

## GAUSSIAN MARKOV RANDOM FIELD PRIORS FOR INVERSE PROBLEMS

JOHNATHAN M. BARDSLEY

Department of Mathematics Sciences  
University of Montana  
Missoula, MT, 59812-0864 USA

**ABSTRACT.** In this paper, our focus is on the connections between the methods of (quadratic) regularization for inverse problems and Gaussian Markov random field (GMRF) priors for problems in spatial statistics. We begin with the most standard GMRFs defined on a uniform computational grid, which correspond to the oft-used discrete negative-Laplacian regularization matrix. Next, we present a class of GMRFs that allow for the formation of edges in reconstructed images, and then draw concrete connections between these GMRFs and numerical discretizations of more general diffusion operators. The benefit of the GMRF interpretation of quadratic regularization is that a GMRF is built-up from concrete statistical assumptions about the values of the unknown at each pixel given the values of its neighbors. Thus the regularization term corresponds to a concrete spatial statistical model for the unknown, encapsulated in the prior. Throughout our discussion, strong ties between specific GMRFs, numerical discretizations of diffusion operators, and corresponding regularization matrices, are established. We then show how such GMRF priors can be used for edge-preserving reconstruction of images, in both image deblurring and medical imaging test cases. Moreover, we demonstrate the effectiveness of GMRF priors for data arising from both Gaussian and Poisson noise models.

### 1. INTRODUCTION

In this paper, we focus on linear inverse problems that can be modeled, after numerical discretization, as

$$\mathbf{b} = \mathbf{A}\mathbf{x}.$$

Here  $\mathbf{b} \in \mathbb{R}^m$  is the response vector (or observed data);  $\mathbf{x}$  is the  $n \times 1$  vector of unknowns; and  $\mathbf{A}$  is the  $m \times n$  design (or forward model) matrix, obtained in our case via a numerical discretization of the forward model.

In practice,  $\mathbf{b}$  contains random noise, and in this paper we are interested in two noise models. The first is the standard independent and identically distributed (iid) Gaussian case:

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

where  $\boldsymbol{\eta}$  is a Gaussian random vector with components satisfying  $\eta_i \sim N(0, \lambda^{-1})$  for all pixels  $i$ ; here the inverse-variance parameter  $\lambda$  is known as the

---

2000 *Mathematics Subject Classification*: Primary: 15A29, 65F22, 65C60; Secondary: 94A08.

*Key words and phrases*: inverse problems, regularization, Gaussian Markov random fields, numerical partial differential equations, Bayesian inference, image reconstruction.

*precision*. The other noise model, which is used in both astronomical and medical imaging, is the Poisson case:

$$(2) \quad \mathbf{b} = \text{Poiss}(\mathbf{A}\mathbf{x} + \boldsymbol{\gamma}),$$

where  $\boldsymbol{\gamma}$  is the  $m \times 1$  vector of background counts and is assumed known.

For these two probability models, the probability density functions  $p(\mathbf{b}|\mathbf{x})$  have the form

$$(3) \quad p(\mathbf{b}|\mathbf{x}) \propto \exp\left(-\frac{\lambda}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2\right),$$

for noise model (1), and

$$(4) \quad p(\mathbf{b}|\mathbf{x}) \propto \exp\left(-\sum_{j=1}^{n^2} \{([\mathbf{A}\mathbf{x}]_j + \gamma_j) + b_j \ln([\mathbf{A}\mathbf{x}]_j + \gamma_j)\}\right),$$

for noise model (2).

In both of these instances, the maximum likelihood estimators (i.e. the maximizers of  $p(\mathbf{b}|\mathbf{x})$ , defined in (3) and (4), with respect to  $\mathbf{x}$ ) are unstable with respect to the noise in the data  $\mathbf{b}$ . Such instability is a characteristic of *inverse problems*, and it has to do with the fact that the matrix  $\mathbf{A}$  is the numerical discretization of a compact operator defined on a function space. The standard technique for overcoming this instability is *regularization*. For general discussions of inverse problems and techniques for regularization, from both numerical and functional analytic points-of-view, see one of the many excellent texts on the subject, e.g., [8, 10, 12, 18].

In the context of Bayesian statistics, regularization corresponds to the choice of the prior probability density function. Bayes' Theorem states that given  $p(\mathbf{b}|\mathbf{x})$ , and an assumed *prior* probability density function  $p(\mathbf{x})$ , then for a particular realization of the data  $\mathbf{b}$ , the *posterior* probability density function  $p(\mathbf{x}|\mathbf{b})$  can be written

$$(5) \quad p(\mathbf{x}|\mathbf{b}) \propto p(\mathbf{b}|\mathbf{x})p(\mathbf{x}).$$

In this paper, our focus is on the use of *Gaussian Markov random fields* (GMRFs) for defining the prior  $p(\mathbf{x})$ . As we will see, this corresponds to assuming that the prior is Gaussian and of the form

$$(6) \quad p(\mathbf{x}) \propto \exp\left(-\frac{\delta}{2}\mathbf{x}^T\mathbf{L}\mathbf{x}\right),$$

where the *precision* (inverse-covariance) matrix  $\delta\mathbf{L}$  is sparse and encodes statistical assumptions regarding the values of the  $x_i$ 's based on the values of its neighbors. Once a suitable prior is chosen, it is standard to then compute the maximum a posteriori (MAP) estimator, which is the minimizer of  $-\ln p(\mathbf{x}|\mathbf{b})$ , i.e.

$$(7) \quad \mathbf{x}_{\text{MAP}} = \arg \min_{\mathbf{x}} \left\{ -\ln p(\mathbf{b}|\mathbf{x}) + \frac{\delta}{2}\mathbf{x}^T\mathbf{L}\mathbf{x} \right\},$$

where  $p(\mathbf{b}|\mathbf{x})$  is given by (3) or (4), depending on the noise model that is assumed. The MAP approach has seen a significant amount of recent interest in the field of inverse problems; see, e.g., the text books [6, 12, 17].

Typically in inverse problems, the regularization matrix  $\mathbf{L}$  is chosen in an ad hoc fashion – two common choices are the identity matrix and the discretized negative-Laplacian matrix – and the value of  $\delta$  is estimated from the data using a regularization parameter selection method, such as the discrepancy principle or generalized cross validation [10, 18]. Here, we instead take a Gaussian Markov random field

(GMRF) approach, in which the joint density function (6) for  $\mathbf{x}$  is obtained from an individual Gaussian probability model for each unknown pixel intensity  $x_i$ . As we will see, this approach leads both to standard choices for  $\mathbf{L}$ , such as the negative-Laplacian, as well as to some non-standard choices.

GMRFs have been studied extensively within the field of spatial statistics; for example, see the excellent text [16] and the references therein. Within the field of inverse problems, a class of GMRF priors is presented in [7] that are similar to what we present here. However in that paper, the corresponding (implicit) probability models for the individual pixel intensities (the  $x_i$ 's), which can be found in [16], are not presented. Given the importance of these pixel-wise probability models, not only for a full understanding of the prior, but for the development of new prior models, we include them here. Moreover, we make an explicit connection between several GMRFs and finite difference discretizations of elliptic partial differential equations (PDEs) on a uniform grid, and suggest further connections, also based on results found in [16], between GMRFs and discretizations of elliptic PDEs using other discretization methods and/or non-uniform computational grids.

The paper is organized as follows. In the next section, we discuss Gaussian Markov random fields in some detail. Then in Section 3, we present some iterative methods for edge-preserving reconstruction which make use of GMRF priors and then test their effectiveness on synthetic examples in image deblurring and positron emission tomography, using the Gaussian and Poisson noise models, respectively. Finally, we end with conclusions in Section 4.

## 2. GAUSSIAN MARKOV RANDOM FIELDS

In the pioneering work of Besag [5], an approach known as *conditional autoregression* was introduced for defining statistical models of a spatially distributed parameter vector  $\mathbf{x} = (x_1, \dots, x_n)$ . In this approach, probability distributions are assigned for the *full conditionals*  $x_i | \mathbf{x}_{-i}$ , where  $\mathbf{x}_{-i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$ , and from these assumptions a joint density for the random vector  $\mathbf{x}$  results, in which case  $\mathbf{x}$  is known as a *Markov random field* (MRF).

In this paper, we consider the special case in which the full conditionals are assumed to be Gaussian and of the form

$$(8) \quad x_i | \mathbf{x}_{-i} \sim \mathcal{N} \left( \sum_{j \neq i} \beta_{ij} x_j, \kappa_i^{-1} \right),$$

where  $\beta_{ij}$  and  $\kappa_i$  are parameters satisfying  $\beta_{ij}\kappa_i = \beta_{ji}\kappa_j$  for all  $i$  and  $j$ .

It can be shown (see [16, Theorem 2.6]) that (8) holds if and only if the joint density function for  $\mathbf{x}$  is Gaussian and of the form

$$(9) \quad p(\mathbf{x}) = (2\pi)^{-n/2} |\mathbf{Q}|^{1/2} \exp \left( -\frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} \right),$$

where

$$(10) \quad [\mathbf{Q}]_{ij} = \begin{cases} \kappa_i, & i = j, \\ -\kappa_i \beta_{ij}, & i \neq j. \end{cases}$$

In this case,  $\mathbf{x}$  is known as a Gaussian MRF (GMRF). Moreover, we must make the additional assumption that the *precision* matrix  $\mathbf{Q}$  is positive definite so that  $\mathbf{x}$  is a proper probability distribution.

In this paper, we are interested in GMRFs in which the matrix  $\mathbf{Q}$  is the numerical discretization of a diffusion operator in either one- or two-dimensions. In our discussion, we will focus on finite difference discretizations and their connections to some standard GMRFs, however we emphasize that other methods of numerical discretization, such as finite elements, could also be used. Provided the resulting discretization matrix  $\mathbf{Q}$  is symmetric positive definite and has the form (10), using the prior (9) corresponds to the pixel-wise Gaussian probability assumption (8).

The computational grid that is used for performing the numerical discretization defines a set of nodes  $i$  and a neighborhood system (or stencil)  $\partial_i$  for  $i = 1, \dots, n$ . It turns out that the elements of  $\partial_i$  are the nodes  $j \neq i$  for which  $[\mathbf{Q}]_{ij} \neq 0$ , and hence

$$\partial_i = \{j \neq i \mid \beta_{ij} \neq 0\}.$$

Thus the sparsity structure of  $\mathbf{Q}$  determines the neighborhood system, and vice versa.

Moreover, if  $\mathbf{x}_{\partial_i} = \{x_j \mid j \in \partial_i\}$ , we have  $p(x_i | \mathbf{x}_{-i}) = p(x_i | \mathbf{x}_{\partial_i})$  for all  $i$ , and hence it suffices to define the *neighborhood conditionals* (see [16, Theorem 2.4])

$$(11) \quad x_i | \mathbf{x}_{\partial_i} \sim \mathcal{N} \left( \sum_{j \in \partial_i} \beta_{ij} x_j, \kappa_i^{-1} \right)$$

to obtain (9), (10).

As an example, suppose  $\kappa_i = \delta n_i$ , where  $n_i = |\partial_i|$  and  $\delta > 0$ , and  $\beta_{ij} = 1/n_i$  for  $j \in \partial_i$  and 0 otherwise. Then (11) takes the form

$$(12) \quad x_i | \mathbf{x}_{\partial_i} \sim \mathcal{N}(\bar{x}_{\partial_i}, (\delta n_i)^{-1}),$$

where  $\bar{x}_{\partial_i} = \frac{1}{n_i} \sum_{j \in \partial_i} x_j$  is the mean of the neighboring values, and the joint density function is given by (9) with

$$(13) \quad [\mathbf{Q}]_{ij} = \delta \begin{cases} n_i & i = j, \\ -1 & j \in \partial_i, \\ 0 & \text{otherwise.} \end{cases}$$

As we will see in a moment, (13) is the finite difference discretization of the negative-Laplacian operator (scaled by a parameter  $\delta$ ) defined on a uniform grid. The resulting prior (9) is commonly used in inverse problems, in which case the discrete negative-Laplacian is the regularization matrix and  $\delta$  is the regularization parameter. Thus we see that (12) is the underlying pixel-wise probability model assumption that leads to this commonly used method of regularization.

**2.1. GMRFs AND THE DISCRETE NEGATIVE-LAPLACIAN WITH DIRICHLET BCs.** We now implement the GMRF defined by (12) in both one-dimensional (1D) and two-dimensional (2D) cases.

In 1D, we assume a uniform grid on  $[0, 1]$  with  $n$  vertices  $\{1, \dots, n\}$  at locations  $\{s_i\}_{i=1}^n$ , where  $s_i = i/(n+1)$ . Moreover, we define  $x_i$  to be the intensity value at  $s_i$  and assume Dirichlet (zero) boundary conditions (BCs), so that  $x_0 = x_{n+1} = 0$ . The neighborhood system is then given by  $\partial_i = \{i-1, i+1\}$  for  $i = 1, \dots, n$ . Thus

$n_i = 2$  for all  $i$  and from (13),  $\mathbf{Q} = \delta\mathbf{L}_{1D}$  with

$$(14) \quad \mathbf{L}_{1D} = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix}_{n \times n}.$$

Note that  $\mathbf{Q} = \delta\mathbf{L}_{1D}$  is symmetric, positive definite (SPD), as required, and that  $\mathbf{L}_{1D}$  is the 1D negative-Laplacian matrix on the computational grid with Dirichlet BCs; that is,  $\mathbf{L}_{1D}\mathbf{x}$  is the finite different discretization of the diffusion operator

$$-\frac{d^2x}{dt^2}, \quad x(0) = x(1) = 0.$$

For details on this basic numerical discretization, see [13, 20] or any of a number of standard texts on numerical analysis.

In 2D, we assume a uniform grid on  $[0, 1] \times [0, 1]$  with  $n^2$  vertices  $\{(i, j)\}_{i,j=1}^n$  at locations  $\{(s_i, t_j)\}_{i,j=1}^n$ , where  $s_k = t_k = k/(n+1)$ . Moreover, we define  $x_{ij}$  to be the intensity value at  $(s_i, t_j)$  for  $i, j = 1, \dots, n$ , and assume Dirichlet BCs so that  $x_{0,j} = x_{n+1,j} = x_{i,0} = x_{i,n+1} = 0$ . The neighborhood system is then given by

$$\partial_{ij} = \{(i-1, j), (i+1, j), (i, j-1), (i, j+1)\}, \quad \text{for } i, j = 1, \dots, n.$$

Hence,  $n_{ij} = |\partial_{ij}| = 4$ , and (12) takes the form

$$(15) \quad x_{ij} | \mathbf{x}_{\partial_{ij}} \sim \mathcal{N}(\bar{x}_{\partial_{ij}}, (\delta n_{ij})^{-1}),$$

where  $\bar{x}_{\partial_{ij}} = \frac{1}{n_{ij}} \sum_{(r,s) \in \partial_{ij}} x_{rs}$ .

After reordering the array  $\{x_{ij}\}_{i,j=1}^n$  by stacking its columns, we obtain the precision matrix  $\mathbf{Q} = \delta\mathbf{L}_{2D}$ , with

$$(16) \quad \mathbf{L}_{2D} = \begin{bmatrix} \tilde{\mathbf{L}} & -\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ -\mathbf{I} & \tilde{\mathbf{L}} & -\mathbf{I} & \ddots & \vdots \\ \mathbf{0} & \ddots & \ddots & \ddots & \mathbf{0} \\ \vdots & \ddots & -\mathbf{I} & \tilde{\mathbf{L}} & -\mathbf{I} \\ \mathbf{0} & \cdots & \mathbf{0} & -\mathbf{I} & \tilde{\mathbf{L}} \end{bmatrix}_{n^2 \times n^2},$$

where  $\mathbf{I}$  the  $n \times n$  identity matrix,  $\mathbf{0}$  the  $n \times n$  zero matrix, and

$$\tilde{\mathbf{L}} = \begin{bmatrix} 4 & -1 & 0 & \cdots & 0 \\ -1 & 4 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 4 & -1 \\ 0 & \cdots & 0 & -1 & 4 \end{bmatrix}_{n \times n}.$$

Note that  $\mathbf{Q}$  is SPD and that  $\mathbf{L}_{2D}$  is the 2D discrete negative-Laplacian matrix on the chosen computational grid with Dirichlet BCs; that is,  $\mathbf{L}_{2D}\mathbf{x}$  is the finite difference discretization of the diffusion operator

$$-\frac{d^2x}{ds^2} - \frac{d^2x}{dt^2}, \quad x(0, t) = x(s, 0) = x(1, t) = x(s, 1) = 0.$$

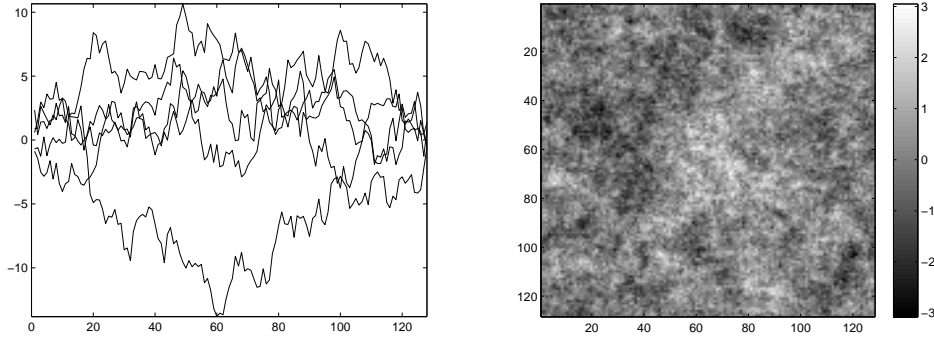


FIGURE 1.

For details on this standard finite difference discretization, see [13].

It is well-known that  $\mathbf{L}_{2D}$  defined in (16) can be alternatively expressed using the Kronecker product ‘ $\otimes$ ’; specifically,

$$(17) \quad \mathbf{L}_{2D} = \mathbf{L}_{1D} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{L}_{1D},$$

where  $\mathbf{L}_{1D}$  is defined in (14). The Kronecker product formula (17) will be used again in what follows.

Note that in both the 1D and 2D cases, the prior (9) takes the form

$$(18) \quad p(\mathbf{x}) \propto \exp\left(-\frac{\delta}{2} \mathbf{x}^T \mathbf{L} \mathbf{x}\right).$$

Thus GMRFs provide a convenient way to relate pixel-wise distributional assumptions of the form (12) and (15) with quadratic regularization functions of the form  $\frac{\delta}{2} \mathbf{x}^T \mathbf{L} \mathbf{x}$ .

On the left in Figure 1, we’ve plotted five realizations from (18) in 1D case, while on the right we’ve plotted one realization from (18) in the 2D case.

**2.2. INTRINSIC GMRFS.** The assumption that  $\mathbf{Q}$  is positive definite is often too restrictive. So-called *intrinsic GMRFs* (IGMRFs) allow for a symmetric positive *semi*-definite matrix  $\mathbf{Q}$ , so that  $\mathbf{Q}$  is allowed to have zero eigenvalues [16]. We will focus on first-order IGMRFs, which have only one zero eigenvalue with eigenvector  $\mathbf{1}$ , i.e.  $\mathbf{Q}\mathbf{1} = \mathbf{0}$ . Moreover, as above, the precision matrix  $\mathbf{Q} = \delta\mathbf{L}$  will be tied directly to pixel-wise conditional densities of the form (12) in 1D and (15) in 2D. In both cases, the prior takes the form (18). In this case,  $\delta\mathbf{L}$  is not technically a precision matrix, since it has a zero eigenvalue, and hence,  $p(\mathbf{x})$  not technically a probability density function, however we will continue to call these the precision and prior, respectively, following the convention of [16].

**2.2.1. IGMRFs and the discrete negative-Laplacian with periodic BCs.** A standard example of a 1D IGMRF prior results if we assume conditional distributions (12) (and hence precision (13)), but with a periodic BC, which for the same computational grid as above assumes  $x_{n+1} = x_1$  and  $x_0 = x_n$ . In this case, the precision matrix then has the form  $\mathbf{Q} = \delta\mathbf{L}_{1D}$  with  $\mathbf{L}_{1D}$  defined as in (14), but with  $[\mathbf{L}_{1D}]_{1n} = [\mathbf{L}_{1D}]_{n1} = -1$ . Note then that  $\mathbf{Q}\mathbf{1} = \mathbf{0}$ , and  $\mathbf{L}_{1D}$  is the 1D negative-Laplacian matrix on the computational grid with periodic BCs; that is,  $\mathbf{L}_{1D}\mathbf{x}$  is the

finite different discretization of the diffusion operator

$$-\frac{d^2x}{dt^2}, \quad x(0) = x(1).$$

In 2D, if we assume conditional distributions (15) and the same computational grid as above, the periodic BC results in the following assumption:  $x_{0,j} = x_{n,j}$ ,  $x_{n+1,j} = x_{1,j}$ ,  $x_{i,0} = x_{i,n}$ , and  $x_{i,n+1} = x_{i,1}$  for  $1 \leq i, j \leq n$ . Then  $\mathbf{Q} = \delta \mathbf{L}_{2D}$ , where  $\mathbf{L}_{2D}$  can be expressed in terms of the Kronecker product formula (17) with  $\mathbf{L}_{1D}$  defined in the previous paragraph. The corresponding prior is given, again, by (18). Note that  $\mathbf{Q}\mathbf{1} = \mathbf{0}$  and that  $\mathbf{L}_{2D}$  is the 2D discrete negative-Laplacian matrix on the chosen computational grid with periodic BCs; that is,  $\mathbf{L}_{2D}\mathbf{x}$  is the finite difference discretization of the diffusion operator

$$-\frac{d^2x}{ds^2} - \frac{d^2x}{dt^2}, \quad x(0, t) = x(1, t), \quad x(s, 0) = x(s, 1).$$

Note also that in this case,  $\mathbf{L}_{2D}$  has block circulant structure, and hence, can be diagonalized by the 2D-DFT [16].

The realizations from these GMRFs are very similar to those appearing in Figure 1, except that they satisfy periodic (rather than zero) boundary conditions.

2.2.2. *IGMRFs and the discrete negative-Laplacian with Neumann BCs.* Another standard example of an IGMRF results when a Neumann BC is assumed. In 1D this BC is defined  $x'(0) = x'(1) = 0$ , while in 2D it has the form  $x'(0, t) = x'(1, t) = x'(s, 0) = x'(s, 1) = 0$ .

In 1D, the Neumann negative-Laplacian also arises from a statistical model of the *increments*  $\Delta x_i = x_{i+1} - x_i$ . In particular, if we assume

$$(19) \quad \Delta x_i \stackrel{iid}{\sim} \mathcal{N}(0, \delta^{-1}), \quad i = 1, \dots, n-1,$$

then the density function for  $\mathbf{x}$  can be written

$$(20) \quad \begin{aligned} p(\mathbf{x}) &\propto \delta^{(n-1)/2} \exp\left(-\frac{\delta}{2} \sum_{i=1}^{n-1} (x_{i+1} - x_i)^2\right) \\ &= \delta^{(n-1)/2} \exp\left(-\frac{\delta}{2} \mathbf{x}^T (\mathbf{D}^T \mathbf{D}) \mathbf{x}\right), \end{aligned}$$

where

$$(21) \quad \mathbf{D} = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix}_{(n-1) \times n}.$$

Note that (20) has the form (18) with  $\mathbf{L} = \mathbf{D}^T \mathbf{D}$ .

It is easy to verify that  $\mathbf{D}^T \mathbf{D}$  has the same form as  $\mathbf{L}_{1D}$  in (14), but with  $[\mathbf{L}_{1D}]_{11} = [\mathbf{L}_{1D}]_{nn} = 1$ , which is the discrete negative-Laplacian matrix on the computational grid with Neumann (zero flux) BCs; that is,  $\mathbf{L}_{1D}\mathbf{x}$  is the finite different discretization of the diffusion operator

$$-\frac{d^2x}{dt^2}, \quad x'(0) = x'(1) = 0.$$

Note that the BCs  $x'(0) = x'(1) = 0$  are approximated on the computational grid via  $(x_1 - x_0)/\Delta t = (x_{n+1} - x_n)/\Delta t = 0$ , implying  $x_0 = x_1$  and  $x_{n+1} = x_n$ .

Alternatively, we can derive this IGMRF prior via the autoregressive model (12) with neighborhood system  $\partial_1 = \{2\}$ ,  $\partial_i = \{i-1, i+1\}$  for  $i = 2, \dots, n-1$ , and  $\partial_n = \{n-1\}$ . Then the matrix  $\mathbf{Q}$  defined by (13) has the form  $\delta \mathbf{D}^T \mathbf{D}$ .

In 2D, we define the horizontal and vertical increments, respectively, as  $\Delta_h x_{ij} = x_{i+1,j} - x_{ij}$  and  $\Delta_v x_{ij} = x_{i,j+1} - x_{ij}$ , and assume

$$(22) \quad \Delta_h x_{ij}, \Delta_v x_{ij} \stackrel{iid}{\sim} \mathcal{N}(0, \delta^{-1}), \quad i, j = 1, \dots, n-1.$$

Then the density function for  $\mathbf{x}$  has the form (see [16, Chapter 3])

$$\begin{aligned} p(\mathbf{x}) &\propto \delta^{(n^2-1)/2} \exp \left( -\frac{\delta}{2} \left( \sum_{j=1}^n \sum_{i=1}^{n-1} (x_{i+1,j} - x_{ij})^2 + \sum_{i=1}^n \sum_{j=1}^{n-1} (x_{i,j+1} - x_{ij})^2 \right) \right) \\ &= \delta^{(n^2-1)/2} \exp \left( -\frac{\delta}{2} (\|(\mathbf{I} \otimes \mathbf{D})\mathbf{x}\|^2 + \|(\mathbf{D} \otimes \mathbf{I})\mathbf{x}\|^2) \right) \\ (23) \quad &= \delta^{(n^2-1)/2} \exp \left( -\frac{\delta}{2} \mathbf{x}^T (\mathbf{I} \otimes (\mathbf{D}^T \mathbf{D}) + (\mathbf{D}^T \mathbf{D}) \otimes \mathbf{I}) \mathbf{x} \right) \end{aligned}$$

where  $\mathbf{D}$  is defined in (21) and ‘ $\otimes$ ’ denotes Kronecker product. Note that (23) has the form (18) with  $\mathbf{L} = \mathbf{I} \otimes (\mathbf{D}^T \mathbf{D}) + (\mathbf{D}^T \mathbf{D}) \otimes \mathbf{I}$ . Moreover, the precision matrix can once again be written in Kronecker product form (17), this time with  $\mathbf{L}_{1D} = \mathbf{D}^T \mathbf{D}$ .

Also as in the 1D case, the matrix  $\mathbf{L}_{2D} = \mathbf{I} \otimes (\mathbf{D}^T \mathbf{D}) + (\mathbf{D}^T \mathbf{D}) \otimes \mathbf{I}$  is the discrete negative-Laplacian matrix on the computational grid with Neumann BCs; that is  $\mathbf{L}_{2D}\mathbf{x}$  is the finite difference discretization of the diffusion operator

$$-\frac{d^2 x}{ds^2} - \frac{d^2 x}{dt^2}, \quad x'(0, t) = x'(1, t) = x'(s, 0) = x'(s, 1) = 0,$$

where the BCs are approximated as in the 1D case.

Alternatively, as in 1D, we can derive this IGMRF prior via the autoregressive model (15). In this case, the neighborhood system has the form

$$\partial_{ij} = \{(i-1, j), (i+1, j), (i, j-1), (i, j+1)\}, \quad \text{for } i, j = 2, \dots, n-1,$$

whereas if  $i$  or  $j$  is 1 or  $n$ , the vertices containing a 0 or  $n+1$  are removed from  $\partial_{ij}$ ; for example,  $\partial_{1j} = \{(2, j), (1, j-1), (1, j+1)\}$  and  $\partial_{11} = \{(1, 2), (2, 1)\}$ . Note then that  $n_{ij} = |\partial_{ij}|$  is one of 2, 3, or 4. Then the matrix  $\mathbf{Q}$  defined by (13) has the form  $\delta(\mathbf{I} \otimes (\mathbf{D}^T \mathbf{D}) + (\mathbf{D}^T \mathbf{D}) \otimes \mathbf{I})$ .

The realizations from these GMRFs are also very similar in appearance to those in Figure 1, except that they satisfy the Neumann boundary boundary condition.

**2.2.3. IGMRFs for more general diffusion operators.** In certain situations, it is advantageous to allow for the increment variance to be larger in some areas than others. For example, if you know that your image  $\mathbf{x}$  has a sharp intensity change at pixel  $i$ , you should allow for a large increment variance (small precision) at that location, whereas in other areas of the image you might want to enforce a smaller increment variance (larger precision). As we will see, allowing for a spatially varying increment variance is equivalent to discretizing a diffusion equation with a spatially-varying diffusion coefficient.

As in the previous sub-section, we will begin by modeling the increments, and then make connections to numerical PDEs. In the 1D case, the assumption of independent increments corresponds to replacing (19) with

$$(24) \quad \Delta x_i \sim \mathcal{N}(0, (w_i \delta)^{-1}), \quad i = 1, \dots, n-1,$$

where  $\Delta x_i = x_{i+1} - x_i$ . Then the probability density function for  $\mathbf{x}$  has the form

$$\begin{aligned}
 p(\mathbf{x}) &\propto \delta^{(n-1)/2} \exp\left(-\frac{\delta}{2} \sum_{i=1}^{n-1} w_i (x_{i+1} - x_i)^2\right) \\
 &\propto \delta^{(n-1)/2} \exp\left(-\frac{\delta}{2} \|\mathbf{W}^{1/2} \mathbf{D} \mathbf{x}\|^2\right) \\
 (25) \quad &= \delta^{(n-1)/2} \exp\left(-\frac{\delta}{2} \mathbf{x}^T (\mathbf{D}^T \mathbf{W} \mathbf{D}) \mathbf{x}\right),
 \end{aligned}$$

where  $\mathbf{W} = \text{diag}(w_1, \dots, w_{n-1})$  and  $\mathbf{D}$  is as in (21).

The prior (25) corresponds to the auto-regressive model (see [16, Chapter 3])

$$(26) \quad x_i | \mathbf{x}_{\partial_i} \sim \mathcal{N}(\bar{x}_{\partial_i}, (\delta n_i w_i)^{-1}),$$

where  $\bar{x}_{\partial_i} = \frac{1}{n_i} \sum_{j \in \partial_i} x_j$  and the neighborhood system is as in the 1D iid increment case. Comparing with (12), we see that it is very similar except that (26) allows for a spatially varying precision.

Finally, it can be shown that  $\mathbf{D}^T \mathbf{W} \mathbf{D} \mathbf{x}$  is the finite different discretization of the operator

$$-\frac{d}{ds} \left( w(s) \frac{dx}{ds} \right), \quad x'(0) = x'(1) = 0,$$

where  $w_i = w(s_i)$  for  $i = 1, \dots, n-1$ .

Note that one can implement periodic or Dirichlet boundary conditions by modifying the first derivative matrix  $\mathbf{D}$  accordingly. We do this in the following 2D example. Assuming periodic, rather than Neumann, boundary conditions the forward difference matrix becomes

$$(27) \quad \mathbf{D} = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & & \ddots & -1 & 1 \\ 1 & 0 & \cdots & 0 & -1 \end{bmatrix}_{n \times n}.$$

Now we model the horizontal and vertical increments:

$$(28) \quad \Delta_h x_{ij} \sim \mathcal{N}(0, (w_{ij}^h \delta)^{-1}) \quad \text{and} \quad \Delta_v x_{ij} \sim \mathcal{N}(0, (w_{ij}^v \delta)^{-1})$$

for  $i, j = 1, \dots, n$ . Then the density function for  $\mathbf{x}$  has the form

$$\begin{aligned}
 p(\mathbf{x}) &\propto \delta^{(n^2-1)/2} \exp\left(-\frac{\delta}{2} \left( \sum_{j=1}^n \sum_{i=1}^{n-1} w_{ij}^h \Delta_h x_{ij}^2 + \sum_{i=1}^n \sum_{j=1}^{n-1} w_{ij}^v \Delta_v x_{ij}^2 \right)\right) \\
 &= \delta^{(n^2-1)/2} \exp\left(-\frac{\delta}{2} \left( \|\mathbf{W}_h^{1/2} (\mathbf{I} \otimes \mathbf{D}) \mathbf{x}\|^2 + \|\mathbf{W}_v^{1/2} (\mathbf{D} \otimes \mathbf{I}) \mathbf{x}\|^2 \right)\right) \\
 (29) \quad &= \delta^{(n^2-1)/2} \exp\left(-\frac{\delta}{2} \mathbf{x}^T (\mathbf{D}_h^T \mathbf{W}_h \mathbf{D}_h + \mathbf{D}_v^T \mathbf{W}_v \mathbf{D}_v) \mathbf{x}\right),
 \end{aligned}$$

where  $\mathbf{D}_h = \mathbf{I} \otimes \mathbf{D}$  and  $\mathbf{D}_v = \mathbf{D} \otimes \mathbf{I}$  are the discrete horizontal and vertical derivatives, respectively. Moreover,

$$\mathbf{W}_h = \text{diag}(\text{vec}(\{w_{ij}^h\}_{ij=1}^n)) \quad \text{and} \quad \mathbf{W}_v = \text{diag}(\text{vec}(\{w_{ij}^v\}_{ij=1}^n)),$$

where the ‘vec’ function creates vectors from 2D arrays by stacking columns.

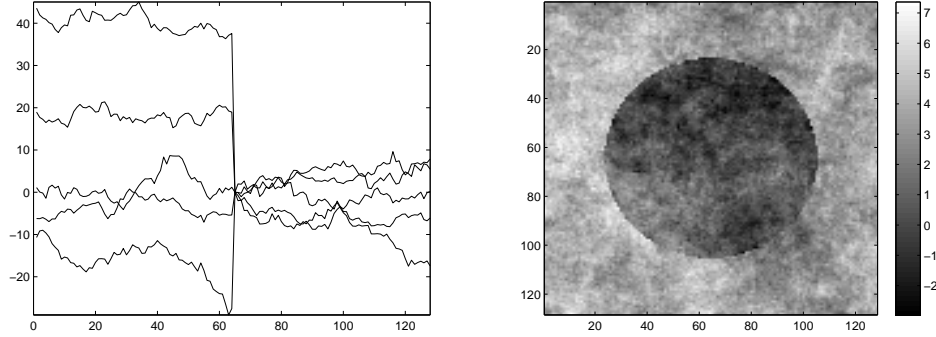


FIGURE 2. On the left, are five realizations from the 1D IGMRF (25) with  $w_{n/2} = 0.05$  and  $w_i = 1$  for all  $i \neq n/2$ . On the right, is a plot of a single realization from the 2D IGMRF (29) with  $w_{ij}^h = w_{ij}^v = 0.001$  along the boundary of a circle contained within  $[0, 1] \times [0, 1]$  and  $w_{ij} = 1$  otherwise.

The prior (29) corresponds to the rather complicated auto-regressive model (see [16, Chapter 3]),

$$x_{ij} | \mathbf{x}_{\partial_{ij}} \sim \mathcal{N} \left( \frac{w_{ij}^h \sum_{(r,s) \in \partial_{ij}^h} x_{rs} + w_{ij}^v \sum_{(r,s) \in \partial_{ij}^v} x_{rs}}{n_{ij}^h w_{ij}^h + n_{ij}^v w_{ij}^v}, \frac{1}{\delta(n_{ij}^h w_{ij}^h + n_{ij}^v w_{ij}^v)} \right),$$

where  $\partial_{ij}^h$  and  $\partial_{ij}^v$  denote the vertical and horizontal neighbors of pixel  $(i, j)$ , respectively, and  $n_{ij}^h = |\partial_{ij}^h|$  and  $n_{ij}^v = |\partial_{ij}^v|$ . This expression simplifies significantly in the case that  $w_{ij} = w_{ij}^h = w_{ij}^v$ , in which case

$$(30) \quad x_{ij} | \mathbf{x}_{\partial_{ij}} \sim \mathcal{N}(\bar{x}_{\partial_{ij}}, (\delta n_{ij} w_{ij})^{-1}),$$

where  $\bar{x}_{\partial_{ij}} = \frac{1}{n_{ij}} \sum_{(r,s) \in \partial_{ij}} x_{rs}$  and  $n_{ij} = |\partial_{ij}|$ . Comparing with (15), we see that as in 1D, (30) is the same, except that it allows for a spatially varying precision.

Finally, it can be shown that  $(\mathbf{D}_h^T \mathbf{W}_h \mathbf{D}_h + \mathbf{D}_v^T \mathbf{W}_v \mathbf{D}_v) \mathbf{x}$  is the finite different discretization of the diffusion operator

$$\frac{\partial}{\partial s} \left( w^h(s, t) \frac{\partial x}{\partial s} \right) + \frac{\partial}{\partial t} \left( w^v(s, t) \frac{\partial x}{\partial t} \right), \quad x(0, t) = x(1, t), \quad x(s, 0) = x(s, 1),$$

with  $w_{ij}^h = w^h(s_i, t_j)$ , and  $w_{ij}^v = w^v(s_i, t_j)$ , for  $i, j = 1, \dots, n$ .

In Figure 2, we plot realizations from examples of such IGMRF priors. On the left, we plot five realizations from the 1D-IGMRF (25) with  $w_{n/2} = 0.05$  and  $w_i = 1$  for all  $i \neq n/2$ . Note that decreasing the precision (increasing the variance) of the increment at pixel  $n/2$  allows for a large jump at pixel  $n/2$ . On the right, we plot a single realization from the 2D-IGMRF (29) with  $w_{ij}^h = w_{ij}^v = 0.001$  along the boundary of a circle contained within  $[0, 1] \times [0, 1]$ , and otherwise let  $w_{ij} = 1$ . This realization shows that the prior allows for large jumps along the boundary of the circle, where the increment variance is larger, while within and without the circle boundary the behavior is the same as in the realization in Figure 1.

Conditional Autoregressive Model	Partial Differential Equation	Regularization or Precision Matrix
$x_i   \mathbf{x}_{\partial_i} \sim \mathcal{N}(\bar{x}_{\partial_i}, (n_i w_i)^{-1})$	$-\frac{d}{ds} \left( w(s) \frac{dx}{ds} \right)$	$\mathbf{D}^T \mathbf{W} \mathbf{D}$
$x_{ij}   \mathbf{x}_{\partial_{ij}} \sim \mathcal{N}(\bar{x}_{\partial_{ij}}, (n_{ij} w_{ij})^{-1})$	$-\frac{\partial}{\partial s} \left( w(s, t) \frac{\partial x}{\partial s} \right) - \frac{\partial}{\partial t} \left( w(s, t) \frac{\partial x}{\partial t} \right)$	$\mathbf{D}_h^T \mathbf{W} \mathbf{D}_h + \mathbf{D}_v^T \mathbf{W} \mathbf{D}_v$

TABLE 1. Conditional autoregressive models and diffusion equations yielding the same regularization/precision matrix. The table does not contain mention of the boundary conditions, the parameter  $\delta$ , or the case when  $w^h \neq w^v$ . Note that  $w \equiv 1$  yields the negative-Laplacian.

Assuming that  $w_v = w_h = w$  in the 2D case, the connection between the above Gaussian conditional autoregressive models, finite difference discretizations of diffusion equations, and regularization matrices is summarized in Table 1.

**Remark 1.** A similar discussion of Gaussian prior models for inverse problems derived from the assumption of independent Gaussian increments can be found in [7]. However, neither the conditional autoregressive models, nor the connection to discretized partial differential equations, outlined in Table 1, appear in that paper.

### 3. EDGE-PRESERVING MAP ESTIMATION USING GMRP PRIORS

In this section, we begin with the Gaussian noise model (1) and the application of image deblurring. We then move on to the Poisson noise model (2) and the application of positron emission tomography (PET).

3.1. GAUSSIAN NOISE MODEL. In this section, we implement the GMRP priors from the previous section for edge-preserving image reconstruction. Edge-preserving priors require the IGMRPs discussed in Section 2.2.3, i.e. those that correspond to a finite difference discretization of the PDE

$$(31) \quad -\frac{\partial}{\partial s} \left( w(s, t) \frac{\partial x}{\partial s} \right) - \frac{\partial}{\partial t} \left( w(s, t) \frac{\partial x}{\partial t} \right).$$

After discretization, (31) takes the form

$$(32) \quad \mathbf{L} \mathbf{x} = (\mathbf{D}_h^T \mathbf{W} \mathbf{D}_h + \mathbf{D}_v^T \mathbf{W} \mathbf{D}_v) \mathbf{x}.$$

The difficulty in using (32) is that a matrix  $\mathbf{L}$  that will preserve edges in the unknown image  $\mathbf{x}$  will depend upon  $\mathbf{x}$  through  $\mathbf{W}$ . This can be overcome by defining  $\mathbf{W}$  using a previously computed estimate  $\mathbf{x}^k$  of  $\mathbf{x}$ . For example, given  $\mathbf{x}^k$ , we can make the total variation choice for  $\mathbf{W} = \mathbf{W}^k$  [18], which has diagonal values

$$(33) \quad [\mathbf{W}^k]_{ii} = 1 / \sqrt{[\mathbf{D}_h \mathbf{x}^k]_i^2 + [\mathbf{D}_v \mathbf{x}^k]_i^2 + \beta}, \quad i = 1, \dots, n,$$

where  $\beta > 0$ . This suggests the following edge-preserving reconstruction algorithm for the Gaussian noise case (1).

**Algorithm 1.** Set  $k = 0$ ,  $\mathbf{W}^0 = \mathbf{I}$ , and iterate the following.

1. Define  $\mathbf{L} = \mathbf{D}_h^T \mathbf{W}^k \mathbf{D}_h + \mathbf{D}_v^T \mathbf{W}^k \mathbf{D}_v$ .
2. Compute  $\alpha$  every  $j$ th iteration using generalized cross validation.
3. Compute the solution  $\mathbf{x}_\alpha^{k+1} = \arg \min_{\mathbf{x} \in \Omega} \left\{ \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 + \frac{\alpha}{2} \mathbf{x}^T \mathbf{L}\mathbf{x} \right\}$ .
4. Set  $\mathbf{W}^{k+1} = \mathbf{W}(\mathbf{x}_\alpha^{k+1})$  using (33) and  $k = k + 1$ . Return to Step 1.

Generalized cross validation (GCV) – the method for estimating  $\alpha$  mentioned in Step 2 – is a well-known regularization parameter selection method for inverse problems [10, 18]. The GCV choice for  $\alpha$  minimizes the function

$$(34) \quad G(\alpha) = \frac{n \|\mathbf{A}\mathbf{x}_\alpha - \mathbf{b}\|^2}{[\text{tr}(\mathbf{I} - \mathbf{A}\mathbf{A}_\alpha)]^2},$$

where ‘tr’ denotes matrix trace,  $\mathbf{A}_\alpha = (\mathbf{A}^T \mathbf{A} + \alpha \mathbf{L})^{-1} \mathbf{A}^T$ , and  $\mathbf{x}_\alpha = \mathbf{A}_\alpha \mathbf{b}$ . When parameters in the prior are estimated from the data, such as is the case here, the approach is called *empirical Bayes* [9].

Note that Algorithm 1 is precisely the *lagged-diffusivity fixed point iteration* of [19], except that the regularization parameter  $\alpha$  is updated at every  $j$ th iteration, whereas in [19]  $\alpha$  is fixed at the outset.

**Remark 2.** Looking at autoregressive model (30), in light of (33), we see that the use of the regularization matrix  $\mathbf{L}$  defined by (32) results in a large increment variance at pixels where the norm of the gradient is large; specifically, note that if  $\mathbf{x}^k$  is the previously computed estimate of  $\mathbf{x}$ , the prior stems from the autoregressive model (30) with  $w_{ij} = ([\mathbf{D}_h \mathbf{x}^k]_{ij}^2 + [\mathbf{D}_v \mathbf{x}^k]_{ij}^2 + \beta)^{-1/2}$ .

3.1.1. *Image Deblurring Test Case.* Now we test the above iteration on a two-dimensional image deblurring problem. The forward model, mapping the unknown  $x$  to the observation  $b$ , both defined on  $[0, 1] \times [0, 1]$ , has convolutions form:

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

For our experiments, we choose a Gaussian convolution kernel  $A$ , and discretize the integral using mid-point quadrature on a  $128 \times 128$  uniform computational grid over  $[0, 1] \times [0, 1]$ . We assume that  $x$  extends periodically outside of  $[0, 1] \times [0, 1]$ , which after discretization yields a linear system of equations  $\mathbf{b} = \mathbf{A}\mathbf{x}$  in which  $\mathbf{A}$  has block circulant structure. Thus  $\mathbf{A}$  can be diagonalized by the discrete Fourier transform (DFT) [18].

The data  $\mathbf{b}$  is generated using (1) with the noise variance  $\lambda^{-1}$  chosen so that the noise strength is 2% that of the signal strength. The image used to generate the data and the data itself are shown in Figure 3.

Next, we perform our reconstruction technique using Algorithm 1. It remains to define the discrete horizontal and vertical derivative matrices:  $\mathbf{D}_h = \mathbf{I} \otimes \mathbf{D}$  and  $\mathbf{D}_v = \mathbf{D} \otimes \mathbf{I}$ , where assuming periodic boundary conditions, we have  $\mathbf{D}$  defined by (27). Nonetheless, the precision matrix  $\mathbf{D}_h^T \mathbf{W} \mathbf{D} + \mathbf{D}_v^T \mathbf{W} \mathbf{D}_v$  does not generally have block circulant structure, and hence cannot be diagonalized by the DFT. Thus we must use the preconditioned conjugate gradient method [18, 20] (PCG) to compute  $\mathbf{x}_\alpha^{k+1}$  in Step 3 of Algorithm 1. For the preconditioner we use  $\mathbf{M} = \mathbf{A}^T \mathbf{A} + \alpha(\mathbf{D}_h^T \mathbf{D}_h + \mathbf{D}_v^T \mathbf{D}_v)$ , which is diagonalizable by the DFT; indeed,

$$\mathbf{M}^{-1} \mathbf{r} = \text{vec} \left( \text{IDFT} \left( \frac{\mathbf{1}}{|\hat{\mathbf{a}}_s|^2 + \alpha \hat{\mathbf{1}}_s} \odot \text{DFT}(\mathbf{R}) \right) \right),$$

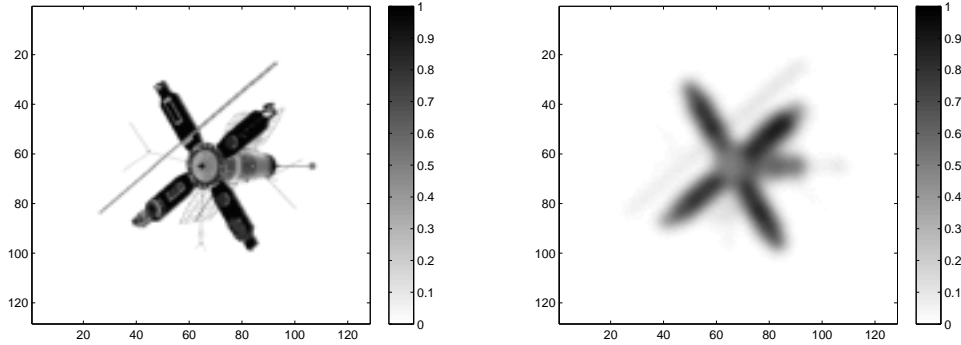


FIGURE 3. On the left is the two-dimensional image used to generate the data, and on the right is the blurred noisy data.

where  $\mathbf{R}$  is an  $n \times n$  array,  $\mathbf{r} = \text{vec}(\mathbf{R})$ ,  $\hat{\mathbf{I}}_s$  is the  $n \times n$  eigenvalue array of  $\mathbf{D}_h^T \mathbf{D}_h + \mathbf{D}_v^T \mathbf{D}_v$ ,  $\hat{\mathbf{a}}_s$  is the  $n \times n$  eigenvalue array of  $\mathbf{A}$ , and IDFT is the inverse discrete Fourier transform. See [18] for more detail on the diagonalization of the block circulant matrices by the DFT.

Finally, implementing the GCV method in Step 2 of Algorithm 1 requires repeated evaluation of the GCV function  $G$  defined in (34), which in turn requires a matrix trace computation that is infeasible for our large-scale example. However, the trace can be estimated using *randomized trace estimation* [18], which is motivated from the fact that if  $\mathbf{v}$  is a white noise random vector and  $\mathbf{C}$  is a symmetric matrix, then  $E(\mathbf{v}^T \mathbf{C} \mathbf{v}) = \text{tr}(\mathbf{C})$ , where  $E$  denotes the expected value function. Thus if we choose  $\mathbf{v}$  to be a realization of a  $n^2 \times 1$  white noise vector,

$$\text{tr}(\mathbf{I} - \mathbf{A} \mathbf{A}_\alpha) \approx n^2 - \mathbf{v}^T \mathbf{A} \mathbf{A}_\alpha \mathbf{v},$$

where  $\mathbf{I}$  is the  $n^2 \times n^2$  identity matrix. The choice of  $\mathbf{v}$  that minimizes the variance in this estimator is the one for which the components of  $\mathbf{v}$  are independent and take on the values of 1 and -1 with equal probability (see [18] and the references therein). Note that to compute  $\mathbf{A}_\alpha \mathbf{v}$  we must solve the linear system  $(\mathbf{A}^T \mathbf{A} + \alpha \mathbf{L}) \mathbf{w} = \mathbf{A}^T \mathbf{v}$  for  $\mathbf{w}$  using PCG. Thus the computation of the trace approximation, and hence each evaluation of the GCV function, requires an additional PCG run. Finally, we note that to minimize the GCV function (34), we use MATLAB's `fminbnd` function.

We are now ready to test Algorithm 1 on the above image deblurring test problem. We choose  $\beta = 0.001$  in (33) and show reconstructions for two different implementations. First, we compute  $\alpha$  only in the first iteration, which can be done very efficiently because in that case  $\mathbf{L}$  has block circulant structure [18], and otherwise iterate the method 10 times. The corresponding reconstruction is given on the upper-left in Figure 4, while the edge map, which is a plot of the diagonal values of  $\mathbf{W}$  defined by (33), is plotted on the upper-right. Finally, the relative error for the final reconstruction was  $\|\mathbf{x}_\alpha - \mathbf{x}_{\text{true}}\| / \|\mathbf{x}_{\text{true}}\| = 0.2056$ . Next, compute 10 iterations of Algorithm 1 with  $\alpha$  computed in the first, and then every third, iteration, i.e. 1, 3, 6, 9. The corresponding reconstruction is given on the lower-left in Figure 4, while the edge map is plotted on the lower-right. Finally, the relative error for the final reconstruction was 0.1959.

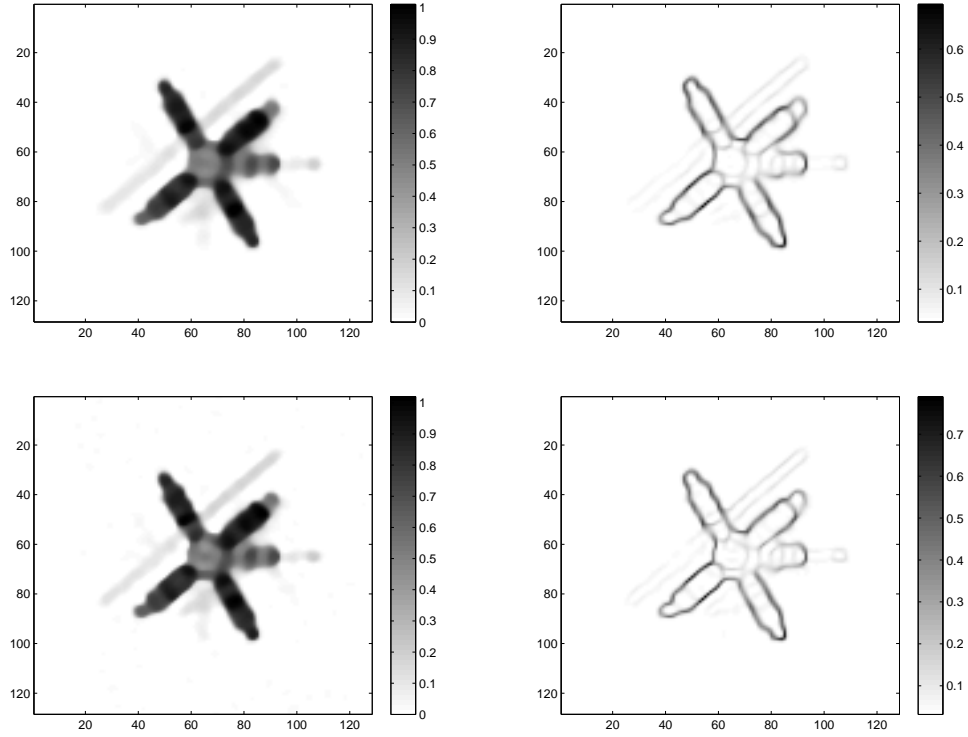


FIGURE 4. Results obtained using Algorithm 1 on an image deblurring test case. The upper plots correspond to using GCV once in iteration 1 and taking 10 iterations, while the lower plots correspond to using GCV in the first, third, sixth, and ninth iterations.

3.2. POISSON NOISE MODEL. For the Poisson noise case, we use the same precision matrix  $\mathbf{L}$  defined by (32), but model  $\mathbf{W}$  differently; this time using the Bayesian hierarchical modelling approach of [7, 1]. The idea is to modify the increment model (28) as follows:

$$(35) \quad \Delta_h x_{ij} \sim \mathcal{N}(0, \theta_{ij}) \quad \text{and} \quad \Delta_v x_{ij} \sim \mathcal{N}(0, \theta_{ij}).$$

Note that the parameter  $\delta$  does not appear in (35) and hence the full precision matrix has the form (32) with

$$\mathbf{W} = \mathbf{W}_\theta = \text{diag}(\theta_1^{-1}, \dots, \theta_n^{-1}).$$

Then the prior probability density has the form

$$(36) \quad \begin{aligned} p(\mathbf{x}|\boldsymbol{\theta}) &\propto \det(\mathbf{W}_\theta) \exp\left(-\frac{1}{2}\mathbf{x}^T(\mathbf{D}_v^T \mathbf{W}_\theta \mathbf{D}_v + \mathbf{D}_h^T \mathbf{W}_\theta \mathbf{D}_h)\mathbf{x}\right) \\ &= \exp\left(-\frac{1}{2}\mathbf{x}^T(\mathbf{D}_v^T \mathbf{W}_\theta \mathbf{D}_v + \mathbf{D}_h^T \mathbf{W}_\theta \mathbf{D}_h)\mathbf{x} - \sum_{i=1}^N \ln \theta_i\right). \end{aligned}$$

Next, following [1, 7], we assume a hierarchical model for the  $\theta_{ij}$ 's:

$$(37) \quad \theta_{ij} \sim \text{Gamma}(\alpha_0, \theta_0),$$

which implies

$$\begin{aligned} p(\boldsymbol{\theta}) &\propto \prod_{i=1}^{n^2} \theta_i^{\alpha_0-1} \exp(\theta_i/\theta_0) \\ &= \exp\left((\alpha_0 - 1) \sum_{i=1}^{n^2} \ln \theta_i + \sum_{i=1}^{n^2} \theta_i/\theta_0\right) \end{aligned}$$

We choose  $\alpha_0$  and  $\theta_0$  so that the mean of the Gamma distribution (37), i.e.  $\alpha_0\theta_0$ , is equal to  $\alpha^{-1}$  where  $\alpha$  is the value of the regularization parameter obtained from the GCV method for Poisson inverse problems described in [3]. This method chooses  $\alpha$  for the regularized solution given by

$$\mathbf{x}_\alpha = \arg \min_{\mathbf{x} \geq \mathbf{0}} \left\{ \sum_{i=1}^{n^2} \{[\mathbf{Ax}]_i + \gamma_i - b_i \ln([\mathbf{Ax}]_i + \gamma_i)\} + \frac{\alpha}{2} \mathbf{x}^T (\mathbf{D}_h^T \mathbf{D}_h + \mathbf{D}_v^T \mathbf{D}_v) \mathbf{x} \right\}$$

using a technique based on the GCV method presented above. Note that this regularization function corresponds to the prior  $p(\mathbf{x}|\boldsymbol{\theta})$  given in (36) with  $\boldsymbol{\theta} = \alpha\mathbf{1}$ . Below we will suggest a specific value for  $\theta_0$  that yields an edge-preserving GMRP prior; then  $\alpha_0 = 1/(\alpha\theta_0)$  and both parameters of the distribution (37) have been defined.

Given all of the above choices, we can now define the posterior density function  $p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{b})$  using Bayes' Law:

$$\begin{aligned} (38) \quad -\ln p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{b}) &= -\ln(p(\mathbf{b}|\mathbf{x})p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})) \\ &\simeq \sum_{i=1}^{n^2} \{([\mathbf{Ax}]_i + \gamma_j) - b_j \ln([\mathbf{Ax}]_j + \gamma_j)\} + \frac{1}{2} \mathbf{x}^T \mathbf{L} \mathbf{x} \\ &\quad + \sum_{i=1}^{n^2} \frac{\theta_i}{\theta_0} - (\alpha_0 - 2) \sum_{i=1}^{n^2} \ln \theta_j, \end{aligned}$$

where  $\simeq$  means ‘‘is equal up to an additive unimportant constant’’, and  $\mathbf{L} = \mathbf{D}_v^T \mathbf{W}_\theta \mathbf{D}_v + \mathbf{D}_h^T \mathbf{W}_\theta \mathbf{D}_h$  with  $\mathbf{W}_\theta = \text{diag}(\theta_1^{-1}, \dots, \theta_{n^2}^{-1})$ .

We obtain our reconstructed image  $\mathbf{x}$  and edge map  $\boldsymbol{\theta}$  by minimizing (38). We do this, as in [1], using the following method.

**Algorithm 2.** Set  $k = 1$  and  $\boldsymbol{\theta}^0 = \alpha^{-1}\mathbf{1}$ , with  $\alpha$  chosen using GCV from [3].

1. Compute  $\mathbf{x}^k = \text{argmin}_{\mathbf{x} \geq \mathbf{0}} -\ln p(\mathbf{x}|\mathbf{b}, \boldsymbol{\theta}^{k-1})$ .
2. Compute  $\boldsymbol{\theta}^k = \text{argmin}_{\boldsymbol{\theta}} -\ln p(\boldsymbol{\theta}|\mathbf{b}, \mathbf{x}^k)$ .
3. Set  $k = k + 1$  and return to Step 1.

In Step 1, we use the nonnegativity constrained convex programming method of [2, 4] for computing  $\mathbf{x}$ . Whereas in Step 2, the  $\boldsymbol{\theta}$  update has the form

$$\theta_j^k = \theta_0 \left( \frac{\alpha_0 - 2}{2} + \sqrt{\frac{[\mathbf{D}_1 \mathbf{x}^k]_j^2 + [\mathbf{D}_2 \mathbf{x}^k]_j^2}{2\theta_0} + \frac{(\alpha_0 - 2)^2}{4}} \right).$$

We choose  $\alpha_0 = 2$ , which yields the total variation update, as in the lagged-diffusivity fixed point iteration [19]. Thus  $\theta_0 = 1/(2\alpha)$ . Substituting these values

into the above equation and adding a parameter  $\beta > 0$  to avoid  $\theta_j = 0$ , yields the Step 2 update

$$(39) \quad \theta_j^k = \frac{1}{2\sqrt{\alpha}} \sqrt{[\mathbf{D}_1 \mathbf{x}^k]_j^2 + [\mathbf{D}_2 \mathbf{x}^k]_j^2 + \beta}.$$

3.2.1. *PET and SPECT Test Cases.* In both positron emission tomography (PET) and single photon emission tomography (SPECT), a radioactive tracer element is injected into the body, which exhibits radioactive decay, resulting in photon emission. The emitted photons that leave the body are recorded by a photon detector, which also determines the line of response (LOR)  $L(\omega, y)$ , along which the photon(s) have propagated; given a fixed coordinate system,  $L(\omega, y)$  is the unique line making an angle  $\omega$  with an axis (e.g. the vertical) that is a perpendicular distance of  $y$  from the origin. We parameterize  $L(\omega, y)$  by  $L(\omega, y) = \{z(s) \mid 0 \leq s \leq S\}$ .

In both PET and SPECT, the data  $b(\omega, y)$  corresponds to the number of detected incidents along  $L(\omega, y)$ . The model relating the tracer density  $x$  to the data is given by

$$b(\omega, y) = \int_{L(\omega, y)} A_{\omega, y}(z(s)) x(z(s)) ds,$$

where the impulse response function  $A_{\omega, y}(z(r))$  can be viewed as the probability that an emission event located at  $z(r)$  along  $L(\omega, y)$  is recorded by the detector system.

For SPECT, single photons are detected by single detectors. Given an emission event at location  $z(r)$  emitted along  $L(\omega, y)$  with the detector located at  $z(S)$ , the impulse response function has the form  $A_{\omega, y}(z(r)) = e^{-\int_r^S \mu(z(t)) dt}$ , where the function  $\mu(z)$  is the mass absorption of the body being imaged and is assumed to be known. The full model then becomes

$$(40) \quad b(\omega, y) = \int_{L(\omega, y)} e^{-\int_s^S \mu(z(t)) dt} x(z(s)) ds,$$

where the interior integral is along the line  $L(\omega, y)$  [14].

For PET, a pair of photons are emitted at a location  $z(r)$  along  $L(\omega, y)$  with detectors located at  $z(0)$  and  $z(S)$ . In this case, the impulse response is the product of probabilities:

$$A_{\omega, y}(z(r)) = e^{-\int_0^r \mu(z(t)) dt} e^{-\int_r^S \mu(z(t)) dt} = e^{-\int_{L(\omega, y)} \mu(z(t)) dt},$$

which doesn't depend on  $r$  and  $\mu(z)$  is as in the SPECT model (40). Hence we have the somewhat simpler mathematical model

$$(41) \quad b(\omega, y) = e^{-\int_{L(\omega, y)} \mu(z(t)) dt} \int_{L(\omega, y)} x(z(s)) ds.$$

In fact, dividing both sides of (41) by  $e^{-\int_{L(\omega, y)} \mu(z(t)) dt}$  yields the Radon transform model, which is what is solved in the computed tomography inverse problem [14].

After discretization, (40) and (41) can be written as a system of linear equations of the form (1). The discretization occurs both in the spatial domain, where  $\mu$  and  $x$  are defined, as well as in the Radon transform  $((\omega, y))$  domain, where the data  $b$  is defined. We use a uniform  $n \times n$  spatial grid, and a uniform grid for the transform domain with  $n$  angles and  $n$  sensors. In our experiments,  $n = 128$  so that (1) has size  $128^2 \times 128^2$ .

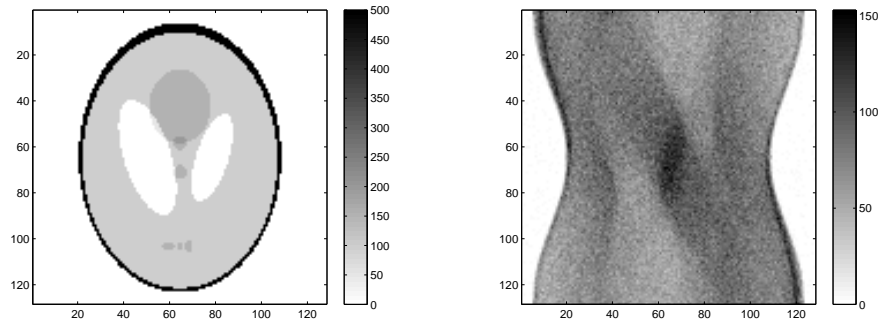


FIGURE 5. On the left is the true image, and on the right is the PET data.

We generate data and perform experiments only for the PET data, as the results in the SPECT case are very similar and the algorithm works equally well. To generate the data  $\mathbf{b}$ , we use the Poisson noise model (2) with  $\gamma = \mathbf{1}$  and MATLAB's `poissrnd` function. The true tracer density, given on the left in Figure 5, is the Shepp-Logan phantom generated using MATLAB's `phantom` function. We take  $\mu = 0$  in (41) to construct our matrix  $\mathbf{A}$ , which is standard for PET numerical experiments [15], and scale the true tracer density  $\mathbf{x}$  so that the percent-noise is approximately 11. The data is shown on the right in Figure 5.

Finally, we test Algorithm 2 on this synthetic PET example. We choose  $\beta = 0.001$  in (39). In Figure 6, we plot the reconstruction  $\mathbf{x}^k$  for  $k = 1$  (which is the solution of (38) with  $\alpha$  chosen using the GCV method of [3]) in the upper-left, and for  $k = 10$  in the lower-left. The edge-maps, which here we define as  $\ln \boldsymbol{\theta}^k$ , are in the upper-right for  $k = 1$  and lower-right for  $k = 10$ , in Figure 6. The relative errors for these reconstructions was 0.3023 for  $k = 1$  and 0.2073 for  $k = 10$ .

#### 4. CONCLUSIONS

A significant portion of our focus in this paper is on the connections between Gaussian Markov random field (GMRF) priors, finite difference discretizations of diffusion equations (elliptic PDEs), and quadratic regularization functions.

The diffusion equations considered include the negative-Laplacian with Dirichlet, Neumann, and periodic boundary conditions, as well as more general diffusion operators with a spatially varying diffusion coefficient and one of the above boundary conditions. We show that numerically discretizing these PDEs on a uniform computational grid yields a GMRF prior or, equivalently, a quadratic regularization function.

The power of drawing the connection between these discretized PDEs, regularization functions, and GMRF priors is that a GMRF corresponds to a set of concrete statistical assumptions regarding the conditional distributions for the unknown intensity  $x_i$  given the intensities of its neighbors  $\mathbf{x}_{\partial_i}$ . In particular, we assume that  $x_i | \mathbf{x}_{\partial_i}$  is Gaussian with mean equal to the mean of the values in  $\mathbf{x}_{\partial_i}$  and some unknown, to-be-determined variance related to the regularization parameter. Thus the regularization term corresponds to a concrete spatial statistical model for the unknown, encapsulated in the prior. These connections are summarized in Table 1.

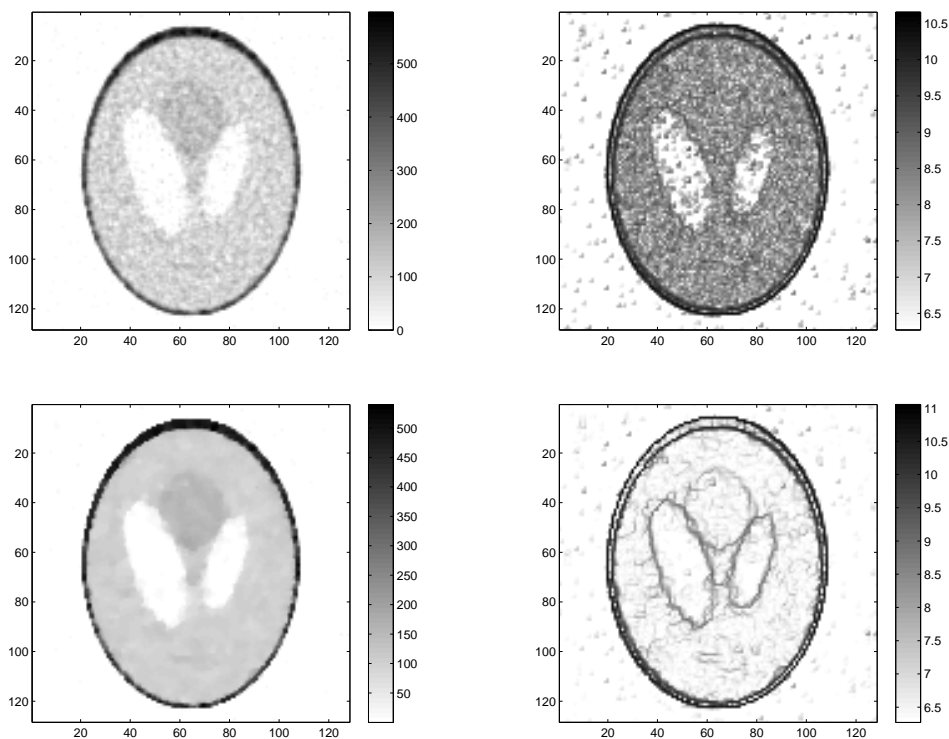


FIGURE 6. Results obtained using Algorithm 2 on a PET synthetic test case. The upper plots correspond to iteration 1 and the lower plots to iteration 10 iteration.

Finally, with various GMRFs in hand, we presented two iterative methods for edge-preserving image reconstruction, one applied to image deblurring in the Gaussian noise case, and the other applied to the positron emission tomography image reconstruction problem with Poisson noise. In the first algorithm, we draw close connections to the well-known lagged-diffusivity fixed point iteration of [18], while in the other we extend the Bayesian paradigm and assume a probability model for the increment variances  $\theta_i$ . We then optimize with respect to both  $\mathbf{x}$  and  $\boldsymbol{\theta}$ , to obtain both a reconstructed image as well as an edge-map. In both cases, the techniques and algorithms work well.

## REFERENCES

- [1] J. M. BARDSLEY, D. CALVETTI, AND E. SOMERSALO, *Hierarchical regularization for edge-preserving reconstruction of PET images*, *Inverse Problems*, **26** (2010), 035010.
- [2] J. M. BARDSLEY AND J. GOLDES, *An Iterative Method for Edge-Preserving MAP Estimation when Data-Noise is Poisson*, *SIAM Journal on Scientific Computing*, **32(1)**, (2010), pp. 171-185.
- [3] J. M. BARDSLEY AND J. GOLDES, *Regularization Parameter Selection Methods for Ill-Posed Poisson Maximum Likelihood Estimation*, *Inverse Problems*, **25** (2009) 095005.
- [4] J. M. BARDSLEY AND C. R. VOGEL, *A Nonnegatively Constrained Convex Programming Method for Image Reconstruction*, *SIAM Journal on Scientific Computing*, 25(4), 2004, pp. 1326-1343.

- [5] J. BESAG, *Spatial Interaction and the Statistical Analysis of Lattice Systems*, Journal of the Royal Statistical Society, Series B, **36(2)** (1974), pp. 192-236.
- [6] D. CALVETTI AND E. SOMERSALO, *Introduction to Bayesian Scientific Computing*, Springer 2007.
- [7] D. CALVETTI AND E. SOMERSALO, *Hypermmodels in the Bayesian Imaging Framework*, Inverse Problems, **24(3)** (2008), 034013.
- [8] H. K. ENGL, M. HANKE, AND A. NEUBAUER, *Regularization of Inverse Problems*, Kluwer, 1996.
- [9] 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.
- [10] P. C. HANSEN, *Discrete Inverse Problems: Insight and Algorithms*, SIAM, Philadelphia, 2010.
- [11] D. HIGDON, *A primer on space-time modelling from a Bayesian perspective*, Los Alamos Nation Laboratory, Statistical Sciences Group, Technical Report, LA-UR-05-3097.
- [12] J. KAIPIO AND E. SOMERSALO, *Statistical and Computational Inverse Problems*, Springer 2005.
- [13] D. KINCAID AND W. CHENEY, *Numerical Analysis: Mathematics of Scientific Computing*, Brooks/Cole, 2002.
- [14] F. NATTERER AND F. WÜBBELING, *Mathematical Methods in Image Reconstruction*, SIAM 2001.
- [15] J. M. OLLINGER AND J. A. FESSLER, *Positron-Emission Tomography*, IEEE Signal Processing Magazine, pp. 43-55, (January 1997).
- [16] H. RUE AND L. HELD, *Gaussian Markov Random Fields: Theory and Applications*, Chapman and Hall/CRC, 2005.
- [17] A. M. STUART, *Inverse problems: a Bayesian approach*, Acta Numerica, **19**, 2010.
- [18] C. R. VOGEL, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, 2002.
- [19] C. R. VOGEL AND M. E. OMAN, *A fast, robust algorithm for total variation based reconstruction of noisy, blurred images*, IEEE Transactions on Image Processing, **7** (1998), pp. 813-824.
- [20] D. WATKINS, *Fundamentals of Matrix Computations*, Wiley, 2010.

*E-mail address:* bardsleyj@mso.umt.edu