

## AN EFFICIENT COMPUTATIONAL METHOD FOR TOTAL VARIATION-PENALIZED POISSON LIKELIHOOD ESTIMATION

JOHNATHAN M. BARDSLEY

Department of Mathematical Sciences  
University of Montana  
Missoula, Montana 59812, USA

ABSTRACT. Approximating non-Gaussian noise processes with Gaussian models is standard in data analysis. This is due in large part to the fact that Gaussian models yield parameter estimation problems of least squares form, which have been extensively studied both from the theoretical and computational points of view. In image processing applications, for example, data is often collected by a CCD camera, in which case the noise is a Gaussian/Poisson mixture with the Poisson noise dominating for a sufficiently strong signal. Even so, the standard approach in such cases is to use a Gaussian approximation that leads to a negative-log likelihood function of weighted least squares type.

In the Bayesian point-of-view taken in this paper, a negative-log prior (or regularization) function is added to the negative-log likelihood function, and the resulting function is minimized. We focus on the case where the negative-log prior is the well-known total variation function and give a statistical interpretation. Regardless of whether the least squares or Poisson negative-log likelihood is used, the total variation term yields a minimization problem that is computationally challenging. The primary result of this work is the efficient computational method that is presented for the solution of such problems, together with its convergence analysis. With the computational method in hand, we then perform experiments that indicate that the Poisson negative-log likelihood yields a more computationally efficient method than does the use of the least squares function. We also present results that indicate that this may even be the case when the data noise is i.i.d. Gaussian, suggesting that regardless of noise statistics, using the Poisson negative-log likelihood can yield a more computationally tractable problem when total variation regularization is used.

### 1. INTRODUCTION

We begin with the discrete linear equation

$$(1) \quad \mathbf{z} = \mathbf{A}\mathbf{u},$$

where  $\mathbf{A} \in \mathbb{R}^{N \times N}$  and  $\mathbf{z} \in \mathbb{R}^N$  are known and  $\mathbf{u} \in \mathbb{R}^N$  is unknown. In this paper,  $\mathbf{z}$  and  $\mathbf{u}$  correspond to lexicographically ordered two-dimensional arrays of size  $\sqrt{N} \times \sqrt{N}$ . We will consider two well-known cases: (i)  $\mathbf{A}$  corresponds to a discretization of a two-dimensional convolution operator, and (ii)  $\mathbf{A} = \mathbf{I}$  is the  $N \times N$  identity matrix. The problem of estimating  $\mathbf{u}$  from  $\mathbf{z}$  given  $\mathbf{A}$  is known in the first case as deconvolution and in the second case as denoising.

---

2000 *Mathematics Subject Classification*: 65J22, 65K10, 65F22.

*Key words and phrases*: total variation, nonnegatively constrained optimization, image reconstruction, Bayesian statistical methods.

Email: [bardsleyj@mso.umt.edu](mailto:bardsleyj@mso.umt.edu). This work was done during this author's visit to the University of Helsinki, Finland in 2006-07 under the University of Montana Faculty Exchange Program.

We start with (1) because it is observations about discrete images that will provide motivation for the Bayesian approach that we will take in the formulation of our inverse problem. The Bayesian statistical point of view has seen increasing use as a means of rigorously formulating regularization techniques for classical inverse problems [10]. In this approach, the incorporation of *a priori* knowledge about the statistical characteristics of the noise contained in the data  $\mathbf{z}$  and of the unknown discrete true image  $\mathbf{u}$  is advocated.

For any image collected by a CCD camera, the noise contained in  $\mathbf{z}$  follows a well-known distribution [15]. In particular,  $\mathbf{z}$  is a realization of the random vector

$$(2) \quad \hat{\mathbf{z}} = \text{Poiss}(\mathbf{A}\mathbf{u}) + \text{Poiss}(\gamma \cdot \mathbf{1}) + N(\mathbf{0}, \sigma^2\mathbf{I}).$$

Here  $\mathbf{1}$  is an  $N \times 1$  vector of all ones. By (2) we mean that  $\hat{\mathbf{z}}$  is the sum of three random vectors: the first two are Poisson with Poisson parameter vectors  $\mathbf{A}\mathbf{u}$  and  $\gamma \cdot \mathbf{1}$  respectively, and the third is Normal with mean vector  $\mathbf{0}$  and covariance matrix  $\sigma^2\mathbf{I}$ . Following arguments found in [4], we first estimate (2) by

$$(3) \quad \hat{\mathbf{z}} + \sigma^2 \cdot \mathbf{1} = \text{Poiss}(\mathbf{A}\mathbf{u} + (\gamma + \sigma^2) \cdot \mathbf{1}),$$

which has probability density function

$$(4) \quad p_{\mathbf{z}}(\mathbf{z}; \mathbf{u}) := \prod_{i=1}^N \frac{([\mathbf{A}\mathbf{u}]_i + \gamma + \sigma^2)^{z_i + \sigma^2} \exp[-([\mathbf{A}\mathbf{u}]_i + \gamma + \sigma^2)]}{z_i + \sigma^2}.$$

We note that since Poisson random variables take on only discrete values,  $p_{\mathbf{z}}(\mathbf{z}; \mathbf{u})$  should, in theory, be positive only for  $\mathbf{z} \in \mathbb{Z}_+^N$ . However to ease in both analysis and computation, we will treat  $p_{\mathbf{z}}$  as a probability density defined on  $\mathbb{R}_+^N \cup \{0\}$ .

In the Bayesian approach, a prior probability density  $p_{\mathbf{u}}(\mathbf{u})$  for  $\mathbf{u}$  is also specified, and the posterior density

$$(5) \quad p_{\mathbf{u}}(\mathbf{u}; \mathbf{z}) := \frac{p_{\mathbf{z}}(\mathbf{z}; \mathbf{u})p_{\mathbf{u}}(\mathbf{u})}{p_{\mathbf{z}}(\mathbf{z})},$$

given by Bayes' Law, is maximized with respect to  $\mathbf{u}$ . The maximizer of  $p_{\mathbf{u}}(\mathbf{u}; \mathbf{z})$  is called the maximum a posteriori (MAP) estimate. We note that since  $p_{\mathbf{z}}(\mathbf{z})$  does not depend on  $\mathbf{u}$ , it is not needed in the computation of the MAP estimate and thus, can be ignored. Maximizing (5) is equivalent to minimizing

$$(6) \quad \begin{aligned} T(\mathbf{u}) &= -\ln p_{\mathbf{z}}(\mathbf{z}; \mathbf{u}) - \ln p_{\mathbf{u}}(\mathbf{u}) \\ &= \sum_{i=1}^N \left\{ ([\mathbf{A}\mathbf{u}]_i + \gamma + \sigma^2) - (z_i + \sigma^2) \ln([\mathbf{A}\mathbf{u}]_i + \gamma + \sigma^2) \right\} - \ln p_{\mathbf{u}}(\mathbf{u}). \end{aligned}$$

Note that we have dropped the  $\ln(z_i + \sigma^2)$  term from the summation for notational simplicity since this term has no effect on the corresponding minimization problem.

In classical inverse problems,  $-\ln p_{\mathbf{u}}(\mathbf{u})$  corresponds to the regularization term. However in the Bayesian setting,  $p_{\mathbf{u}}(\mathbf{u})$  is probability density known as the *prior* from which the unknown  $\mathbf{u}$  is assumed to arise. Thus if we can formulate our prior knowledge regarding the characteristics of  $\mathbf{u}$  in the form of a probability density  $p_{\mathbf{u}}(\mathbf{u})$ , we have a natural, and statistically rigorous, motivation for our regularization method. In this paper, we will provide a statistical interpretation for the use of the total variation for the negative-log prior  $-\ln p_{\mathbf{u}}(\mathbf{u})$ .

Once the cost function  $T$  has been explicitly defined, our task will be to solve the nonnegatively constrained problem

$$(7) \quad \min_{\mathbf{u} \geq \mathbf{0}} T(\mathbf{u}).$$

The nonnegativity constraint arises from the prior knowledge that the true intensity vector  $\mathbf{u}$  has nonnegative components (intensities). The algorithm that we present for solving (7) is an extension of the nonnegatively constrained convex programming algorithm of [3]. The method intersperses gradient projection and conjugate gradient (CG) iterations in such a way that global convergence is guaranteed for strictly convex functions in an efficient manner. Due to the use of CG, preconditioning strategies can also be employed, and this we explore as well. A connection between our approach and the quasi-Newton formulation of the lagged-diffusivity fixed point iteration [16, 17] is illustrated.

The paper is organized as follows. We begin in Section 2 with a statistical interpretation for taking  $-\ln p_{\mathbf{u}}(\mathbf{u})$  to be a discrete total variation function. In Section 3, we analyze the resulting posterior density and make explicit the close connection between the Poisson negative-log likelihood and a certain weighted least squares function. The computational method that is the focus of the paper is presented in Section 4 and numerical tests are made in Section 5. We end with conclusions in Section 6.

## 2. A STATISTICAL INTERPRETATION FOR TOTAL VARIATION REGULARIZATION

In the discussion that follows,  $\mathbf{u}$  is a two-dimensional array. Recall that in (1) and in our computations it is the corresponding lexicographically ordered one-dimensional vector.

In [8], it is noted that for a large number of two-dimensional signals  $[\mathbf{u}]_{i,j} = u_{i,j}$ , the values of  $u_{i+1,j} - u_{i,j}$  and  $u_{i,j+1} - u_{i,j}$  tend to be realistically modeled by a Laplace distribution with mean 0, which has the form

$$p(x) = \frac{\alpha}{2} e^{-\alpha|x|}.$$

An analysis of 4000 digital images in [9] provides further support for this claim. We note that the Laplace distribution has heavy tails, which means that the probability of  $u_{i+1,j} - u_{i,j}$  and  $u_{i,j+1} - u_{i,j}$  being large (such as is the case at a jump discontinuity in the image) is not prohibitively small.

The main result of [8] is that for one-dimensional signals, the total variation method of [14], with appropriate choice of regularization parameter, provides the MAP estimate given the assumptions that the measurement noise is i.i.d. Gaussian and that adjacent pixel differences are independent and satisfy a Laplacian distribution. For images of dimensions two and higher, however, an analogous result has not been given. This is due in large part to the fact that the one-dimensional Laplacian distribution does not extend in a straightforward manner – as do the Gaussian and Poisson distributions – to the multivariate case. To see this, we note that in [6] it is shown that if

$$[\mathbf{\Gamma}\mathbf{u}]_{i,j} = ([\mathbf{\Gamma}_x\mathbf{u}]_{i,j}, [\mathbf{\Gamma}_y\mathbf{u}]_{i,j}) := (u_{i+1,j} - u_{i,j}, u_{i,j+1} - u_{i,j})$$

is an i.i.d. Laplacian random vector, its probability density function is given by

$$\frac{\alpha^2}{2\pi} K_0 \left( \alpha \sqrt{[\mathbf{\Gamma}_x\mathbf{u}]_{i,j}^2 + [\mathbf{\Gamma}_y\mathbf{u}]_{i,j}^2} \right),$$

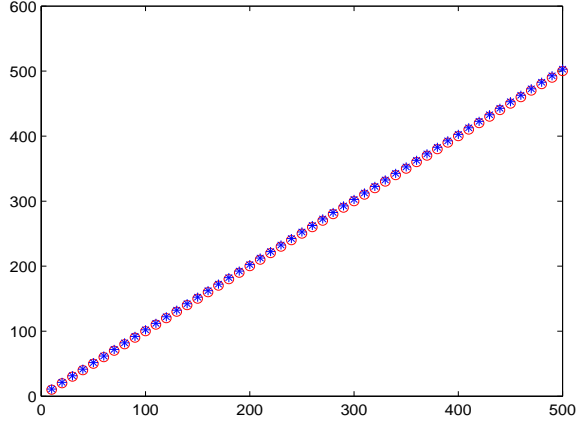


FIGURE 1. Plot of  $x$  (o) and  $-\ln K_0(x)$  (\*) on the interval  $[0, 500]$ .

where  $K_0$  is the order zero, modified second kind Bessel function. Assuming independence, this yields the negative-log prior

$$(8) \quad -\ln p_{\mathbf{u}}(\mathbf{u}) = c - \sum_{i,j=1}^N \ln K_0 \left( \alpha \sqrt{[\mathbf{\Gamma}_x \mathbf{u}]_{i,j}^2 + [\mathbf{\Gamma}_y \mathbf{u}]_{i,j}^2} \right)$$

where  $c$  is a constant. Ignoring  $c$  and using the approximation

$$-\ln K_0(x) \approx x,$$

whose accuracy is illustrated in Figure 1, we obtain

$$-\ln p_{\mathbf{u}}(\mathbf{u}) \approx \alpha \sum_{i,j=1}^N \sqrt{[\mathbf{\Gamma}_x \mathbf{u}]_{i,j}^2 + [\mathbf{\Gamma}_y \mathbf{u}]_{i,j}^2},$$

which is the discrete total variation function.

Thus we see that for two-dimensional discrete signals, the use of total variation regularization together with the statistically correct choice of regularization parameter gives a close approximation of the MAP estimate given the assumption that the data noise model is given by (3) and that adjacent pixel differences in both the horizontal and vertical directions in the unknown object  $\mathbf{u}$  are independent and satisfy a two-dimensional i.i.d. Laplacian distribution.

### 3. AN ANALYSIS OF THE POSTERIOR DENSITY FUNCTION

For notational convenience, we express (6) with the above negative-log prior approximation as

$$(9) \quad T_{\alpha}(\mathbf{u}) = T_0(\mathbf{u}) + \alpha J_{\beta}(\mathbf{u}),$$

where  $\alpha > 0$ ,

$$(10) \quad T_0(\mathbf{u}) := \sum_{i=1}^N \left\{ ([\mathbf{A}\mathbf{u}]_i + \gamma + \sigma^2) - (z_i + \sigma^2) \ln([\mathbf{A}\mathbf{u}]_i + \gamma + \sigma^2) \right\},$$

and

$$(11) \quad J_\beta(\mathbf{u}) := \sum_{i,j=1}^N \sqrt{[\Gamma_x \mathbf{u}]_{i,j}^2 + [\Gamma_y \mathbf{u}]_{i,j}^2 + \beta},$$

where  $\beta > 0$  is included in order to ensure the differentiability of  $J_\beta$ .

The gradient and Hessian of  $T_\alpha(\mathbf{u})$  are given by

$$(12) \quad \begin{aligned} \nabla T_\alpha(\mathbf{u}) &= \nabla T_0(\mathbf{u}) + \alpha \nabla J_\beta(\mathbf{u}), \\ \nabla^2 T_\alpha(\mathbf{u}) &= \nabla^2 T_0(\mathbf{u}) + \alpha \nabla^2 J_\beta(\mathbf{u}). \end{aligned}$$

The gradient and Hessian of the Poisson likelihood functional  $T_0(\mathbf{u})$  have expressions

$$(13) \quad \nabla T_0(\mathbf{u}) = \mathbf{A}^T \begin{pmatrix} \mathbf{A}\mathbf{u} - (\mathbf{z} - \gamma) \\ \mathbf{A}\mathbf{u} + \gamma + \sigma^2 \end{pmatrix},$$

$$(14) \quad \nabla^2 T_0(\mathbf{u}) = \mathbf{A}^T \text{diag} \left( \frac{\mathbf{z} + \sigma^2}{(\mathbf{A}\mathbf{u} + \gamma + \sigma^2)^2} \right) \mathbf{A},$$

where  $\text{diag}(\mathbf{v})$  is the diagonal matrix with  $\mathbf{v}$  as its diagonal. Here, and in what follows, we will use  $\mathbf{x}/\mathbf{y}$ , where  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ , to denote Hadamard, or component-wise, division, and  $\mathbf{x}^2 := \mathbf{x} \odot \mathbf{x}$ , where “ $\odot$ ” denotes the Hadamard product.

Note that for moderate to large values of  $\sigma^2$ , say  $\sigma^2 \geq 3^2$ , it is extremely unlikely for  $z_i$  to be negative. Then, since Poisson random variables take on only nonnegative integer values, the random vector  $\mathbf{z} + \sigma^2 \mathbf{1}$  is also highly unlikely to have nonpositive components. We will assume that  $\mathbf{A}$  is positive definite and that  $\mathbf{A}\mathbf{u} \geq \mathbf{0}$  whenever  $\mathbf{u} \geq \mathbf{0}$ . It immediately follows that  $\nabla^2 T_0(\mathbf{u})$  is positive definite for all  $\mathbf{u} \geq \mathbf{0}$ , and hence  $T_0$  is strictly convex on  $\mathbf{u} \geq \mathbf{0}$ .

The gradient and Hessian of  $J_\beta(\mathbf{u})$  are given by

$$(15) \quad \nabla J_\beta(\mathbf{u}) = \mathbf{L}_1(\mathbf{u})\mathbf{u},$$

$$(16) \quad \nabla^2 J_\beta(\mathbf{u}) = \mathbf{L}_1(\mathbf{u}) + 2\mathbf{L}_2(\mathbf{u}),$$

where, if  $\psi(t) := \sqrt{t + \beta}$ ,  $\Gamma \mathbf{u}^2 := (\Gamma_x \mathbf{u})^2 + (\Gamma_y \mathbf{u})^2$ , and  $\Gamma_{xy} \mathbf{u} := \Gamma_x \mathbf{u} \odot \Gamma_y \mathbf{u}$ ,

$$(17) \quad \mathbf{L}_1(\mathbf{u}) = \begin{bmatrix} \Gamma_x \\ \Gamma_y \end{bmatrix}^T \begin{bmatrix} \text{diag}(\psi'(\Gamma \mathbf{u}^2)) & \mathbf{0} \\ \mathbf{0} & \text{diag}(\psi'(\Gamma \mathbf{u}^2)) \end{bmatrix} \begin{bmatrix} \Gamma_x \\ \Gamma_y \end{bmatrix},$$

$$\mathbf{L}_2(\mathbf{u}) = \begin{bmatrix} \Gamma_x \\ \Gamma_y \end{bmatrix}^T \begin{bmatrix} \text{diag}((\Gamma_x \mathbf{u})^2 \odot \psi''(\Gamma \mathbf{u}^2)) & \text{diag}(\Gamma_{xy} \mathbf{u} \odot \psi''(\Gamma \mathbf{u}^2)) \\ \text{diag}(\Gamma_{xy} \mathbf{u} \odot \psi''(\Gamma \mathbf{u}^2)) & \text{diag}((\Gamma_y \mathbf{u})^2 \odot \psi''(\Gamma \mathbf{u}^2)) \end{bmatrix} \begin{bmatrix} \Gamma_x \\ \Gamma_y \end{bmatrix}.$$

For a more detailed treatment of these computations see [16]. We note that  $\nabla^2 J_\beta(\mathbf{u})$  is positive semi-definite for all  $\mathbf{u}$ , and hence,  $J_\beta$  is a convex function.

We can now prove that  $T_\alpha$  has a unique minimizer on  $\mathbf{u} \geq \mathbf{0}$ . This follows if  $T_\alpha$  is strictly convex and coercive on  $\mathbf{u} \geq \mathbf{0}$  [16, Chapter 2]. Recall that  $T_\alpha$  is coercive on  $\mathbf{u} \geq \mathbf{0}$  provided

$$\|\mathbf{u}\|_2 \rightarrow \infty \quad \text{implies} \quad T_\alpha(\mathbf{u}) \rightarrow \infty.$$

**Theorem 3.1.** *Assume that  $\mathbf{z} + \sigma^2 \mathbf{1} > \mathbf{0}$ ,  $\mathbf{A}$  is positive definite and  $\mathbf{A}\mathbf{u} \geq \mathbf{0}$  for all  $\mathbf{u} \geq \mathbf{0}$ . Then  $T_\alpha$  is strictly convex and coercive on  $\mathbf{u} \geq \mathbf{0}$ , and hence has a unique nonnegative minimizer.*

*Proof.* First, we note that given our assumptions and discussion above,  $T_0$  is strictly convex on  $\mathbf{u} \geq \mathbf{0}$ . We have also argued that  $J_\beta$  is convex. Thus  $T_\alpha$  is strictly convex on  $\mathbf{u} \geq \mathbf{0}$ .

The coercivity of  $T_\alpha$  is proved using the following application of Jensen's inequality:

$$(18) \quad T_0(\mathbf{u}) \geq \|\mathbf{A}\mathbf{u} + (\gamma + \sigma^2)\mathbf{1}\|_1 - \|\mathbf{z} + \sigma^2\|_\infty \ln \|\mathbf{A}\mathbf{u} + (\gamma + \sigma^2)\mathbf{1}\|_1$$

for  $\mathbf{u} \geq \mathbf{0}$ . Since  $\mathbf{A}$  is positive definite  $\|\mathbf{u}\|_2 \rightarrow \infty$  implies that  $\|\mathbf{A}\mathbf{u}\|_1 \rightarrow \infty$ , which in turn implies, via (18), that  $T_0(\mathbf{u}) \rightarrow \infty$ . Coercivity then follows from the fact that  $J_\beta(\mathbf{u}) \geq \mathbf{0}$  for all  $\mathbf{u}$ .

We have the result, finally, from the fact that  $\mathbf{u} \geq \mathbf{0}$  is a convex set.  $\square$

**3.1. THE WEIGHTED LEAST SQUARES CONNECTION.** From (3) we have  $\mathbf{z} + \sigma^2\mathbf{1} \approx \mathbf{A}\mathbf{u} + \gamma + \sigma^2$ , and hence from (13) and (14),

$$\begin{aligned} \nabla T_0(\mathbf{u}) &\approx \mathbf{A}^T \text{diag}(\mathbf{z} + \sigma^2\mathbf{1})^{-1} (\mathbf{A}\mathbf{u} - (\mathbf{z} - \gamma\mathbf{1})), \\ \nabla^2 T_0(\mathbf{u}) &\approx \mathbf{A}^T \text{diag}(\mathbf{z} + \sigma^2\mathbf{1})^{-1} \mathbf{A}, \end{aligned}$$

which are the gradient and Hessian, respectively, for the weighted least squares function

$$(19) \quad T_0^{\text{wls}}(\mathbf{u}) = \|\mathbf{A}\mathbf{u} - (\mathbf{z} - \gamma\mathbf{1})\|_{\text{diag}(\mathbf{z} + \sigma^2\mathbf{1})^{-1}}^2,$$

where  $\|\mathbf{v}\|_{\mathbf{C}}^2 := \mathbf{v}^T \mathbf{C} \mathbf{v}$ . The function (19) is also the negative-log likelihood that results when the following Gaussian approximation of (3) is used.

$$\begin{aligned} \text{Pois}(\mathbf{A}\mathbf{u} + (\gamma + \sigma^2)\mathbf{1}) &\approx N(\mathbf{A}\mathbf{u} + \gamma + \sigma^2, \text{diag}(\mathbf{A}\mathbf{u} + (\gamma + \sigma^2)\mathbf{1})), \\ &\approx \mathbf{A}\mathbf{u} + (\gamma + \sigma^2)\mathbf{1} + N(\mathbf{0}, \text{diag}(\mathbf{z} + \sigma^2\mathbf{1})). \end{aligned}$$

This approximation is commonly used in practice when data noise is known to be of Poisson type. It was also used in the development of a statistically motivated algorithm in [2].

Finally, we note that Theorem 3.1 will also hold if  $T_0$  is replaced either by  $T_0^{\text{wls}}$  or by the regular least squares function  $\|\mathbf{A}\mathbf{u} - \mathbf{z}\|_2^2$  in the definition of  $T_\alpha$  in (9).

#### 4. A NONNEGATIVELY CONSTRAINED CONVEX PROGRAMMING METHOD

In this section, we outline a computationally efficient method for solving

$$(20) \quad \min_{\mathbf{u} \geq \mathbf{0}} T_\alpha(\mathbf{u}),$$

where  $T_\alpha$  is defined in (9). We follow the approach set forth in [3], which was developed for use on Tikhonov-regularized Poisson likelihood estimation problems.

**4.1. PRELIMINARIES.** The projection of a vector  $\mathbf{u} \in \mathbb{R}^N$  onto the feasible set  $\Omega := \{\mathbf{u} \in \mathbb{R}^N \mid \mathbf{u} \geq \mathbf{0}\}$  can be conveniently expressed as

$$\mathcal{P}(\mathbf{u}) := \arg \min_{\mathbf{v} \in \Omega} \|\mathbf{v} - \mathbf{u}\| = \max\{\mathbf{u}, \mathbf{0}\},$$

where  $\max\{\mathbf{u}, \mathbf{0}\}$  is the vector whose  $i$ th component is zero if  $u_i < 0$  and is  $u_i$  otherwise. The active set for a vector  $\mathbf{u} \geq \mathbf{0}$  is defined

$$(21) \quad \mathcal{A}(\mathbf{u}) = \{i \mid u_i = 0\},$$

and the complementary set of indices,  $\mathcal{I}(\mathbf{u})$ , is known as the inactive set.

The reduced gradient of  $T_\alpha$  at  $\mathbf{u} \geq \mathbf{0}$  is given by

$$[\nabla_{\text{red}} T_\alpha(\mathbf{u})]_i = \begin{cases} [\nabla T_\alpha(\mathbf{u})]_i, & i \in \mathcal{I}(\mathbf{u}) \\ 0, & i \in \mathcal{A}(\mathbf{u}), \end{cases}$$

the projected gradient of  $T_\alpha$  by

$$[\nabla_{\text{proj}} T_\alpha(\mathbf{u})]_i = \begin{cases} [\nabla T_\alpha(\mathbf{u})]_i, & i \in \mathcal{I}(\mathbf{u}), \text{ or } i \in \mathcal{A}(\mathbf{u}) \text{ and } \frac{\partial T_\alpha(\mathbf{u})}{\partial x_i} < 0, \\ 0, & \text{otherwise,} \end{cases}$$

and the reduced Hessian by

$$(22) \quad [\nabla_{\text{red}}^2 T_\alpha(\mathbf{u})]_{ij} = \begin{cases} [\nabla^2 T_\alpha(\mathbf{u})]_{i,j}, & \text{if } i \in \mathcal{I}(\mathbf{u}) \text{ and } j \in \mathcal{I}(\mathbf{u}) \\ \delta_{ij}, & \text{otherwise.} \end{cases}$$

Finally, we define  $\mathbf{D}_{\mathcal{I}}(\mathbf{u})$  to be the diagonal matrix with components

$$(23) \quad [\mathbf{D}_{\mathcal{I}}(\mathbf{u})]_{ii} = \begin{cases} 1, & i \in \mathcal{I}(\mathbf{u}) \\ 0, & i \in \mathcal{A}(\mathbf{u}), \end{cases}$$

and  $\mathbf{D}_{\mathcal{A}}(\mathbf{u}) := \mathbf{I} - \mathbf{D}_{\mathcal{I}}(\mathbf{u})$ . Note then that

$$(24) \quad \nabla_{\text{red}} T_\alpha(\mathbf{u}) = \mathbf{D}_{\mathcal{I}}(\mathbf{u}) \nabla T_\alpha(\mathbf{u}),$$

$$(25) \quad \nabla_{\text{red}}^2 T_\alpha(\mathbf{u}) = \mathbf{D}_{\mathcal{I}}(\mathbf{u}) \nabla^2 T_\alpha(\mathbf{u}) \mathbf{D}_{\mathcal{I}}(\mathbf{u}) + \mathbf{D}_{\mathcal{A}}(\mathbf{u}),$$

4.2. THE GRADIENT PROJECTION ITERATION. A key component of the iterative method introduced in [3] is the gradient projection iteration [11], which we present now: given  $\mathbf{u}_k \geq \mathbf{0}$ , we compute  $\mathbf{u}_{k+1}$  via

$$(26) \quad \mathbf{p}_k = -\nabla T_\alpha(\mathbf{u}_k),$$

$$(27) \quad \lambda_k = \arg \min_{\lambda > 0} T_\alpha(\mathcal{P}(\mathbf{u}_k + \lambda \mathbf{p}_k)),$$

$$(28) \quad \mathbf{u}_{k+1} = \mathcal{P}(\mathbf{u}_k + \lambda_k \mathbf{p}_k).$$

In practice, subproblem (27) is solved inexactly using a projected backtracking line search. In the implementation used here, we take the initial step length parameter to be

$$(29) \quad \lambda_k^0 = \frac{\|\mathbf{p}_k\|^2}{\langle \nabla^2 T_\alpha(\mathbf{u}_k) \mathbf{p}_k, \mathbf{p}_k \rangle}.$$

The quadratic backtracking line search algorithm found in [12] is then used to create a sequence of line search parameters  $\{\lambda_k^j\}_{j=0}^m$ , where  $m$  is the smallest positive integer such that the sufficient decrease condition

$$(30) \quad T_\alpha(\mathbf{u}_k(\lambda_k^j)) \leq T_\alpha(\mathbf{u}_k) - \frac{\mu}{\lambda_k^j} \|\mathbf{u}_k - \mathbf{u}_k(\lambda_k^j)\|^2$$

is satisfied, where  $\mu \in (0, 1)$  and

$$(31) \quad \mathbf{u}_k(\lambda) = \mathcal{P}_\Omega(\mathbf{u}_k + \lambda \mathbf{p}_k).$$

Then we take  $\lambda_k := \lambda_k^m$  in (27).

**Theorem 4.1.** *Let  $\{\mathbf{u}_k\}$  be a sequence generated by the gradient projection iteration as discussed above. Then, if  $\bar{\mathbf{u}}$  is a limit point of  $\{\mathbf{u}_k\}$ ,  $\nabla_{\text{proj}} T_\alpha(\bar{\mathbf{u}}) = \mathbf{0}$ . Furthermore, the optimal active set is identified in finitely many iterations. More precisely, if  $\mathbf{u}_k \rightarrow \bar{\mathbf{u}}$  there exists an integer  $m_0$  such that for all  $k \geq m_0$ ,  $\mathcal{A}(\mathbf{u}_k) = \mathcal{A}(\bar{\mathbf{u}})$ .*

The first half of this Theorem is proved in [1] and the second half is proved in [5, Theorem 4.1]. Under the hypotheses of Theorem 3.1, we can prove a much stronger convergence result.

**Theorem 4.2.** *Given that the assumptions of Theorem 3.1 hold, the gradient projection iterates  $\mathbf{u}_k$  converge to the unique minimizer  $\mathbf{u}^*$  of  $T_\alpha$  on  $\Omega$ .*

*Proof.*  $\{T_\alpha(\mathbf{u}_k)\}$  is a decreasing sequence that is bounded below by  $T_\alpha(\mathbf{u}^*)$ , where  $\mathbf{u}^*$  is the unique solution of (20) given by Theorem 4.1. Hence it converges to some  $\bar{T}_\alpha \geq T_\alpha(\mathbf{u}^*)$ . Since  $T_\alpha$  is coercive,  $\{\mathbf{u}_k\}$  is bounded, and hence there exists a subsequence  $\{\mathbf{u}_{k_j}\}$  converging to some  $\bar{\mathbf{u}}$ . By Theorem 4.1,  $\nabla_{\text{proj}} T_\alpha(\bar{\mathbf{u}}) = \mathbf{0}$ , and since  $T_\alpha$  is strictly convex,  $\bar{\mathbf{u}} = \mathbf{u}^*$ . Thus  $T_\alpha(\mathbf{u}_k) \rightarrow T_\alpha(\mathbf{u}^*)$ . Computing a Taylor expansion of  $T_\alpha$  about  $\mathbf{u}^*$ , we obtain

$$(32) \quad T_\alpha(\mathbf{u}_k) - T_\alpha(\mathbf{u}^*) = (\mathbf{u}_k - \mathbf{u}^*)^T \nabla T_\alpha(\mathbf{u}^*) + \frac{1}{2} (\mathbf{u}_k - \mathbf{u}^*)^T \nabla^2 T_\alpha(\mathbf{u}^*) (\mathbf{u}_k - \mathbf{u}^*) + \mathcal{O}(\|\mathbf{u}_k - \mathbf{u}^*\|_2^3).$$

Now, by Theorem 4.1 we have that  $\mathcal{A}(\mathbf{u}_k) = \mathcal{A}(\mathbf{u}^*)$  for  $k > m_0$ , which allows us to express (32) as

$$(33) \quad \begin{aligned} T_\alpha(\mathbf{u}_k) - T_\alpha(\mathbf{u}^*) \\ = \frac{1}{2} (\mathbf{u}_k - \mathbf{u}^*)^T \nabla_{\text{red}}^2 T_\alpha(\mathbf{u}^*) (\mathbf{u}_k - \mathbf{u}^*) + \mathcal{O}(\|\mathbf{u}_k - \mathbf{u}^*\|_2^3). \end{aligned}$$

for  $k > m_0$ . Furthermore, since  $\{\mathbf{u}_k\}$  is bounded, and given our hypotheses,

$$\min_k \left( \frac{\mathbf{z} + \sigma^2 \mathbf{1}}{\mathbf{A} \mathbf{u}_k + (\sigma^2 + \gamma) \mathbf{1}} \right) > 0.$$

Thus, since  $\mathbf{A}$  is positive definite, the minimum eigenvalue  $\lambda_{k,\min}$  of  $T_\alpha(\mathbf{u}_k)$  is positive and  $\lambda_{\min} := \min_k \{\lambda_{k,\min}\} > 0$ . Finally, from (33) we have

$$T_\alpha(\mathbf{u}_k) - T_\alpha(\mathbf{u}^*) \geq \frac{\lambda_{\min}}{2} \|\mathbf{u}_k - \mathbf{u}^*\|_2^2 + \mathcal{O}(\|\mathbf{u}_k - \mathbf{u}^*\|_2^3),$$

and hence  $\mathbf{u}_k \rightarrow \mathbf{u}^*$ . □

**4.3. THE REDUCED LAGGED-DIFFUSIVITY STEP.** In practice, the gradient projection iteration is very slow to converge. However, a robust method with much better convergence properties results if gradient projection iterations are interspersed with steps computed from the reduced Newton system

$$(34) \quad \nabla_{\text{red}}^2 T_\alpha(\mathbf{u}_k) \mathbf{p} = -\nabla_{\text{red}} T_\alpha(\mathbf{u}_k).$$

This is the approach taken in [3], however when total variation regularization is used, a more computationally efficient method results if (34) is replaced by the reduced quasi-Newton system

$$(35) \quad (\nabla_{\text{LD}}^2)_{\text{red}} T_\alpha(\mathbf{u}_k) \mathbf{p} = -\nabla_{\text{red}} T_\alpha(\mathbf{u}_k),$$

where  $(\nabla_{\text{LD}}^2)_{\text{red}} T_\alpha(\mathbf{u})$  is defined as in (22) with

$$(36) \quad \nabla_{\text{LD}}^2 T_\alpha(\mathbf{u}) = \nabla^2 T_0(\mathbf{u}) + \alpha \mathbf{L}_1(\mathbf{u})$$

and  $\mathbf{L}_1(\mathbf{u})$  defined in (17). We note that if in (36)  $T_0$  is replaced by the regular least squares likelihood function, the unreduced linear system

$$\nabla_{\text{LD}}^2 T_\alpha(\mathbf{u}) \mathbf{p} = -\nabla T_\alpha(\mathbf{u}_k)$$

is that which is solved in each iteration of the lagged-diffusivity fixed point iteration of [17]. Thus we call (35), (36) the reduced lagged-diffusivity system. We now present a method for its approximate solution.

Approximate solutions of (35), (36) can be efficiently obtained using conjugate gradient iteration (CG) [13] applied to the problem of minimizing

$$(37) \quad q_k(\mathbf{p}) = T_\alpha(\mathbf{u}_k) + \langle \nabla_{\text{red}} T_\alpha(\mathbf{u}_k), \mathbf{p} \rangle + \frac{1}{2} \langle (\nabla_{\text{LD}}^2)_{\text{red}} T_\alpha(\mathbf{u}_k) \mathbf{p}, \mathbf{p} \rangle.$$



The result is a sequence  $\{\mathbf{p}_k^j\}$  that converges to the minimizer of (37). Even with rapid CG convergence, for large-scale problems it is important to choose effective stopping criteria to reduce overall computational cost. We have found that the following stopping criterion from Moré and Toraldo [12] is very effective:

$$(38) \quad q_k(\mathbf{p}_k^{j-1}) - q_k(\mathbf{p}_k^j) \leq \gamma_{CG} \max\{q_k(\mathbf{p}_k^{i-1}) - q_k(\mathbf{p}_k^i) \mid i = 1, \dots, j-1\},$$

where  $0 < \gamma_{CG} < 1$ . Then the approximate solution of (37) is taken to be the  $\mathbf{p}_k^{m_{CG}}$  where  $m_{CG}$  is the smallest integer such that (38) is satisfied.

With  $\mathbf{p}_k := \mathbf{p}_k^{m_{CG}}$ , we again apply a projected backtracking line search, only this time we use the much less stringent acceptance criteria

$$(39) \quad T_\alpha(\mathbf{u}_k(\lambda_k^m)) < T_\alpha(\mathbf{u}_k).$$

4.4. THE NUMERICAL ALGORITHM. In the first stage of our algorithm we need stopping criteria for the gradient projection iterations. Borrowing from Moré and Toraldo [12], we stop when

$$(40) \quad T_\alpha(\mathbf{u}_{k-1}) - T_\alpha(\mathbf{u}_k) \leq \gamma_{GP} \max\{T_\alpha(\mathbf{u}_{i-1}) - T_\alpha(\mathbf{u}_i) \mid i = 1, \dots, k-1\},$$

where  $0 < \gamma_{GP} < 1$ .

#### Gradient Projection-Reduced-Lagged-Diffusivity (GPRLD) Iteration

**Step 0:** Select initial guess  $\mathbf{u}_0$ , and set  $k = 0$ .

**Step 1:** Given  $\mathbf{u}_k$ .

- (1) Take gradient projection steps until either (40) is satisfied or  $\text{GP}_{\max}$  iterations have been computed. Return updated  $\mathbf{u}_k$ .

**Step 2:** Given  $\mathbf{u}_k$ .

- (1) Do CG iterations to approximately minimize the quadratic (37) until either (38) is satisfied or  $\text{CG}_{\max}$  iterations have been computed. Return  $\mathbf{p}_k = \mathbf{p}_k^{m_{CG}}$ .
- (2) Find  $\lambda_k^m$  that satisfies (39), and return  $\mathbf{u}_{k+1} = \mathbf{u}_k(\lambda_k^m)$ .
- (3) Update  $k := k + 1$  and return to Step 1.

Since at each outer GPRLD iteration at least one gradient projection step, with sufficient decrease condition (39), is taken, by Theorem 4.2 we have the following result.

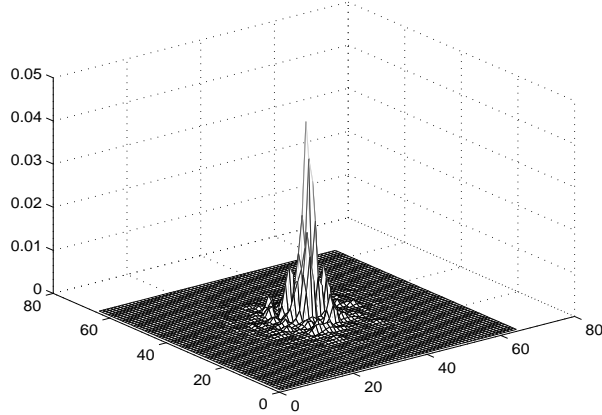
**Theorem 4.3.** *The iterates  $\{\mathbf{u}_k\}$  generated by GPRLD are guaranteed to converge to the unique solution of problem (20).*

4.5. PRECONDITIONING STRATEGIES. The converge properties of the CG iterations in GPRLD can be improved if an effective preconditioner is used. This requires the solution of a linear system of the form

$$(41) \quad \mathbf{M}_k \mathbf{v} = \nabla_{\text{red}} T_\alpha(\mathbf{u}_k)$$

at each inner CG iteration. Here  $\mathbf{M}_k$  is the symmetric positive definite preconditioning matrix, or, simply, the preconditioner. Roughly speaking, for the resulting method to be efficient, solutions of (41) must be efficiently computable, with  $\mathbf{M}_k \approx (\nabla_{\text{LD}}^2)_{\text{red}} T_\alpha(\mathbf{u}_k)$ .

The primary difficulty with preconditioning CG iterations within GPRLD is that since  $(\nabla_{\text{LD}}^2)_{\text{red}} T_\alpha(\mathbf{u}_k)$  changes in a nontrivial way with  $k$ , the preconditioner must also change with  $k$ . In addition, due to the form of  $(\nabla_{\text{LD}}^2)_{\text{red}} T_\alpha(\mathbf{u}_k)$  (see (22)), most standard preconditioning techniques are unusable. However, following [3] we note

FIGURE 2. PSF  $\mathbf{a}$  used in the numerical experiments.

that the discrete PSF  $\mathbf{a}$  – which completely determines the blurring matrix  $\mathbf{A}$  [16] – is typically localized in the sense that except for a few components near the center with high intensity, its components are relatively small. This can be clearly seen in Figure 2, where the PSF used in our numerical experiments below is plotted. Such a PSF can be accurately approximated by zeroing out the smaller components. We select the truncated PSF  $\hat{\mathbf{a}}$  to be the array with entries

$$(42) \quad \hat{a}_{ij} = \begin{cases} a_{ij}, & a_{ij} \geq \tau, \\ 0 & \text{otherwise,} \end{cases}$$

where the truncation parameter

$$(43) \quad \tau = r \max_{i,j} a_{ij}, \quad 0 < r < 1.$$

We then take  $\hat{\mathbf{A}}$  to be the blurring matrix determined by  $\hat{\mathbf{a}}$ . For our experiments, we used  $r = 0.1$ . With lexicographical ordering of the unknowns,  $\hat{\mathbf{A}}$  is a banded, sparse matrix. The bandwidth decreases and the sparsity increases as either the PSF becomes more concentrated about its central peak, or as the truncation parameter  $\tau$  increases.

Motivated by the form of  $(\nabla_{LD}^2)_{\text{red}} T_\alpha(\mathbf{u})$  we take the preconditioning matrix at outer iteration  $k$  to be

$$(44) \quad \mathbf{M}_k = \mathbf{D}_I^k \hat{\mathbf{A}}^T \text{diag} \left( \frac{\mathbf{z} + \sigma^2}{(\mathbf{A}\mathbf{u} + \gamma + \sigma^2)^2} \right) \hat{\mathbf{A}} \mathbf{D}_I^k + \alpha \mathbf{D}_I^k \mathbf{L}_1(\mathbf{u}_k) \mathbf{D}_I^k + \mathbf{D}_A^k,$$

where  $\mathbf{D}_I^k = \mathbf{D}_I(\mathbf{u}_k)$ ,  $\mathbf{D}_A^k = \mathbf{I} - \mathbf{D}_I(\mathbf{u}_k)$  and  $\mathbf{L}_1(\mathbf{u}_k)$  is defined in (17). Note that since  $\hat{\mathbf{A}}$  and  $\mathbf{L}_1(\mathbf{u}_k)$  are banded, sparse matrices, each  $\mathbf{M}_k$  will also be a banded, sparse matrix. Moreover, if the size of the active set increases with  $k$ , the number of nonzero (diagonal) entries in  $\mathbf{D}_I^k$  decreases, and  $\mathbf{M}_k$  becomes even more sparse.

To solve (41), (44), we compute a Cholesky factorization of  $\mathbf{M}_k$  after a sparse reordering of indices using MATLAB's `symamd` function. Note that this Cholesky factorization need only be computed once per outer GPRLD iteration. Furthermore, if the active set is large, the computation of the Cholesky factorization will be very efficient. In order to optimize the efficiency of the resulting method, we use the

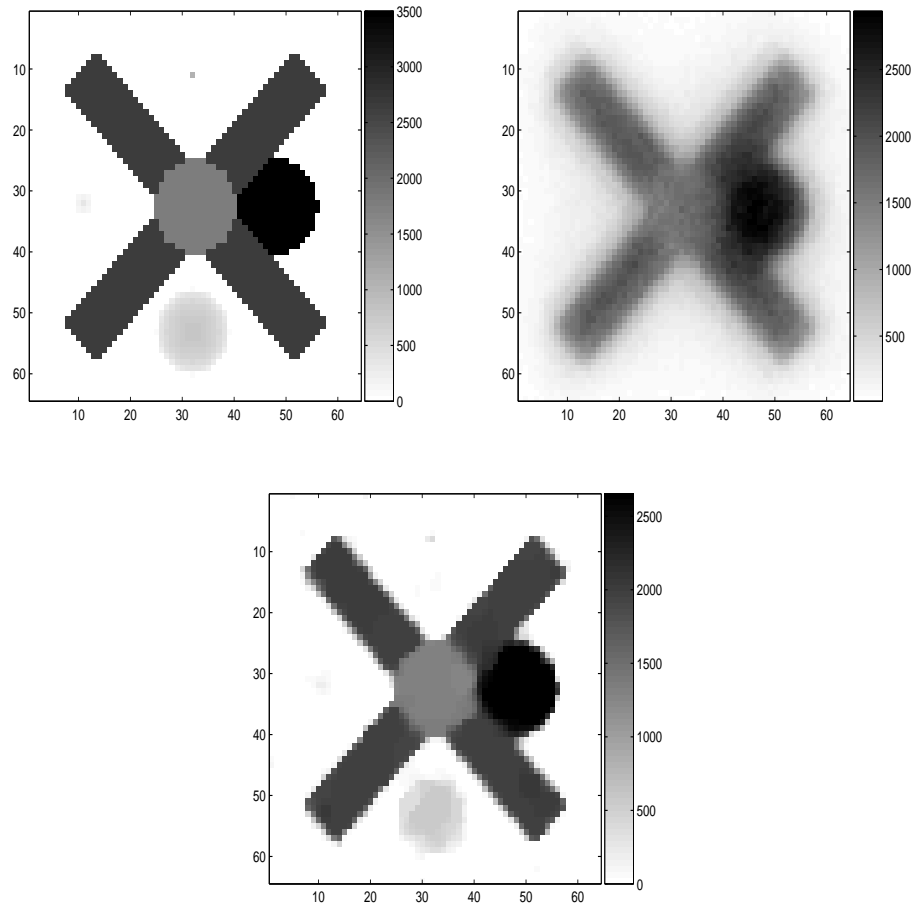


FIGURE 3. On the upper-left is the true object. On the upper-right, is the blurred, noisy image  $\mathbf{z}$ . The lower image is the reconstruction obtained by GPRLD applied to (20) with  $\alpha = 5 \times 10^{-5}$ .

preconditioner only from the 5th iteration onwards, since in first few iterations it expensive to implement and has relatively little benefit.

## 5. NUMERICAL EXPERIMENTS

In this section, we present results obtained when GPRLD is used for solving (20) with simulated data generated according to statistical model (2).

5.1. A DEBLURRING EXAMPLE. Our first tests are performed on the  $64 \times 64$  simulated satellite seen on the upper-left side in Figure 3. Generating corresponding blurred noisy data requires a discrete PSF  $\mathbf{a}$ , which we compute using the Fourier optics PSF model

$$\mathbf{a} = \left| \mathbf{F}^{-1} \left\{ \mathbf{p} \odot e^{i\phi} \right\} \right|^2,$$

where  $\mathbf{p}$  is the  $N \times N$  indicator array for the telescopes pupil;  $\phi$  is the  $N \times N$  array that represents the aberrations in the incoming wavefronts of light;  $\hat{i} = \sqrt{-1}$ ; and  $\mathbf{F}$  is the two-dimensional discrete Fourier transform matrix. The PSF used in our experiments can be seen in Figure 2. The  $64^2 \times 64^2$  blurring matrix  $\mathbf{A}$  obtained from  $\mathbf{a}$  is block Toeplitz with Toeplitz blocks (BTTB) [16]. For efficient computations,  $\mathbf{A}$  is embedded in a  $128^2 \times 128^2$  block circulant with circulant block (BCCB) matrix, which can be diagonalized by the two-dimensional discrete Fourier and inverse discrete Fourier transform matrices [16]. Data  $\mathbf{z}$  with a signal-to-noise ratio of approximately 30 is then generated using (2) with  $\sigma^2 = 25$  and  $\gamma = 10$  – physically realistic values for these parameters. To generate Poisson noise, the `poissrnd` function in MATLAB’s Statistics Toolbox is used. The corresponding blurred, noisy data  $\mathbf{z}$  is given on the upper-right in Figure 3.

With the blurred, noisy data in hand, we estimate the object by solving (20) using GPRLD with  $\text{GP}_{\max} = 1$  (note that then a value for  $\gamma_{\text{GP}}$  is not needed),  $\gamma_{\text{CG}} = 0.25$  with preconditioning and 0.1 without, and  $\text{CG}_{\max} = 40$ , which is only ever satisfied if preconditioning is *not* used. We stop iterations once

$$(45) \quad \|\nabla_{\text{proj}} T_{\alpha}(\mathbf{u}_k)\| / \|\nabla_{\text{proj}} T_{\alpha}(\mathbf{u}_0)\| < \text{GradTol},$$

where  $\text{GradTol} = 10^{-5}$ . We chose these parameter values in order to balance computational efficiency with good convergence properties of the method. Our initial guess was  $\mathbf{u}_0 = \mathbf{1}$ , and the regularization parameter was taken to be  $\alpha = 5 \times 10^{-5}$ . This choice of parameter approximately minimizes  $\|\mathbf{u}_{\alpha} - \mathbf{u}_{\text{exact}}\|$ . The reconstruction is given in Figure 3.

To illustrate the effectiveness of the preconditioner, in Figure 4 we plot the projected gradient norm versus the total number of FFTs, which is required when computing multiplication by  $\mathbf{A}$  and is the dominant cost in the implementation of the algorithm. The effectiveness of the preconditioner is evident. GPRLD without preconditioning required 37720 FFTs before (45) was satisfied, whereas when (44) was used only 504 FFTs were needed.

In order to demonstrate the efficiency of GPRLD, we also compare its convergence properties with those of the projected lagged-diffusivity (PLD) method of [4]. PLD has the form of the projected gradient iteration (26)-(28) with (27) replaced by an approximate solution of

$$(46) \quad (\nabla_{\text{LD}}^2)_{\text{red}} T_{\alpha}(\mathbf{u}_k) \mathbf{p} = -\nabla_{\text{proj}} T_{\alpha}(\mathbf{u}_k)$$

obtained using a truncated CG iteration (see [4] for more details). One important difference between these two methods is that PLD uses the standard (c.f. [13]) CG stopping criteria

$$(47) \quad \begin{aligned} & \|(\nabla_{\text{LD}}^2)_{\text{red}} T_{\alpha}(\mathbf{u}_k) \mathbf{p}_k^j + \nabla_{\text{red}} T_{\alpha}(\mathbf{u}_k)\| \\ & \leq \min \left\{ \frac{1}{2}, \|\nabla_{\text{red}} T_{\alpha}(\mathbf{u}_k)\| \right\} \cdot \|\nabla_{\text{red}} T_{\alpha}(\mathbf{u}_k)\|, \end{aligned}$$

where  $\mathbf{p}_k^j$  is the  $j$ th CG iterate at outer iteration  $k$  and  $\mathbf{p}_k^0 = \mathbf{0}$ . In addition, we allow a maximum of 40 CG iterations. This is due to the fact that more CG iterations yields little benefit given the additional expense. We note that  $\mathbf{M}_k$  in (44) can also be used as a preconditioner for CG iterations in PLD (no preconditioner was presented in [4]). We use the preconditioned version of the algorithm here. The convergence of the preconditioned PLD method for this example is illustrated in

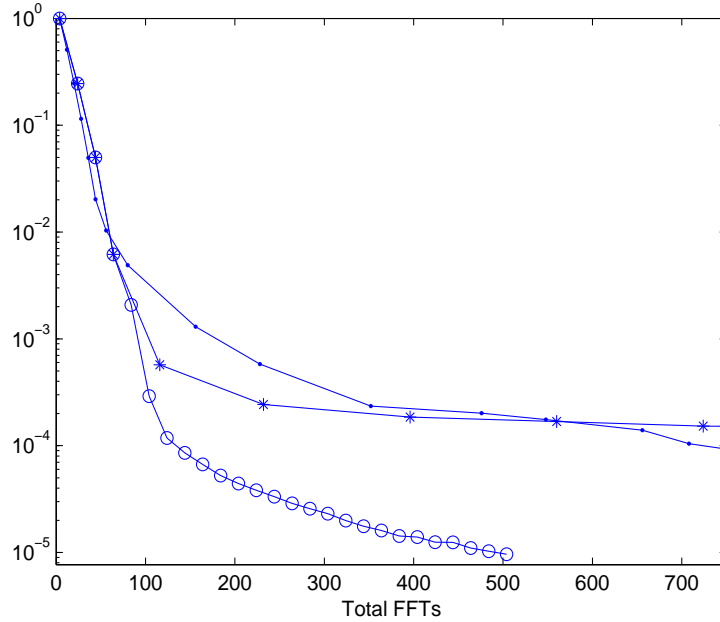


FIGURE 4. Plots of  $\|\nabla_{\text{proj}}T_{\alpha}(\mathbf{u}_k)\|_2/\|\nabla_{\text{proj}}T_{\alpha}(\mathbf{u}_0)\|_2$  versus cumulative FFTs for gradient projection-reduced-lagged-diffusivity both with (o-) and without (\*) preconditioning and for projected lagged-diffusivity with preconditioning (·-).

Figure 4, where it is readily seen that preconditioned GPRLD is the more efficient method.

5.2. A COMPARISON WITH LEAST SQUARES COST FUNCTIONS. Preconditioned GPRLD can also be implemented on nonnegatively constrained, total variation penalized least squares problems. In this subsection, we implement the method on both regular least squares and weighted least squares problems with total variation regularization; that is, we solve (20) with  $T_0$  replaced by  $T_0^{\text{wls}}$ , which is defined in (19), and also by  $T_0^{\text{ls}} := \|\mathbf{A}\mathbf{u} - \mathbf{z}\|_2^2$ . We use the same algorithmic parameters as those above. The regularization parameters were also chosen in the same fashion and are given by  $\alpha = 5 \times 10^{-5}$  for both the Poisson and weighted least squares likelihood and  $\alpha = 5 \times 10^{-3}$  for the regular least squares likelihood. The preconditioners for the least squares likelihoods both have the form (compare with (44))

$$(48) \quad \mathbf{M}_k = \mathbf{D}_{\mathcal{I}}^k \hat{\mathbf{A}}^T \mathbf{W} \hat{\mathbf{A}} \mathbf{D}_{\mathcal{I}}^k + \alpha \mathbf{D}_{\mathcal{I}}^k \mathbf{L}_1(\mathbf{u}_k) \mathbf{D}_{\mathcal{I}}^k + \mathbf{D}_{\mathcal{A}}^k,$$

where  $\mathbf{W} = \mathbf{I}$  for the regular least squares function and  $\mathbf{W} = \text{diag}(\mathbf{z} + \sigma^2 \mathbf{1})^{-1}$  for the weighted least squares function.

We plot  $\|\nabla_{\text{proj}}T_{\alpha}(\mathbf{u}_k)\|/\|\nabla_{\text{proj}}T_{\alpha}(\mathbf{u}_0)\|$  versus cumulative FFTs for each method in Figure 5. We first note that GPRLD is most efficient when the Poisson likelihood is used and is least efficient when the regular least squares function is used. This contradicts the following standard assumption: when a more statistically accurate, but also more complex, likelihood function is used, one can expect more accurate

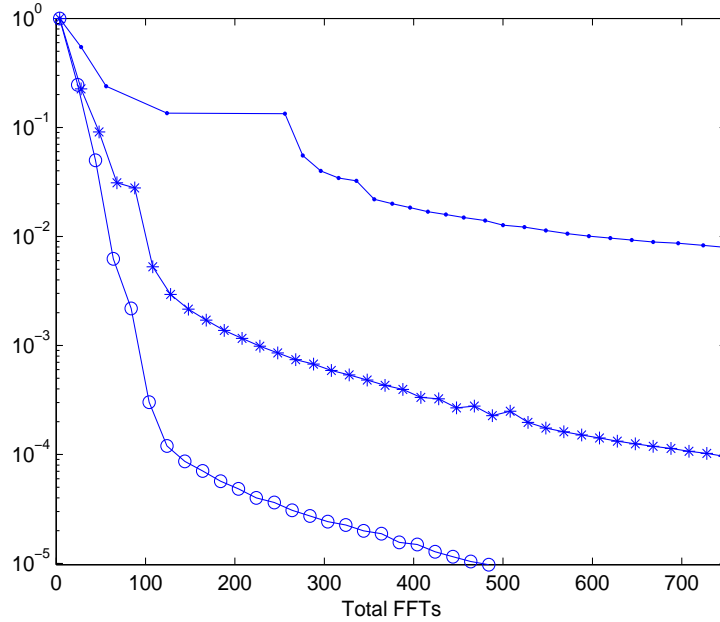


FIGURE 5. Plots of  $\|\nabla_{\text{proj}}T_\alpha(\mathbf{u}_k)\|_2/\|\nabla_{\text{proj}}T_\alpha(\mathbf{u}_0)\|_2$  versus cumulative FFTs for preconditioned gradient projection-reduced-lagged-diffusivity with Poisson likelihood (o-), weighted least squares likelihood (\*-), and regular least squares likelihood (-).

estimates, however, the resulting computational problem will also be more demanding. This example shows that this assumption is not always correct, since the most complex likelihood yields the most computationally efficient method. Regarding the accuracy of the estimates, the Poisson likelihood estimate was slightly more accurate than the weighted least squares estimate, which in turn was substantially more accurate than the regular least squares estimate. Finally, we note that when the regular least squares likelihood was used, the active set at the computed estimate (recall (21)) was empty, which suggests that in this case the unconstrained and non-negatively constrained minimizers coincide. Thus the nonnegativity constraints can not be said to be inhibiting the convergence properties of the regular least squares method.

5.3. A DENOISING EXAMPLE. Since total variation is often used in denoising applications, we present a denoising example now. We begin with the standard object represented in Figure 6. We do not use preconditioning in the implementation of any of the methods.

In our experiment, we use noise model (2) with  $\sigma^2 = 25$  and  $\gamma = 0$ . The noisy data, which has a signal to noise ratio of approximately 11, is given on the upper-left in Figure 7. We then apply GPRLD without preconditioning and with parameter values  $\text{GP}_{\max} = 10$ ,  $\gamma_{\text{GP}} = 0.1$ ,  $\text{CG}_{\max} = 40$ , and  $\gamma_{\text{CG}} = 0.1$ . The regularization parameter used for both the Poisson and weighted least squares likelihood was  $\alpha = 3 \times 10^{-4}$ , and for regular least squares it was  $\alpha = 3 \times 10^{-3}$ . In all cases, it was chosen



FIGURE 6. Test Image.

(as above) to approximately minimize the solution error. Iterations were stopped once (45) with  $\text{GradTol} = 10^{-3}$  was satisfied. The reconstruction obtained using the Poisson likelihood is given on the upper-right in Figure 7. The reconstructions using the other methods were similar and yielded similar values for the approximation error. The number of FFTs required in the implementation of each method was 2680 when the Poisson likelihood was used, 6272 when the weighted least squares likelihood was used, and 10172 when the regular least squares likelihood was used. Finally, we implement the lagged-diffusivity fixed point iteration with the quasi-Newton system approximately solved using 40 iterations of CG together with a stopping tolerance of the type (47). To reach (45) with  $\text{GradTol} = 10^{-3}$  required 10164 FFTs, which is nearly identical with that of regular least squares GPRLD. This can be explained by the fact that once again the nonnegativity constraint is inactive at the approximate solution computed by GPRLD.

The previous example shows that if data noise satisfies (2), using the Poisson likelihood yields the most efficient method. However, if instead the data noise model has the form  $\hat{\mathbf{z}} = N(\mathbf{u}, \sigma^2 \mathbf{I})$ , as is the standard assumption, the question is, will the Poisson likelihood still yield good results? The answer is yes, at least in the experiment presented now, where we take  $\sigma^2 = 25$ , which yields the image on the lower left in Figure 7 with a signal to noise ratio of approximately 27. Applying GPRLD to (20) with the Poisson likelihood using  $\sigma^2 = 25$  and  $\gamma = 0$ , with the regularization parameter  $\alpha = 5 \times 10^{-5}$  and the above algorithmic parameters, with stopping criteria  $\text{GradTol} = 10^{-4}$ , we obtain the reconstruction on the lower right in Figure 7. This reconstruction yields a very similar value of approximation error as when the regular least squares function was used. Furthermore, we can see from the convergence analysis in Figure 8 that using GPRLD with the Poisson likelihood is much more efficient than is the lagged-diffusivity fixed point iteration implemented as in the previous example. Thus it seems that in this case using a Poisson approximation of Gaussian statistics yields an estimate of comparable accuracy to that obtained with the statistically correct regular least squares likelihood, but in a much more efficient manner.

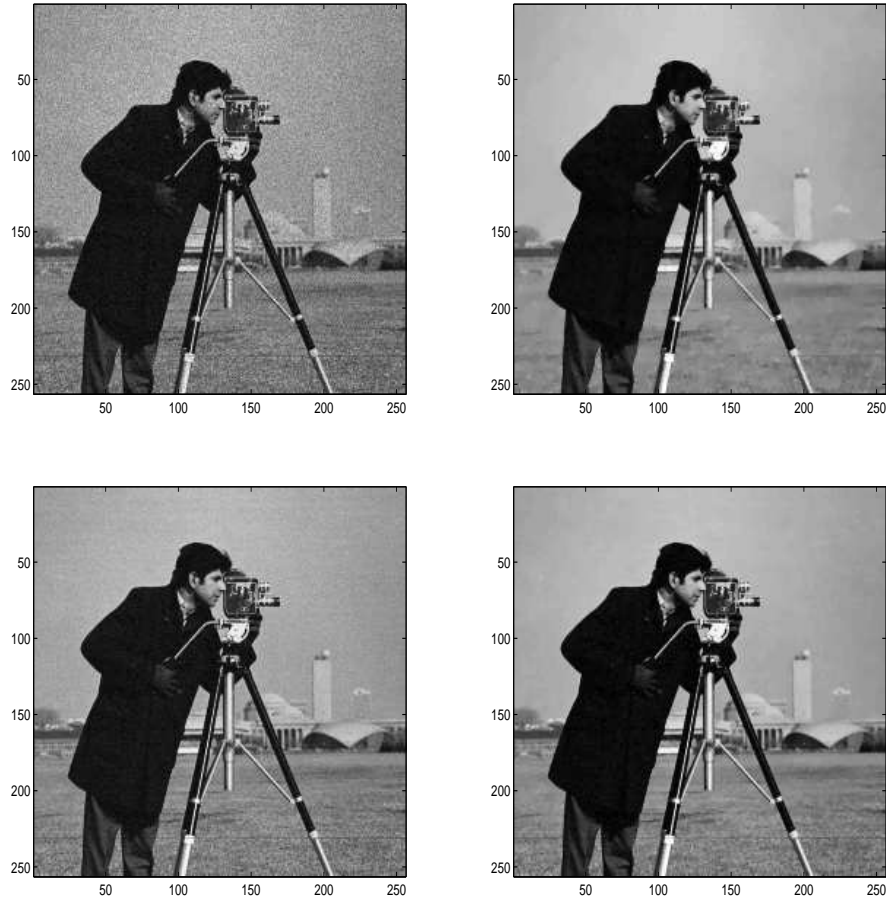


FIGURE 7. noisy image on the upper-right; reconstruction using preconditioned gradient projection-reduced-lagged-diffusivity with  $\alpha = 3 \times 10^{-4}$  on the bottom.

## 6. CONCLUSIONS

We have motivated total variation-penalized Poisson likelihood estimation in a statistically rigorous fashion from the Bayesian perspective. We first derived the Poisson likelihood function (4), and then showed that the total variation penalty well-approximates the negative-log prior that results when the horizontal and vertical pixel differences in the image being estimated are assumed to have an independent, bivariate Laplacian distribution, and hence that for an appropriately chosen value for the regularization parameter  $\alpha$ , the solution of (20) well-approximates the maximum a posteriori (MAP) estimator assuming statistical model (3) and negative-log prior (8).

We then made explicit the connection between the Poisson likelihood and the weighted least squares function (19) – an approximation that is often used in the literature when Poisson noise is known to be present in collected data. The weighted



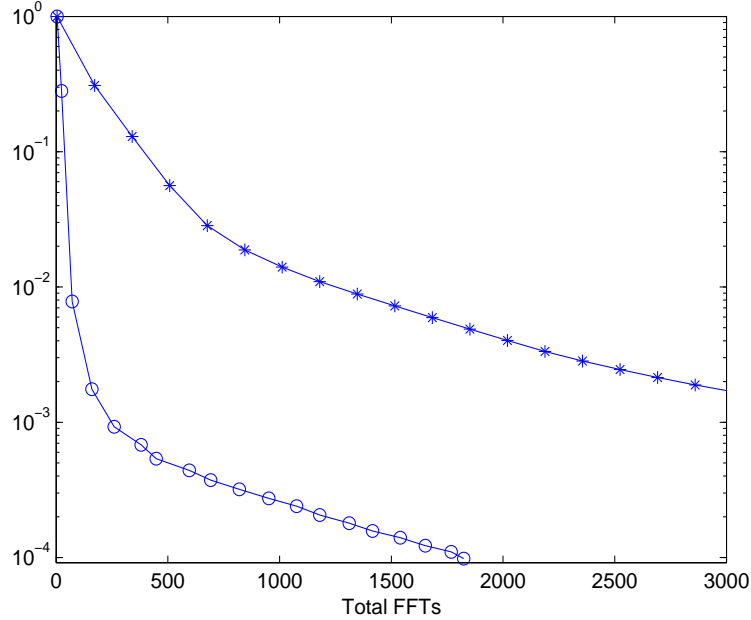


FIGURE 8. Plots of  $\|\nabla_{\text{proj}}T_{\alpha}(\mathbf{u}_k)\|_2/\|\nabla_{\text{proj}}T_{\alpha}(\mathbf{u}_0)\|_2$  versus cumulative FFTs for gradient projection-reduced-lagged-diffusivity with Poisson likelihood and  $\alpha = 7 \times 10^{-5}$  (o-) and lagged-diffusivity fixed point iteration with regular least squares likelihood and  $\alpha = 7 \times 10^{-4}$  (\*-).

least squares likelihood was then included in the numerical comparisons later in the paper.

A large portion of the paper was devoted to the presentation of a computationally efficient method for solving (20), which we have named gradient projection-reduced-lagged-diffusivity (GPRLD) due to its close connection to the lagged-diffusivity fixed point iteration of [17]. The method is sufficiently general so as to be applicable to problem (20) with either Poisson, weighted least squares, or regular least squares likelihood. Although our convergence theory was presented for the Poisson likelihood, the results extend in a straightforward manner to any likelihood that is coercive and whose Hessian matrix has uniformly bounded eigenvalues over bounded subsets of  $\{\mathbf{u} \mid \mathbf{u} \geq \mathbf{0}\}$ , e.g.  $T_0^{\text{wls}}$  and  $T_0^{\text{ls}}$  given our assumptions.

The method is an extension of the method presented in [3] for nonnegatively constrained, Tikhonov regularized Poisson likelihood estimation. What is new is that when total variation prior is used, the lagged-diffusivity, quasi-Newton form (35), (36) is more efficient than the Newton form (34). We also presented the sparse preconditioner (44), which was shown to substantially improve the convergence properties of GPRLD.

GPRLD was tested on synthetically generated astronomical data, and it was shown that the use of the Poisson likelihood yields a noticeably more efficient method and accurate than when either the weighted least squares or regular least

squares likelihood is used. However, the use of the weighted least squares likelihood yields a noticeably more efficient method than when the regular least squares likelihood is used. Thus in this case – in agreement with results in [2] – the incorporation of prior knowledge of noise statistics via the appropriate choice of likelihood function yields a more efficient method. This is notable given the fact that the Poisson likelihood is non-quadratic, and hence more complex, than the quadratic least squares likelihoods.

Finally, we compared GPRLD with the previously mentioned likelihood functions and the standard lagged-diffusivity fixed point iteration on a denoising example. Once again, the use of the Poisson likelihood yielded the most efficient method when noise model (2) was used. However, we also found that when the data error was i.i.d. Gaussian, the Poisson negative-log likelihood still yielded a more efficient method, which suggests that its use may improve convergence in general when total variation regularization is used.

## 7. ACKNOWLEDGEMENTS

The author would like to acknowledge the support of the University of Montana International Exchange Program and of the Department of Mathematics and Statistics at the University of Helsinki.

## REFERENCES

- [1] D. P. Bertsekas, *On the Goldstein-Levitin-Poljak Gradient Projection Method*, IEEE Transactions on Automatic Control, **21** (1976), pp. 174–184.
- [2] Johnathan M. Bardsley and James G. Nagy, *Covariance-Preconditioned Iterative Methods for Nonnegatively Constrained Astronomical Imaging*, SIAM Journal on Matrix Analysis and Applications, Vol. 27, No. 4, 2006, pp. 1184-1198.
- [3] 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.
- [4] Johnathan M. Bardsley and Aaron Luttmann, *Total Variation-Penalized Poisson Likelihood Estimation for Ill-Posed Problems*, submitted, University of Montana Technical Report #8, 2006.
- [5] P. H. Calamai and J. J. Moré, *Projected Gradient Methods for Linearly Constrained Problems*, Mathematical Programming, **39** (1987), pp. 93–116.
- [6] Torbjørn Eltoft and Taesu Kim, *On the Multivariate Laplace Distribution*, IEEE Signal Processing Letters, Vol. 13, No. 5, May 2006, pp. 300-303.
- [7] J. W. Goodman, *Introduction to Fourier Optics, 2nd Edition*, McGraw-Hill, 1996.
- [8] M. Green, *Statistics of images, the TV algorithm of Rudin-Osher-Fatemi for image denoising, and an improved denoising algorithm*, CAM Report 02-55, UCLA, October 2002.
- [9] Jिंगgang Huang and David Mumford, *Statistics of Natural Images and Models*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999, pp. 541-547.
- [10] Jari Kaipio and Erkki Somersalo, *Statistical and Computational Inverse Problems*, Springer 2005.
- [11] C. T. Kelley, *Iterative Methods for Optimization*, SIAM, Philadelphia, 1999.
- [12] 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.
- [13] J. Nocedal and S. Wright, *Numerical Optimization*, Springer 1999.
- [14] L. I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Physica D, 60 (1992), pp. 259–268.
- [15] D. L. Snyder, A. M. Hammoud, and R. L. White, *Image recovery from data acquired with a charge-coupled-device camera*, Journal of the Optical Society of America A, **10** (1993), pp. 1014–1023.
- [16] C. R. Vogel, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, 2002.

- [17] 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.