

16

Posterior Exploration for Computationally Intensive Forward Models

David Higdon, C. Shane Reese, J. David Moulton, Jasper A. Vrugt, and Colin Fox

16.1 Introduction

In a common inverse problem, we wish to infer about an unknown spatial field $x = (x_1, \dots, x_m)^T$, given indirect observations $y = (y_1, \dots, y_n)^T$. The observations, or data, are linked to the unknown field x through some physical system

$$y = \zeta(x) + \epsilon,$$

where $\zeta(x)$ denotes the physical system and ϵ is an n -vector of observation errors. Examples of such problems include medical imaging (Kaipio and Somersalo, 2004), geologic and hydrologic inversion (Stenerud et al., 2008), and cosmology (Jimenez et al. 2004). When a forward model, or simulator, of the physical process $\eta(x)$ is available, one can model the data using the simulator

$$y = \eta(x) + e,$$

where e includes observation error as well as error due to the fact that the simulator $\eta(x)$ may be systematically different from reality $\zeta(x)$ for input condition x . Our goal is to use the observed data y to make inference about the spatial input parameters x —predict x and characterize the uncertainty in the prediction for x .

The likelihood $L(y|x)$ is then specified to account for both mismatch and sampling error. We will assume zero-mean Gaussian errors so that

$$L(y|x) \propto \exp \left\{ -\frac{1}{2} (y - \eta(x))^T \Sigma_e^{-1} (y - \eta(x)) \right\}, \quad (16.1)$$

with Σ_e known. It is worth noting that the data often come from only a single experiment. So while it is possible to quantify numerical errors, such as those due to discretization (see Kaipio and Somersalo, 2004; Nissinen et al., 2008), there is no opportunity to obtain data from additional experiments for which some controllable inputs have been varied. Because of this limitation, there is little hope of determining the sources of error in e due to model inadequacy. Therefore, the likelihood specification will often need to be done with some care, incorporating the modeler's judgment about the appropriate size and nature of the mismatch term.

In many inverse problems we wish to reconstruct x , an unknown process over a regular two-dimensional lattice. We consider systems for which the model input parameters x denote a spatial field or image. The spatial prior is specified for x , $\pi(x)$, which typically takes into account modeling, and possibly computational considerations.

The resulting posterior is then given by

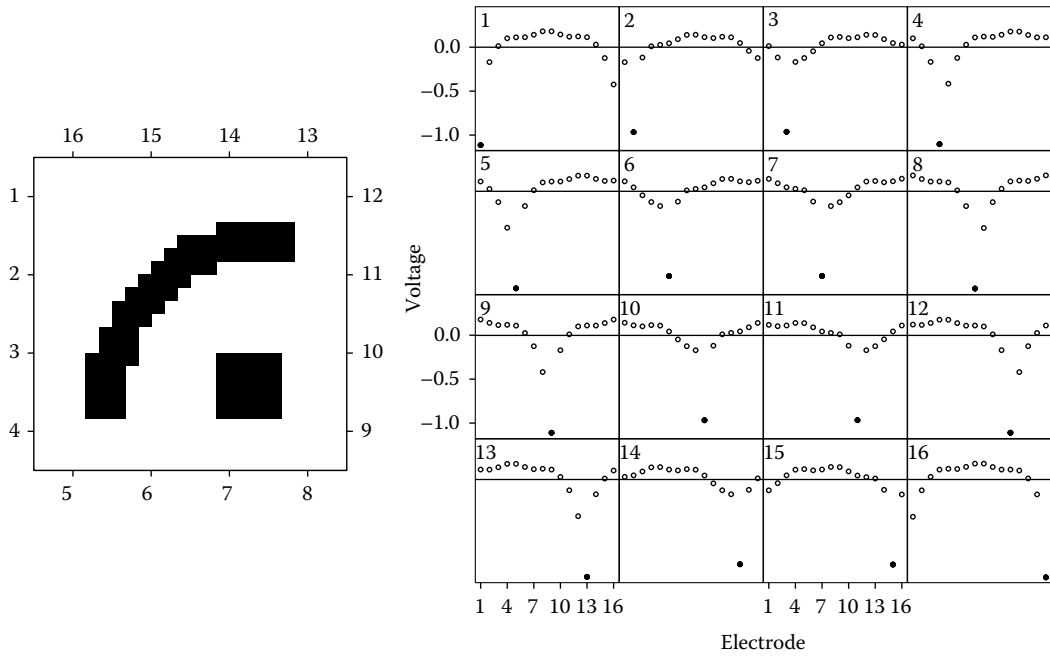
$$\pi(x|y) \propto L(y|\eta(x)) \times \pi(x).$$

This posterior can, in principle, be explored via Markov chain Monte Carlo (MCMC). However the combined effects of the high dimensionality of x and the computational demands of the simulator make implementation difficult, and often impossible, in practice. By itself, the high dimensionality of x is not necessarily a problem. MCMC has been carried out with relative ease in large image applications (Rue, 2001; Weir, 1997). However, in these examples, the forward model was either trivial or non-existent. Unfortunately, even a mildly demanding forward simulation model can greatly affect the feasibility of doing MCMC to solve the inverse problem.

In this chapter we apply a standard single-site updating scheme that dates back to Metropolis et al. (1953) to sample from this posterior. While this approach has proven effective in a variety of applications, it has the drawback of requiring hundreds of thousands of calls to the simulation model. In Section 16.3 we consider two MCMC schemes that use highly multivariate updates to sample from $\pi(x|y)$: the multivariate random-walk Metropolis algorithm (Gelman et al., 1996) and the *differential evolution* MCMC (DE-MCMC) sampler of ter Braak (2006). Such multivariate updating schemes are alluring for computationally demanding inverse problems since they have the potential to update many (or all) components of x at once, while requiring only a single evaluation of the simulator. Next, in Section 16.4, we consider augmenting the basic posterior formulation with additional formulations based on faster, approximate simulators. The faster, approximate simulators are created by altering the multigrid solver used to compute $\eta(x)$. These approximate simulators can be used in a delayed acceptance scheme (Christen and Fox, 2005; Fox and Nicholls, 1997), as well as in an augmented formulation (Higdon et al., 2002). Both of these recipes can be utilized with any of the above MCMC schemes, often leading to substantial improvements in efficiency. In each section we illustrate the updating schemes with an electrical impedance tomography (EIT) application described in the next section, where the values of x denote electrical conductivity of a two-dimensional object. The chapter concludes with a discussion and some general recommendations.

16.2 An Inverse Problem in Electrical Impedance Tomography

Bayesian methods for EIT applications have been described in Fox and Nicholls (1997), Kaipio et al. (2000), and Andersen et al. (2003). A notional inverse problem is depicted in Figure 16.1; this setup was given previously in Moulton et al. (2008). Here a two-dimensional object composed of regions with differing electrical conductivity is interrogated by 16 electrodes. From each electrode, in turn, a current I is injected into the object and taken out at a rate of $I/(16 - 1)$ at the remaining 15 electrodes. The voltage is then measured at each of the 16 electrodes. These 16 experimental configurations result in $n = 16 \times 16$ voltage observations which are denoted by the n -vector y . The measurement error is simulated by adding independent and identically distributed mean-zero Gaussian noise to each of the voltage


FIGURE 16.1

A synthetic EIT application. A two-dimensional object is surrounded by electrodes at 16 evenly spaced locations around its edge. The conductivity of the object is 3 in the white regions, and 4 in the black regions (the units are arbitrary since the data are invariant to scalings of the conductivity). First, a current of I is injected at electrode 1, and extracted evenly at the other 15 electrodes. The voltage is measured at each electrode. This data is shown in the plot labeled 1 on the right. Similar experiments are carried out with each electrode taking a turn as the injector. The resulting voltages are shown in the remaining 15 plots. In each plot, the voltage corresponding to the injector electrode is given by a black plotting symbol.

measurements. The standard deviation σ of this noise is chosen so that the signal to noise ratio is about 1000 : 3, which is typical of actual EIT measurements. The resulting simulated data is shown on the right in Figure 16.1—one plot for each of the 16 circuit configurations. In each of those plots, the injector electrode is denoted by the black plotting symbol.

We take s to denote spatial locations within the object $\Omega = [0, 1] \times [0, 1]$, and take $x(s)$ to denote the electrical conductivity at site s . We also take $v(s)$ to be the potential at location s , and $j(s)$ to be the current at boundary location s . A mathematical model for the measurements is then the Neumann boundary-value problem

$$\begin{aligned} -\nabla \cdot x(s) \nabla v(s) &= 0, \quad s \in \Omega, \\ x(s) \frac{\partial v(s)}{\partial n(s)} &= j(s), \quad s \in \partial\Omega, \end{aligned}$$

where $\partial\Omega$ denotes the boundary of the object Ω and $n(s)$ is the unit normal vector at the boundary location $s \in \partial\Omega$. The conservation of current requires that the sum of the currents at each of the 16 electrodes be 0.

In order to numerically solve this problem for a given set of currents at the electrodes and a given conductivity field, $x(s)$, the conductivity field is discretized into an $m = 24 \times 24$ lattice. We use a robust multigrid solver called Black Box MG (Dendy, 1987). In addition to

being rather general and fast, we can also exploit the multigrid nature of the algorithm to develop fast approximations using the MCMC scheme described in Section 16.4.

Now, for any specified conductivity configuration x and current configuration, the multigrid solver produces 16 voltages. For all 16 current configurations, 16 forward solves produces an $n = 256$ -vector of resulting voltages $\eta(x)$. Hence, the sampling model for the data y given the conductivity field x is given by Equation 16.1, where $\Sigma_e = \sigma^2 I_n$.

For the conductivity image prior, we adapt a Markov random field (MRF) prior from Geman and McClure (1987). This prior has the form

$$\pi(x) \propto \exp \left\{ \beta \sum_{i \sim j} u(x_i - x_j) \right\}, \quad x \in [2.5, 4.5]^m, \quad (16.2)$$

where β and s control the regularity of the field, and $u(\cdot)$ is the tricube function of Cleveland (1979):

$$u(d) = \begin{cases} \frac{1}{s} (1 - [d/s]^3)^3, & \text{if } -s < d < s, \\ 0, & \text{if } |d| \geq s. \end{cases}$$

The sum is over all horizontal and vertical nearest neighbors, denoted by $i \sim j$, and given by the edges in the Markov random field graph in Figure 16.2. Hence, this prior encourages neighboring x_i to have similar values, but once x_i and x_j are more than s apart, the penalty does not grow. This allows occasional large shifts between neighboring x_i . For this chapter, we fix $(\beta, s) = (0.5, 0.3)$. A realization from this prior is shown on the right in Figure 16.2. A typical prior realization shows patches of homogeneous values, along with abrupt changes in intensity at patch boundaries. This prior also allows an occasional, isolated, extreme single pixel value.

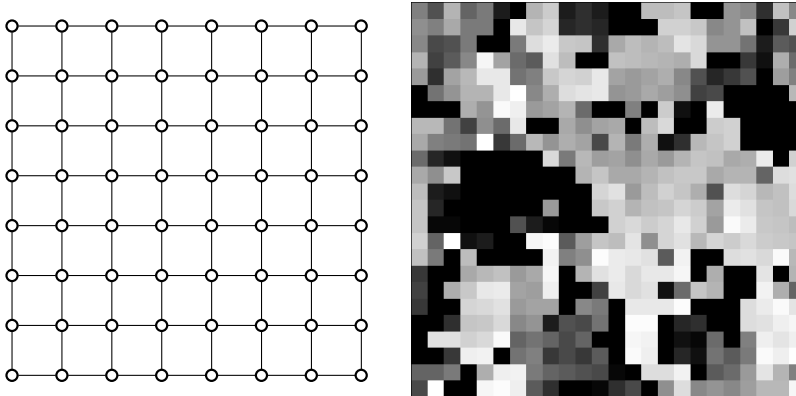


FIGURE 16.2

(Left) First-order neighborhood MRF graph corresponding to the prior in Equation 16.2; each term in the sum corresponds to an edge in the MRF graph. (Right) A realization from this gray level prior.

The resulting posterior density has the form

$$\pi(x|y) \propto \exp \left\{ -\frac{1}{2\sigma^2} (y - \eta(x))^T (y - \eta(x)) \right\} \times \exp \left\{ \beta \sum_{i \sim j} u(x_i - x_j) \right\}, \quad x \in [2.5, 4.5]^m. \quad (16.3)$$

The patchiness and speckle allowed by this prior, and the rather global nature of the likelihood, make posterior exploration for this inverse problem rather challenging, and a good test case for various MCMC schemes that have been developed over the years. We note that the nature of the posterior can be dramatically altered by changing the prior specification for x . This is discussed later in this section.

This chapter considers a number of MCMC approaches for sampling from this posterior distribution. We start at the beginning.

16.2.1 Posterior Exploration via Single-Site Metropolis Updates

A robust and straightforward method for computing samples from the posterior $\pi(x|y)$ is the single-site Metropolis scheme, originally carried out in Metropolis et al. (1953) on the world's first computer with addressable memory, the MANIAC. A common formulation of this scheme is summarized in Algorithm 1 using pseudocode. This scheme is engineered to maintain detailed balance—so that the relative movement between any two states x and x^* is done in proportion to the posterior density at these two points. The width of the proposal distribution σ_z should be adjusted so that inequality in line 5 is satisfied roughly half the time (Gelman et al., 1996), but an acceptance rate between 70% and 30% does nearly as well for single-site updates. After scanning through each of the parameter elements (for loop, steps 3–7), one typically records the current value of x . We do so every 10 scans through the parameter vector.

ALGORITHM 1 SINGLE-SITE METROPOLIS

```

1: initialize  $x$ 
2: for  $k = 1 : \text{niter}$  do
3:   for  $i = 1 : m$  do
4:      $x'_i = x_i + z$ , where  $z \sim N(0, \sigma_z)$ 
5:     if  $u < \frac{\pi(x'|y)}{\pi(x|y)}$ , where  $u \sim U(0, 1)$  then
6:       set  $x_i = x'_i$ 
7:     end if
8:   end for
9: end for
    
```

This single-site scheme was originally intended for distributions with very local dependencies within the elements of x so that the ratio in line 5 simplifies dramatically. In general, this simplification depends on the full conditional density of x_i ,

$$\pi(x_i | x_{-i}, y), \quad \text{where } x_{-i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)^T.$$

This density is determined by keeping all of the product terms in $\pi(x|y)$ that contain x_i , and ignoring the terms that do not. Hence the ratio in line 5 can be rewritten as

$$\frac{\pi(x'|y)}{\pi(x|y)} = \frac{\pi(x'_i|x_{-i}, y)}{\pi(x_i|x_{-i}, y)}.$$

In many cases this ratio becomes trivial to compute. However, in the case of this particular inverse problem, we must still evaluate the simulator to compute this ratio. This is exactly what makes the MCMC computation costly for this problem.

Nonetheless, this straightforward sampling approach does adequately sample the posterior, given sufficient computational effort. Figure 16.3 shows realizations produced by the single-site Metropolis algorithm, separated by 1000 scans through each element of x . Inspection of these realizations makes it clear that posterior realizations yield a crisp distinction between the high- and low-conductivity regions, as was intended by the MRF prior for x . Around the boundary of the high conductivity region, there is a fair amount of uncertainty as to whether or not a given pixel has high or low conductivity.

Figure 16.4 shows the resulting posterior mean for x and the history of three pixel values over the course of the single-site updating scheme. The sampler was run until $40,000 \times m$ forward simulations were carried out. An evenly spaced sample of 6000 values for three of the m pixels is shown on the left in Figure 16.4. Note that for the middle pixel (blue circle), the marginal posterior distribution is bimodal—some realizations have the conductivity value near 3, others near 4. Being able to move between these modes is crucial for a well-mixing chain. Getting this pixel to move between modes is not simply a matter of getting that one pixel to move by itself; the movement of that pixel is accomplished by getting the entire image x to move between local modes of the posterior.

This local multimodality is largely induced by our choice of prior. For example, if we alter the prior model in Equation 16.2 so that

$$u(d) = -d^2, \quad (16.4)$$

we have a standard Gaussian Markov random field (GMRF) prior for x . If, in addition, the simulator is a linear mapping from inputs x to outputs $\eta(x)$, the resulting posterior is necessarily Gaussian, and hence unimodal. While this is not true for nonlinear forward models/simulators, the GMRF prior still has substantial influence on the nature of the posterior. Figure 16.5 shows two realizations and the posterior mean resulting from such a prior with $\beta = 2$. Here posterior realizations are locally more variable—the difference between neighboring pixels is generally larger. However, the global nature of the posterior realizations is far more controlled than those in Figure 16.3 since the GMRF prior suppresses

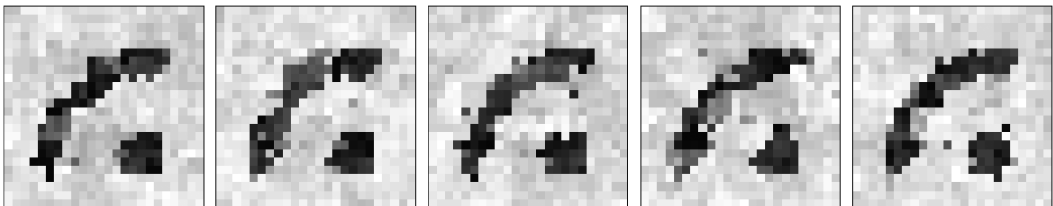
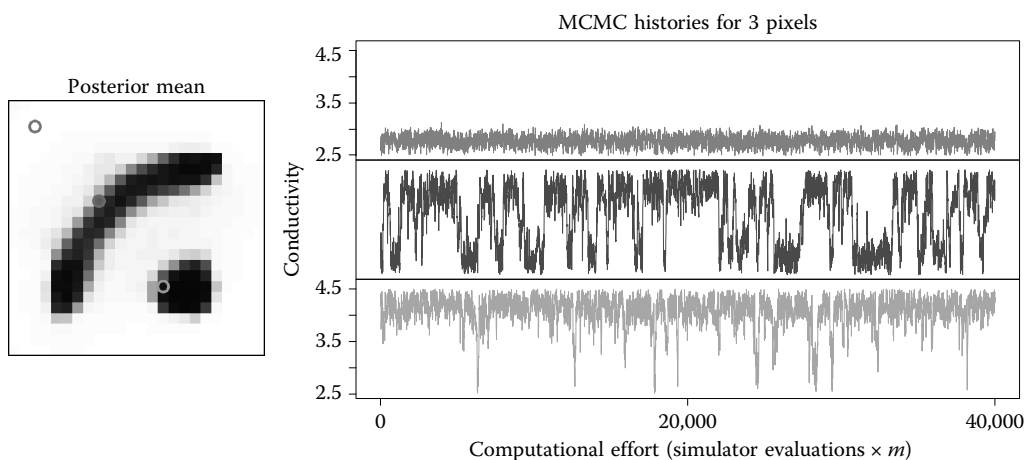


FIGURE 16.3

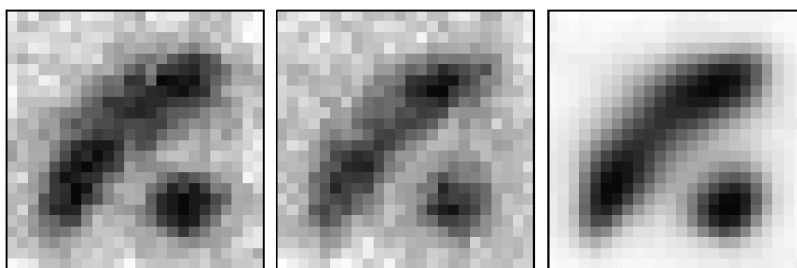
Five realizations from the single-site Metropolis scheme. Realizations are separated by 1000 scans through the m -dimensional image parameter x .


FIGURE 16.4

Posterior mean image for x and MCMC traces of three pixels: one which is predominantly light (small conductivity); one which is predominantly dark (high conductivity) and one which is on the edge of the object. This MCMC run carries out $40,000 \times m$ forward simulator evaluations. The value of x_i is given every 10th iteration (i.e. every $10 \times m$ single-site updates).

local modes that appear under the previous formulation. This resulting formulation is also far easier to sample, requiring about one tenth of the effort needed for formulation in Equation 16.3. An alternate, controlling prior formulation uses a process convolution prior for x is given in the Appendix to this chapter. In addition to yielding a more easily sampled posterior, the prior also represents the image x with far fewer parameters than the m used in the MRF specifications.

While these alternative specifications lead to simpler posterior distributions, they do so while giving overly smooth posterior realizations. Still, such realizations may be useful for exploratory purposes, and for initializing other samplers; we do not further pursue such formulations here. Instead, we focus on comparison of various MCMC schemes to sample the original gray level posterior in Equation 16.3. We use the sample traces from the three pixels circled in Figure 16.4 to make comparisons between a variety of samplers which are discussed in the next sections—the movement of these three pixels is representative of all the image pixels. In particular, we focus on the frequency of movement between high and low conductivity at these sites.


FIGURE 16.5

Two realizations and the posterior mean from the single-site Metropolis scheme run on the posterior resulting from the GMRF prior. Realizations are separated by 1000 scans through the m -dimensional image x .

16.3 Multivariate Updating Schemes

Schemes that propose to update more than just a single component of x at a time have the potential to reduce the computational burden of producing an MCMC sample from $\pi(x|y)$. The single-site scheme above is also applicable when the proposal for x' changes some or all of the components of x . However, producing a multivariate candidate x' that has an appreciable chance of being accepted (i.e. satisfying the inequality in line 5 of Algorithm 1) while allowing appreciable movement, is very difficult. This highlights a very appealing aspect of the single-site Metropolis scheme: even fairly thoughtless one-dimensional proposals have an appreciable chance of being accepted while adequately exploring the posterior.

There are clustering MCMC algorithms from statistical physics that allow for many pixels in x to be updated at once (Edwards and Sokal, 1988). Such methods can be adapted to this particular problem as in Higdon (1998); however, such methods typically show decreased efficiency relative to single-site updating when the likelihood is strong relative to the prior. This is certainly the case with our attempts on this application whose results are not worth discussing here. Instead, we look to multivariate random-walk Metropolis updating and the DE-MCMC scheme of ter Braak (2006) as competitors to the costly single-site Metropolis updating for our EIT application.

16.3.1 Random-Walk Metropolis

The multivariate random-walk Metropolis scheme (RWM) has been the focus of a number of theoretical investigations (Gelman et al., 1996; Tierney, 1994). But to date this scheme has not been widely used in applications, and has proven advantageous only in simple, unimodal settings. The preference for single-site, or limited multivariate updates in practice may be attributed to how the full conditionals often simplify computation, or may be due to the difficulty in tuning highly multivariate proposals. In our EIT application, the univariate full conditionals do not lead to any computational advantages. If there is ever an application for which RWM may be preferable, this is it. Single-site updating is very costly, and may be inefficient relative to multivariate updating schemes for this multimodal posterior.

A multivariate Gaussian RMW scheme for the m -vector x is summarized in Algorithm 2 using pseudocode.

ALGORITHM 2 RANDOM-WALK METROPOLIS

```

1: initialize  $x$ 
2: for  $k = 1 : \text{niter}$  do
3:    $x' = x + z$ , where  $z \sim N_m(0, \Sigma_z)$ 
4:   if  $u < \frac{\pi(x'|y)}{\pi(x|y)}$ , where  $u \sim U(0, 1)$  then
5:     set  $x = x'$ 
6:   end if
7: end for

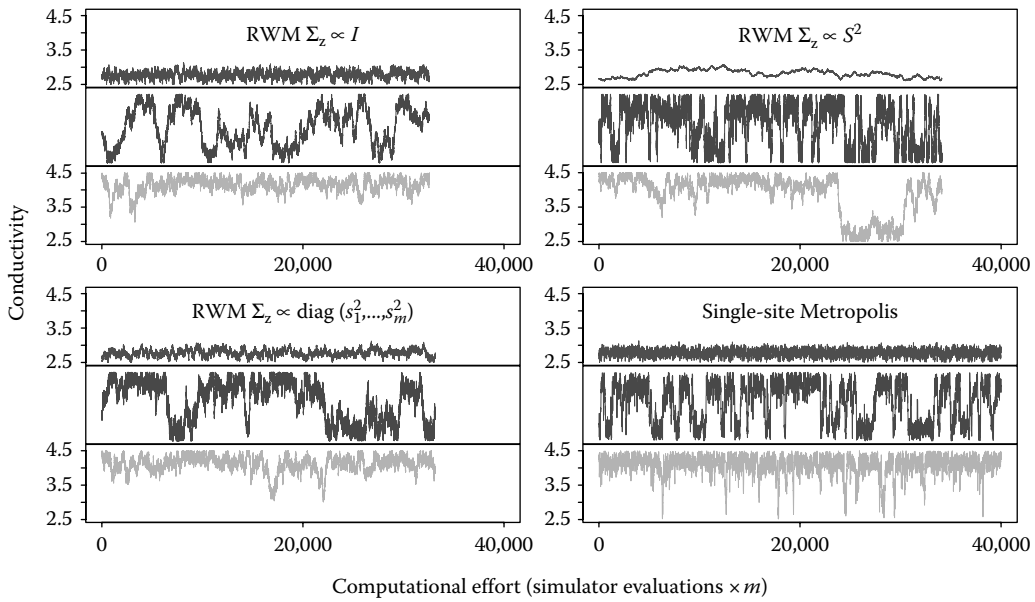
```

We consider three different proposals for this scheme:

$$\Sigma_z \propto \Sigma_1 = I_m,$$

$$\Sigma_z \propto \Sigma_2 = \text{diag}(s_1^2, \dots, s_m^2),$$

$$\Sigma_z \propto \Sigma_3 = S^2,$$


FIGURE 16.6

MCMC traces of three pixels circled in Figure 16.4 under three multivariate random-walk Metropolis schemes, and single-site Metropolis. For each run, $40,000 \times m$ forward simulator evaluations are carried out. While the RWM scheme with $\Sigma_z \propto S^2$ results in good movement for the central pixel, the movement of the top and bottom pixels is clearly inferior to that of single-site Metropolis.

where s_i^2 is the posterior marginal sample variance for the conductivity x_i , and S^2 is the $m \times m$ sample covariance matrix—both estimated from the previously obtained single-site MCMC run. In each case we set $\Sigma_z = \alpha_i \Sigma_i$, where the scalar α_i is chosen so that the candidate x' is accepted 30% of the time, which is close to optimal in a Gaussian setting.

MCMC traces for these three implementations of RWM are shown in Figure 16.6. The traces from the single-site Metropolis scheme are also given for comparison. Interestingly, the behavior of the traces varies with the choice of Σ_z . The scheme with $\Sigma_z \propto S^2$ shows the most movement for the central pixel, which moves between high and low conductivity over the run. However, its performance for the top, low-conductivity pixel is noticeably worse. None of the RWM schemes do as well as single-site Metropolis when looking at the bottom, high conductivity pixel. These results suggest that a scheme that utilizes both single-site and RWM updates with $\Sigma_z \propto S^2$ might give slightly better posterior exploration than single-site Metropolis alone.

16.3.2 Differential Evolution and Variants

In ter Braak's DE-MCMC algorithm, a collection of independent chains $\{x^1, \dots, x^P\}$ are constructed, each sampling from the posterior. Chain x^p is updated according to a multivariate Metropolis step where the candidate is a perturbation of x^p based on the difference between two randomly chosen chains in the collection. This DE-MCMC scheme is described in Algorithm 3. In the original implementation, σ^2 is chosen to be quite small so that a proposal of the form $x' = x + e$ would nearly always be accepted; γ is chosen so that the proposal is accepted about 30% of the time. Hence, it is the $\gamma(x^q - x')$ part of the proposal that accounts for nearly all of the movement from the current location x .

ALGORITHM 3 DIFFERENTIAL EVOLUTION

```

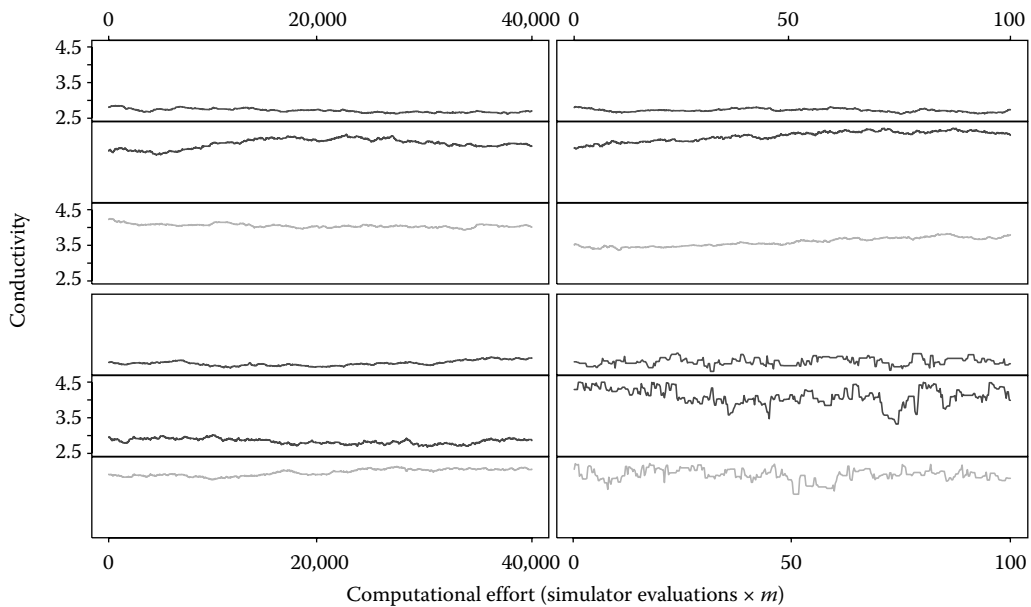
1: initialize  $P$  copies  $\{x^1, \dots, x^P\}$ 
2: for  $k = 1 : \text{niter}$  do
3:   for  $p = 1 : P$  do
4:     chose indices  $q$  and  $r$  without replacement from  $\{1, \dots, p-1, p+1, \dots, P\}$ 
5:      $x^{p'} = x^p + \gamma(x^q - x^r) + e$ , where  $e \sim N(0, \sigma^2 I_n)$ 
6:     if  $u < \frac{\pi(x^{p'}|y)}{\pi(x^p|y)}$ , where  $u \sim U(0, 1)$  then
7:        $x^p = x^{p'}$ 
8:     end if
9:   end for
10: end for

```

One interpretation of the DE-MCMC algorithm is as an empirical version of the RWM algorithm. The proposal distribution is a scaled difference between random draws from $\pi(x|y)$; the dependence between the parallel chains means that these draws are not independent. Theoretical considerations make $2.38/\sqrt{m}$ a useful starting choice for γ (Gelman et al., 1996). However, some tuning of γ is usually appropriate. An obvious appeal of this DE scheme is that it avoids the difficult task of determining the appropriate Σ_z used in the Gaussian RWM implementation from earlier in this section. By carrying P copies of the chain, fruitful multivariate candidates can be generated on the fly. Such schemes have proven useful in difficult, low-dimensional posterior distributions, but the utility of such an approach has yet to be demonstrated on highly multivariate posteriors resulting from applications such as this.

As a first step in illustrating DE-MCMC on the EIT application, we initialized the $P = 400$ chains by taking equally spaced realizations from the first $6000 \times m$ iterations from the single-site Metropolis scheme described earlier. Then each of the 400 chains were updated in turn according to the DE-MCMC algorithm. The sampler continued until $40,000 \times m$ simulator evaluations were carried out. Thus each chain was updated $100 \times m$ times. The resulting MCMC traces for the three representative pixels are shown in Figure 16.7 for three of the 400 chains used in our implementation. For comparison, the trace from $100 \times m$ single-site Metropolis is also given on the bottom right of the figure. Also, the mean and (marginal) standard deviation for the central pixel (marked by the blue circle in Figure 16.4) are shown in Figure 16.8 for each of the 400 chains. Within a given chain, the pixels show very little movement; the final value of the 400 chains is not far from the starting point, as is clear from Figure 16.8.

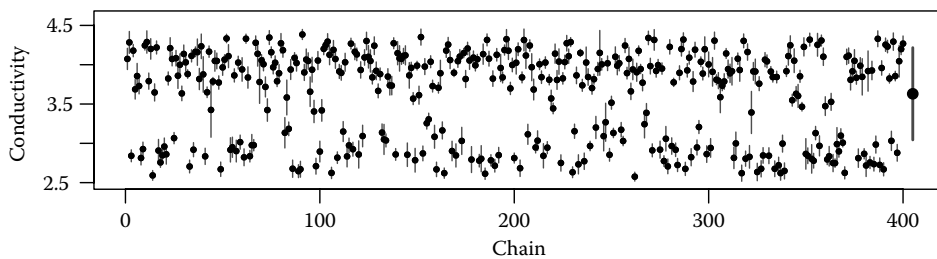
We also consider an alteration to the basic formulation of ter Braak in which the scalar γ is drawn from a $U(-a, a)$ distribution. We set $a = 0.02$ so that the proposal is accepted about 30% of the time. For this alteration, we set $e = 0$ since these small steps had very little impact on the sampler. While this alteration leads to noticeably better movement than our standard DE-MCMC implementation, the movement of this chain is still clearly inferior to single-site Metropolis. Given the less than stellar performance of the multivariate RWM scheme, the lack of success here is not a big surprise since both schemes make use of highly multivariate updates based on $\pi(x|y)$. The larger surprise is that the general failure of these multivariate updating schemes to provide any improvement over single-site Metropolis updating, even when there are no computational savings to be had by considering univariate full conditionals. We note that the poor performance of these multivariate updating schemes does not preclude the existence of some modification that will eventually prove beneficial for this application; we simply did not find one.


FIGURE 16.7

MCMC traces of the same three pixels shown in Figure 16.4 for three of the 400 chains used in the DE-MCMC scheme. The algorithm ran until $40,000 \times m$ forward simulator evaluations so that the computational effort matches the other MCMC schemes. For each of the 400 chains, $100 \times m$ updates are carried out. The bottom right plot shows movement from $100 \times m$ single-site Metropolis iterations for comparison. The resulting movement is clearly inferior to that of the standard single-site scheme when normalized by computational effort.

16.4 Augmenting with Fast, Approximate Simulators

In many applications, a faster, approximate simulator is available for improving the MCMC sampling. There are a limited number of rigorous approaches for utilizing fast, approximate simulators: delayed acceptance schemes that limit the number of calls to the expensive,


FIGURE 16.8

The marginal posterior mean and lines extending ± 1 standard deviation for the central pixel circled in Figure 16.4. The mean and standard deviation is estimated separately from each of the 400 copies in the DE-MCMC scheme. The chains were initialized from evenly spaced realizations taken from the single-site Metropolis scheme. The spread of the means relative to the estimated standard deviation indicates a very poor mixing, or posterior movement for the DE-MCMC scheme. For comparison, the marginal mean ± 1 standard deviation of single-site Metropolis is shown at the rightmost edge of the plot.

exact simulator (Christen and Fox, 2005; Efendiev et al., 2006; Fox and Nicholls, 1997); and augmented or tempering schemes (Higdon et al., 2002, 2003). For this chapter, we consider simple implementations of both of these approaches and discuss their implementation in context of the EIT application.

For the multigrid EIT simulator $\eta(x)$, an approximate simulator can be created by altering how the multigrid steps are carried out during the solve. Specifically, multigrid algorithms achieve their efficiency through the recursive use of a local smoothing iteration and successively coarser discrete problems; see (Briggs et al., 2000) for an overview of multigrid principles. The most common multigrid cycle, the V-cycle, is shown in Figure 16.9. Here a local smoothing iteration, such as a Gauss–Seidel relaxation, smooths the error of the current iterate. In turn, the smoothed error is represented on a coarser grid through the restriction (weighting) of the current residual. The coarser grid provides a means to find an inexpensive correction to the current iterate; however, this grid may still be too large for a direct solve. In this case the process is repeated until the coarsest grid is reached and a direct solve may be performed. The correction is then interpolated and smoothed, repeatedly, until the finest grid is reached. If a single smoothing iteration is applied at each grid level of the coarsening and refining phases, then the multigrid cycle is denoted as $V(1, 1)$.

The complementarity of the smoothing and coarse-grid correction processes leads to multigrid’s optimal algorithmic scaling (i.e. solution cost grows only linearly with number of unknowns), and to a uniform reduction in the error with each cycle. It is this latter property that creates the opportunity to develop efficient approximate solvers using elements of robust variational multigrid algorithms. For example, MacLachlan and Moulton (2006) developed the Multilevel Upscaling (MLUPS) algorithm to efficiently model flow through highly heterogeneous porous media. MLUPS leveraged the hierarchy of discrete operators provided by the operator-induced variational coarsening of the Black Box Multigrid (BoxMG) algorithm (see Dendy, 1982), and eliminated the smoothing iterations from the finest few levels.

In this work, we produce approximate solvers by limiting the number of $V(1, 1)$ cycles carried out. Starting with a fixed initial solution, the approximate solvers will produce

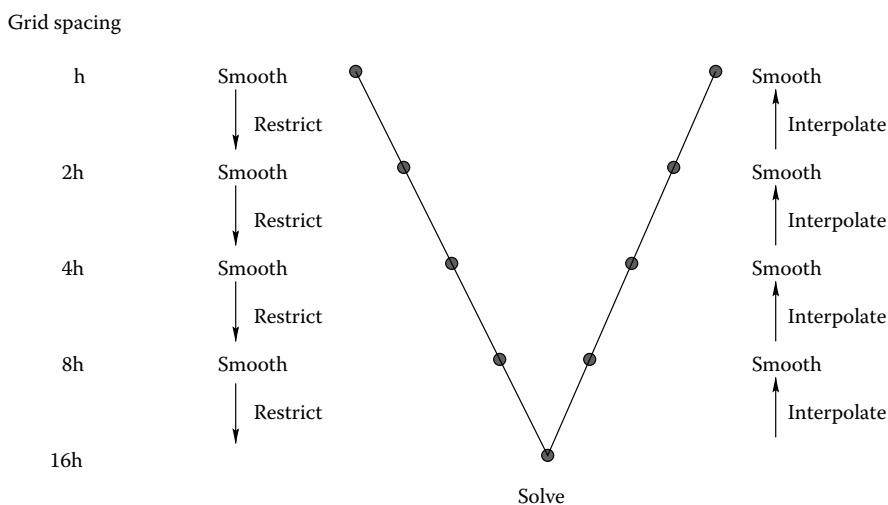


FIGURE 16.9

Schematic of the V-cycle multigrid iterative algorithm.

ALGORITHM 4 DIFFERENTIAL ACCEPTANCE METROPOLOIS

```

1: initialize  $x$ 
2: for  $k = 1 : \text{niter}$  do
3:   for  $i = 1 : m$  do
4:      $x'_i = x_i + z$ , where  $z \sim N(0, \sigma_z)$ 
5:     if  $u_1 < \frac{\pi_1(x'_i|y)}{\pi_1(x_i|y)}$ , where  $u_1 \sim U(0, 1)$  then
6:       if  $u_2 < \frac{\pi_0(x'_i|y)\pi_1(x_i|y)}{\pi_0(x_i|y)\pi_1(x'_i|y)}$ , where  $u_2 \sim U(0, 1)$  then
7:         set  $x_i = x'_i$ 
8:       end if
9:     end if
10:  end for
11: end for
    
```

solutions of fitted voltages which depend on the conductivity field x . These approximate solutions are obtained more quickly, but do not match the voltages obtained from the “exact” solve. Here we consider two approximate solvers: one that stops after two $V(1, 1)$ cycles; and another that stops after a single $V(1, 1)$ cycle. The resulting approximate simulators $\eta_1(x)$ (two V cycles) and $\eta_2(x)$ (a single V cycle) are less accurate overall, but faster. In this case, $\eta_1(x)$ typically takes a third of the computing time to evaluate relative to the exact solver $\eta_0(x)$, while $\eta_2(x)$ typically takes a quarter of the time.

16.4.1 Delayed Acceptance Metropolis

The delayed acceptance approach of Fox and Nicholls (1997) and Christen and Fox (2005) uses a fast, approximate simulator to “pretest” a proposal. This approach adapts ideas from the surrogate transition method of Liu (2001) for dealing with complex forward models. For now, we define different posterior formulations, depending on which simulator is used:

$$\pi_v(x|y) \propto L_v(y|x) \times \pi(x) \propto \exp \left\{ -\frac{1}{2\sigma^2} (y - \eta_v(x))^T (y - \eta_v(x)) \right\} \times \exp \left\{ \beta \sum_{i \sim j} u(x_i - x_j) \right\} \times I[x \in [2.5, 4.5]^m].$$

We note that one could alter the sampling model for the approximate formulations, though it is not done here.

A simple Metropolis-based formulation of this scheme is given in Algorithm 4, where $\pi_0(x|y)$ and $\pi_1(x|y)$ denote the posteriors using the exact and approximate simulators, respectively. Notice that the exact simulator need only be run if the pretest condition ($u_1 < \frac{\pi_1(x'_i|y)}{\pi_1(x_i|y)}$) involving the faster, approximate simulator is satisfied. Hence, if the proposal width is chosen so that the pretest condition is satisfied only a third of the time, the exact simulator is only run for a third of the MCMC iterations. If we use the first approximate simulator $\eta_1(x)$, then the $40,000 \times m$ iterations required for our original single-site Metropolis scheme take about 66% of the computational effort using this delayed acceptance approach.

If $\eta_1(x)$ is a very good approximation to the exact simulator $\eta_0(x)$, then this delayed acceptance sampler is equally efficient if one normalizes by iteration. This is the case for the first approximate simulator in this example—the difference in log likelihood is typically no more than ± 0.2 over the range of posterior samples. However, if the approximate simulator

poorly matches the exact one, any savings obtained by reducing the number of exact simulator evaluations will be more than offset by reduced efficiency in the delayed acceptance sampler. In our application here, $|\eta_2(x) - \eta_0(x)|$ can be as large as 2 for some realizations x from the posterior. In using $\eta_2(x)$ in the delayed acceptance scheme, we detect a slight increase in autocorrelation which is more than offset by gains in computational efficiency from evaluating $\eta_2(x)$ in place of $\eta_0(x)$.

Apparently, this potential loss of efficiency is not present in this application since $\eta_2(x)$ is still an adequate approximation to the exact simulator $\eta_0(x)$. This loss of efficiency due to poor approximation is readily apparent if one takes a univariate example in which $\pi_0(x|y)$ is the standard normal density, and $\pi_1(x|y)$ is the normal density with mean one and standard deviation 0.5. In this case, the delayed acceptance sampler must take occasional, slow-moving excursions in the negative numbers to offset the lack of support in $\pi_1(x|y)$ in that region, reducing the efficiency of the sampler.

Finally, we note that Christen and Fox (2005) give a more general formulation for the delayed acceptance sampler for which the approximate simulator can depend on the current state x of the chain. While a bit more demanding computationally, the more general algorithm can make use of local approximations which are available in some applications.

16.4.2 An Augmented Sampler

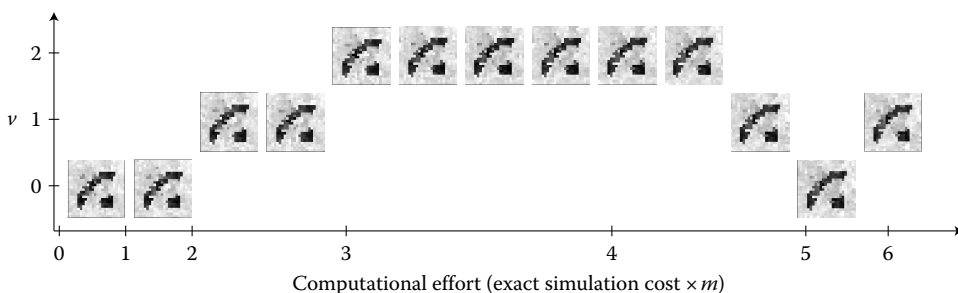
By augmenting the posterior of interest with auxiliary distributions one can use Metropolis coupling (Geyer, 1991), simulated tempering (Marinari and Parisi, 1992), or related schemes (Liu and Sabatti, 1999). Here we augment our posterior with additional posteriors based on the two approximate simulators. We introduce the auxiliary variable $v \in \{0, 1, 2\}$ to our formulation, which indexes the simulator to be used, and treat v as an additional parameter in a larger formulation. We specify a uniform prior for v over $\{0, 1, 2\}$ resulting in the augmented formulation

$$\begin{aligned} \pi(x, v|y) &\propto L(y|x, v) \times \pi(x) \times \pi(v) \\ &\propto \exp \left\{ -\frac{1}{2\sigma^2} (y - \eta_v(x))^T (y - \eta_v(x)) \right\} \times \exp \left\{ \beta \sum_{i \sim j} u(x_i - x_j) \right\} \\ &\times I[x \in [2.5, 4.5]^m] \times I[v \in \{0, 1, 2\}]. \end{aligned}$$

This augmented formulation can be sampled as before, except that after scanning through the elements of x to carry out single-site Metropolis updates, a simple Metropolis update is then carried out for v by making a uniform proposal over $\{0, 1, 2\} \setminus v$. Ideally, this chain should move somewhat often between the states of v .

A small subsequence from this chain is shown in Figure 16.10. As this sampler runs, the draws for which $v = 0$ are from the posterior of interest. While $v = 1$ or 2 , the chain is using one of the faster, approximate simulators. Hence, it can more quickly carry out the single-site Metropolis updates, so that the chain moves more rapidly through this auxiliary posterior. By the time the chain returns to $v = 0$, the realizations of x will generally show more movement than a sampler based solely on the exact simulator $\eta_0(x)$.

Marginally, the augmented sampler spends about 20% of its iterations at $v = 2$, 42% at $v = 1$, and 38% at $v = 0$. This augmented formulation allows about twice the number of single-site Metropolis updates as compared to the the standard single-site Metropolis chain


FIGURE 16.10

A sequence from the augmented sampler chain. A scan of m of single-site Metropolis updates is followed by a Metropolis update of the formulation index variable v . Here the sequence of images x starts with $v = 0$, then moves up to $v = 2$, and then back down to $v = 0$. While the chain is not using the exact simulator ($v > 0$), the computational cost of carrying out the m single-site Metropolis updates for each x_i is substantially reduced.

on the exact formulation alone. In all this nearly doubles the efficiency when normalized by computing effort. The efficiency of the sampler could be improved slightly by altering the prior for v so that the chain spends more time at $v = 2$ and less at $v = 0$.

A feature of both the delayed acceptance algorithm and the augmented formulation is that they utilize the most efficient MCMC scheme available. Both of these methods could be used with an alternative to single-site Metropolis if it is found to be more efficient. For the augmented example above, we could improve the computational efficiency by employing delayed acceptance, using $\eta_1(x)$ when carrying out the m single-site Metropolis updates for x_i when $v = 0$. There is no practical benefit in using delayed acceptance using $\eta_2(x)$ when $v = 1$ since the relative speed of the two simulators is not that different.

16.5 Discussion

For the EIT example, single-site Metropolis requires about 2 million simulator evaluations to effectively sample this posterior distribution. Multivariate updating schemes such as random-walk Metropolis or DE-MCMC—as we implemented them here—do not offer any real relief. Utilizing fast approximations through delayed acceptance and/or tempering schemes may reduce the computational burden by a factor of 4 or so, more if a very fast and accurate approximation is available. This means that sampling this $m = 576$ -dimensional posterior is going to require at least a half a million simulator evaluations. This number will most certainly increase as the dimensionality m increases. Hence, a very fast simulator is required if one hopes to use such an image-based MCMC approach for a three-dimensional inverse problem.

One challenging feature of this application is the multimodal nature of the posterior which is largely induced by our choice of prior. By specifying a more regularizing prior, such as the GMRF (Equation 16.4) or the process convolution (Equation A.1), the resulting posterior will more likely be unimodal, so that standard MCMC schemes will be more efficient. Of course, the sacrifice is that one is now less able to recover small-scale structure that may be present in the inverse problem.

In some applications the simulator is sufficiently complicated that one can only hope to run it a few hundred times. In such cases, there is no possibility of reconstructing an entire

image of unknown pixel intensities. However, one can construct a very fast surrogate by replacing the simulator by a response surface model built from the limited number of simulations that have been carried out. Craig et al. (2001) and Kennedy and O'Hagan (2001) are two examples of applications which utilize a response surface to aid the resulting simulation-based inference. Of course, this requires a low-dimensional representation of the unknown parameters to be input to the simulator. It also requires that the simulation output be amenable to a response surface model.

Finally, we note that the traditional way to speed up the computation required to solve an inverse problem is to speed up the simulator $\eta(x)$. A substantial amount of progress has been made in creating simulators that run on highly distributed computing machines. Comparatively little progress has been made in utilizing modern computing architectures to speed up MCMC-based posterior exploration in difficult inverse problems. Clearly schemes such as Metropolis coupling chains and DE-MCMC are quite amenable to distributed implementations. The integration of modern computing architecture with MCMC methods will certainly extend the reach of MCMC based solutions to inverse problems.

Appendix: Formulation Based on a Process Convolution Prior

An alternative to treating each pixel in the image as a parameter to be estimated is to use a lower-dimensional representation for the prior. Here we describe a process convolution (Higdon, 2002) prior for the underlying image x .

We define $x(s)$, $s \in \Omega$, to be a mean-zero Gaussian process. But rather than specify $x(s)$ through its covariance function, it is determined by a latent process u and a smoothing kernel $k(s)$. The latent process $u = (u_1, \dots, u_p)^T$ is located at the spatial sites $\omega_1, \dots, \omega_p$, also in Ω (shown in Figure 16.11). The u_j are then modeled as independent draws from a $N(0, \sigma_u^2)$ distribution. The resulting continuous Gaussian process model for $x(s)$ is then

$$x(s) = \sum_{j=1}^p u_j k(s - \omega_j), \quad (\text{A.1})$$

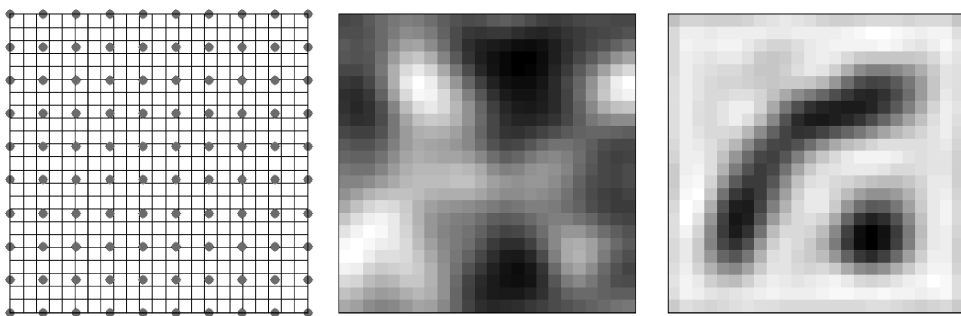


FIGURE 16.11

(Left) A 10×10 lattice of locations $\omega_1, \dots, \omega_p$, for the u_j of the process convolution prior; the 24×24 image pixels are shown for reference. (Middle) A realization from the process convolution prior for $x(s)$. (Right) Posterior mean from the single-site Metropolis scheme run on the u vector that controls the image x .

where $k(\cdot - \omega_j)$ is a kernel centered at ω_j . For the EIT application, we define the smoothing kernel $k(\cdot)$ to be a radially symmetric bivariate Gaussian density, with standard deviation $\sigma_u = 0.11$. Figure 16.11 shows a prior draw from this model over the 24×24 pixel sites in Ω . Under this formulation, the image x is controlled by $p = 100$ parameters in u . Thus a single-site Metropolis scan of u takes less than 20% of the computational effort required to update each pixel in x . In addition, this prior enforces very smooth realizations for x . This makes the posterior distribution better behaved, but may make posterior realizations of x unreasonably smooth. The resulting posterior mean for x is shown in Figure 16.11. For a more detailed look at process convolution models, see Higdon (2002); Paciorek and Schervish (2004) give non-stationary extensions of these spatial models.

Acknowledgments

This work was partially supported by the Los Alamos National Laboratory LDRD/ER research grant 20080300ER, *Multilevel Adaptive Sampling for Multiscale Inverse Problems*, and the DOE Office of Science Advanced Computing Research (ASCR) program in Applied Mathematical Sciences.

References

- Andersen, K., Brooks, S., and Hansen, M. 2003. Bayesian inversion of geoelectrical resistivity data. *Journal of the Royal Statistical Society, Series B*, 65(3):619–642.
- Briggs, W. L., Henson, V. E., and McCormick, S. F. 2000. *A Multigrid Tutorial*, 2nd edn. SIAM Books, Philadelphia.
- Christen, J. and Fox, C. 2005. Markov chain Monte Carlo using an approximation. *Journal of Computational and Graphical Statistics*, 14(4):795–810.
- Cleveland, W. S. 1979. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74:829–836.
- Craig, P. S., Goldstein, M., Rougier, J. C., and Seheult, A. H. 2001. Bayesian forecasting using large computer models. *Journal of the American Statistical Association*, 96:717–729.
- Dendy, J. E. 1982. Black box multigrid. *Journal of Computational Physics*, 48:366–386.
- Dendy Jr, J. 1987. Two multigrid methods for three-dimensional problems with discontinuous and anisotropic coefficients. *SIAM Journal on Scientific and Statistical Computing*, 8:673.
- Edwards, R. G. and Sokal, A. D. 1988. Generalization of the Fortuin-Kasteleyn-Swendsen-Wang representation and Monte Carlo algorithm. *Physical Review Letters*, 38:2009–2012.
- Efendiev, Y., Hou, T., and Luo, W. 2006. Preconditioning Markov chain Monte Carlo simulations using coarse-scale models. *SIAM Journal on Scientific Computing*, 28(2):776–803.
- Fox, C. and Nicholls, G. 1997. Sampling conductivity images via MCMC. In K. V. Mardia, C. A. Gill and R. G. Aykroyd (eds), *The Art and Science of Bayesian Image Analysis. Proceedings of the Leeds Annual Statistics Research Workshop*, pp. 91–100. Leeds University Press, Leeds.
- Gelman, A., Roberts, G., and Gilks, W. 1996. Efficient Metropolis jumping rules. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith (eds), *Bayesian Statistics 5: Proceedings of the Fifth Valencia International Meeting*, pp. 599–607. Oxford University Press, Oxford.
- Geman, S. and McClure, D. 1987. Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute*, 52(4):5–21.

- Geyer, C. J. 1991. Monte Carlo maximum likelihood for dependent data. In E. M. Keramidas (ed.), *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, pp. 156–163. American Statistical Association, New York.
- Higdon, D. 2002. Space and space-time modeling using process convolutions. In C. Anderson, V. Barnett, P. C. Chatwin, and A. H. El-Shaarawi (eds), *Quantitative Methods for Current Environmental Issues*, pp. 37–56. Springer, London.
- Higdon, D., Lee, H., and Bi, Z. 2002. A bayesian approach to characterizing uncertainty in inverse problems using coarse and fine scale information. *IEEE Transactions in Signal Processing*, 50:389–399.
- Higdon, D. M. 1998. Axiliary variable methods for Markov chain Monte Carlo with applications. *Journal of the American Statistical Association*, 93:585–595.
- Higdon, D. M., Lee, H., and Holloman, C. 2003. Markov chain Monte Carlo-based approaches for inference in computationally intensive inverse problems. In J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West (eds), *Bayesian Statistics 7. Proceedings of the Seventh Valencia International Meeting*, pp. 181–197. Oxford University Press, Oxford.
- Jimenez, R., Verde, L., Peiris, H., and Kosowsky, A. 2004. Fast cosmological parameter estimation from microwave background temperature and polarization power spectra. *Physical Review D*, 70(2):23005.
- Kaipio, J., Kolehmainen, V., Somersalo, E., and Vauhkonen, M. 2000. Statistical inversion and Monte Carlo sampling methods in electrical impedance tomography. *Inverse Problems*, 16(5):1487–1522.
- Kaipio, J. P. and Somersalo, E. 2004. *Statistical and Computational Inverse Problems*. Springer, New York.
- Kennedy, M. and O'Hagan, A. 2001. Bayesian calibration of computer models (with discussion). *Journal of the Royal Statistical Society, Series B*, 68:425–464.
- Liu, J. 2001. *Monte Carlo Strategies in Scientific Computing*. Springer, New York.
- Liu, J. and Sabatti, C. 1999. Simulated sintering: Markov chain Monte Carlo with spaces of varying dimensions. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith (eds), *Bayesian Statistics 6: Proceedings of the Sixth Valencia International Meeting*. Oxford University Press, Oxford.
- MacLachlan, S. P. and Moulton, J. D. 2006. Multilevel upscaling through variational coarsening. *Water Resources Research*, 42.
- Marinari, E. and Parisi, G. 1992. Simulated tempering: a new Monte Carlo scheme. *Europhysics Letters*, 19:451–458.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. 1953. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092.
- Moulton, J. D., Fox, C., and Svyatskiy, D. 2008. Multilevel approximations in sample-based inversion from the Dirichlet-to-Neumann map. *Journal of Physics: Conference Series*, 124:012035 (10 pp.).
- Nissinen, A., Heikkinen, L. M., and Kaipio, J. P. 2008. The Bayesian approximation error approach for electrical impedance tomography—experimental results. *Measurement Science and Technology*, 19(1):015501 (9 pp.).
- Paciorek, C. J. and Schervish, M. J. 2004. Nonstationary covariance functions for Gaussian process regression. In S. Thrun, L. Saul, and B. Schölkopf (eds), *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA.
- Rue, H. 2001. Fast sampling of Gaussian Markov random fields. *Journal of the Royal Statistical Society, Series B*, 63(2):325–338.
- Stenerud, V., Kippe, V., Lie, K., and Datta-Gupta, A. 2008. to appear. Adaptive multiscale streamline simulation and inversion for high-resolution geomodels. *SPE J*.
- ter Braak, C. F. J. 2006. A Markov chain Monte Carlo version of the genetic algorithm Differential Evolution: Easy Bayesian computing for real parameter spaces. *Statistics and Computing*, 16(3):239–249.
- Tierney, L. 1994. Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics*, 21:1701–1762.
- Weir, I. 1997. Fully Bayesian reconstructions from single photon emission computed tomography. *Journal of the American Statistical Association*, 92:49–60.