

RESEARCH ARTICLE

An MCMC Method for Uncertainty Quantification in Nonnegativity Constrained Inverse Problems

Johnathan M. Bardsley[†] and Colin Fox^{*}

(Received 00 Month 200x; in final form 00 Month 200x)

The development of computational algorithms for solving inverse problems is, and has been, a primary focus of the inverse problems community. Less studied, but of increased interest, is uncertainty quantification for solutions of inverse problems obtained using computational methods. In this paper, we present a method of uncertainty quantification for linear inverse problems with nonnegativity constraints. We present a Markov chain Monte Carlo (MCMC) method for sampling from a particular probability distribution over the unknowns. From the samples, estimation and uncertainty quantification for both the unknown (image in our case) and regularization parameter are performed. The primary challenge of the approach is that for each sample a large-scale nonnegativity constrained quadratic minimization problem must be solved. We perform numerical tests on both one- and two-dimensional image deconvolution problems, as well as on a computed tomography test case. Our results show that our nonnegativity constrained sampler is effective and computationally feasible.

Keywords: inverse problems, image reconstruction, bound constrained optimization, Markov chain Monte Carlo, uncertainty quantification.

AMS Subject Classification: 15A29, 65F22, 65C05, 65C60, 94A08

1. Introduction

In this paper, we focus on linear models with independent and identically distributed Gaussian noise; that is we assume the data-noise model

$$\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\eta}, \quad (1)$$

where $\mathbf{b} \in \mathbb{R}^m$ corresponds to observed data; \mathbf{x} is the $n \times 1$ vector of unknowns; \mathbf{A} is the $m \times n$ forward model matrix obtained via a numerical discretization of the forward map; and $\boldsymbol{\eta}$ is an $m \times 1$ independent and identically distributed (iid) Gaussian random vector with variance σ^2 across all pixels.

We are interested in linear models of the form (1) that arise from the numerical discretization of an ill-posed linear operator equation $b = Ax$, where b and x are functions, and A is a compact operator. Such equations arise in a number of important image processing applications, such as computed tomography (CT) and image deconvolution.

The goal is to obtain an estimate of the unknown \mathbf{x} , given the data \mathbf{b} , the forward map \mathbf{A} , and statistical model (1). The most common approach for obtaining such an estimate is to write down a variational problem, or corresponding Euler-Lagrange equation, that is then solved using a numerical method, the solution being the

[†] Department of Mathematical Sciences, University of Montana, USA. Email: bardsleyj@mso.umt.edu. This work was supported by the NSF under grant DMS-0915107

^{*} Department of Physics, University of Otago, Dunedin, New Zealand, fox@physics.otago.ac.nz

desired estimate. For general discussions of ill-posed problems and regularization, see one of the many excellent texts on the subject, e.g., [9, 19].

However, applications exist in which the values of estimated parameters have financial, or other, consequences attached to them. In such cases, the problem of uncertainty quantification (UQ) is extremely important. In the recent paper [1], a Bayesian hierarchical model is presented, and an efficient MCMC method is introduced for sampling from the resulting posterior distribution. The samples from the posterior density are then used to compute an estimator (reconstructed image), e.g., the sample median, as well as for uncertainty quantification, e.g., by computing quantiles for each sampled parameter. Similar approaches have been studied by researchers in the field of inverse problems and imaging; see, e.g., [5, 6, 11, 12, 15, 20].

In this paper, we modify the MCMC method from [1] to incorporate a non-negativity constraint for \mathbf{x} , which arises in many applications, e.g., in CT, where the unknown is tissue density, and in image deblurring, where the unknown is light intensity. The modification results from replacing the quadratic minimization problem (or equivalent linear system) solved when computing the image samples with the corresponding nonnegativity constrained quadratic minimization problem. The modification results in an MCMC method for sampling from an implicitly defined distribution and is no longer strictly Bayesian. Nonetheless, we will provide a mathematical justification for, and interpretation of, this modification.

While adding the nonnegativity constraint is simple in theory, efficiently solving the resulting nonnegativity constrained quadratic program is a nontrivial problem. For this, we use the gradient projection–conjugate gradient (GPCG) method of [14]. GPCG is extremely efficient for large-scale problems and inherits finite convergence from CG (when exact arithmetic is assumed).

Nonnegativity constrained sampling is different from positivity constrained sampling. In the later, the probability of the unknown being zero is zero, whereas in the former the unknown is allowed to be zero with some positive probability, in our case determined by the likelihood and prior. For imaging examples, where zero intensity values are common, a nonnegativity constraint seems natural. Nevertheless, we compare results obtained with both approaches.

The paper is organized as follows. In Section 2, we derive the posterior density function in the unconstrained case from Bayes' Law, and then present the MCMC method of [1] for computing samples from the posterior density. Then in Section 3, we focus on sampling with constraints, first presenting a method for positivity constrained sampling, and then presenting our nonnegativity constrained sampler obtained via a modification of the unconstrained sampler from Section 2. Finally, in Section 4, we present a method for determining MCMC chain convergence and then apply our nonnegativity constrained MCMC method to examples from image deconvolution in both one- and two-dimensions and from computed tomography. Conclusions are given in Section 5.

2. The posterior density function

We begin by presenting an MCMC for sampling from a particular posterior density function, which we derive using Bayes' Law, as in [1, 10], and which has the form

$$p(\mathbf{x}, \lambda, \delta | \mathbf{b}) \propto p(\mathbf{b} | \mathbf{x}, \lambda) p(\lambda) p(\mathbf{x} | \delta) p(\delta). \quad (2)$$

The first term on the right is called the likelihood function, which is determined by (1) and has the form

$$p(\mathbf{b}|\mathbf{x}, \lambda) \propto \lambda^{n/2} \exp\left(-\frac{\lambda}{2}\|\mathbf{Ax} - \mathbf{b}\|^2\right),$$

where $\lambda = 1/\sigma^2$ is the noise precision.

The third term on the right in equation (2) is the prior probability density function and is assumed to have the form

$$p(\mathbf{x}|\delta) \propto \delta^{n/2} \exp\left(-\frac{\delta}{2}\mathbf{x}^T \mathbf{Cx}\right), \quad (3)$$

where δ is a scaling parameter for the prior precision matrix $\delta\mathbf{C}$. Here δ and \mathbf{C} correspond to the regularization parameter and matrix, respectively. Note that (3) is also used in Bayesian spacial statistics [10, 17].

Finally, the hyper-prior probability densities for λ and δ are taken to be Gamma distributions:

$$p(\lambda) \propto \lambda^{\alpha_\lambda - 1} \exp(-\beta_\lambda \lambda), \quad (4)$$

$$p(\delta) \propto \delta^{\alpha_\delta - 1} \exp(-\beta_\delta \delta). \quad (5)$$

Note that the Gamma distribution $\Gamma(\alpha, \beta)$ has a probability density function satisfying $g(t|\alpha, \beta) \propto t^{\alpha-1} \exp(-\beta t)$. Following [10], we take $\alpha_\lambda = \alpha_\delta = 1$ and $\beta_\lambda = \beta_\delta = 10^{-4}$. Then the hyper-priors can be deemed to be “uninformative”, since the mean and variance of the corresponding Gamma distributions is $\alpha/\beta = 10^4$ and $\alpha/\beta^2 = 10^8$, respectively. Uninformative hyper-priors are chosen so that their effect on the sampled values for λ and δ are negligible. Note that these choices of parameters require no tuning whatsoever and work on a variety of examples. Moreover, no other parameters remain to be defined. However, if one has a reasonable *a priori* notion of what λ and/or δ should be, the corresponding hyper-prior parameter choices could be modified accordingly.

Thus (2) has the form

$$p(\mathbf{x}, \lambda, \delta|\mathbf{b}) \propto \lambda^{n/2+\alpha_\lambda-1} \delta^{n/2+\alpha_\delta-1} \exp\left(-\frac{\lambda}{2}\|\mathbf{Ax} - \mathbf{b}\|^2 - \frac{\delta}{2}\mathbf{x}^T \mathbf{Cx} - \beta_\lambda \lambda - \beta_\delta \delta\right). \quad (6)$$

In [1, 10], a MCMC method is introduced for sampling from (6), and uncertainty quantification is performed [1] for image deconvolution examples in one- and two-dimensions.

2.1. Defining the matrix \mathbf{C} via Gaussian Markov random fields

It remains to define the prior precision (regularization) matrix \mathbf{C} . Staying with the statistical flavor of the presentation, we use Gaussian Markov random fields (MRF's), as in [10, 17]. Some of the notations set in this subsection will be used later in the paper.

A MRF on the pixel grid is defined by specifying a neighborhood system at each pixel, as well as a set of n conditional densities $\{p(x_i|\mathbf{x}_{\partial_i}), i = 1, \dots, n\}$, where $\mathbf{x}_{\partial_i} = \{x_j|j \in \partial_i\}$, with $j \in \partial_i$ if $i \neq j$ and pixels i and j are neighbors. We note that MRF's have the property that for all i , the full conditional density $p(x_i|\mathbf{x}_{-i}) = p(x_i|\mathbf{x}_{\partial_i})$, where $\mathbf{x}_{-i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$, i.e. it depends only

upon it's neighbors.

For a given neighborhood system, we define the conditional densities to be Gaussian and of the form

$$x_i | \mathbf{x}_{\partial_i} \sim N(\bar{x}_{\partial_i}, 1/n_i),$$

where n_i is the number of neighbors belonging to x_i and \bar{x}_{∂_i} is the average of the x_j 's neighboring x_i . The resulting joint Gaussian density for \mathbf{x} is then given by [10, 17]

$$p(\mathbf{x}) \propto |\mathbf{C}|^{1/2} \exp\left(-\frac{1}{2} \mathbf{x}^T \mathbf{C} \mathbf{x}\right),$$

where

$$[\mathbf{C}]_{ij} = \begin{cases} n_i & i = j, \\ -1 & j \in \partial_i, \\ 0 & \text{otherwise.} \end{cases}$$

Thus if we define \mathbf{C}_{-i} to be the matrix that results if the i th column of \mathbf{C} is removed, we have $n_i x_{\partial_i} = -[\mathbf{C}_{-i} \mathbf{x}_{-i}]_i$.

Note that if zero boundary pixels are assumed and a standard first-order neighborhood is used – which in 1D includes the pixels to the left and right, and in 2D those to the left, right, above, and below – the discrete negative-Laplacian with Dirichlet boundary conditions results. If, however, periodic boundary conditions are assumed, the discrete negative-Laplacian with periodic boundary conditions results. Finally, recall that in practice, we include the scaling parameter δ (see (3)), which is necessary in any practical application [10, 17].

2.2. MCMC sampling of the posterior distribution

In this section, we present the MCMC method of [1]. Our choice of prior (3) on \mathbf{x} , and hyper-priors (4) on λ and (5) on δ were made with conjugacy relationships in mind [8], i.e. so that the full conditional densities have the same form as the corresponding prior/hyper-prior. To see this, note that the full conditional densities have the form

$$\begin{aligned} p(\mathbf{x} | \lambda, \delta, \mathbf{b}) &\propto \exp\left(-\frac{\lambda}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 - \frac{\delta}{2} \mathbf{x}^T \mathbf{C} \mathbf{x}\right), \\ p(\lambda | \mathbf{x}, \delta, \mathbf{b}) &\propto \lambda^{n/2 + \alpha_\lambda - 1} \exp\left(\left[-\frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 - \beta_\lambda\right] \lambda\right), \\ p(\delta | \mathbf{x}, \lambda, \mathbf{b}) &\propto \delta^{n/2 + \alpha_\delta - 1} \exp\left(\left[-\frac{1}{2} \mathbf{x}^T \mathbf{C} \mathbf{x} - \beta_\delta\right] \delta\right), \end{aligned}$$

and hence,

$$\mathbf{x}|\lambda, \delta, \mathbf{b} \sim N((\lambda \mathbf{A}^T \mathbf{A} + \delta \mathbf{C})^{-1} \lambda \mathbf{A}^T \mathbf{b}, (\lambda \mathbf{A}^T \mathbf{A} + \delta \mathbf{C})^{-1}), \quad (7)$$

$$\lambda|\mathbf{x}, \delta, \mathbf{b} \sim \Gamma\left(n/2 + \alpha_\lambda, \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 + \beta_\lambda\right), \quad (8)$$

$$\delta|\mathbf{x}, \lambda, \mathbf{b} \sim \Gamma\left(n/2 + \alpha_\delta, \frac{1}{2} \mathbf{x}^T \mathbf{C}\mathbf{x} + \beta_\delta\right). \quad (9)$$

The following Gibbs sampler follows immediately from (7)-(9).

An Unconstrained MCMC Method for Sampling from $p(\mathbf{x}, \delta, \lambda|\mathbf{b})$.

0. Initialize δ_0 and λ_0 , and set $k = 0$;
1. Compute $\mathbf{x}^k \sim N((\lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{C})^{-1} \lambda_k \mathbf{A}^T \mathbf{b}, (\lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{C})^{-1})$;
2. Compute $\lambda_{k+1} \sim \Gamma(n/2 + \alpha_\lambda, \frac{1}{2} \|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|^2 + \beta_\lambda)$;
3. Compute $\delta_{k+1} \sim \Gamma(n/2 + \alpha_\delta, \frac{1}{2} (\mathbf{x}^k)^T \mathbf{C}\mathbf{x}^k + \beta_\delta)$;
4. Set $k = k + 1$ and return to Step 1.

Since the parameters λ and δ are scalar, the scalar random draws required in Steps 2 and 3 are very efficient and easy to compute given the appropriate random number generator. Note that this is precisely the MCMC method presented in [1].

For inverse problems, the computational bottleneck occurs in Step 1, where the following linear system must be solved at each iteration for \mathbf{x}^k :

$$(\lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{C})\mathbf{x}^k = \lambda_k \mathbf{A}^T \mathbf{b} + \mathbf{w}, \quad \text{where } \mathbf{w} \sim N(\mathbf{0}, \lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{C}). \quad (10)$$

Thus in order for the MCMC method to be efficient, solutions of (10) must be efficiently computable. In [1], for example, a two-dimensional image deconvolution test case with periodic boundary conditions is considered. In this instance, $(\lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{C})$ can be diagonalized via the discrete Fourier transform, and hence, (10) can be efficiently solved.

3. MCMC sampling with constraints

Computing Gaussian samples as in (10) clearly does not impose constraints on the computed samples. However, in many imaging applications a nonnegativity constraint is natural.

The problem of sampling from a Gaussian with a positivity constraint has been studied by other researchers; see, e.g., [5] for the scalar, standard normal case. The negative portions of the Gaussian are simply truncated, yielding no mass at the boundary of the constraints. As we will see, positivity constrained sampling can be incorporated into (10), within a Gibbsian framework, in a straightforward manner.

Nonnegativity constrained sampling, on the other hand, results from projecting the negative portion of the distribution onto the boundary of the constraints, yielding mass at the boundary. In some imaging applications, e.g. looking at dark regions of the night sky in astronomy, one can imagine a positive probability that certain pixels have a zero value, yielding mass at the boundary.

In what follows, we show how to perform both types of sampling schemes and compare the two methods later in the paper.

3.1. Positivity constrained sampling

A convergent sampler also results if we replace Step 1 of the above MCMC method by a component-wise Gibbs sampler (see [10]), as follows:

- (1) Compute a sample x_i^k from the random variable with probability density $p(x|\mathbf{x}_{-i}^k, \lambda_k, \delta_k)$ proportional to

$$\exp \left\{ -\frac{\lambda_k[\mathbf{A}^T \mathbf{A}]_{ii} + \delta_k \mathbf{C}_{ii}}{2} \left(x - \frac{\lambda_k([\mathbf{A}^T \mathbf{b}]_i - [\mathbf{A}^T \mathbf{A}_{-i} \mathbf{x}_{-i}^k]_i) - \delta_k \mathbf{C}_{-i} \mathbf{x}_{-i}}{\lambda_k[\mathbf{A}^T \mathbf{A}]_{ii} + \delta_k \mathbf{C}_{ii}} \right)^2 \right\},$$

for $i = 1, \dots, n$, where \mathbf{A}_{-i} and \mathbf{C}_{-i} are the matrices that result after the i th column of \mathbf{A} and \mathbf{C} , respectively, are removed.

The direct method (10) is preferable to this component-wise Gibbs sampler provided it is computationally feasible. However, (11) has the benefit that a positivity constraint is easily added. This can be done if Step 1 is replaced by

- (1) Compute a sample x_i^k from the random variable with probability density

$$p_+(x|\mathbf{x}_{-i}^k, \lambda_k, \delta_k) \propto p_+(x)p(x|\mathbf{x}_{-i}^k, \lambda_k, \delta_k) \tag{11}$$

for $i = 1, \dots, n$, where $p(x|\mathbf{x}_{-i}^k, \lambda_k, \delta_k)$ is given in (11), and $p_+(x)$ is 1 for positive values of x and 0 otherwise.

The resulting sampler for the posterior probability density $p(\mathbf{x}, \delta, \lambda|\mathbf{b})$ is provably convergent. Moreover, we can directly sample from (11) mimicking the approach of [5] for sampling from the positivity constrained standard normal density. There the Inverse Cumulative Distribution Function (CDF) Method is used.

Inverse CDF Method for sampling from a random variable x given its CDF ϕ :

- (1) Compute $t \sim \text{Uniform}([0, 1])$;
- (2) Calculate $x = \phi^{-1}(t)$.

To implement the Inverse CDF Method, we need ϕ^{-1} for x with probability density function (11).

To simplify notation, let μ and γ be the mean and precision of the Gaussian random variable (11). Then the cumulative density of (11) has the form

$$\begin{aligned} \phi(z) &= C \int_0^z \exp \left\{ -\frac{\gamma}{2}(x - \mu)^2 \right\} dx, \\ &= C \sqrt{\frac{2}{\gamma}} \left(\int_0^{\sqrt{\frac{\gamma}{2}}(z-\mu)} e^{-s^2} ds + \int_0^{\sqrt{\frac{\gamma}{2}}\mu} e^{-s^2} ds \right), \\ &= C \sqrt{\frac{\pi}{2\gamma}} \left(\text{erf} \left(\sqrt{\frac{\gamma}{2}}(z - \mu) \right) + \text{erf} \left(\sqrt{\frac{\gamma}{2}}\mu \right) \right), \end{aligned}$$

where the first inequality follows from the change of variables $s = \frac{\gamma}{2}(x - \mu)$, and $\text{erf}(t) = \frac{2}{\sqrt{\pi}} \int_0^t e^{-s^2} ds$. Now since $\lim_{z \rightarrow \infty} \phi(z) = \lim_{z \rightarrow \infty} \text{erf}(z) = 1$, we can solve

for the normalizing constant C :

$$C = \left(\sqrt{\frac{\pi}{2\gamma}} \left(1 + \operatorname{erf} \left(\sqrt{\frac{\gamma}{2}} \mu \right) \right) \right)^{-1}$$

Hence the cumulative density function for (11) is given by

$$\phi(z) = \frac{\operatorname{erf} \left(\sqrt{\frac{\gamma}{2}} (z - \mu) \right) + \operatorname{erf} \left(\sqrt{\frac{\gamma}{2}} \mu \right)}{1 + \operatorname{erf} \left(\sqrt{\frac{\gamma}{2}} \mu \right)}.$$

Finally, solving the equation $t = \phi(z)$ for z yields

$$\phi^{-1}(t) = \mu + \sqrt{\frac{2}{\gamma}} \operatorname{erfinv} \left\{ t \left(1 + \operatorname{erf} \left(\sqrt{\frac{\gamma}{2}} \mu \right) \right) - \operatorname{erf} \left(\sqrt{\frac{\gamma}{2}} \mu \right) \right\}. \quad (12)$$

Built-in functions for evaluating erf and erfinv are contained in MATLAB. Thus we have shown how to sample from (11), yielding a positivity constrained, component-wise Gibbs sampler for the posterior density $p(\mathbf{x}, \delta, \lambda | \mathbf{b})$. For large-scale problems, given the component-wise nature of this sampler, the resulting MCMC will be computationally demanding. On the other hand, the method samples from a specific posterior density and remains Bayesian.

3.2. Nonnegativity Constrained Sampling

Note that for the conditional distributions defined by (11), the probability of $x_i = 0$ is zero for all i . However, one can imagine applications in image processing where $x_i = 0$ with some positive probability. For example, in the application of astronomy, we would expect a reconstruction of an image with a dark background to have a large number of zero valued pixels. Similarly, in medical imaging, e.g. computed tomography, we would expect zero tissue density values for pixels in regions where there is no tissue, and hence, where the x-ray signal is not attenuated.

We now introduce a probability model that will have such characteristics. We obtain our MCMC method by making modification to the unconstrained MCMC method presented above, or equivalently, by modifying the conditional densities (7)-(9). We begin by noting, as in [1], that (10) can be equivalently written as the solution of the variational problem

$$\mathbf{x}^k = \arg \min_{\mathbf{x}} \left\{ \frac{1}{2} \mathbf{x}^T (\lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{C}) \mathbf{x} - \mathbf{x}^T (\lambda_k \mathbf{A}^T \mathbf{b} + \mathbf{w}) \right\}, \quad (13)$$

where $\mathbf{w} \sim N(\mathbf{0}, \lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{C})$. This suggests the following obvious modification of (13) for computing nonnegativity constrained samples:

$$\mathbf{x}^k = \arg \min_{\mathbf{x} \geq \mathbf{0}} \left\{ \frac{1}{2} \mathbf{x}^T (\lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{C}) \mathbf{x} - \mathbf{x}^T (\lambda_k \mathbf{A}^T \mathbf{b} + \mathbf{w}) \right\}, \quad (14)$$

where $\mathbf{w} \sim N(\mathbf{0}, \lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{C})$.

The reason for using (14) is that highly efficient computational methods exist for its solution. Unfortunately, the inclusion of the nonnegativity constraint modifies the probability model in a non-trivial way. In particular, it implicitly imposes a probability model on the zero set of the solution $\mathcal{A}(\mathbf{x}) = \{i | x_i = 0\}$, which in turn

modifies the prior, defined only on the indices $i \notin \mathcal{A}(\mathbf{x})$ for a given \mathbf{x} . Because of this, the approach can no longer be said to be strictly Bayesian.

Nonetheless, the resulting conditional density $p(\mathbf{x}|\mathbf{b}, \lambda, \delta)$, implicitly defined by (14), has an intuitive interpretation. To see this, we note that (14) can be shown to be equivalent to

$$\mathbf{x}^k = \arg \min_{\mathbf{x} \geq \mathbf{0}} \|\mathbf{x} - \mathbf{x}_{\text{UC}}^k\|_E^2, \tag{15}$$

where \mathbf{x}_{UC}^k is an unconstrained sample defined by (10), with

$$\|\mathbf{w}\|_E^2 = \langle \mathbf{w}, \mathbf{w} \rangle_E = \mathbf{w}^T (\lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{C}) \mathbf{w}. \tag{16}$$

Thus we have the following result, which gives an intuitive description of the sampling scheme for $p(\mathbf{x}|\mathbf{b}, \lambda, \delta)$ defined by (14).

Theorem 3.1: *Given the norm $\|\cdot\|_E$ defined by (16) and a sample \mathbf{x}_{UC}^k drawn from*

$$N \left((\lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{C})^{-1} \lambda_k \mathbf{A}^T \mathbf{b}, (\lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{C})^{-1} \right),$$

the corresponding vector \mathbf{x}^k defined by (14) is the minimum-norm projection of \mathbf{x}_{UC}^k onto the set of all nonnegative vectors, i.e. it satisfies (15).

Note that the probability density/mass function defined by (14) will have positive mass along the boundaries of $\mathbf{x} \geq \mathbf{0}$. Note also that (14) yields implicit estimates of the zero set $\mathcal{A}(\mathbf{x}) = \{i | x_i = 0\}$ and that these estimates are dependent upon both the data \mathbf{b} and the forward map \mathbf{A} . In a standard Bayesian approach, an explicit probability model for the zero set would be defined, using, for example, a spike and slab prior [18].

Finally, an additional modification must be made to the unconstrained MCMC method presented in Section 2.2 to obtain our nonnegativity constrained MCMC method. Recall that in the unconstrained case, the prior has the form (3). However, for $\mathbf{x} \geq \mathbf{0}$, if $\mathbf{D}_{\mathbf{x}}$ is a diagonal matrix with diagonal values $[\mathbf{D}_{\mathbf{x}}]_{ii} = 0$ for $i \in \mathcal{A}(\mathbf{x})$ and 1 otherwise, then

$$\frac{\delta}{2} \mathbf{x}^T \mathbf{C} \mathbf{x} = \frac{\delta}{2} \mathbf{x}^T (\mathbf{D}_{\mathbf{x}} \mathbf{C} \mathbf{D}_{\mathbf{x}}) \mathbf{x},$$

which corresponds to a degenerate precision matrix $\delta \mathbf{D}_{\mathbf{x}} \mathbf{C} \mathbf{D}_{\mathbf{x}}$, which is nonsingular only for the indices $i \notin \mathcal{A}(\mathbf{x})$. Because of this, the normalizing constant for the prior must also be modified, yielding

$$p(\mathbf{x}|\delta) = \delta^{n_p/2} \exp \left(\frac{\delta}{2} \mathbf{x}^T \mathbf{C} \mathbf{x} \right),$$

where n_p is the number of positive elements in \mathbf{x} . Thus the full conditional density for δ also changes slightly, from (7) to

$$p(\delta|\mathbf{x}, \lambda, \mathbf{b}) \propto \delta^{n_p/2 + \alpha_\delta - 1} \exp \left(\left[-\frac{1}{2} \mathbf{x}^T \mathbf{C} \mathbf{x} - \beta_\delta \right] \delta \right),$$

and Step 3 of our MCMC method will need to be modified accordingly. In our experience, the resulting MCMC method will yield a credibility interval for λ that

contains the true value of $1/\sigma^2$, whereas without the change in Step 3, $1/\sigma^2$ will lie outside the credibility interval for λ .

We can now present our nonnegativity constrained MCMC method.

A Nonnegativity Constrained MCMC Method for Sampling from $p(\mathbf{x}, \delta, \lambda | \mathbf{b})$.

0. Initialize δ_0 and λ_0 , and set $k = 0$;
1. Compute \mathbf{x}^k defined by (14) with $\mathbf{w} \sim N(\mathbf{0}, \lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{C})$;
2. Compute $\lambda_{k+1} \sim \Gamma(n/2 + \alpha_\lambda, \frac{1}{2} \|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|^2 + \beta_\lambda)$;
3. Compute $\delta_{k+1} \sim \Gamma(n_p^k/2 + \alpha_\delta, \frac{1}{2} (\mathbf{x}^k)^T \mathbf{C}\mathbf{x}^k + \beta_\delta)$, where n_p^k is the number of nonzero entries in \mathbf{x}^k ;
4. Set $k = k + 1$ and return to Step 1.

3.2.1. An Iterative Method for Computing Nonnegative Samples

By solving (14) as accurately as possible at every step, we obtain independent samples of the image \mathbf{x} . This requires a sophisticated, fast converging numerical method.

A different approach, which we do not pursue here, is to use a numerical method that is slow to converge, yielding approximate solutions of (14) that aren't quite independent.

The standard iterative method for solving (14) is gradient projection (GP) iteration [4, 13]. However, GP is very slow to converge for large-scale problems. GP does, however, have the desirable property that it identifies the zero set of the solution of (14) $\mathcal{A}(\mathbf{x}^k)$ in finitely many iterations [4, 13].

The more efficient and faster converging methods for solving (14) incorporate second-order information. Such methods include the projected-Newton (PN) method [2, 13] and the gradient projection-conjugate gradient (GPCG) algorithm of Moré and Torello [14]. Bound constrained limited memory BFGS (secant-type) methods also exist [3, 13] and can be used. Of these methods, we have found GPCG to be the most efficient for quadratic problems. In fact, it is shown in [14] that GPCG converges to the unique solution of (14) in finitely many iterations, assuming exact arithmetic [14]. Moreover, for large-scale problems, the use of gradient projection iterations allows for large numbers of active constraints to be added, or dropped, at each iteration. This, together with the use of CG, yields rapid convergence of the method.

Direct modifications of CG have also been used for solving bound constrained quadratic minimization problems. These include the method of [16], which is slow to converge for large-scale problems due to the fact that it only allows the addition of one index to the zero set per iteration. Another algorithm, with intuitive appeal, results from simply projecting CG iterations – at every iteration or every few iterations – onto the set of nonnegative vectors $\{\mathbf{x} | \mathbf{x} \geq \mathbf{0}\}$. Unfortunately, convergence of the resulting algorithm to the unique solution of (14) cannot be proved, and hence we do not recommend its use.

For the sake of completeness, we now provide a description of the GPCG algorithm for solving problems of the form

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \geq \mathbf{0}} \left\{ q(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{B}\mathbf{x} - \mathbf{x}^T \mathbf{c} \right\}, \quad (17)$$

where $\mathbf{B} \in \mathbb{R}^{n \times n}$ is symmetric positive definite, and $\mathbf{c} \in \mathbb{R}^n$. In order to keep the presentation of the algorithm brief, we leave a significant amount of detail to the paper [14].

GPCG Algorithm for solving (17):

Step 0. Set $j = 0$ and choose initial guess $\mathbf{x}^0 = \mathbf{1}$.

Step 1. Apply gradient projection iterations (see below for more detail) to (17) with initial guess \mathbf{x}^j until stopping criteria are satisfied. Output the updated $\mathbf{x}^j \geq 0$.

Step 2. Compute a quadratic Taylor series approximation q_j of q centered at \mathbf{x}^j and restricted to the indices r such that $x_r^j > 0$ (see below for more detail). Use the conjugate gradient (CG) iteration to minimize q_j until stopping criteria are satisfied. Use the most recent CG iterate \mathbf{p}^j as a search direction in a projected backtracking line search. Output \mathbf{x}^{j+1} .

Step 3. If the outer iteration stopping criteria have been met, end the GPCG iterations. Otherwise, set $j = j + 1$ and return to Step 1.

Step 1, gradient projection iteration: The gradient projection (GP) iteration [4, 13] used in Step 1 of GPCG is defined as follows: given $\mathbf{x}^i \geq 0$ compute \mathbf{x}^{i+1} via

$$\begin{aligned} \mathbf{p}^i &= \mathbf{c} - \mathbf{B}\mathbf{x}^i, \\ \alpha^i &= \arg \min_{\alpha > 0} q(\mathcal{P}(\mathbf{x}^i + \alpha\mathbf{p}^i)), \\ \mathbf{x}^{i+1} &= \mathcal{P}(\mathbf{x}^i + \delta^i\mathbf{p}^i). \end{aligned} \tag{18}$$

Here $\mathcal{P}(\mathbf{x}) = \max\{\mathbf{x}, \mathbf{0}\}$, where the maximum is computed component-wise. In practice, the subproblem (18) is solved inexactly using a projected backtracking line search algorithm.

GP is stopped once a sufficient decrease condition, given in [14], is satisfied. For more detail on the GP implementation within GPCG, see [14].

Step 2, conjugate gradient iteration: The quadratic Taylor series approximation of q used in Step 2 of iteration k of the GPCG algorithm is of the form

$$q_j(\mathbf{p}) = q(\mathbf{x}^j) + \langle \nabla_{\text{red}} q(\mathbf{x}^j), \mathbf{p} \rangle + \frac{1}{2} \langle \nabla_{\text{red}}^2 q(\mathbf{x}^j) \mathbf{p}, \mathbf{p} \rangle, \tag{19}$$

where

$$[\nabla_{\text{red}} q(\mathbf{x}^j)]_r = \begin{cases} [\mathbf{B}\mathbf{x}^j - \mathbf{c}]_i, & x_r > 0 \\ 0, & x_r = 0, \end{cases}$$

and

$$[\nabla_{\text{red}}^2 q(\mathbf{x}^j)]_{rs} = \begin{cases} [\mathbf{B}]_{rs}, & \text{if } x_r > 0 \text{ and } x_s > 0 \\ \delta_{rs}, & \text{otherwise.} \end{cases}$$

where $\delta_{rs} = 1$ if $r = s$ and is 0 otherwise.

After an approximate minimizer \mathbf{p}^j of q_j has been computed by the CG iterative method, equipped with the stopping rule of [14], a backtracking line search is performed, with \mathbf{p}^j as the search direction, to guarantee that $q(\mathbf{x}^{j+1}) < q(\mathbf{x}^j)$.

4. Numerical Experiments

In this section, we show how the output from our MCMC algorithm can be used, both to give point estimates as well as to allow for the quantification of uncertainty

in those estimates. First, however, we need a means for assessing MCMC chain convergence.

4.1. Assessing MCMC chain convergence

As in [1], we follow the recommended approach presented in [8], which requires the computation of multiple MCMC chains, with randomly chosen starting points. With multiple chains in hand, a statistic for each sampled parameter is then computed, whose value provides a measure of convergence.

This statistic is defined as follows. Suppose we compute n_r parallel chains, each of length n_s (after discarding the first half of the simulations), and that $\{\psi_{ij}\}$, for $i = 1, \dots, n_s$ and $j = 1, \dots, n_r$, is the collection of samples of a single parameter. Then we define

$$B = \frac{n_s}{n_r - 1} \sum_{j=1}^{n_r} (\bar{\psi}_{\cdot j} - \bar{\psi}_{\cdot\cdot})^2, \quad \text{where} \quad \bar{\psi}_{\cdot j} = \frac{1}{n_s} \sum_{i=1}^{n_s} \psi_{ij}, \quad \text{and} \quad \bar{\psi}_{\cdot\cdot} = \frac{1}{n_r} \sum_{j=1}^{n_r} \bar{\psi}_{\cdot j};$$

and

$$W = \frac{1}{n_r} \sum_{j=1}^{n_r} s_j^2, \quad \text{where} \quad s_j^2 = \frac{1}{n_s - 1} \sum_{i=1}^{n_s} (\psi_{ij} - \bar{\psi}_{\cdot j})^2.$$

Note that $\bar{\psi}_{\cdot j}$ and $\bar{\psi}_{\cdot\cdot}$ are the individual chain mean and overall sample mean, respectively. Thus B provides a measure of the variance between the n_r chains, while W provides a measure of the variance within individual chains.

The marginal posterior variance of the estimand $\text{var}(\psi|\mathbf{b})$ can then be estimated by

$$\widehat{\text{var}}^+(\psi|\mathbf{b}) = \frac{n_s - 1}{n_s} W + \frac{1}{n_s} B,$$

which is an unbiased estimate under stationarity [8]. The statistic of interest to us, however, is

$$\widehat{R} = \sqrt{\frac{\widehat{\text{var}}^+(\psi|\mathbf{b})}{W}}, \quad (20)$$

which declines to 1 as $n_s \rightarrow \infty$.

Once \widehat{R} is ‘near’ 1 for all sampled parameters, the $n_s n_r$ samples from the last half of all of the sequences together can be treated as samples from the target distribution [8]. We will see that in our examples, \widehat{R} will be much nearer to 1 than 1.1, which is deemed acceptable in [8].

4.2. A one-dimensional image deblurring example

We begin with a one-dimensional example (from [19]), because in one-dimension it is more straightforward to visualize uncertainty from the MCMC samples.

The data model is of the form (1), with $\mathbf{b} = \mathbf{A}\mathbf{x}$ obtained via mid-point quadra-

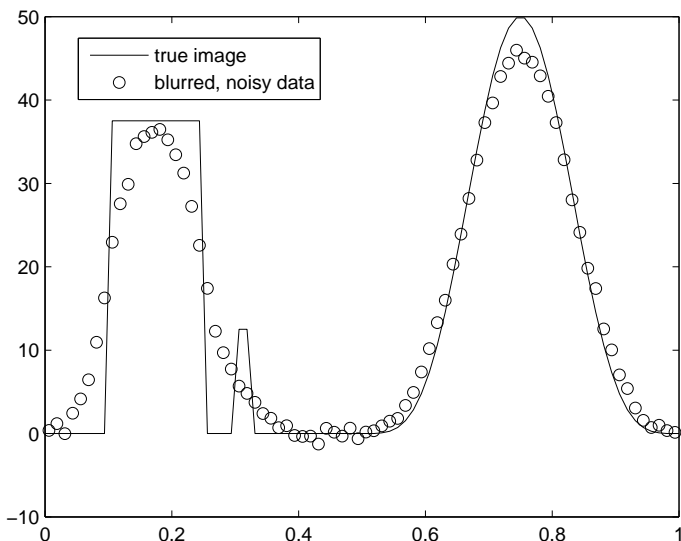


Figure 1. The one-dimensional true image and blurred noisy data.

ture applied to the convolution equation

$$b(s) = \int_0^1 A(s - s')x(s')ds',$$

with a Gaussian convolution kernel $A(s) = \exp(-s^2/(2\gamma^2))/\sqrt{\pi\gamma^2}$, $\gamma > 0$. Then \mathbf{A} has the form

$$[\mathbf{A}]_{ij} = h \exp(-((i - j)h)^2/(2\gamma^2)) / \sqrt{\pi\gamma^2}, \quad 1 \leq i, j \leq n, \quad (21)$$

where $h = 1/n$ with n the number of grid points in $[0, 1]$. We use $n = 80$, and the resulting \mathbf{A} has full column rank with condition number on the order of 10^{16} , resulting in a severely ill-conditioned problem. The true image is given by the solid line in Figure 1, and the data \mathbf{b} is generated using (1), with the noise variance σ^2 chosen so that the noise strength is 2% that of the signal strength.

In each of the three tests below, we compute 5 MCMC chains each of length 1000, as is suggested in [8]. Moreover, the initial values δ_0 and λ_0 in Step 0 are chosen randomly from the uniform distributions $U(0, 0.5)$ and $U(2, 8)$, respectively.

We present the unconstrained results from [1] for comparison. The computation of the samples in this case took approximately 8 seconds. The maximum \hat{R} value at the end of the run was 1.026. Direct image sampling (10) was used in Step 1 of the (unconstrained) MCMC algorithm via a Cholesky factorization of $\lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{C}$. From the image samples, on the left in Figure 2, we plot the median (or 0.5 quantile) as our reconstruction, and 95% credibility images given by the 0.025 and 0.975 quantiles of the samples at each pixel; these three quantiles were computed using MATLAB's `quantile` function. From the samples for λ and δ , on the right in Figure 2, we plot histograms for λ , δ , and the regularization parameter $\alpha = \delta/\lambda$. Finally, as a verification of our statistical model, we note that the true noise precision $1/\sigma^2 \approx 5.35$ is contained within the 95% credibility interval for λ , [3.85, 8.12], computed using MATLAB's `quantile` function.

For the positivity constrained case, the computation of the samples took approximately 25 seconds. The maximum \hat{R} value at the end of the run was 1.001. From

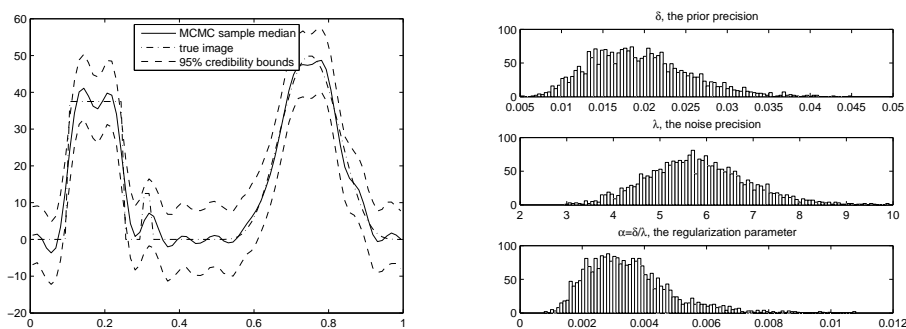


Figure 2. One-dimensional unconstrained example. On the left are plots of the median, and the 0.025 and 0.975 quantiles of the image samples. On the right are histograms of the samples of the precision parameters δ and λ , as well as of the regularization parameter $\alpha = \delta/\lambda$.



Figure 3. One-dimensional positivity constrained example. On the left are plots of the median, and the 0.025 and 0.975 quantiles of the image samples. On the right are histograms of the samples of the precision parameters δ and λ , as well as of the regularization parameter $\alpha = \delta/\lambda$.

the image samples, on the left in Figure 3, we plot the median image, the 0.025 quantile image, and the 0.975 quantile image. From the samples for λ and δ , on the right in Figure 3, we plot histograms for λ , δ , and the regularization parameter $\alpha = \delta/\lambda$. The reconstructed image appears to be reasonably accurate, however, for this example, we note that the true noise precision $1/\sigma^2 \approx 5.35$ is *not* contained within the 95% credibility interval for λ , $[1.55, 3.75]$, computed using MATLAB's `quantile` function, suggesting that the positivity constrained statistical model is not correct. In particular, it seems reasonable that, for this example, there is a positive probability of certain pixel values being zero.

For the nonnegativity constrained case, the computation of the samples took approximately 25 seconds. The maximum \hat{R} value at the end of the run was 1.002. We use the GPCG method to approximately solve the nonnegativity constrained quadratic programming problem in Step 1, with a maximum of 5 gradient projection and 20 CG iterations per outer GPCG iteration, and stopping tolerances of 10^{-6} for the relative decrease in the norm of the project gradient, and a maximum of 50 outer GPCG iterations. Output analogous to the previous two examples is given in Figure 4. Note that the value of $1/\sigma^2 \approx 5.35$ is within the 95% credibility interval $[4.76, 9.62]$, which validates the approach, and over repeated simulations, although the credibility interval changes, it always contains 5.35. In addition, the reconstruction appears to be of a higher resolution than in the positivity constrained case.

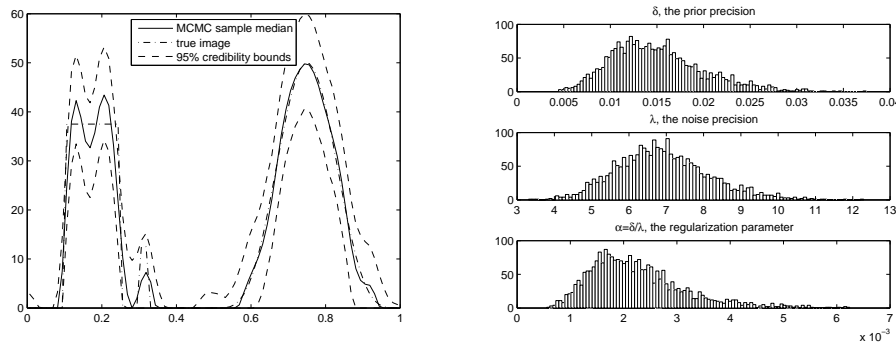


Figure 4. One-dimensional nonnegativity constrained example. On the left are plots of the median, and the 0.025 and 0.975 quantiles of the image samples. On the right are histograms of the samples of the precision parameters δ and λ , as well as of the regularization parameter $\alpha = \delta/\lambda$.

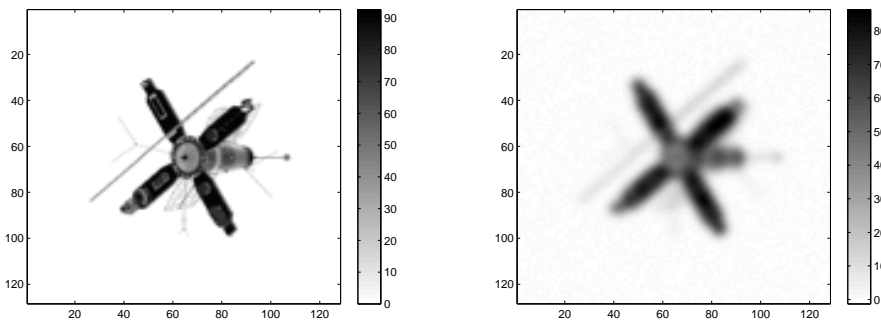


Figure 5. On the left is the true image, and on the right is the blurred noisy data.

4.3. A two-dimensional image deblurring example

In many inverse problems applications the spatial domain is two-dimensional. Thus we must show that our method is also effective on two-dimensional problems. Two-dimensional convolution has the form

$$b(s, t) = \int_0^1 \int_0^1 A(s - s', t - t')x(s', t')ds'dt'.$$

As above, we choose a Gaussian convolution kernel A , and discretize using mid-point quadrature on an 128×128 uniform computational grid over $[0,1] \times [0,1]$. Periodic boundary conditions for the image are assumed, so that \mathbf{A} is an $n^2 \times n^2$ block circulant with circulant blocks matrix, and hence is diagonalizable by the two-dimensional discrete Fourier transform (DFT) [9, 19]. Finally, the data \mathbf{b} is generated using (1) with the noise variance σ^2 chosen so that the noise strength is 2% that of the signal strength. The true image and data are shown in Figure 5.

Once again, following [8], we compute 5 MCMC chains each of length 1000. However, given that we are receiving independent samples for \mathbf{x} , shorter chains are likely sufficient; indeed, our initial tests seem indicate this. The initial values δ_0 and λ_0 in Step 0 are chosen randomly from the uniform distributions $U(0, 0.5)$ and $U(5, 10)$, respectively.

For the unconstrained constrained sampler of [1], the computation took approximately 38 seconds. The maximum \widehat{R} value at the end of the run was $\widehat{R} = 1.0054$. The diagonalization of $\lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{C}$ obtained via the discrete Fourier transform was used to make direct image sampling (10) efficient. We plot the mean of the sampled images as the reconstruction in Figure 6. From the samples for λ and δ ,

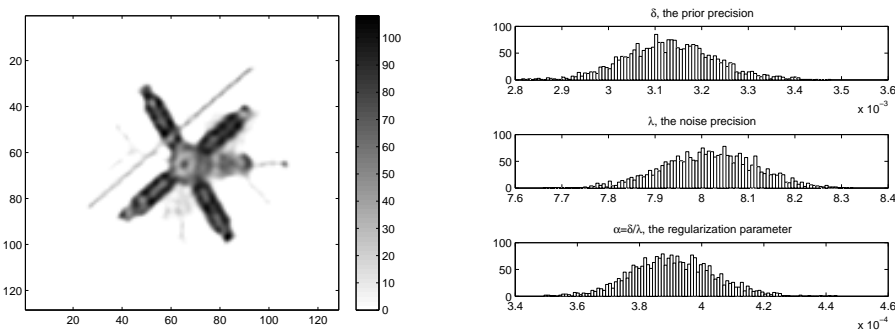


Figure 6. Two-dimensional unconstrained example. On the left is the mean image. On the right are histograms of the samples of the precision parameters δ and λ , as well as of the regularization parameter $\alpha = \lambda/\delta$.

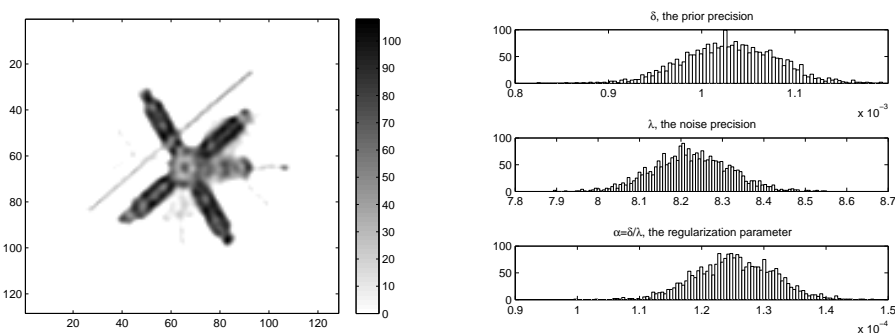


Figure 7. Two-dimensional nonnegativity constrained example. On the left is the mean image. On the right are histograms of the samples of the precision parameters δ and λ , as well as of λ/δ .

on the right in Figure 6, we plot histograms for λ , δ , and the regularization parameter $\alpha = \delta/\lambda$. The approach is validated by the fact that the true noise precision $1/\sigma^2 = 8.09$ is contained within the sample 95% credibility interval [7.82, 8.21].

For the nonnegativity constrained sampler, the computation took approximately 1 hour 15 minutes. The maximum \widehat{R} value at the end of the run was $\widehat{R} = 1.0065$. We implement GPCG for the nonnegativity constrained image sampling step with the same stopping criteria as in the one-dimensional case. We plot the mean of the sampled images as the reconstruction in Figure 7. From the samples for λ and δ , on the right in Figure 7, we plot histograms for λ , δ , and the regularization parameter $\alpha = \delta/\lambda$. The approach is validated by the fact that the true noise precision $1/\sigma^2 = 8.09$ is contained within the sample 95% credibility interval [8.043, 8.410]. Note that visually, the nonnegativity constrained mean image seems to be of higher resolution.

It remains to quantify the uncertainty in the median image estimates for both of the two-dimensional examples. This is more difficult than in one-dimension; indeed, satisfactory techniques for visualizing uncertainty in two- and three-dimensions are not known to the authors. Thus, for both the unconstrained and nonnegativity constrained problems, we simply plot the standard deviation of the sampled values at each pixel. To give the reader some sense of the variability suggested by these images, we note that for a Gaussian, the 95% confidence interval is approximately two standard deviations either side of the mean. Also note that, in the unconstrained case (on the left), the uncertainty is uniform throughout the image, whereas in the nonnegativity constrained case (on the right), the uncertainty is higher in the higher intensity regions, and is very small (in fact, in many cases 0) in the zero-intensity regions of the true image.

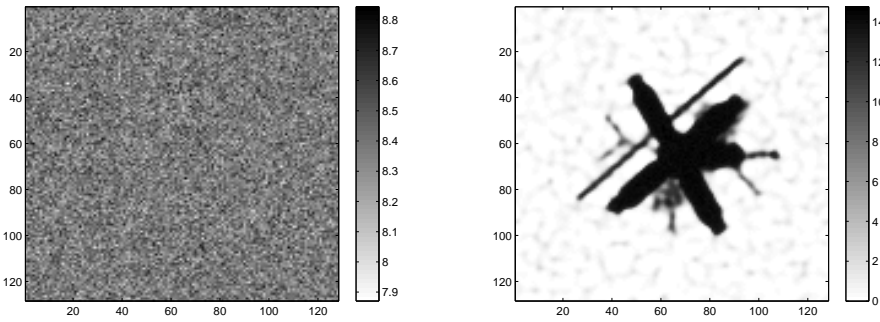


Figure 8. Plot of the standard deviation of the computed samples at each pixel. On the left is the plot for the unconstrained case, and on the right is a plot for the nonnegativity constrained case.

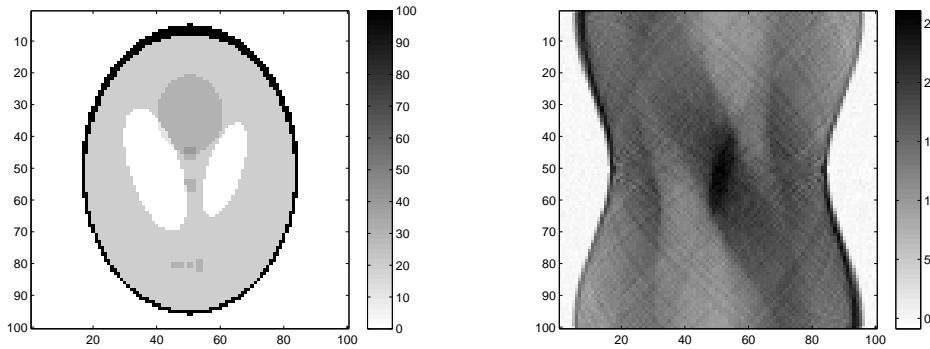


Figure 9. On the left is the two-dimensional true image, and on the right is the blurred noisy data.

4.4. Computed tomography

Computed tomography (CT) involves the reconstruction of the mass absorption function x of a body from one-dimensional projections of that body. A particular one-dimensional projection is obtained by integrating x along all parallel lines making a given angle ω with an axis in a fixed coordinate system. Each line L can be uniquely represented in this coordinate system by ω together with its perpendicular distance d to the origin. Suppose $L(\omega, d) = \{z(s) \mid 0 \leq s \leq S\}$, with an X-ray source located at $z(0)$ and a sensor at $z(S)$. The Radon transform model for CT has the form

$$b(\omega, d) = \int_{L(\omega, d)} x(z(s)) ds. \tag{22}$$

A discretized version of (22) is what is solved in the CT inverse problem, where b corresponds to collected data, and x is the unknown. The discretization occurs both in the spatial domain, where x is defined, as well as in the Radon transform domain, where b is defined and the independent variables are ω and d . We will use a uniform $n \times n$ spatial grid and a uniform $n_\omega \times n_d$ grid in the Radon transform domain. Then, after column-stacking the resulting two-dimensional arrays, we obtain a matrix-vector system of the form (1), where the data vector $\mathbf{b} \in \mathbb{R}^{n_\omega n_d}$, the unknown vector $\mathbf{x} \in \mathbb{R}^{n^2}$, and the forward model matrix $\mathbf{A} \in \mathbb{R}^{n_\omega n_d \times (n^2)}$. We generate data, once again, using (1) with $n = n_\omega = n_d = 100$ and σ^2 chosen so that the noise power is 2% that of the image. The truth image used is on the left in Figure 9 (generated using MATLAB's `phantom` function) and the data (called a sinogram) is given on the right.

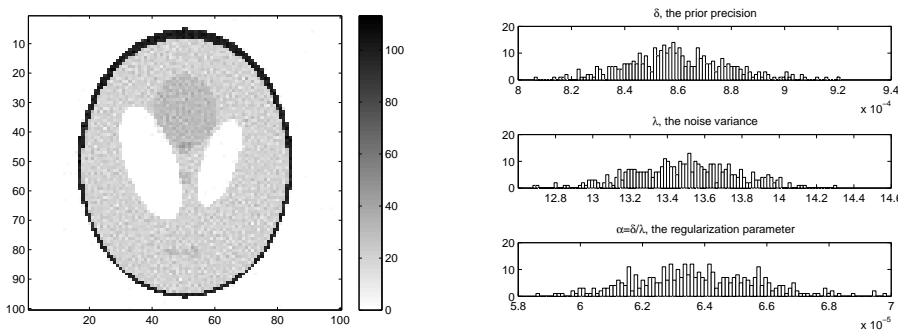


Figure 10. Computed tomography nonnegativity constrained example, 5 parallel chains of length 150. On the left is the mean image. On the right are histograms of the samples of the precision parameters δ and λ , as well as of λ/δ .

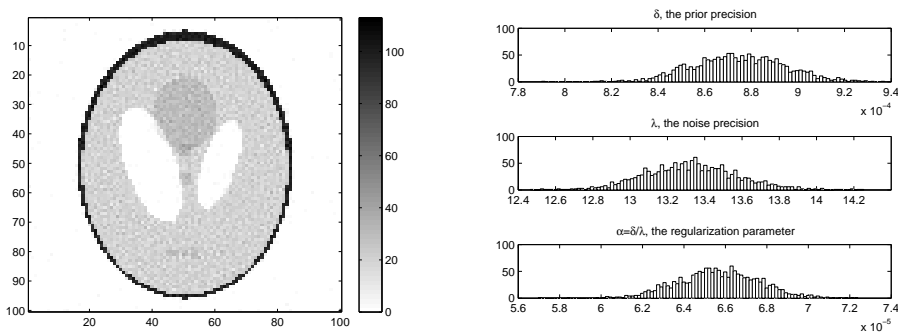


Figure 11. Computed tomography nonnegativity constrained example, 5 parallel chains of length 650. On the left is the mean image. On the right are histograms of the samples of the precision parameters δ and λ , as well as of λ/δ .

Once again, we run two separate MCMC simulations. However, here we compare the results obtained from a short and longer chain, in order to test the hypothesis that shorter chains may be sufficient since we are computing independent image samples. As above, we compute 5 MCMC chains each of length 150, and then, separately, 5 MCMC chains each of length 650. In both cases, the initial values δ_0 and λ_0 were chosen in Step 0 from the uniform distributions $U(0, 0.5)$ and $U(5, 10)$, respectively. The chains of length 150 took approximately 2 hours 18 minutes to compute, while the chains of length 650 took approximately 9 hours 45 minutes.

The maximum \widehat{R} value at the end of the runs was $\widehat{R} = 1.053$ for the chains of length 150, and $\widehat{R} = 1.008$ for the chains of length 650. In both cases, the projected gradient norm stopping tolerance of 10^{-6} is satisfied in every step by GPCG (with the iteration parameter values as in the above image deblurring examples), suggesting that the method is yielding approximately independent samples for \mathbf{x} .

We plot the mean of the sampled images as the reconstruction in Figures 10 and 11. From the samples for λ and δ , on the right in Figures 10 and 11, we plot histograms for λ , δ , and the regularization parameter $\alpha = \delta/\lambda$. A 95% credibility interval for λ , computed using MATLAB's quantile function, is given by [12.61, 13.53] for the chain of length 150 and [12.85, 13.83] for the chain of length 650, both of which contain the true value $1/\sigma^2 = 13.24$. The results are very similar for the samples computed from these two chains, suggesting that, due to the fact that independent samples are computed from the method, short chains yield satisfactory results.

Finally, we compare the standard deviation images for the two chains in Figure 12 and see that there are very similar. Note that as in the deblurring example,

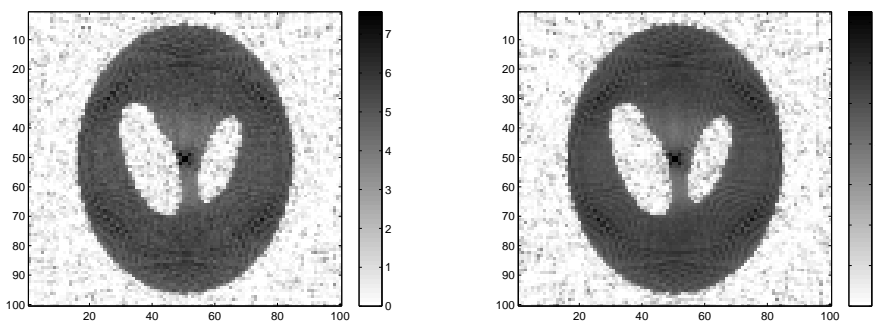


Figure 12. The standard deviation images for the chain of length 150, on the left, and the chain of length 650, on the right.

the uncertainty is higher in the higher intensity regions, and is very small (in fact, in many cases 0) in the zero-intensity regions of the true image, including in the corners and in the lungs.

5. Conclusions

We consider linear inverse problems with nonnegativity constraints and present an MCMC method for such problems, whose output (samples) can be used to obtain both estimates of unknowns, as well as for quantifying uncertainty in those estimates. Significantly, *a priori* choice of tuning parameters, such as the regularization parameter in classical inverse problems, is not required. However, each sample (and hundreds must be computed) requires the solution of a large-scale nonnegativity constrained quadratic program, so the approach is computationally intensive.

We advocate the use of the gradient projection-conjugate gradient method (GPCG) [14] for solving the large-scale nonnegativity constrained quadratic program; to our knowledge it is the most efficient method for such problems, though other efficient methods could also be used.

For comparison, we also present a positivity constrained MCMC sampler closely related to previous literature and show that the nonnegativity constrained sampler yields better results. Moreover, we provide a theorem interpreting the probability density implicitly defined by the addition of the nonnegativity constraint.

Our tests focus on image deconvolution examples in both one- and two-dimensions, as well as on computed tomography. The results in one-dimension are used to argue that the nonnegativity constrained sampler gives better results than the positivity constrained sampler in the image deblurring case. Due to the component-wise nature of the positivity constrained sampler, its use on two-dimensional problems is not feasible on the laptop used for the tests. In addition, in one-dimension, uncertainty quantification is more easily visualized.

In order to show that the approach is efficient enough to be used on large-scale problems, we present two-dimensional image deblurring and computed tomography results. In all cases, the approach is effective, and the credibility intervals for the samples of the noise precision λ contain the true value used for the simulations. Moreover, results in the CT case indicate that since the sampler yields independent image samples, short length chains will yield satisfactory results.

Acknowledgements

This work was supported by the National Science Foundation under grant DMS-0915107. The first author would like to thank the University of Montana and the Department of Physics at the University of Otago, New Zealand, and Dr. Colin Fox in particular, for their support during his 2010-11 sabbatical year.

References

- [1] J. M. Bardsley, *An MCMC Method for Estimation and Uncertainty Quantification in Linear Inverse Problems*, submitted, Math Sciences, University of Montana, Tech. Report #28, 2010.
- [2] D. Bertsekas, *Projected Newton methods for optimization problems with simple constraints*, SIAM Journal on Control and Optimization, **20** (1982), pp. 221-246.
- [3] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, *A limited memory algorithm for bound constrained optimization*, SIAM Journal on Scientific Computing, **16** (1995), pp. 1190-1208.
- [4] P. H. Calamai and J. J. Moré, *Projected Gradient Methods for Linearly Constrained Problems*, Mathematical Programming, **39** (1987), pp. 93-116.
- [5] D. Calvetti and E. Somersalo, *Introduction to Bayesian Scientific Computing*, Springer 2007.
- [6] D. Calvetti and E. Somersalo, *Hypermodels in the Bayesian Imaging Framework*, Inverse Problems, **24**(3), 2008, doi: 10.1088/0266-5611/24/3/034013.
- [7] T. Cui and C. Fox, *Computational Inference for Inverse Problems*, International Society for Bayesian Analysis Bulletin, **17**(3), 2010, pp. 12-15.
- [8] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis, Second Edition*, Chapman & Hall/CRC, Texts in Statistical Science, 2004.
- [9] P. C. Hansen, J. Nagy, and D. O’Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, SIAM, Philadelphia, 2006.
- [10] Dave Higdon, *A primer on space-time modelling from a Bayesian perspective*, Los Alamos National Laboratory, Statistical Sciences Group, Technical Report, LA-UR-05-3097.
- [11] J. P. Kaipio, V. Kolehmainen, E. Somersalo, and M. Vauhkonen, *Statistical inversion and Monte Carlo sampling methods in electrical impedance tomography*, Inverse Problems, **16**(5), 2000, pp. 14871522.
- [12] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Springer 2005.
- [13] C. T. Kelley, *Iterative Methods for Optimization*, SIAM, Philadelphia, 1999.
- [14] J. J. Moré and G. Toraldo, *On the Solution of Large Quadratic Programming Problems with Bound Constraints*, SIAM Journal on Optimization, **1** (1991), pp. 93-113.
- [15] G. Nicholls and C. Fox, *Prior modelling and posterior sampling in impedance imaging*, Bayesian Inference for Inverse Problems, Proc. SPIE 3459, 1998, pp. 116-127.
- [16] D. P. O’Leary, *A generalized conjugate gradient algorithm for solving a class of quadratic programming problems*, Linear Algebra and Its Applications **34** (1980), pp. 371-399.
- [17] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*, Chapman and Hall/CRC, 2005.
- [18] H. Ishwaran and J. Sunil Rao, *Spike and Slab Variable Selection: Frequentist and Bayesian Strategies*, The Annals of Statistics, **33**(2) (2005), pp. 730773.
- [19] C. R. Vogel, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, 2002.
- [20] D. Watzenig and C. Fox, *A review of statistical modeling and inference for electrical capacitance tomography*, Measurement Science and Technology, **20**(5), 2009, doi:10.1088/0957-0233/20/5/052002.