AGU PUBLICATIONS

JGR

Journal of Geophysical Research: Solid Earth

RESEARCH ARTICLE

10.1002/2014JB011010

Key Points:
- MCMC sampling methods for estimating posteriors often yield biased results
- A recursive algorithm to sample exactly from the posterior avoids bias issues
- Algorithm outperforms Gibbs sampling for synthetic application to seismic data

Correspondence to:
M. Walker,
matt.walker@ed.ac.uk

Citation:
Walker, M., and A. Curtis (2014), Spatial Bayesian inversion with localized likelihoods: An exact sampling alternative to MCMC, *J. Geophys. Res. Solid Earth*, *119*, 5741–5761, doi:10.1002/2014JB011010.

Received 31 JAN 2014
Accepted 23 MAY 2014
Accepted article online 3 JUN 2014
Published online 23 JUL 2014

Spatial Bayesian inversion with localized likelihoods: An exact sampling alternative to MCMC

Matthew Walker[1] and Andrew Curtis[1]

[1] School of GeoSciences, Grant Institute, University of Edinburgh, Edinburgh, UK

Abstract Geoscientists often use spatially discretized cellular models of the Earth where data in each grid cell provide independent information about the model parameters of interest at that location. In Bayesian inference this information is given as a set of likelihoods describing the (unnormalized) probability of the parameters, given only the data in each cell. Preexisting information about the model parameters' values and their spatial correlations may be described by a prior probability distribution. The prior, likelihoods, and Bayes' rule together specify a posterior probability distribution that describes the resultant state of information over all model parameters. However, due to the high dimensionality of typical models, the posterior is usually only known up to a multiplicative constant and only at specific, numerically evaluated points in the model space (i.e., it is not known analytically). Markov chain Monte Carlo (MCMC) methods are typically used to produce an ensemble of correlated samples from the posterior. These ensembles are slow to converge in distribution to the posterior; indeed, they may not converge in finite time, and detecting their state of convergence is often impossible in practice. Thus, estimates of the posterior obtained in this way may be biased. We derive a recursive algorithm which samples the posterior exactly, so as to avoid these convergence issues. Its computational cost scales with the size of the parameters' sample space, the prior's spatial range of dependency, and the shortest edge dimension of the grid. We develop an approximation to the algorithm such that it may be used on large 2-D (and potentially 3-D) model grids. We apply it to synthetic seismic attribute data and obtain results which compare favorably to the results of MCMC (Gibbs) sampling—which exhibits convergence problems.

1. Introduction

Inversion in geophysics often involves using data distributed over a region of space to infer values of model parameters distributed over that same space. It is then common to parameterize space into a (1-D, 2-D, or 3-D) grid of M cells and assign an index i to each cell such that $H = \{1, ..., i, ...M\}$ is the set of all indices in the grid. Furthermore, a model parameter variable, G_i, at each of these cells would be used to describe quantities of interest there. In certain circumstances it is then appropriate to collocate a data variable, d_i, in each cell. In the simplest case where each datum d_i is only dependent upon the local model parameter value G_i we could then invert the data at each cell individually for the model parameter at that cell (it should be noted that, in general, G_i and/or d_i could be vectors, but for notational convenience we will denote them as scalars herein).

However, it is nearly always recognized that uncertainty is present in the inversion at each grid cell. This uncertainty can be caused either by noise in the data, by intrinsic uncertainty in the relationship between the model parameter and the data, or by both [*Mukerji et al.*, 2001]. Fortunately, we often also have prior information about the values and spatial correlation of the model parameters which helps to constrain their inferred values. Bayesian inversion methods then permit us to integrate all of these sources of information into a solution which correctly characterizes the resulting uncertainty on the model parameters in all cells jointly. This approach is commonly taken, for example, when inverting seismic attribute data determined across a reservoir model [*Bosch et al.*, 2010], or when inverting remote sensing data [*Datcu et al.*, 1998] determined over an Earth surface model. Unfortunately, such Bayesian methods can be very computationally costly.

In Bayesian inversion, the uncertainty on the "cell-wise" relation between the model parameter and the data can be encapsulated in the form of the likelihood probability distribution, $p(\mathbf{d}|\mathbf{G})$, where $\mathbf{d} = [d_1, ..., d_i, ..., d_M]$ and $\mathbf{G} = [G_1, ..., G_i, ..., G_M]$. As suggested above, we are interested in cases where the

datum d_i is independent of all other variables (other than G_i) given G_i. We call this the *local likelihood property* henceforth, and write it mathematically as

$$p(d_i|G_i, \mathbf{G}_{\subseteq H\backslash i}, \mathbf{d}_{\subseteq H\backslash i}) = p(d_i|G_i) \tag{1}$$

where subscripts are used to reference subsets of cells in the grid and their associated variables as a vector. The notation $\subseteq H\backslash i$ signifies *any* set of indices in H (i.e., within the grid) which does not include cell i. G_i and d_i are now interpreted as random variables; thus, we write their sample spaces as $\mathcal{G}$ and $\mathcal{D}$, respectively. Accordingly, $\mathbf{G}$ and $\mathbf{d}$ are the vectors of all such random variables in the grid, and their sample spaces can be written as $\mathcal{G}^M$ and $\mathcal{D}^M$, respectively: the M exponent implies that the sample space for a single cell is taken to the power of the number of cells in the grid M (we assume that all G_i variables in the grid have the same sample space $\mathcal{G}$). Henceforth, we frame this work entirely in the context of a discrete G_i variable, with sample space $G_i \in \mathcal{G} = \{1, 2, ..., |\mathcal{G}|\}$. We discuss extensions to continuous G_i later.

As mentioned above, apart from likelihood information, we may also have preexisting information about the model parameters' values and their spatial interdependency. This is described by a prior probability distribution, $p(\mathbf{G})$, joint over the set of model parameters [*Eidsvik et al.*, 2002]. In spatial inverse problems it is usually desirable to specify the prior information (within $p(\mathbf{G})$) in terms of the relative relationships between the variables across cells and not the values of the variable at absolute positions. A natural way to pose such information probabilistically is to use distributions where the variable at a single cell is conditioned upon the variables in the surrounding cells, which is written

$$p(G_i|\mathbf{G}_{H\backslash i}) = \frac{p(\mathbf{G})}{p(\mathbf{G}_{H\backslash i})} \tag{2}$$

where the notation $H\backslash i$ signifies all indices in H (i.e., within the grid) except that of cell i. Such distributions are referred to as full conditionals [*Besag*, 1974]. It is often assumed that the dependency can be limited to a certain subset of the surrounding cells called the neighborhood of cell i, $\mathrm{Ne}(i)$. In this case the full conditional can be written as

$$p(G_i|\mathbf{G}_{H\backslash i}) = p(G_i|\mathbf{G}_{\mathrm{Ne}(i)}) = \frac{p(G_i, \mathbf{G}_{\mathrm{Ne}(i)})}{p(\mathbf{G}_{\mathrm{Ne}(i)})}. \tag{3}$$

It is important to note that the definition of the neighborhood as such means that a cell is not a member of its own neighborhood, $i \notin \mathrm{Ne}(i)$. A single, duplicate full conditional is then often used to characterize the prior as a whole, which is to say that $p(G_i|\mathbf{G}_{\mathrm{Ne}(i)})$ is invariant to i (except at the edge of the model grid, where simple modifications can be made to compensate for any absent neighbors specified by $\mathrm{Ne}(i)$). Henceforth, we refer to this property, i.e., that we can specify the prior using a full conditional as in equation (3), as the *local prior property*.

The aim of Bayesian inversion is to determine a so-called posterior probability distribution over $\mathbf{G}$, which combines the information in the likelihood and prior distributions. Mathematically, we combine our prior and likelihood using Bayes' rule [*Scales and Tenorio*, 2001] as

$$p(\mathbf{G}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{G})p(\mathbf{G})}{\sum_{\mathbf{G}\in\mathcal{G}^M} p(\mathbf{d}|\mathbf{G})p(\mathbf{G})}, \tag{4}$$

where $p(\mathbf{G}|\mathbf{d})$ is the posterior probability distribution and the term in the denominator $\sum_{\mathbf{G}\in\mathcal{G}^M} p(\mathbf{d}|\mathbf{G})p(\mathbf{G})$ is called the evidence which herein is interpreted as a normalizing constant. In general, there are fundamental challenges to obtaining the posterior distribution. It is clear that the size of the sample space of $\mathbf{G}$, $|\mathcal{G}^M| = |\mathcal{G}|^M$ (i.e., the size of the sample space for a single cell taken to the power of the number of cells in the grid). For 2-D or 3-D grids, M will often be greater than 10^3, thus $|\mathcal{G}^M|$ can be extremely large even if the sample space of the model variable at a single cell, $\mathcal{G}$, is small. For example, consider a discrete model of rock type which describes the location of a fluid reservoir in the Earth's subsurface. At each cell we might have $G_i \in \mathcal{G} = [\mathrm{reservoir}, \mathrm{nonreservoir}]$. This implies that $|\mathcal{G}| = 2$. However, even for small models $M > 10^3$; thus, $|\mathcal{G}^M| > 10^{301}$, and more typical industrial-scale models have $M \sim 10^6 - 10^9$. In general, it is very difficult to determine a posterior over such large model spaces [*Biegler et al.*, 2011].

If the prior and likelihood are known parametrically, then it may be that the product $p(\mathbf{d}|\mathbf{G})p(\mathbf{G})$ itself has a convenient parametric form which permits the normalizing constant in equation (4) to be calculated

analytically [*George et al.*, 1993]. In such cases the posterior can be expressed parametrically and thus characterized with ease, so the size of **G** is not problematic. However, it is much more common that the product $p(\mathbf{d}|\mathbf{G})p(\mathbf{G})$ can be evaluated for a particular given combination of **d** and **G** values but does not yield a parametric form [*Tarantola*, 2002]. Practical, spatial inverse problems almost always have this characteristic, even if both the local likelihood and prior properties are assumed: likelihood and prior probabilities may be evaluated from equations (1) and (3), respectively, but there is no clear way in which the two may be combined analytically to yield a parametric $p(\mathbf{G}|\mathbf{d})$ distribution.

Thus, if we wish to characterize the whole posterior in such cases then we might be forced to discretize the entire space $\mathcal{G}^M$ and systematically evaluate and store the product of prior and likelihood throughout this discretization. Clearly, as $|\mathcal{G}^M|$ is large such an operation would be extremely inefficient because it requires exploration of the whole of this space.

It is preferable instead to use a method which does not explore the entire posterior but which is nevertheless sensitive to the distribution of probability mass/density in the posterior [*Biegler et al.*, 2011]. The most common strategy to estimate the posterior distribution is therefore to obtain a set of samples from it, then use those samples to characterize it [*Mosegaard and Sambridge*, 2002]. Characterization might include probability estimation, or calculating point estimates or moments of the distribution. Obtaining samples from a distribution, for which one only knows the unnormalized density or probability, may be achieved using Markov Chain Monte Carlo (MCMC) methods. However, MCMC methods can suffer from bias issues since they rely on the assumption that the distribution of a chain of correlated samples (which these methods produce) converges to the posterior distribution within a *finite* set of samples; generally, there are no proofs that suggest this is true.

In this work, we derive a recursive algorithm for computing a decomposition of the posterior into a set of conditional distributions, which permits direct sequential sampling of **G** from $p(\mathbf{G}|\mathbf{d})$. Thus, this allows independent, rather than correlated, samples to be made from the posterior, and no assumptions need to be made regarding convergence. Henceforth, this is referred to as *exact* sampling, and the method may be a useful alternative to MCMC sampling methods. The derivation assumes both the local likelihood and local prior properties. Thus, the applicability of our methodology is limited to problems where we have cell-wise likelihoods and where we are able to specify the prior using a full conditional distribution (as in equation (3)). The method cannot be used as a useful alternative to MCMC methods for problems which do not fulfill these properties. The ability to specify the prior using a full conditional is central to the derivation of the algorithm, but the limitation of the conditional dependency to a certain range of cells is not (theoretically, $\mathrm{Ne}(i)$ may be any size). However, we will show later that the computational cost of the algorithm scales exponentially with the size of $\mathrm{Ne}(i)$ and the (minimum) dimension of the model grid. Thus, in practice limitations on the size of the neighborhood must be considered; such assumptions about limited (conditional) spatial dependency in **G** are often made in spatial inverse problems, so this does not obviate practical application of the algorithm. However, the effect of the dimension of the model grid on computational cost is not so easily reduced, and we therefore also develop an approximate version of the recursive algorithm to insure that the algorithm is computationally feasible for large grids. We also find that the cost of the algorithm scales with $|\mathcal{G}|$; thus, there must also be limitations to the size of the sample space, but, again, these are not so strong as to prevent the practical use of the algorithm.

In the next section we describe the convergence problems of MCMC methods in detail since this motivates the construction of the exact sampling methods and provides a bench mark against which the new algorithm should be compared. Then in the methodology we first describe the decomposition of the posterior which we use to create the recursive algorithm. We then describe the assumptions we make about the likelihood and prior in more detail, and how such distributions are determined in practice, before deriving the recursive algorithm for a 2-D grid. After a discussion of the algorithm's computational cost we discuss possible limitations on $\mathrm{Ne}(i)$ and $|\mathcal{G}|$ and define the approximate algorithm which permits application to realistically sized grids. Finally, we apply the approximate algorithm to the inversion of synthetic seismic attribute data and compare the results to that of Gibbs sampling, a MCMC algorithm.

2. Convergence Problems of MCMC Methods

In MCMC methods a chain of correlated samples is created from a target distribution. If the chain is long enough, the set of samples converges in distribution to this target distribution [*Gilks et al.*, 1996]. For example, if we wish to sample from the posterior $p(\mathbf{G}|\mathbf{d})$, we could use the archetypal MCMC algorithm, the Metropolis-Hastings algorithm [*Metropolis et al.*, 1953; *Hastings*, 1970] summarized in algorithm 1.

Algorithm 1 The Metropolis-Hastings algorithm to obtain n samples from $p(\mathbf{G}|\mathbf{d})$, where $\mathcal{U}[\mathcal{L}]$ is a Uniform distribution which is nonzero only over the set $\mathcal{L}$.

Obtain the initial ($t = 0$) sample $\mathbf{G}^{t=0} \sim \mathcal{U}[\mathcal{G}^M]$;
For $t = 1, 2, ..., n$
Obtain a candidate by sampling $\mathbf{G}'$ from a "proposal distribution":

$$\mathbf{G}' \sim q(\mathbf{G}'|\mathbf{G}^{(t-1)}); \tag{5}$$

Calculate probability α of transitioning to the candidate:

$$\alpha = \min\left\{1, \frac{p(\mathbf{G}'|\mathbf{d}) \cdot q(\mathbf{G}^{(t-1)}|\mathbf{G}')}{p(\mathbf{G}^{(t-1)}|\mathbf{d}) \cdot q(\mathbf{G}'|\mathbf{G}^{(t-1)})}\right\}; \tag{6}$$

With probability α, set $\mathbf{G}^t = \mathbf{G}'$, otherwise set $\mathbf{G}^t = \mathbf{G}^{t-1}$;
End For

The proposal distribution q used in algorithm 1 (equation (5)) is chosen on the basis of how well it promotes convergence to the desired distribution. Generally speaking, it should be as similar to the posterior distribution itself as possible [*Haario et al.*, 1999]. This is problematic since the posterior is not known a priori, and using a proposal distribution which is very dissimilar to the target can lead to slow convergence and bias. For example, consider a posterior probability density function with one maximum, which has a small support within which most of the probability mass is contained. Because of its small support, it might take many iterations of algorithm 1 to find the peak if we do not use a similar proposal distribution from which to draw candidates (this is the so-called Witch's Hat problem [see *Kass et al.*, 1998]). This can be remedied by choosing a proposal distribution which promotes the so-called random walk behavior by making the proposal distribution conditionally dependent upon the current member of the chain $\mathbf{G}^{(t-1)}$ (as is explicitly written in equation (5)); proposed candidates tend to be close to the current sample and tend to be selected preferentially by equation (6) if they too have high probability. This heuristic enforces our intuition that high-probability areas will be "close" together within the model space and encourages the chain to follow gradients toward regions of high probability.

The division in equation (6) implies that the normalization constant (in equation (4)) is never explicitly required for such an algorithm. The only requirement for convergence to the posterior distribution is that the Markov chain, which is induced by the use of the proposal distribution, be *irreducible*. Irreducibility means that all parts of the model space $\mathcal{G}^M$ may be reached by the chain starting from any position in the model space [*Gilks et al.*, 1996]. However, there is no assurance of convergence for finite n, and convergence is difficult to diagnose even if it occurs [*Besag and Green*, 1993]. The chain may be biased toward its starting position, so the initial part of the chain may exhibit "transient" (nonstationary) behavior. If the chain has converged, it will exhibit some "dynamic stationarity" and this in some cases may be used as a diagnostic of convergence. If the onset of stationarity can be detected, samples from this transient period (the so-called burn-in period) may be ignored in order to remove this bias from the ensemble.

Unfortunately, observing apparent dynamic stationarity over a finite set of samples does not imply that the ensemble has truly converged to the target distribution. This is problematic because it implies that the posterior distribution, which we estimate from the ensemble of samples, would be incomplete and biased (even if we remove the burn-in samples). For example, consider the case of a probability distribution having two distinct peaks, each with small support as in the example above. Suppose that the chain of samples were currently confined within one of those high-probability peaks. The probability of moving to the other peak is low since not only must the proposal distribution produce a sample within the other peak but the

probability of transition to that sample may then also tend to be low (since the chain is already within a high-probability region). This problem can be compounded by the use of local random walk proposal distributions if the probability of samples being chosen in between the peaks is low, since they may require that the chain traverse areas of low probability in order to move from one peak to another. This problem is similar to the problem of convergence to local maxima in optimization problems [*Saul and Roweis*, 2003]. However, in Bayesian inversion the objective is to determine the whole posterior distribution, and thus, it is a problem if the chain becomes stuck in *any* maxima (whether it be global or local) since this implies that the rest of the distribution may be inadequately sampled. We cannot easily diagnose this problem because the chain may nevertheless exhibit dynamic stationarity within the region of the maxima. Thus, in practice when we use MCMC techniques, it is hard to guarantee convergence to the posterior and hence ensure that the ensemble of samples is unbiased (unless we have a good idea of what the posterior should be like a priori).

There are many existing strategies which aim to detect or ensure convergence to the posterior by using heuristic rules to enhance mobility (or "mixing") of the chain around the model space. Well-known examples include simulated annealing [*Kirkpatrick et al.*, 1983] and hybrid MCMC [*Chen et al.*, 2001]. Such methodologies have been used successfully in a wide range of applications, but they do not ensure nor detect convergence: they only make it more probable that a nonbiased estimate of the posterior will be found within a practical number of iterations.

To a large extent then, both making a choice of proposal distribution and our ability to correctly detect stationarity depend on the form and strength of our prior information. As suggested above, in spatial inverse problems it is usual to specify much of the prior information in terms of relative spatial relationships between the variables in different cells, rather than in terms of values of the variable at absolute positions. In other words, probabilities are assigned to certain patterns or variations which occur across the model grid. The prior distribution naturally has high variance: there are many possible configurations of $\mathbf{G}$ which contain relative relationships or patterns which are acceptable, but the distance between such configurations within $\mathcal{G}$ may be large. An example is if a variogram is used to describe porosity heterogeneity in a subsurface reservoir: generally, there is a large range of configurations of porosity which would be consistent with any particular variogram [*Olea*, 1999, p.154]. Furthermore, in many cases the prior and likelihood distributions are multimodal [*Shahraeeni et al.*, 2012]. It is rarely discussed in the geophysical literature but we suppose that given these properties of the prior coupled with the local likelihood distributions (which may be arbitrarily complex), it is quite common for considerable complexity (and multimodality) to occur in the posterior distribution. Thus, the problems associated with bias in the convergence of MCMC sampling are highly relevant to a range of spatial problems that invoke MCMC methods.

3. Methodology

In this paper we derive a sampling methodology which avoids the use of MCMC sampling techniques altogether. The methodology estimates the conditional decomposition of the posterior distribution as

$$p(\mathbf{G}|\mathbf{d}) = \prod_{i=1}^{M} p(G_i|\mathbf{d}, \mathbf{G}_{<i}). \tag{7}$$

where $<i$ denotes the set of indices $1, \ldots, i-1$ which for $i = 1$ represents the empty set. We refer to the $p(G_i|\mathbf{d}, \mathbf{G}_{<i})$ distributions as the *partial conditionals*. Obtaining these distributions allows sequential sampling from the posterior [*Journel et al.*, 1998]. This refers to the process of first sampling G_1 from $p(G_1|\mathbf{d})$, then G_2 from $p(G_2|\mathbf{d}, G_1)$, then G_3 from $p(G_3|\mathbf{d}, G_1, G_2)$ and so forth until G_M is sampled from $p(G_M|\mathbf{d}, \mathbf{G}_{<M})$, each time using the previously sampled $\mathbf{G}_{<i}$ values as the conditioning variables. If each of the partial conditionals are of closed form, then each can be sampled exactly and the vector of samples for all cells $\mathbf{G}$ is itself an exact sample from the posterior. One then need only repeat the sequential sampling process to obtain another independent sample from the posterior; in this way we avoid the problems of convergence associated with the use of correlated MCMC sampling.

We use a recursive algorithm to determine the partial conditional distributions in closed form based on the algorithm of *Bartolucci and Besag* [2002]. Such recursive algorithms have their roots in hidden Markov chains [*Baum et al.*, 1970; *Scott*, 2002] and have been applied to spatial inverse problems [*Ulvmoen and Hammer*, 2010]. However, such methods require significant computational resources and as such in the past have only been applied to small problems [*Friel et al.*, 2009]. We believe that computational advances now

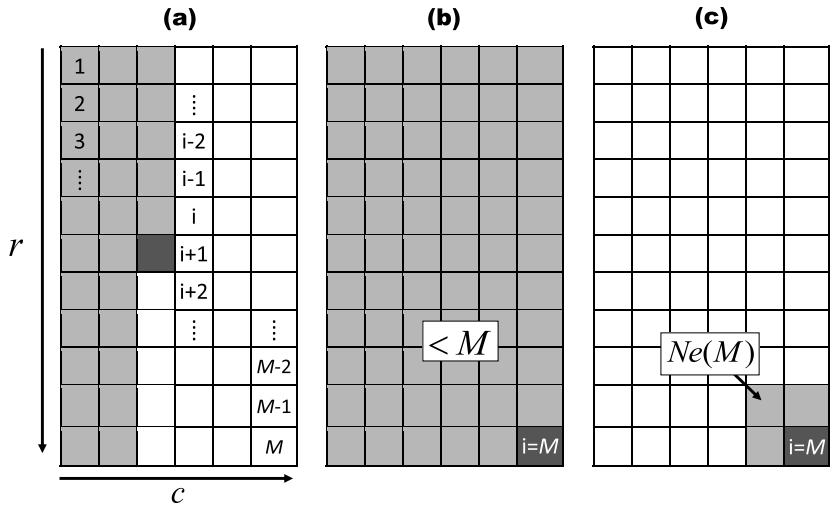

Figure 1. (a) Indexing of the 2-D grid with dimensions r (number of rows) and c (number of columns). The total number of cells $M = c \times r$. Also depicted is the dependency structure of the partial conditionals, $p(G_i|\mathbf{d}, \mathbf{G}_{<i})$, in equation (7): the dark gray cell is the variate G_i, and the light gray cells are those containing the conditioning variables $\mathbf{G}_{<i}$. These distributions are also conditioned upon data in all cells, $\mathbf{d}$. (b) The dependency of $p(G_M|\mathbf{d}, \mathbf{G}_{<M})$ (i.e., when $i = M$). (c) When $i = M$ the set $\{<M\}$ must contain the neighborhood of M; thus, the dependency of $p(G_i|\mathbf{d}, \mathbf{G}_{<i})$ is limited to the neighborhood of M (one possible example of such a neighborhood is shown here; other examples are shown in Figure 2).

make practical applications of these algorithms possible, when appropriate approximations are made to the conditional decomposition in equation (7). Indeed, *Arnesen* [2010] and *Tjelmeland and Austad* [2012] have already shown this to be true. However, the derivation of their recursive algorithm and the required approximations for its practical application are based on the representation of the posterior as a Gibbs potential [*Friel and Rue*, 2007]. We present a more pragmatic approach and develop our approximation using a probabilistic terminology (developed initially by *Bartolucci and Besag* [2002]). Importantly, this permits the exact sampling algorithm to be implemented easily and adapted for use in certain geophysical inverse problems.

In the following, we develop the recursive algorithm for a 2-D grid with r rows and c columns indexed as shown in Figure 1 (the algorithm can easily be generalized to 3-D grids or collapsed to 1-D grids). We first describe the likelihood and prior terms in more detail and then derive the recursive algorithm itself. We discuss its computational cost with respect to the parameters of the inversion and appropriate approximations which permit it to be applied to large grids.

3.1. The Likelihood Term

As a consequence of the local likelihood property, we interpret the likelihood term as an unnormalized probability distribution describing the probability of $\mathbf{G}$ given $\mathbf{d}$. Using equation (1) and elementary probability identities, the likelihood $p(\mathbf{d}|\mathbf{G})$ may be decomposed as

$$p(\mathbf{d}|\mathbf{G}) = \prod_{i=1}^{M} p(d_i|\mathbf{G}, \mathbf{d}_{<i}) = \prod_{i=1}^{M} p(d_i|G_i). \tag{8}$$

To use equation (8) we require the likelihood distributions for each datum, that is, for each individual cell in the grid, $p(d_i|G_i)$. The likelihood distributions should therefore be understood as functions of G_i since d_i is fixed (i.e., as stated above they are interpreted as unnormalized probability distributions over G_i). Effectively, each of these distributions represents the likelihood for an individual inverse problem in each grid cell. Of course, it can be computationally demanding to obtain the solution to all of these. However, it is often the case that these inverse problems are themselves very similar between cells (e.g., the same forward physics often applies at each point in the subsurface). Furthermore, both $\mathcal{G}$ and $\mathcal{D}$ may be of limited size, thus parametric estimation of the cell-wise posterior $p(G_i|d_i)$ as a *function* of d_i may be feasible. For example, *Shahraeeni et al.* [2012] trained neural networks to map from seismic attribute data (S and P wave impedances) to the complete posterior distribution over rock-physical parameters. The same neural network could be used to determine the posterior for each cell in the subsurface extremely rapidly. A similar

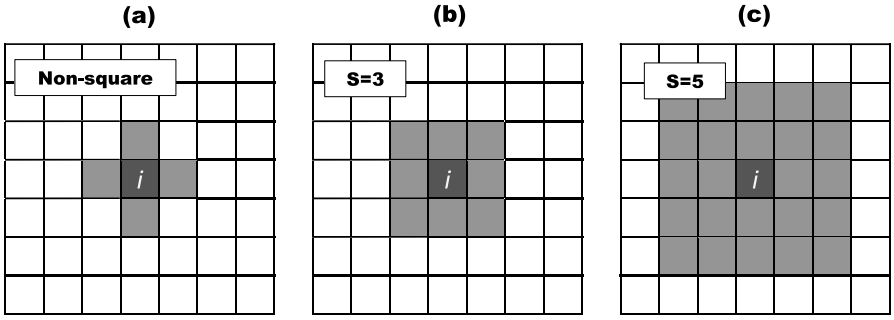

Figure 2. Possible neighborhood arrangements. (a) A "nonsquare" neighborhood commonly used in image processing. (b) A square neighborhood with side of length 3 cells (which we denote $S = 3$). (c) A square neighborhood with $S = 5$.

method was used for global tomographic inversions by *Meier et al.* [2007]. Such posterior distributions can be manipulated using Bayes' rule to obtain the required likelihood distributions by removing the influence of the prior [*Walker and Curtis*, 2014]. Since these other methods exist, we do not focus here on how best to obtain the individual likelihood distributions. Henceforth, we simply assume that we have the appropriate likelihood distributions available and readily accessible for use in the recursive algorithm.

3.2. The Prior Term

The local prior property (equation (3)) defines the form of prior information that we use. However, we have not discussed the neighborhood structure or how the actual full conditional distribution itself may be determined. Typical neighborhood structures are illustrated in Figure 2. A common choice for $\mathrm{Ne}(i)$ is a square centered on i. These neighborhoods can be defined by the length of the square's sides, S (see Figures 2b and 2c). Simple modifications are made to such neighborhoods when i is close to boundaries (i.e., where there are no neighbors beyond boundaries). We will henceforth consider only such square neighborhoods for derivation of the recursive algorithm.

A joint distribution $p(\mathbf{G})$ only gives rise to a valid set of full conditionals (where we are now considering one such distribution for each cell or element of $\mathbf{G}$) if the so-called *positivity* condition is fulfilled. This requires that if the individual marginal probability of each G_i is nonzero over its entire sample space (i.e., $P(G_i) > 0 \; \forall \; G_i \in \mathcal{G}, \; \forall \; i$ which we assume to be the case here), then the joint probability of all the G_i variables must be nonzero over their entire joint sample space (i.e., $P(\mathbf{G}) > 0 \; \forall \; \mathbf{G} \in \mathcal{G}^M$). The necessity of the positivity requirement can be motivated by attempting to apply Brook's lemma [*Brook*, 1964] to obtain the joint distribution from the full conditionals (see, e.g., *Rue and Held* [2005, pp.30-31]).

Even if positivity is fulfilled, an arbitrary set of full conditionals does not necessarily define a *valid* joint probability distribution $p(\mathbf{G})$. This is because the full conditionals may not be self-consistent. J. M. Hammersley and P. Clifford (unpublished manuscript, 1971) were the first to describe the necessary conditions on $p(\mathbf{G})$ which must be met for it to yield a set of full conditionals with a certain neighborhood structure. The Hammersley-Clifford theorem as proven by *Besag* [1974] states that $p(\mathbf{G})$ must factorize over sets of indices called "cliques." A clique is defined as a set of indices, $\Lambda = [\lambda_1, ..., \lambda_{|\Lambda|}]$, where each element $\lambda_i \in \{\mathrm{Ne}(\lambda_q), \; \forall \; \{q \in 1, ..., |\Lambda|\} \backslash i\}$: in words, it is a set comprising indices which are all neighbors of each other. Defined by the chosen neighborhood structure on the grid, $p(\mathbf{G})$ must factorize over all cliques. This ensures that when full conditionals are calculated from the joint distribution (i.e., using equation (3)), the correct neighborhood dependency structure is induced. In turn, this implies that the prior must have the form

$$p(\mathbf{G}) = \prod_{j=1}^{C} f_j(\mathbf{G}_{\Lambda_j}) \tag{9}$$

where C is the number of cliques on the grid, f_i are functions of the cliques, and $\mathbf{G}_{\Lambda_j}$ is all G_i values within clique j. This equation embodies the *factorization* condition which must be met by the joint distribution to yield full conditionals with a certain neighborhood structure. Since the full conditionals are derived from the joint distribution, it is possible to determine the appropriate factorization conditions on the full conditionals which yield a valid joint distribution [*Besag*, 1974].

In the case of spatial inversion, we stipulated that the full conditionals are invariant to i, that is, that we specify the prior by a single, duplicate full conditional (except at the edges of the grid). Regardless, the full conditional(s) must still meet the above conditions. Appropriate full conditional probabilities which meet these conditions can be derived from training images (e.g., *Varma and Zisserman* [2003]). It is easy to see that the factorization requirement is irrelevant if the neighborhoods are not restricted as in equation (2); since then each G_i variable is a neighbor of all others (then the cliques are the size of the grid, and no factorization is required). However, if the neighborhoods are limited in extent (which, as suggested in the introduction, we must apply for computational efficiency), this factorization requirement reduces the flexibility of the full conditional distribution [*Besag*, 1974]. We return to this point later.

3.3. The Recursive Algorithm

In order to determine the posterior and to sample from it efficiently, we develop a recursive algorithm based on the work of *Bartolucci and Besag* [2002]. Set notation is used in the derivation, and brackets ({ }) are used to enclose sets for clarity. Sets will be used to reference subsets of cells in the grid and their associated variables as a vector, for example, $\mathbf{G}_{\{<4\}\setminus 1} = [G_2, G_3]$.

Our goal is to calculate the posterior distribution $p(\mathbf{G}|\mathbf{d})$ on the left of equation (7) by evaluating the partial conditionals $p(G_i|\mathbf{d}, \mathbf{G}_{<i})$ on the right-hand side. These distributions can be found efficiently by using the recursive algorithm of *Bartolucci and Besag* [2002]. Overall in the algorithm the partial conditionals are calculated in the order $i = M, M-1, ..., 2, 1$. To calculate the partial conditional for cell i one must first calculate

$$p(G_i|\mathbf{d}, \mathbf{G}_{\{\leqslant k\}\setminus i}) \tag{10}$$

where

$$k = \max(\mathrm{Ne}(i)). \tag{11}$$

Given the definition of k in equation (11), the set $\{\leqslant k\}\setminus i$ will contain the neighborhood of i. Thus, because of the local prior property (equation (3)), there can be no dependence on G_i values outside of the neighborhood in equation (10). Also there is no dependency on data apart from that located at cell i (in equation (10)), because of the local likelihood property (equation (1)). Thus, we may rewrite equation (10) as

$$p(G_i|\mathbf{d}, \mathbf{G}_{\{\leqslant k\}\setminus i}) = p(G_i|d_i, \mathbf{G}_{\mathrm{Ne}(i)}), \tag{12}$$

and this expression can be decomposed, using Bayes' rule, into two terms:

$$p(G_i|\mathbf{d}, \mathbf{G}_{\{\leqslant k\}\setminus i}) = \mathcal{Z}_i p(d_i|G_i) p(G_i|\mathbf{G}_{\mathrm{Ne}(i)}) \tag{13}$$

where $p(d_i|G_i)$ is a likelihood distribution as described above, $p(G_i|\mathbf{G}_{\mathrm{Ne}(i)})$ is the prior full conditional as described above, and $\mathcal{Z}_i = \left(\sum_{G_i \in \mathcal{G}} p(d_i|G_i) p(G_i|\mathbf{G}_{Ne(i)}) \right)^{-1}$ is a normalizing constant. If we assume that the likelihood distributions have been determined and that we have obtained the prior full conditional, then equation (13) can be determined immediately. $\mathcal{Z}_i$ must be calculated by summation, but this will be an undemanding task if both $\mathcal{G}$ and $|\mathrm{Ne}(i)|$ are not prohibitively large. Once equation (13) is determined then the identity

$$p(G_i|\mathbf{G}_{\{\leqslant j-1\}\setminus i}, \mathbf{d}) = \left\{ \sum_{G_j \in \mathcal{G}} \frac{p(G_j|\mathbf{G}_{<j}, \mathbf{d})}{p(G_i|\mathbf{G}_{\{\leqslant j\}\setminus i}, \mathbf{d})} \right\}^{-1} \tag{14}$$

from *Bartolucci and Besag* [2002], may be applied recursively, for $j = k, k-1, ..., i+2, i+1$. At $j = i+1$ the result gives the desired partial conditional at cell i. The application of this identity represents a secondary backward recursion within the algorithm. It should be understood that since $i = M, M-1, ..., 2, 1$, the $p(G_j|\mathbf{G}_{<j}, \mathbf{d})$ distributions in equation (14) will have been determined in the previous iterations. Consequently, the algorithm must be initiated at $i = M$ where the partial conditional term can be calculated immediately since the neighborhood of cell M, $\mathrm{Ne}(M)$ is entirely contained within the conditioning cells in the partial conditional, i.e.,

$$p(G_M|\mathbf{G}_{<M}, \mathbf{d}) = p(G_M|d_M, \mathbf{G}_{\mathrm{Ne}(M)}) = \mathcal{Z}_M p(d_M|G_M) p(G_M|\mathbf{G}_{\mathrm{Ne}(M)}), \tag{15}$$

where $\mathcal{Z}_M$ again denotes the normalizing constant required by Bayes' rule. Once $p(G_M|\mathbf{d}, \mathbf{G}_{<M})$ is determined, then $p(G_{(M-1)}|\mathbf{d}, \mathbf{G}_{<(M-1)})$ can be calculated and so forth, until the posterior (equation (7)) is determined. Sequential sampling from $p(\mathbf{G}|\mathbf{d})$ can then be performed using the determined partial conditionals. The complete recursive algorithm is summarized in algorithm 2.

It should be noted that the conditional distributions as written in all equations above are strictly correct. However, there may be conditional independence from some of the written conditioning variables. We do not explicitly indicate this conditional independence here in order to make it clear that these distributions are conditioned by these variables (even if they may be conditionally independent); thus, these distributions then cannot be confused for marginals over those conditioning variables. This is important because the domain of the numerator and denominator must be compatible for the division in equation (14) to be valid. Details of the conditional independence structure are given in Appendix A.

Algorithm 2 can be used almost without modification for 3-D grids; only a change must be made to the indexing of the grid such that it runs over a third dimension (in addition to the rows and columns of the 2-D case). A cubic 3-D neighborhood structure would also have to be defined (using this indexing).

Algorithm 2 Recursive algorithm for a 2-D grid with r rows and c columns with $M = r \times c$ cells and neighborhood structure $\mathrm{Ne}(i)$.

Calculate $p(G_M|\mathbf{G}_{<M}, \mathbf{d}) = \mathcal{Z}_M p(d_M|G_M)p(G_M|\mathbf{G}_{\mathrm{Ne}(M)})$;
For $i = M-1, M-2, ..., 2, 1$
  Calculate $k = \max(\mathrm{Ne}(i))$;
  Calculate $p(G_i|\mathbf{d}, \mathbf{G}_{\{\leqslant k\}\setminus i}) = \mathcal{Z}_i p(d_i|G_i)p(G_i|\mathbf{G}_{\mathrm{Ne}(i)})$
  For $j = k, k-1, ..., i+2, i+1$
    Calculate the recursive identity

$$p(G_i|\mathbf{G}_{\{\leqslant j-1\}\setminus i}, \mathbf{d}) = \left\{ \sum_{G_j \in \mathcal{G}} \frac{p(G_j|\mathbf{G}_{<j}, \mathbf{d})}{p(G_i|\mathbf{G}_{\{\leqslant j\}\setminus i}, \mathbf{d})} \right\}^{-1};$$

  End For
  Retain $p(G_i|\mathbf{G}_{<i}, \mathbf{d})$;
End For

3.4. Computational Limitations and Approximations

Bartolucci and Besag [2002] derived an expression for the number of floating point operations required to calculate the partial conditionals, and hence determine the posterior, using algorithm 2 for the nonsquare neighborhood structure illustrated in Figure 2a. It can be derived by applying the conditional dependency structure discussed in Appendix A. We use a slightly modified version of the expression which gives an upper limit to the number of floating point operations required to calculate all the partial conditionals, for a grid with square neighborhood structure of side S, as

$$r \times c \times S \times r \times |\mathcal{G}|^{S \times r} \tag{16}$$

where r is the number of rows, c is the number of columns, S is the dimension of the square template, and $|\mathcal{G}|$ is the size of the sample space of G_i. Since the direction of indexing is arbitrary, r and c are interchangeable (i.e., we could run the algorithm on a grid with indexing in the perpendicular direction to that in Figure 1). Thus, if the dimensions are unequal, then the direction should be chosen such that the lowest dimension appears in the exponent. Despite the exponentiation of $|\mathcal{G}|$ in equation (16), the size of the sample space would not cause computational problems for the recursive algorithm in many real applications. For example, in the inversion of seismic attributes for reservoir parameters we often invert for discrete parameters like rock lithology-fluid class. The number of such classes can be low (see, e.g., *Rimstad and Omre* [2010] where $|\mathcal{G}| = 4$) or geological considerations can allow us to reduce the number of classes by implementing "nesting" of lithologies within one another.

Equation (16) illustrates the importance of the local prior property for efficient computation of the recursive algorithm: it is clear that since S appears in the exponent, the size of the square neighborhood must be

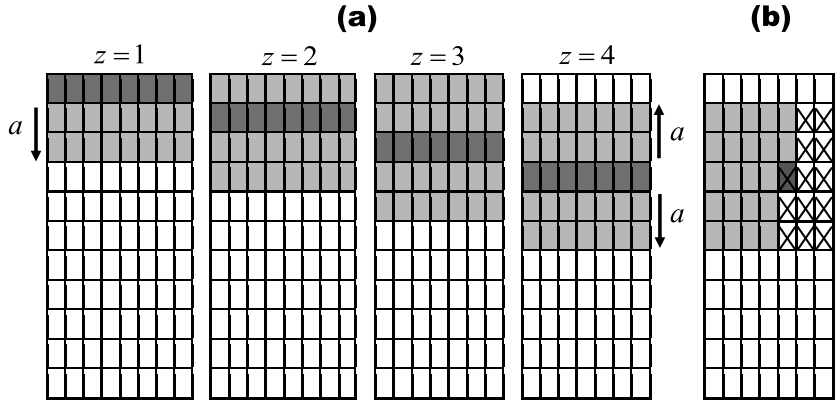

Figure 3. Illustration of the approximation (algorithm 3) to the recursive algorithm (algorithm 2) with approximation parameter $a = 2$. The full recursive algorithm is run on subgrids centered on each row of the complete grid. Subgrids comprise a rows above and below the current row of the complete grid. When a rows do not exist in either direction the subgrid is truncated to include only the available rows. Partial conditionals are determined for each cell of each subgrid. (a) The subgrids centered on rows $z = \{1, 2, 3, 4\}$, where gray cells are members of the subgrids. The partial conditionals determined in the dark gray cells (i.e., for row z of each subgrid) are retained as approximations to $p(G_i|\mathbf{G}_{<i}, \mathbf{d})$ and are thus used for exact sampling from $p(\mathbf{G}|\mathbf{d})$. (b) The dependency structure of the resulting approximate partial conditionals. The dark gray cell is the cell containing the variate, G_i. The light gray cells are those containing the conditioning G_i variables (note that we have not taken into account the conditional independence implied by the global Markov property given in equations (A1) and (A2)). The cells containing crosses are those containing data which are involved in the evaluation of the corresponding partial conditional.

limited to permit efficient application of the algorithm. In many real applications S is assumed to be quite low (see, e.g., *Rimstad and Omre* [2010] where $S = 3$). Thus, this limitation does not obviate the practical application of the algorithm. Unfortunately, however, realistically sized grids have a minimum dimension of at least hundreds of cells [*Caers*, 2005]. Since this number appears in the exponent (r in equation (16)), it is clear that the algorithm, as presented, would be computationally infeasible even with sufficiently low S and $|\mathcal{G}|$ parameters. This motivates us to define an approximation that permits the algorithm to be applied to realistically sized grids by reducing the number which appears in the exponent. To do this we henceforth assume that we have indexing as defined in Figure 1a. Then, roughly speaking, the approximation is to take smaller bands of the grid and run algorithm 2 on these bands.

More precisely, for each row in the grid, $z = 1, 2, ..., r - 1, r$, the set of rows $l(z) = \{\max(z - a, 1), ..., z, ..., \min(z + a, r)\}$ is selected, where a is the so-called approximation parameter. Note that by definition $l(z)$ ignores nonexistent rows. This defines a so-called subgrid for each z, denoted $\left[\mathbf{G}^{*z}, \mathbf{d}^{*z}\right]$, whose elements are defined by $\left[G_i, d_i\right] \in \left[\mathbf{G}^{*z}, \mathbf{d}^{*z}\right] \; \forall \, i : \mathcal{R}(i) \in l(z)$, where the operator $\mathcal{R}(i)$ returns the row to which the cell with index i belongs. Algorithm 2 is run on each $\left[\mathbf{G}^{*z}, \mathbf{d}^{*z}\right]$ subgrid. Once run, each cell in each subgrid has a partial conditional distribution associated with it. For each subgrid, only the partial conditionals for the cells in row z are retained as approximations to the partial conditionals in the *complete* grid. In mathematical terms we set $p(G_i|\mathbf{G}_{<i}, \mathbf{d}) \approx p(G_i|\mathbf{G}^{*z}_{<i}, \mathbf{d}^{*z}) \; : z = \mathcal{R}(i)$. These are approximate because they are only dependent upon cells within the range of the smaller subgrid used in algorithm 3, and likewise only conditioned upon data in that grid. Figure 3a illustrates the use of subgrids for calculating the approximate partial conditionals, and Figure 3b illustrates the resulting dependency structure in one of these distributions. In effect, the approximation reduces the range at which data, d_i, is incorporated into the calculation of the partial conditionals in one direction (e.g., here the range is limited in the vertical direction). Also, the range of the conditioning cells (in terms of the G_i variable) is limited. The result of this approximation is that the computational upper bound in equation (16) is reduced to

$$r \times c \times S \times a \times |\mathcal{G}|^{S \times a}. \tag{17}$$

This approximate algorithm is summarized in algorithm 3. An analogue of this algorithm for 3-D grids would consist of defining cubic subgrids (rather than rectangular subgrids, as in the 2-D case) and then running the 3-D version of algorithm 2 on these subgrids. However, expansion to three dimensions may be computationally expensive since the exponent in equation (17) would become $S \times a \times b$ where the approximation parameters a and b now describe the (limited) size of the cubic subgrid in two dimensions.

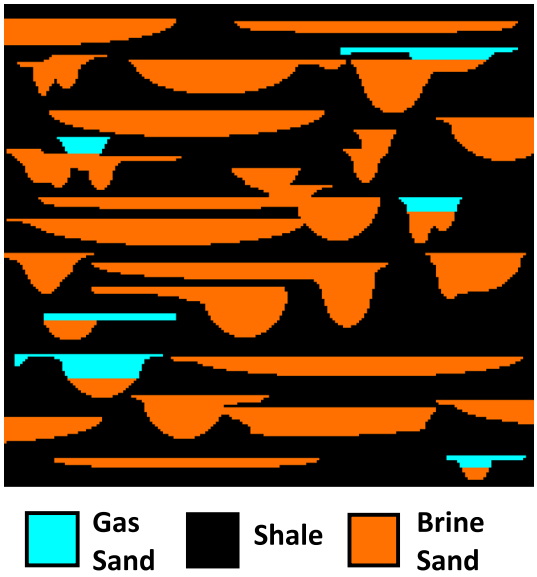

Figure 4. Training image used to obtain the probabilities in the prior full conditional, $p(G_i|\mathbf{G}_{\mathrm{Ne}(i)})$. The training image represents a 2-D cross section from the 3-D result of a geological process model. It contains sand-filled channels with overbank deposits, emplaced within shale. Gas has been injected into some of the channels.

Algorithm 3 Approximate recursive algorithm for a 2D grid with r rows and c columns with $M = r \times c$ cells, and approximation parameter a. The operator $\mathcal{R}(i)$ returns the row to which the cell with index i belongs.

For $z = 1, 2, ..., r-1, r$

Select rows $l(z) = \{\max(z-a, 1), ..., z, ..., \min(z+a, r)\}$;

Define subgrid $\left[G_i, d_i\right] \in \left[\mathbf{G}^{*z}, \mathbf{d}^{*z}\right] \forall\, i : \mathcal{R}(i) \in l(z)$;

Run algorithm 2 with subgrid $\left[\mathbf{G}^{*z}, \mathbf{d}^{*z}\right]$ to obtain $p(G_i|\mathbf{G}^{*z}_{<i}, \mathbf{d}^{*z}) \forall\, i : \mathcal{R}(i) \in l(z)$

End For

Retain approximations $p(G_i|\mathbf{G}_{<i}, \mathbf{d}) \approx p(G_i|\mathbf{G}^{*z}_{<i}, \mathbf{d}^{*z}) : z = \mathcal{R}(i)$;

4. Synthetic Application

We tested the approximate recursive algorithm by applying it to a synthetic inverse problem involving the inversion of seismic attribute data for lithology-fluid class (see, e.g., *Bosch et al.* [2010]). A categorical variable is used to represent three lithology-fluid classes in the subsurface:

$$G_i \in \mathcal{G} = \{\text{Shale, Gas sand, Brine sand}\} \tag{18}$$

where G_i is a univariate discrete-valued variable. Two 2-D cross sections of this variable were generated using a simple geological process model. The model generated channel shapes and overbank deposits. These were filled with brine sand and emplaced within a shale lithology. Gas was then emplaced in some of the channels, in a manner consistent with gas saturation in such geological features (i.e., obeying gravitational ordering). One of these cross sections, shown in Figure 4, was used to define the full conditional distributions and hence the prior. The other, shown in Figure 5a, was used to generate synthetic seismic attribute data; these data will be inverted using the proposed approximate recursive algorithm. The synthetic seismic attribute data were generated by considering each cell in the grid independently and using a probabilistic forward model, $p(\mathbf{d}_i|G_i)$ to generate collocated S and P wave impedances, $\mathbf{d}_i$, at each cell (where we now use bold vector notation for the data at each cell since there are two data values assigned to each cell).

To define $p(\mathbf{d}_i|G_i)$, we chose an appropriate rock physics forward function, $\mathbf{f}(\mathbf{m}_i)$. We used the Yin-Marion shaly sand model, which can predict the P and S wave impedances ($\mathbf{d}_i$) for a given shale-sand mixture and pore fluid. In our definition of this model, three rock-physical parameters ($\mathbf{m}_i$) were allowed to vary: clay volume content V_{clay}, sandstone matrix porosity ϕ_{sand}, and water saturation S_{wt} (hence gas saturation,

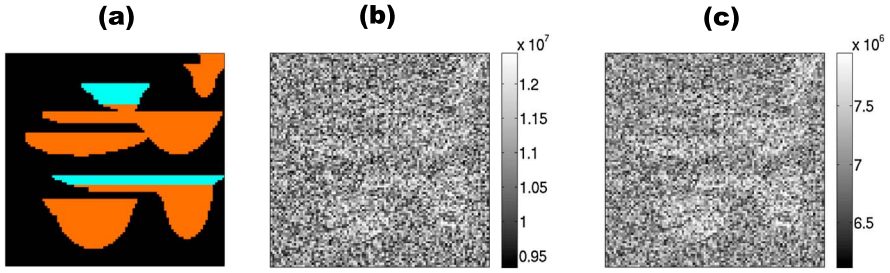

Figure 5. (a) The true target cross section. (b and c) *S* and *P* wave impedance data, respectively, generated using the Yin-Marion shaly sand model (described in Appendix B).

$S_g = 1 - S_{wt}$), such that $\mathbf{m}_i = [V_{\text{clay}}, \phi_{\text{sand}}, S_{\text{wt}}]_i$. A component of random Gaussian noise was added to the output of the rock physics forward function; thus, it could be written as a probability distribution, $p(\mathbf{d}_i|\mathbf{m}_i)$. Full definitions of the Yin-Marion shaly sand model, $\mathbf{f}(\mathbf{m}_i)$, and this distribution are given in Appendix B. The next part of defining $p(\mathbf{d}_i|G_i)$ required definition of a relationship between G_i and $\mathbf{m}_i$. This should be uncertain (probabilistic) as we would expect each lithology-fluid class to have a range of possible different rock-physical parameters [*Avseth et al.*, 2005]. Thus, each lithology-fluid class (in equation (18)) was assigned a distribution describing the probability of these parameters for that particular class $p(\mathbf{m}_i|G_i)$. A full description of $p(\mathbf{m}_i|G_i)$ is given in Appendix B.

Now the full probabilistic forward function can be defined since we have a mapping between G_i and $\mathbf{m}_i$, and between $\mathbf{m}_i$ and $\mathbf{d}_i$:

$$p(\mathbf{d}_i|G_i) = \int_0^1 \int_0^1 \int_0^1 p(\mathbf{d}_i|\mathbf{m}_i)p(\mathbf{m}_i|G_i)d\mathbf{m}_i. \tag{19}$$

This distribution can be sampled from without performing the integration analytically (which would be very difficult given the form of $\mathbf{f}(\mathbf{m}_i)$) by sampling sequentially first $\mathbf{m}_i$ from $p(\mathbf{m}_i|G_i)$ and then $\mathbf{d}_i$ from $p(\mathbf{d}_i|\mathbf{m}_i)$. Thus, we may sample from the distribution and obtain the synthetic data $\mathbf{d}_i$ from the lithology-fluid class G_i in each cell in the target section. The resulting data are shown in Figures 5b and 5c. As can be observed, the distribution of sand facies in Figure 5a is just discernible in these plots; however, there is little or no visual distinction between gas and brine sand facies.

Furthermore, given the ease of sampling from equation (19), we can also obtain the likelihood at each cell efficiently. To do this we begin by defining the prior distribution $p(G_i)$ to be Uniform (over $\mathcal{G}$). Sampling G_i from this distribution and then sampling $\mathbf{d}_i$ from equation (19) allows us to sample from the joint distribution $p(\mathbf{d}_i, G_i) = p(\mathbf{d}_i|G_i)p(G_i)$. Such samples can be used to estimate $p(\mathbf{d}_i, G_i)$ parametrically, and this parametric distribution can be used to obtain the likelihood as $p(\mathbf{d}_i|G_i) = p(\mathbf{d}_i, G_i)/p(G_i)$. The results are shown in Figure 6. This parameteric estimation is computationally simple since G_i is discrete and is small in terms of its sample space (i.e., $|\mathcal{G}| = 3$ from equation (18)) and can be performed by fitting a Gaussian mixture model over the data space for each lithology-fluid class.

It is important to note that we could not have obtained the likelihood without estimating $p(\mathbf{d}_i, G_i)$ first. Initially, although we could sample from $p(\mathbf{d}_i|G_i)$, we did not know it *parametrically*. The latent (or "nuisance") parameters $\mathbf{m}_i$ prevented us from doing so; thus, sampling was required. However, the estimation of the joint distribution (and hence the sampling) need only be done once, and obtaining the likelihood distribution at each cell in the target cross section then only requires fixing $\mathbf{d}_i$ at the appropriate value in $p(\mathbf{d}_i|G_i) = p(\mathbf{d}_i, G_i)/p(G_i)$.

Later we will apply another prior within the recursive algorithm (see equation (13)) to these likelihoods. This represents a so-called *prior replacement* calculation, which is the algebraic replacement of a prior implicit within one posterior distribution by a new, different prior distribution using Bayes' rule, to form a new posterior [*Walker and Curtis*, 2014]. As described in the latter referenced publication, to avoid undefined probabilities in the new posterior arising from division by zero in this calculation, the Uniform prior distribution $p(G_i)$ used to estimate $p(\mathbf{d}_i, G_i)$ must have nonzero probabilities wherever we expect the new, replacing prior to have nonzero probabilities. In this case the Uniform distribution (over the entirety) of the discrete sample space $\mathcal{G}$ satisfies this requirement.

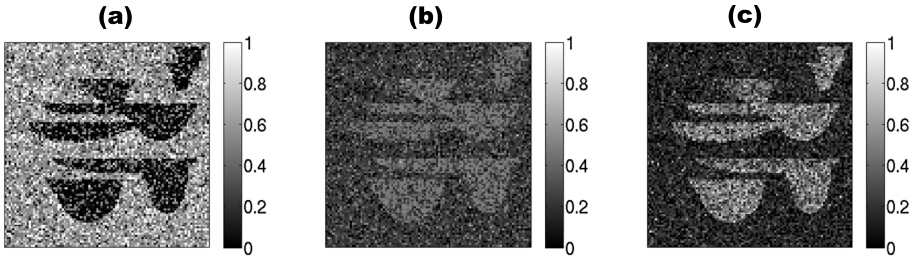

Figure 6. The likelihood of (a) shale, (b) gas sand, and (c) brine sand, determined using a parameterized version of $p(G_i, \mathbf{d}_i)$. The likelihoods are normalized such that (in each cell) we have $p(\mathbf{d}_i|\text{shale}) + p(\mathbf{d}_i|\text{gas sand}) + p(\mathbf{d}_i|\text{brine sand}) = 1$.

The neighborhood for the prior full conditional is square with $S = 3$ as in Figure 2; the actual distribution $p(G_i|\mathbf{G}_{\text{Ne}(i)})$ is derived from the training image, by visiting each cell in the training image grid which has appropriate neighbors available, and counting the occurrences of each conditional event. This method does not necessarily return a full conditional which is consistent with a valid joint distribution $p(\mathbf{G})$ over the entire grid. However, positivity can be ensured by adding a small number to any zero probabilities calculated in the full conditional by the above counting method. It is not easy to ensure that the calculated full conditional satisfies the factorization condition. We have found that attempting to use a full conditional which does definitely satisfy the factorization condition does not produce sufficiently realistic results. Full conditionals which satisfy the factorization requirement cannot contain the same amount of information as our more flexible definition of the full conditionals since they are much more parsimonious in their parameterization (see equation (9)).

Thus, we simply assume that the full conditional obtained using the above method is *approximately* correct. This leads to equation (14) being approximate, which in practice means that equation (14) yields probability distributions which are not normalized. Typically, the error in probability mass is <0.1, and we simply renormalize equation (14) to correct for this. This approximation is an added source of error in the results of the recursive algorithm in this particular example. However, below we compare its results to those obtained using an MCMC algorithm which uses exactly the same prior information and show that it compares favorably.

4.1. Results

With the likelihood distributions at each cell and the full conditional determined, the recursive algorithm can be applied and the approximate partial conditionals calculated. The approximation length used was $a = 4$. Once the partial conditionals have been found, independent samples from the posterior can be

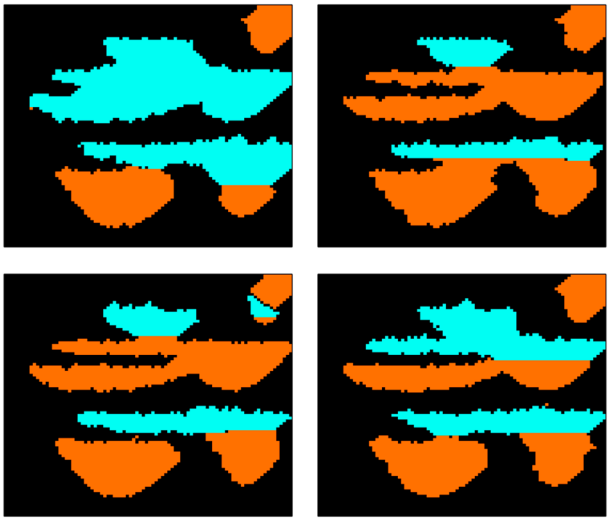

Figure 7. Four samples of **G** from the posterior $p(\mathbf{G}|\mathbf{d})$, obtained using the approximate recursive algorithm.

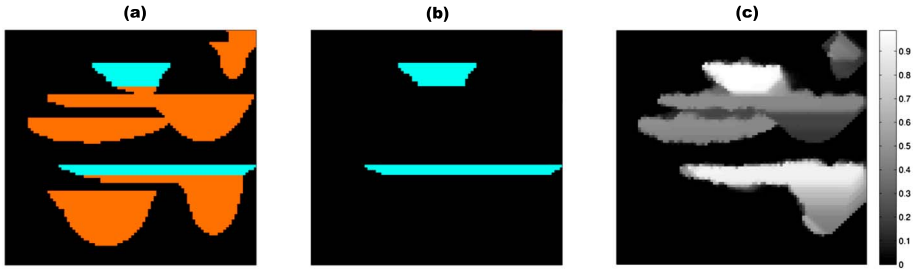

Figure 8. (a) The true cross section used to generate the data. (b) The true distribution of gas sand in this section for comparison. (c) The posterior cell-wise marginal probability of gas sand occurrence (i.e., $p(G_i = \text{gas sand}|\mathbf{d})$ at each cell) generated from the ensemble of samples from the posterior $p(\mathbf{G}|\mathbf{d})$, obtained using the approximate recursive algorithm.

determined rapidly. Using the recursive algorithm to find the partial conditionals took approximately 10,800 s on a standard desktop computer. Making each independent sample of $\mathbf{G}$ from the approximate posterior (specified using the partial conditionals) took approximately 0.1 s to produce the entire model cross section.

An ensemble of 1×10^4 samples from the posterior was made using the recursive algorithm. Figure 7 shows four example realizations from the ensemble. The ensemble of realizations was used to calculate the posterior cell-wise marginal probability of gas sand occurrence (i.e., $p(G_i = \text{gas sand}|\mathbf{d})$ at each cell) as an example of the kind of statistics that are then calculable. This is plotted in Figure 8 along with the target section for comparison.

5. Discussion

The results show that reasonable results can be obtained using the approximate recursive algorithm. The realizations in Figure 7 from the approximate posterior exhibit similarities to the target section in Figure 8a. The cell-wise posterior mean of gas sand occurrence in Figure 8c is consistent both with the true gas sand distribution in Figure 8b and the uncertainty which we might expect: it is nearly certain that the two gas accumulations exist, but there remains some uncertainty as to their exact extent. The quality of the estimate in Figure 8c compared to the information content of the likelihood in Figure 6 shows the additional value of the prior information contained in Figure 4 and embodied in the full conditionals.

These approximate results are somewhat difficult to appraise since we do not have an exact posterior result with which to compare. An alternative estimate for the posterior can be made using MCMC methods. However, such a method of sampling may suffer from the convergence and bias problems described earlier which motivated us to develop the new algorithm in the first place. Nevertheless, we used an MCMC methodology called Gibbs sampling [*Geman and Geman*, 1993] to obtain samples from the posterior and hence obtain an alternative posterior estimate for comparison. The Gibbs sampler, summarized in algorithm 4, uses a full conditional to update each cell at a time in the grid.

Algorithm 4 Gibbs sampling algorithm for sampling from $p(\mathbf{G}|\mathbf{d})$, where $\mathcal{U}[\mathcal{L}]$ is a Uniform distribution which is non-zero only over the set $\mathcal{L}$

Obtain initial sample $\mathbf{G}^{t=0} \sim \mathcal{U}[\mathcal{G}^M]$;
For $t = 1, 2, \ldots, n$
  Set $\mathbf{G}^t = \mathbf{G}^{t-1}$;
  Choose a cell i at random in the grid, $i \sim \mathcal{U}[1, M]$;
  Sample from $G_i' \sim p(G_i'|\mathbf{G}^t_{Ne(i)}, \mathbf{d})$;
  Set $G_i^t = G_i'$;
  Retain $\mathbf{G}^t$;
End for

The full conditional probability distribution required in algorithm 4, $p(G_i|\mathbf{G}_{\text{Ne}(i)}, \mathbf{d})$, is not the full conditional seen earlier in equation (3); it is dependent on the data $\mathbf{d}$ and is therefore derived in the same manner as equations (12) and (13). In itself, it is a simple expression which may be calculated immediately if the

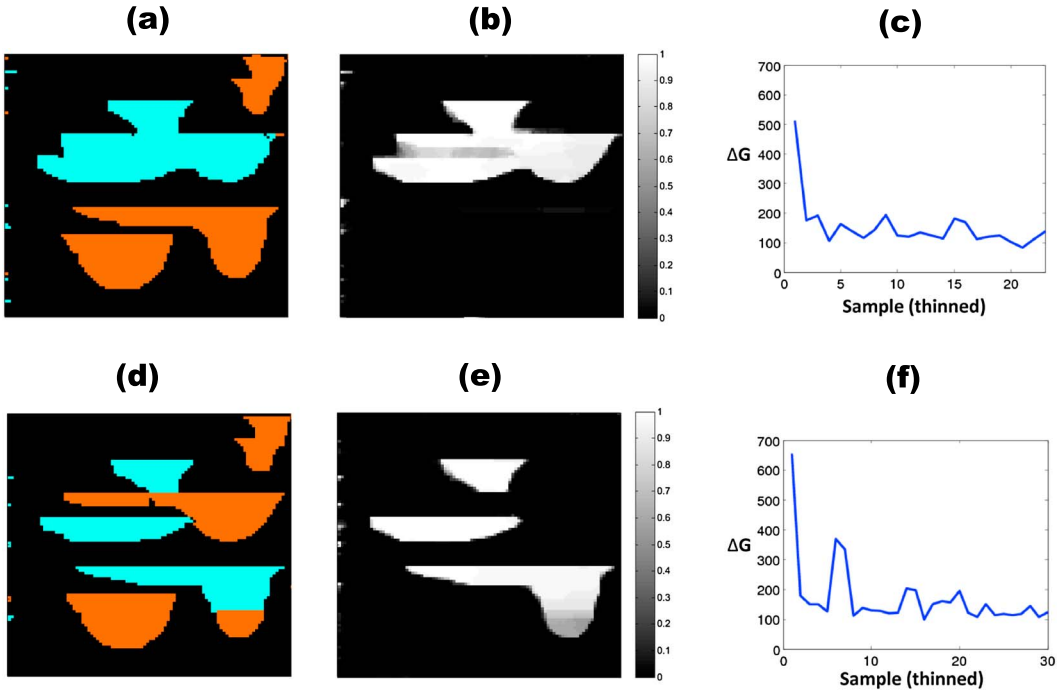

Figure 9. (a–f) The results of running the Gibbs sampling algorithm (a Markov chain Monte Carlo method) described in algorithm 4, to sample from the posterior $p(\mathbf{G}|\mathbf{d})$, with two different starting realizations shown. Figures 9a and 9d show the final realization after 1×10^8 iterations, Figures 9b and 9e show the posterior cell-wise marginal probability of gas sand occurrence (i.e., $p(G_i = \text{gas sand}|\mathbf{d})$ at each cell), and Figures 9c and 9e show the total number of changes in facies between consecutive retained (post-thinning) realizations.

cell-wise likelihood and prior full conditional are known. It can be shown that algorithm 4 is a special case of the Metropolis-Hastings algorithm (and as such it will converge to the target distribution as $n \rightarrow \infty$ if the chain is irreducible) where the proposal distribution is $p(G_i|\mathbf{G}_{\text{Ne}(i)}, \mathbf{d})$ and the probability of transition is always unity [*Geman and Geman*, 1993]. Importantly, it can be proved that if the full conditionals satisfy the positivity condition, the chain is irreducible, and hence, convergence as $n \rightarrow \infty$ is assured [*Robert and Casella*, 2004, p. 376].

The Gibbs sampler is popular because it does not require that either the full prior or likelihood distributions be calculated in each iteration. As with our recursive algorithm, the limited spatial dependency of the full conditionals is useful since it reduces the complexity of calculations required at each iteration to determine the proposal distribution in algorithm 4. However, because the Gibbs sampler is a random walk algorithm which only updates one cell at a time, it moves very slowly around the posterior, and thus convergence to the target distribution can be slow [*Belisle*, 1998; *Van Dyk and Park*, 2008]. Furthermore, the risk of becoming stuck in a maxima is increased because changes in the current state are incremental (only ever at a single cell). This problem can, to a certain extent, be addressed by rerunning the algorithm from different starting points. However, this may be of limited use if the model space is large [*Brooks and Gelman*, 1998] and does not in any case solve the fundamental problem which is the difficulty in ensuring that the Gibbs sampler will be able to visit all important parts of the model space within a practical time period and hence produce a chain of samples which will determine the posterior without bias.

We used the same cell-wise likelihood distributions and prior (defined using the same full conditionals) in algorithm 4 to sample from $p(\mathbf{G}|\mathbf{d})$. Initially, we ran the algorithm for 1×10^8 iterations which took approximately 9×10^9 s. We removed many of the resulting realizations by only retaining a sample every 4×10^6 iterations (this process of "thinning" removes highly correlated samples). From the 25 realizations retained, a cell-wise posterior probability of gas sand occurrence was calculated. This estimate and the final realization retained are plotted in Figures 9a and 9b. As discussed above, the Gibbs sampler has the tendency, in practice, to become "stuck" in probability maxima (and therefore can yield biased results). Thus, we reinitiated the algorithm with a different random starting point $\mathbf{G}^{t=0}$ and repeated the procedure. The results

are plotted in Figures 9d and 9e; note that slightly more realizations (30) were retained after thinning in the second run of Gibbs sampling. The two results are remarkably different. It seems that each has become stuck in a different probability maxima. This conclusion is reinforced when we inspect the sequential difference between the retained realizations (plotted in Figures 9c and 9f): the realizations change greatly at the start of the algorithm, but as the number of iterations increases these changes become increasingly small. Indeed, even when the first chain in Figure 9 was run for 10^9 iterations (taking approximately 24 h) there was little change in the retained realizations (e.g., only one accumulation of gas was ever realized).

Thus, the Gibbs sampling result cannot be trusted—it is clearly highly biased by the starting point because the chain induced is not practically recurrent. For example, if we ran the algorithm just once and got the upper results in Figure 9, we would only detect one of the accumulations of gas, while the lower results in Figure 9 contradict this conclusion.

However, it is clear that Gibbs sampling delivers individual realizations which are more consistent in some ways with the true model: for example, the shape of the channels is better defined in the Gibbs sampling results (the recursive algorithm produces "rough" channel edges). It should also be noted that if we take the two Gibbs results together, there are similarities with the results of the approximate recursive algorithm (which shows that both gas accumulations are almost certainly present simultaneously). The results of the recursive algorithm are therefore in part consistent with those of Gibbs sampling, but they seem to be more reliable since there is no bias induced by the starting point of the algorithm to (local) probability maxima.

Errors in the results of the recursive algorithm may be attributed to the approximations used to determine the partial conditionals. Not only will this approximation error be a function of the approximation parameter but also of the characteristics of the posterior distribution itself (controlled by the forward relation and the prior). We have not derived a method to obtain the approximation error a priori, or even a bound on it. This is a general problem with such approximation methods [*Friel and Rue*, 2007]. Even the rigorously derived, graph theory-based approximation of *Arnesen* [2010] cannot predict or bound the error a priori. Thus, either (i) an extensive empirical study of the relationship between approximation quality and those parameters mentioned above should be made, or (ii) the approximation should be rephrased in order to admit some way of finding a bound on the error. It is not clear how (ii) could be accomplished; thus, option (i) seems a more likely starting point for future work.

Another possible source of error is that we have used a full conditional which may not be consistent with a valid joint distribution. However, we argue that this is probably not the cause of the errors in the realizations produced by the recursive algorithm: the Gibbs sampler used exactly the same prior full conditional and did not produce realizations with such poor definition of the channels' edges. It should be noted that although the factorization condition was not satisfied, the positivity condition was. Thus, the chain induced in the Gibbs sampling algorithm was certainly, at least theoretically, recurrent if it had continued to an infinite number of samples.

In summary, the results obtained using the new sampling algorithm seem good and robust, and we argue that the approximation errors discussed above appear at least no worse than the errors associated with the results of Gibbs sampling. Neither errors can be quantified. With Gibbs sampling we are consoled by the fact that in the infinite limit the ensemble of samples will converge to the desired distribution, but for practical finite chains of samples this may never be the case. Extension of the algorithm to continuous variables G_i (such as those inverted for by *Shahraeeni et al.* [2012]) may be possible. However, it is likely that a sparse parametrization of the prior and likelihood would need to be chosen such that the computational cost may be controlled.

In addition to attempting to estimate the approximation error a priori, future work on our recursive algorithm should concentrate on its practical application to 3-D problems. We have discussed briefly how the recursive algorithm, and the subgrid approximation, may be applied to 3-D grids. However, it is concerning that the number of floating point operations required increases exponentially with the approximation parameter (i.e., b) in the third dimension. We propose that for practical application to 3-D grids, a different approximation scheme should be developed which further reduces the number of floating point operations required. The fundamental control on the computational expense of algorithm 2 is the size of the sample space of $\mathbf{G}$ (i.e., $\mathcal{G}^M$). This sample space is not explicitly chosen but forced upon us by the choice of spatial parameterization as a grid. It may not be optimal if a large part of the model space can be disregarded as

a geological impossibility. This is often the case, given the spatially structured nature of naturally occurring geology. If we call this segment of the model space—which may be assigned zero probability—$\mathcal{N}$, then the effective size of the model space should be $\mathcal{G}' = \mathcal{G}^M - \mathcal{N}$. It is clear that if we were able to somehow run algorithm 2 on $\mathcal{G}'$ rather than $\mathcal{G}^M$, then significant efficiency savings could be made. However, we found that implementation of this in practice is difficult since the division in equation (14) must be carried out with different irregular sample spaces for the denominator and numerator. Further work must be carried out before this approximation can be used effectively.

There are similarities between our recursive algorithm technique and multipoint geostatistical simulation techniques [*Remy et al.*, 2009, pp. 69–73]. These techniques can be interpreted as trying to determine a priori the partial conditionals $p(G_i|\mathbf{d}, \mathbf{G}_{<i})$ [*Strebelle*, 2002]. This means that training images are produced from the $\mathbf{G}$ and corresponding $\mathbf{d}$ variables. Then the partial conditionals are determined empirically from these by using either machine-learning techniques [*Caers*, 2001] or parametric estimation [*Strebelle*, 2002]. The advantage of this approach is that, in theory, no computation is required to obtain the partial conditionals: they are simply learnt from "examples" of $[\mathbf{G}, \mathbf{d}]$ and are ready for use immediately. In reality these examples are created by first generating a training image for the geological variable $\mathbf{G}$ and then using forward modeling to obtain $\mathbf{d}$. This is a significant computational burden, especially if the data generated by $p(d_i|G_i)$ has a high variance (as in the synthetic example presented above) or is costly to generate. Indeed, it may even be impossible to obtain enough samples in finite time, or with finite resources, to determine the partial conditionals sufficiently well. Consequently, the $\mathbf{d}$ variable is often referred to as "soft data" in such inversions, implying that it only constrains G_i locally, e.g., d_i may only constrain G_i [*Zhang et al.*, 2008]. As in these geostatistical learning strategies, the recursive algorithm requires a training image of $\mathbf{G}$ to be generated so that the prior full conditionals can be determined. However, a corresponding training image for $\mathbf{d}$ is not required: the recursive algorithm *analytically* incorporates the observed data into the computation of the partial conditionals using the cell-wise likelihood distributions. Thus, the recursive algorithm may be a useful alternative to current geostatistical learning-based strategies.

6. Conclusion

We have shown that the Bayesian solution for spatial inverse problems can be sampled using a recursive algorithm, if the problem is specified by a grid of model parameters with coincident, independent likelihood information, and spatially correlated prior information specified using a full conditional distribution. The recursive algorithm calculates the conditional probability distributions which allow the posterior distribution to be written as a decomposition which can be sampled exactly. We have developed approximations to the recursive algorithm such that it may be applied efficiently to a large 2-D grid of data. Because the posterior can be sampled exactly, the well-known convergence problems of random walk Markov chain Monte Carlo sampling algorithms are avoided. These algorithms (such as the Metropolis algorithm or Gibbs sampler) may not produce a set of samples which converge to the posterior (target) distribution in a practical time period. We successfully applied the recursive algorithm to a synthetic data set: we inverted seismic attribute data for lithology-fluid class. The synthetic data comprised noisy S and P wave impedances estimated at each cell in the 2-D grid. A training image was used to determine a suitable prior defined using a full conditional. From these two elements we estimated the posterior distribution of brine sand, shale sand, and gas sand throughout the grid. The results of the recursive algorithm compared well to the results of Gibbs sampling on the same synthetic data. The results of Gibbs sampling showed significant bias: the use of the Gibbs sampling results would likely have led to one very significant gas accumulation being completely unidentified. Both gas accumulations are reliably identified by the new recursive algorithm.

Appendix A: Details of Conditional Independence

The local properties of the prior and likelihood distributions induce conditional independence in certain conditional distributions. In terms of dependence upon the data, counterintuitively, the partial conditionals must incorporate nonlocal likelihood information even if the local likelihood property is assumed. Consider the general case of the partial conditional at cell i, $p(G_i|\mathbf{d}, \mathbf{G}_{<i})$; it is easy to show that because of the local likelihood property described by equation (1) we may rewrite this as being independent of the data $\mathbf{d}_{<i}$ which coincides with the conditioning $\mathbf{G}_{<i}$ variables. In mathematical terms $p(G_i|\mathbf{d}, \mathbf{G}_{<i}) = p(G_i|\mathbf{d}_{\geq i}, \mathbf{G}_{<i})$.

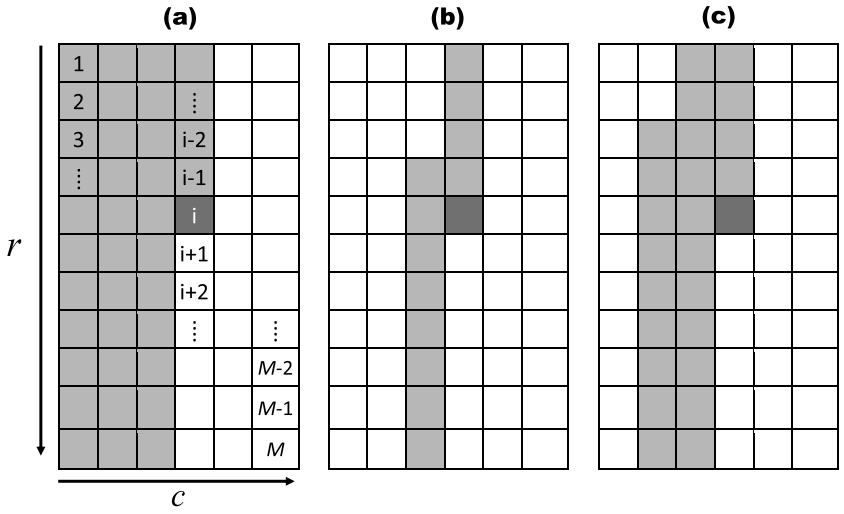

Figure A1. Illustration of conditional dependency structures of partial conditionals induced on a 2-D grid with square neighborhood structures by the global Markov property. (a) The dependency structure of the partial conditional distribution, $p(G_i|\mathbf{d}, \mathbf{G}_{<i})$, without consideration of conditional independence induced by a neighborhood structure. (b) The dependency structure with consideration of the square neighborhood structure (with side $S = 3$), i.e., $p(G_i|\mathbf{d}, \mathbf{G}_{<i}) = p(G_i|\mathbf{d}, \mathbf{G}_J)$ where $J = \{j|j < i \wedge \max(\mathrm{Ne}(j)) \geqslant i\}$. (c) Same as for Figure A1b but with a square neighborhood with side $S = 5$.

By equation (1) it is also obvious that G_i is dependent upon d_i in the partial conditional. However, it is less obvious that G_i must also be dependent upon the data $\mathbf{d}_{>i}$, i.e., $p(G_i|\mathbf{d}, \mathbf{G}_{<i}) \neq p(G_i|d_i, \mathbf{G}_{<i})$. The reason for this is that $\mathbf{d}_{>i}$ yields information about $\mathbf{G}_{>i}$. Furthermore, the prior specifies correlation between the elements of $\mathbf{G}$. Thus, $\mathbf{d}_{>i}$ must yield indirect information about G_i and therefore cannot be ignored in the partial conditional. Therefore, the recursive algorithm is designed to efficiently incorporate the nonlocal likelihood information (i.e., from $\mathbf{d}_{>i}$) into the calculation of the partial conditional distributions.

For the G_i variables, if we have assumed the local prior property, i.e., we assumed equation (3) with some $\mathrm{Ne}(i)$, then the dependency in the partial conditional may be limited to a subset of $\mathbf{G}_{<i}$. This subset is determined by the global Markov property [*Rue and Held*, 2005, p. 24], which can be stated by supposing that we have three mutually exclusive subsets $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{S}$ of cells (indices) in the grid, and some neighborhood structure for the G_i variables in the grid. The property then states that if starting at any cell in $\mathcal{A}$, and only by passing between neighbors, one cannot reach any cell in $\mathcal{B}$ without passing through a cell within $\mathcal{S}$, then $\mathbf{G}_{\mathcal{B}}$ is conditionally independent of $\mathbf{G}_{\mathcal{A}}$ given $\mathbf{G}_{\mathcal{S}}$. For square neighborhood structures on a 2-D grid (e.g., Figure 1a), this can be used to show that the partial conditionals are limited in dependency such that it is possible to write

$$p(G_i|\mathbf{d}, \mathbf{G}_{<i}) = p(G_i|\mathbf{d}, \mathbf{G}_{J(i)}) \tag{A1}$$

where

$$J(i) = \{j|j < i \wedge \max(\mathrm{Ne}(j)) \geqslant i\}\,. \tag{A2}$$

The resulting reduced dependency structure is demonstrated for a partial conditional in Figure A1 for the case of square neighborhoods with $S = 3$ and $S = 5$. Application of the global Markov property to the distributions generated by equations (13) and (14) in the recursive algorithm yields distributions with similarly reduced dependency structure. Thus, the domain of these distributions can be calculated, which permits the

Table A1. Table Describing Bounds Used to Define $p(\mathbf{m}_i|G_i)$

Lithology-Fluid Class	Clay Content by Volume (C)	Sandstone Matrix Porosity (ϕ_s)	Water Saturation (S_{wt})
Shale	[0.20, 0.40]	[0.20, 0.40]	[1.00, 1.00]
Gas sand	[0.00, 0.20]	[0.20, 0.40]	[0.05, 0.60]
Brine sand	[0.00, 0.20]	[0.20, 0.40]	[0.60, 1.00]

number of operations required to calculate equations (13) and (14) in algorithm 2 to be determined. This, in turn, permits the computational cost of the recursive algorithm to be estimated.

Appendix B: The Forward Model: The Yin-Marion Shaly Sand Model

The forward petrophysical model which we use is the Yin-Marion shaly sand model [*Marion*, 1990; *Yin et al.*, 1993; *Avseth et al.*, 2005]. In this model two distinct domains are defined for sand-shale mixtures: sandstones with a secondary shale component called *shaly sands*, and shales with secondary sand component called *sandy shales*. In the former domain, clay particles are assumed to be within the pore space of a sandstone frame. Increasing shale content fills this pore space, decreasing porosity linearly. Thus, in this case the porosity varies according to

$$\phi = \phi_s - C\left(1 - \phi_{\text{sh}}\right), \ \forall\, C < \phi_s \tag{B1}$$

where C is the shale volume fraction, ϕ_s is porosity of the clean sandstone frame and ϕ_{sh} is the intrinsic porosity of the shale. In the sandy-shale domain, the shale volume fraction is greater than the porosity of the clean sandstone frame. In this case the rock is no longer considered to consist of a sandstone frame with a pore space but instead is considered to be shale with sand inclusions. There is no sandstone porosity, only isolated grains, and the only porosity which exists is within the intrinsic pore space of the shale. The total porosity is then

$$\phi = C\phi_{\text{sh}}, \ \forall\, C \geq \phi_s. \tag{B2}$$

The volume fractions of the components (i.e., shale, sand, and pore fluid) predicted by these equations can then be treated in a number of different ways to predict the *S* wave impedance (d_1) and *P* wave impedance (d_2) of the bulk rock. To do this, we chose to use the upper Hashin-Shtrikman bound for the mixture in the shaly sand case and the lower bound in the sandy-shale case (following *Avseth et al.* [2005]) to approximately simulate the two different assumed microgeometries of the domains (see *Mavko et al.* [2009] for an explanation of the microgeometry implied by these bounds). The densities can be calculated with the volume fractions and the known densities of the constituents.

We assumed a constant mineralogy of the shale and sand components in this model. However, we assumed that the pore fluid consisted of a water and a gas phase so a third model parameter is introduced: the water saturation, $S_{\text{wt}} \in [0, 1]$. The elastic moduli and densities of the shale, sand, and pore fluid could be taken from examples in the literature [e.g., *Mavko et al.*, 2009]. Note that the intrinsic porosity of shale is kept constant, so in total only three model parameters could vary, and we write the rock-physical parameter vector, at a point (cell) in the subsurface as $\mathbf{m}_i = [C, \phi_s, S_{\text{wt}}]$.

Using the parameter vector, we symbolically write the Yin-Marion shaly sand model (at a single cell) described above as $\mathbf{d}_i = \mathbf{f}(\mathbf{m}_i)$. This is a deterministic relationship, but we included a random element by adding random Gaussian noise ($\mathbf{e}$) to its output. Thus, the full forward model is written

$$\mathbf{d}_i = \mathbf{f}(\mathbf{m}_i) + \mathbf{e}, \quad \mathbf{e} \sim \phi\left(\mathbf{e}; \mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{d}}\right), \quad \boldsymbol{\Sigma}_{\mathbf{d}} = \begin{bmatrix} \sigma_P^2 & 0 \\ 0 & \sigma_S^2 \end{bmatrix} \tag{B3}$$

where $\mathbf{f}(\mathbf{m}_i)$ represents the Yin-Marion shaly sand model, $\phi()$ is a Gaussian function, $\mathbf{m}_i = [C, \phi_s, S_{\text{wt}}]$ is the vector of model parameters, and $\mathbf{d}_i = \left[d_1, d_2\right]$ is the data vector of impedances. The random Gaussian noise is specified by the standard deviations in the data covariance matrix, $\boldsymbol{\Sigma}_{\mathbf{d}}$: $\sigma_P = 1.5 \times 10^4\ s^{-1}\ \text{m}^{-2}$ kg and $\sigma_S = 1.0 \times 10^4\ s^{-1}\ \text{m}^{-2}$ kg. Thus, the probabilistic forward relation can be written

$$p(\mathbf{d}_i|\mathbf{m}_i) = \frac{1}{\sqrt{(2\pi)^2|\boldsymbol{\Sigma}_{\mathbf{d}}|}} \exp\left(-\left(\mathbf{d}_i - \mathbf{f}(\mathbf{m}_i)\right)^T \boldsymbol{\Sigma}_{\mathbf{d}}{}^{-1} \left(\mathbf{d}_i - \mathbf{f}(\mathbf{m}_i)\right)\right) \tag{B4}$$

where $\mathbf{f}(\mathbf{m}_i)$ represents the Yin-Marion shaly sand model, $\mathbf{m}_i = \left[m_1, m_2\right]$ is the vector of model parameters, and $\mathbf{d}_i = \left[d_1, d_2\right]$ is the data vector of impedances at any point; $\boldsymbol{\Sigma}_{\mathbf{d}}$ is a diagonal covariance matrix describing the (uncorrelated) random noise applied to the data.

However, this distribution only permits the generation of a set of impedances ($\mathbf{d}_i$) once the rock-physical parameters are specified. As implied by equation (B4) the conditional distribution $p(\mathbf{m}_i|G_i)$ describes the probabilistic (i.e., uncertain) relation between lithology-fluid class and rock-physical parameters. We described these relationships using simple bounds [*lower*, *upper*] on the possible values of each parameter. Thus, each parameter was assigned bounds for each lithology-fluid class (see Table A1). The probability distribution of the rock-physical parameters within these bounds was Uniform.

Acknowledgment
We would like to thank TOTAL E&P UK for supporting this work.

References

Arnesen, P. (2010), Approximate recursive calculations of discrete Markov random fields, PhD thesis, Norwegian University of Science and Technology.
Avseth, P., T. Mukerji, and G. Mavko (2005), *Quantitative Seismic Interpretation*, vol. 1, Cambridge Univ. Press, Cambridge, U. K.
Bartolucci, F., and J. Besag (2002), A recursive algorithm for Markov random fields, *Biometrika*, *89*(3), 724–730.
Baum, L. E., T. Petrie, G. Soules, and N. Weiss (1970), A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, *Annals of Math. Stat.*, *41*(1), 164–171.
Belisle, C. (1998), Slow convergence of the Gibbs sampler, *Can. J. Stat.*, *26*(4), 629–641.
Besag, J. (1974), Spatial interaction and the statistical analysis of lattice systems, *J. R. Stat. Soc. Ser. B*, *36*, 192–236.
Besag, J., and P. J. Green (1993), Spatial statistics and Bayesian computation, *J. R. Stat. Soc. Ser. B*, *55*, 25–37.
Biegler, L., G. Biros, O. Ghattas, M. Heinkenschloss, D. Keyes, B. Mallick, L. Tenorio, B. van Bloemen Waanders, K. Willcox, and Y. Marzouk (2011), *Large-Scale Inverse Problems and Quantification of Uncertainty*, vol. 712, John Wiley, Chichester, U. K.
Bosch, M., T. Mukerji, and E. F. Gonzalez (2010), Seismic inversion for reservoir properties combining statistical rock physics and geostatistics: A review, *Geophysics*, *75*(5), 75A165–75A176.
Brook, D. (1964), On the distinction between the conditional probability and the joint probability approaches in the specification of nearest-neighbor systems, *Biometrika*, *51*(3/4), 481–483.
Brooks, S. P., and A. Gelman (1998), General methods for monitoring convergence of iterative simulations, *J. Comput. Graph. Stat.*, *7*(4), 434–455.
Caers, J. (2001), Geostatistical reservoir modelling using statistical pattern recognition, *J. Pet. Sci. Eng.*, *29*(3), 177–188.
Caers, J. (2005), *Petroleum Geostatistics*, Society of Petroleum Engineers, Richardson, TX.
Chen, L., Z. Qin, and J. S. Liu (2001), Exploring hybrid Monte Carlo in Bayesian computation, *Sigma*, *2*, 2–5.
Datcu, M., K. Seidel, and M. Walessa (1998), Spatial information retrieval from remote-sensing images. I. Information theoretical perspective, *IEEE Trans. Geosci. Remote Sens.*, *36*(5), 1431–1445.
Eidsvik, J., H. Omre, T. Mukerji, G. Mavko, P. Avseth, and N. Hydro (2002), Seismic reservoir prediction using Bayesian integration of rock physics and Markov random fields: A north sea example, *Leading Edge*, *21*(3), 290–294.
Friel, N., and H. Rue (2007), Recursive computing and simulation-free inference for general factorizable models, *Biometrika*, *94*(3), 661–672.
Friel, N., A. Pettitt, R. Reeves, and E. Wit (2009), Bayesian inference in hidden Markov random fields for binary data defined on large lattices, *J. Comput. Graph. Stat.*, *18*(2), 243–261.
Geman, S., and D. Geman (1993), Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, *J. Appl. Stat.*, *20*(5-6), 25–62.
George, E. I., U. Makov, and A. Smith (1993), Conjugate likelihood distributions, *Scand. J. Stat.*, *20*, 147–156.
Gilks, W. R., S. Richardson, and D. J. Spiegelhalter (1996), *Markov Chain Monte Carlo in Practice*, vol. 2, CRC Press, Boca Raton, Fla.
Haario, H., E. Saksman, and J. Tamminen (1999), Adaptive proposal distribution for random walk Metropolis algorithm, *Comput. Stat.*, *14*(3), 375–396.
Hastings, W. K. (1970), Monte Carlo sampling methods using Markov chains and their applications, *Biometrika*, *57*(1), 97–109.
Journel, A., R. Gundeso, E. Gringarten, and T. Yao (1998), Stochastic modelling of a fluvial reservoir: A comparative review of algorithms, *J. Pet. Sci. Eng.*, *21*(1), 95–121.
Kass, R. E., B. P. Carlin, A. Gelman, and R. M. Neal (1998), Markov chain Monte Carlo in practice: A roundtable discussion, *Am. Stat.*, *52*(2), 93–100.
Kirkpatrick, S., C. D. Gelatt Jr., and M. P. Vecchi (1983), Optimization by simulated annealing, *Science*, *220*(4598), 671–680.
Marion, D. P. (1990), Acoustical, mechanical, and transport properties of sediments and granular materials, PhD thesis, Stanford University, Department of Geophysics.
Mavko, G., T. Mukerji, and J. Dvorkin (2009), *The Rock Physics Handbook: Tools for Seismic Analysis of Porous Media*, Cambridge Univ. Press, Cambridge, U. K.
Meier, U., A. Curtis, and J. Trampert (2007), Fully nonlinear inversion of fundamental mode surface waves for a global crustal model, *Geophys. Res. Lett.*, *34*, L16304, doi:10.1029/2007GL030989.
Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953), Equation of state calculations by fast computing machines, *J. Chem. Phys.*, *21*, 1087.
Mosegaard, K., and M. Sambridge (2002), Monte Carlo analysis of inverse problems, *Inverse Prob.*, *18*(3), R29.
Mukerji, T., A. Jørstad, P. Avseth, G. Mavko, and J. Granli (2001), Mapping lithofacies and pore-fluid probabilities in a north sea reservoir: Seismic inversions and statistical rock physics, *Geophysics*, *66*(4), 988–1001.
Olea, R. (1999), *Geostatistics for Engineers and Earth Scientists*, Kluwer Academic Publishers, Boston, Mass.
Remy, N., A. Boucher, and J. Wu (2009), *Applied Geostatistics With SGeMS: A User's Guide*, Cambridge Univ. Press, Cambridge, U. K.
Rimstad, K., and H. Omre (2010), Impact of rock physics depth trends and Markov random fields on hierarchical Bayesian lithology fluid prediction, *Geophysics*, *75*(4), R93–R108.
Robert, C. P., and G. Casella (2004), *Monte Carlo Statistical Methods*, vol. 319, Springer, New York.
Rue, H., and L. Held (2005), *Gaussian Markov random fields: Theory and applications*, vol. 104, CRC Press/Chapman & Hall, Boca Raton, Fla.
Saul, L. K., and S. T. Roweis (2003), Think globally, fit locally: Unsupervised learning of low dimensional manifolds, *J. Mach. Learn. Res.*, *4*, 119–155.
Scales, J. A., and L. Tenorio (2001), Prior information and uncertainty in inverse problems, *Geophysics*, *66*(2), 389–397.
Scott, S. L. (2002), Bayesian methods for hidden Markov models, *J. Am. Stat. Assoc.*, *97*(457), 337–351.
Shahraeeni, M. S., A. Curtis, and G. Chao (2012), Fast probabilistic petrophysical mapping of reservoirs from 3D seismic data, *Geophysics*, *77*(3), O1–O19.
Strebelle, S. (2002), Conditional simulation of complex geological structures using multiple-point statistics, *Math. Geol.*, *34*(1), 1–21.
Tarantola, A. (2002), *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation*, Elsevier Science, Amsterdam, Netherlands.
Tjelmeland, H., and H. M. Austad (2012), Exact and approximate recursive calculations for binary Markov random fields defined on graphs, *J. Comput. Graph. Stat.*, *21*(3), 758–780.
Ulvmoen, M., and H. Hammer (2010), Bayesian lithology fluid inversion comparison of two algorithms, *Comput. Geosci.*, *14*(2), 357–367.
Van Dyk, D. A., and T. Park (2008), Partially collapsed Gibbs samplers: Theory and methods, *J. Am. Stat. Assoc.*, *103*(482), 790–796.
Varma, M., and A. Zisserman (2003), Texture classification: Are filter banks necessary? in *Proceedings. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003*, vol. 2, pp. II–691, IEEE, New York.

Walker, M., and A. Curtis (2014), Varying prior information in Bayesian inversion, *Inverse Prob.*, *30*, 065002, doi:10.1088/0266-5611/30/6/065002.

Yin, H., A. Nur, and G. Mavko (1993), Critical porosity a physical boundary in poroelasticity, *Int. J. Rock Mech. Min. Sci. Geomech. Abstr.*, *30*(7), 805–808.

Zhang, T., D. Lu, and D. Li (2008), A statistical information reconstruction method of images based on multiple-point geostatistics integrating soft data with hard data, in *ISCSCT'08. International Symposium on Computer Science and Computational Technology, 2008*, vol. 1, pp. 573–578, IEEE, New York.