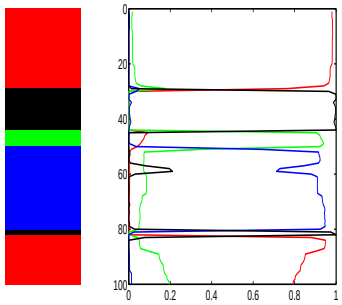
discussed in Remark 2. We use blocks of 100 nodes and in each iteration randomly choose one of the intervals $[1, 100], [51, 150], \ldots, [301, 400]$ to update. Again we run with the two initial states mentioned above. Figure 7 corresponds to Figures 3 to 6, except that the two lower rows both show simulation results when condition in d, one for each of the two initial states mentioned. We see that the convergence is much slower in this case, we can easily see the burn-in phase in the plots. The mixing of the chain is also somewhat less favourable in this case.

5.4 Inversion results

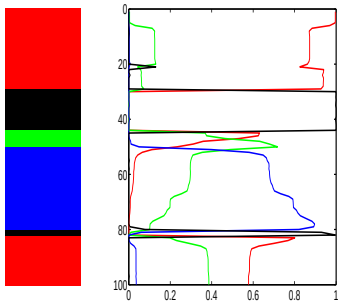
In this section we summarise the inversion results for the simulation exercises introduced above. We limit the attention to the runs where we condition on the simulated seismic data d. Comparing the marginal posterior probabilities in the bottom right corner in Figures 3 to 7 with the corresponding "true" x-values we see how the inversion results, as expected, depend on the signal-to-noise ratio $\text{SN}^\star$. For each of the parameter sets P1 to P4 we repeat the analysis for ten different realisations of y, z and d, still keeping the same realisation for x to allow easier comparison. Figure 8 shows the resulting marginal posterior probabilities for three cases with parameter set P1. Using the maximum marginal posterior probability classifier this figure also gives the resulting confusion matrices. As one should expect, the prediction ability varies somewhat between the different simulated data sets. Figure 9 shows corresponding results for each of P1 to P4 averaged over the ten runs. Again we see how the prediction ability deteriorates as $\text{SN}^\star$ becomes lower. Still, we find the results quite impressive given that the data is as shown in the upper right corners of Figures 3 to 6. It is close to impossible to identify the true x-values by just looking at the data. Studying the confusion matrices more in detail we see shale (black) are best identified, that gas-saturated sandstone (red) are mostly confused with oil-saturated sandstone (green), and oil-saturated sandstone (green) mostly confused with brine-saturated sandstone (blue). This is all as one should expect from Figure 2, the cloud of shale is well separated from the others and the clouds of gas- and oil-saturated sandstone are closer then the clouds of gas- and brine-saturated sandstone.

5.5 A small sensitivity study

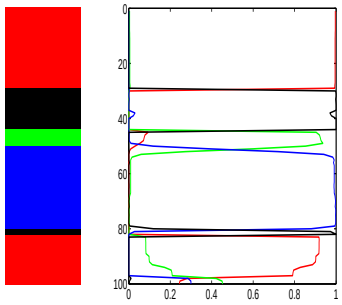
Clearly the inversion results above are better than what one can expect for real seismic data. In the simulation exercises we know, and use, the correct model and parameter values that generated the data. It is beyond the scope of this paper to do inversion of real seismic data, our focus is the algorithms for handling the hidden Markov chain with two hidden layers. However, as a first small test of the consequences of using a wrong model in the inversion we have done a simulation exercise where the data is generated according to our forward model with parameter set P1, whereas parameter set P4 is used in the inversion. Our interest is both how this influences the inversion results and what the consequences are for the convergence and mixing of the Metropolis–Hastings algorithm. After a little try and error, as discussed above, we settled on a value $\varepsilon = 2 \cdot 10^{-3}$ for the algorithm tuning parameter, which resulted in an acceptance rate of 0.49. Again we made two Markov chain runs, one initiated with all $x_i = 1$ and one with with all $x_i = 4$. Convergence seems to occur after very few iterations and the mixing is good, see one of the simulated chains in Figure 10. Corresponding to Figure 9, Figure 11 gives inversion results averaged over ten realisations of y, z and d. The shale (black) and brine-saturated sandstone are still well identified, whereas this is no longer true for gas-



	gas	oil	brine	shale
gas	0.9271	0.0675	0.0034	0.0020
oil	0.0670	0.9253	0.0072	0.0006
brine	0.0030	0.0775	0.8997	0.0198
shale	0	0.0001	0.0070	0.9929



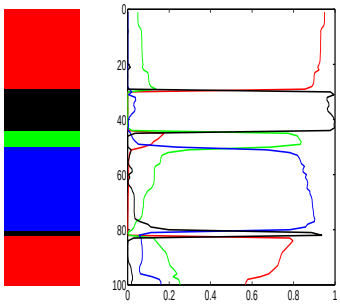
	gas	oil	brine	shale
gas	0.7985	0.1862	0.0121	0.0031
oil	0.4485	0.5394	0.0120	0.0002
brine	0.0200	0.2450	0.7320	0.0030
shale	0.0002	0.0007	0.0042	0.9950



	gas	oil	brine	shale
gas	0.9117	0.0690	0.0188	0.0004
oil	0.0666	0.9108	0.0225	0.0002
brine	0.0004	0.0364	0.9585	0.0047
shale	0	0.0000	0.0050	0.9950

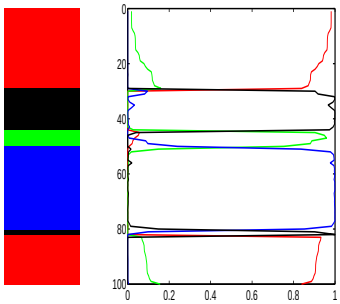
Figure 8: For different simulated data sets d in each row, the figure shows, from left to right, the "true" x, resulting marginal probabilities and confusion matrix using the maximum marginal probability classifier. Red, green, blue and black represent gas-, oil- and brine-saturated sandstone and shale, respectively.

P1:



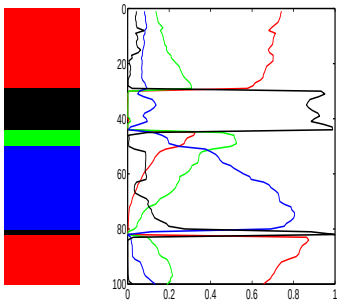
	gas	oil	brine	shale
gas	0.8413	0.1214	0.0328	0.0045
oil	0.1329	0.7969	0.0638	0.0063
brine	0.0037	0.1077	0.8568	0.0318
shale	0.0002	0.0020	0.0260	0.9718

P2:



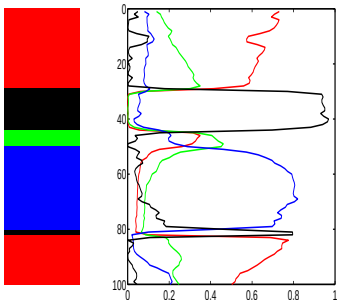
	gas	oil	brine	shale
gas	0.9267	0.0729	0.0003	0.0000
oil	0.0324	0.8889	0.0719	0.0069
brine	0	0.0064	0.9852	0.0083
shale	0	0.0017	0.0212	0.9772

P3:



	gas	oil	brine	shale
gas	0.7169	0.1936	0.0690	0.0205
oil	0.2879	0.5049	0.1873	0.0199
brine	0.0568	0.2049	0.6411	0.0973
shale	0.0010	0.0025	0.0782	0.9183

P4:



	gas	oil	brine	shale
gas	0.6354	0.2382	0.0982	0.0282
oil	0.3162	0.4182	0.2198	0.0458
brine	0.0575	0.1203	0.7278	0.0943
shale	0.0162	0.0234	0.0611	0.8992

Figure 9: Inversion results for each of the parameter sets P1 to P4: ‘ From left to right, the figures shows the “true” x-values, marginal posterior probabilities and confusion matrix, averaged over ten realisations of y, z and d. Red, green, blue and black represent gas-, oil- and brine-saturated sandstone and shale, respectively.

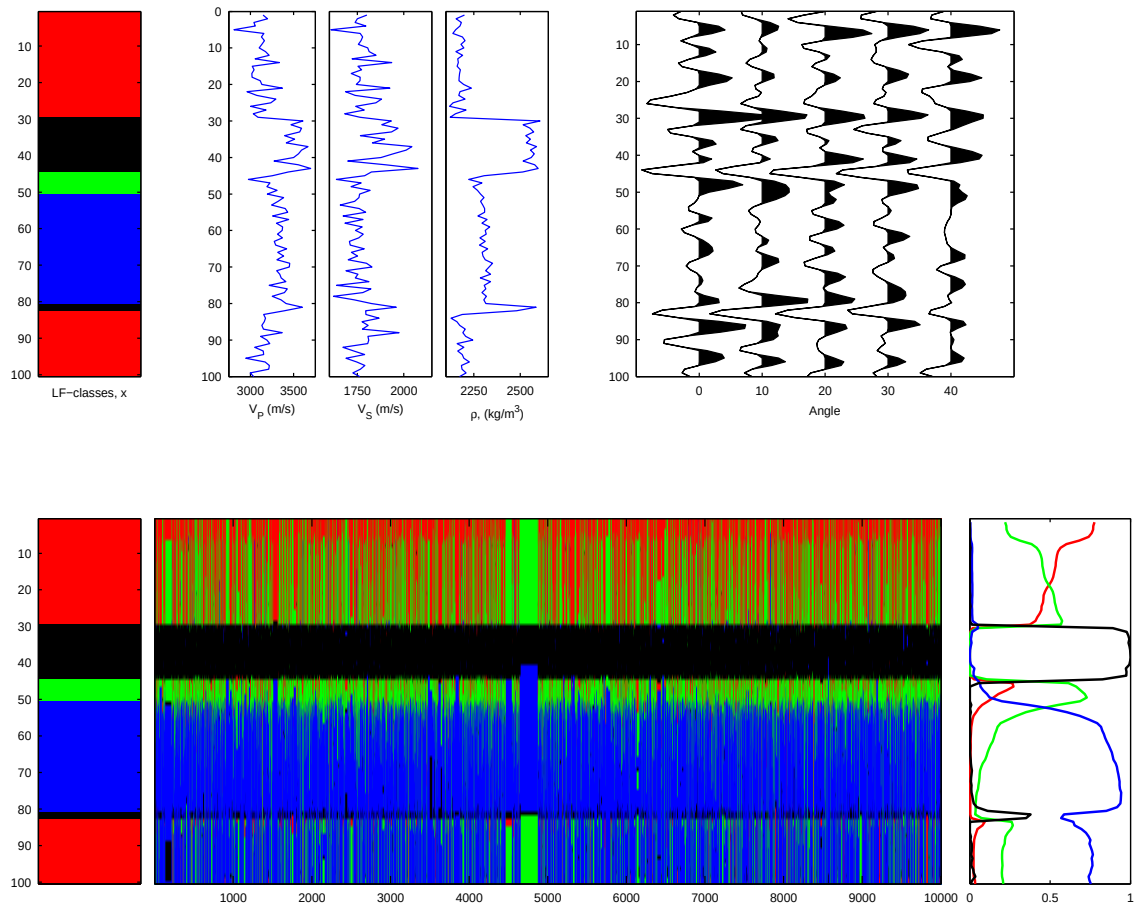
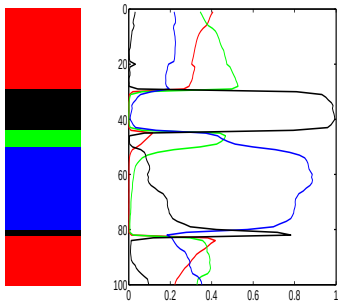


Figure 10: Sensitivity study: In the first row from left to right, the truth x, the true elastic parameters and the synthetic seismic data d. In the second row from left to right, the truth x, all the samples of x from a simulation x, y and z given d, with initial state $x = [0, \ldots, 0]^T$ and the resulting marginal probabilities from the simulation.



	gas	oil	brine	shale
gas	0.3216	0.4169	0.2378	0.0237
oil	0.0793	0.4304	0.4722	0.0181
brine	0.0017	0.0317	0.8120	0.1546
shale	0.0030	0.0109	0.0584	0.9277

Figure 11: Inversion results for the sensitivity study. We have the average result for all the ten inversions. Red, green, blue and black represent gas-, oil-, brine-saturated sandstone and shale, respectively.

and oil-saturated sandstone.

6 Closing remarks

We have revisited the seismic inversion problem as a hidden Markov model with both continuous and discrete hidden variables. We split the model into a switching linear Gaussian model and a Gaussian linear model. To handle the first part computationally we propose an approximate forward-backward algorithm. In a number of simulation exercises we demonstrate the effectiveness of the approximation and how this makes inversion of the seismic model computationally feasible. The approximate algorithm includes a tuning parameter ε. To choose a value for ε one must compromise between memory usage and computation time on one side and approximation accuracy on the other. We have found no automatic way to set the value of ε, but our experience is that it is relatively easy to find a reasonable value by try and error. What makes the choice of ε non-trivial is that it is used to decide what terms to drop in the forward recursions when information from the data is available from one side only. The correct importance of the various terms becomes available first after the following backward recursions.

We think the inversion problem in the switching linear Gaussian model for seismic inversion is harder than the problems previously considered for switching linear dynamical systems (Zoeter and Heskes, 2006; Bar-Shalom and Li, 1998) and switching state space models (Barber, 2006). Within an interval with the same value for x, the seismic data does not depend on the mean value of the continuous variable. By the difference taken in (11) the mean value of the continuous variable influences the data only when the value of x is changing. Otherwise the data only depends on x via the covariance matrix of the continuous variable. This induces larger posterior uncertainty in x and it becomes correspondingly more important to have an approximate forward-backward algorithm that realistically represents this uncertainty. Thus, we think the importance of including more Gaussian terms in the forward recursion is larger for the seismic model than for the cases previously considered in Zoeter and Heskes (2006), Bar-Shalom and Li (1998) and Barber (2006).

We define an approximate forward recursion by dropping Gaussian terms with small weights. In the references mentioned above an approximation is obtained by taking a single Gaussian density that (approximately) represents the whole Gaussian mixture. It is clearly also possible to define an approximate forward recursion by following an intermediate strategy, finding groups of terms in the Gaussian mixture that have similar mean and covariance and approximate these by a single Gaussian term. However, the computational cost of finding what terms to merge is quadratic in the number of terms, whereas the cost of finding what terms to drop is linear in the number of terms. Thus, unless the number of Gaussian terms necessary to obtain a sufficient good approximation is dramatically reduced when using the merging strategy, our simple dropping strategy is preferable. We have done a little experimentation with the merging strategy for our seismic inversion model, but without any success. However, we think the merging strategy may have a potential if the continuous variable y is univariate.

The focus of this paper is the computational problem associated with the hidden Markov seismic model. We have not considered inversion of real seismic data. To answer a real inversion problem one must also solve the associated parameter estimation problem. Preliminary experimentation with maximum likelihood estimation from simulated data indicates that it

is not possible to estimate all the model parameters from available seismic data. Either one must adopt a Bayesian view with informative priors, or information about (at least some of) the parameters must be obtained from other sources.

References

Aki, K. and Richards, P. G. (1980). *Quantitative seismology: Theory and methods*, W. H. Freeman and Company.

Bar-Shalom, Y. and Li, X.-R. (1998). *Estimation and Tracking: Principles, Techniques and Software*, Artech House, Norwood, MA.

Barber, D. (2006). Expectation correction for smoothed inference in switching linear dynamical systems, *Journal of machine learning research* **7**: 2515 – 2540.

Berryman, J. (1995). Mixture theories for rock properties, *in* T. J. Ahrens (ed.), *Rock Physics and Phase Relations, A Handbook of Physical Constants*, Am. Geophys. Union, Washington, D.C., pp. 205–228.

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**: 185–198.

Cappé, O., Moulines, E. and Rydén, T. (2005). *Inference in Hidden Markov Models*, Springer.

Han, X. L. and Green, P. J. (1992). Metropolis methods, Gaussian proposals, and antithetic variables *in* P. Barone, A. Frigessi and M. Piccioni (eds), *Stochastic Models, Statistical Methods and Algorithms in Image Analysis, number 74 in Lecture Notes in Statistics, Springer, Berlin,* pp. 142–164.

Hastings, W. (1970). Monte Carlo sampling using Markov chains and their applications, *Biometrica* **57**: 97–109.

Künsch, H. (2000). State space models and hidden Markov models, *in* O. Barndorff-Nielsen, D. Cox and C. Klüppelberg (eds), *Complex Stochastic Systems*, number 87 in *Monographs on Statsitics and Applied Probability*, Chapman & Hall/CRC.

Larsen, A. L., Ulvmoen, M., Omre, H. and Buland, A. (2006). Bayesian lithology/fluid prediction and simulation on the basis of a Markov-chain prior model, *Geophysics* **71 issue 5**: R69–R78.

MacDonald, I. and Zucchini, W. (1997). *Hidden Markov and Other Models for Discrete-Valued Time Series*, Chapman and Hall.

Scott, A. L. (2002). Bayesian methods for hidden Markov models: Recursive compution in the 21st century, *Journal of the American Statistical Association* **97**: 337–351.

Sheriff, R. E. and Geldart, L. P. (1995). *Exploration Seismology*, Cambridge university press.

Zoeter, O. and Heskes, T. (2006). Deterministic approximate inference techniques for conditionally Gaussian state space models, *Statistics and Computing* **16**: 279–292.