

AN ITERATIVE METHOD FOR EDGE-PRESERVING MAP ESTIMATION WHEN DATA-NOISE IS POISSON

JOHNATHAN M. BARDSLEY^{†‡} AND JOHN GOLDES[†]

Abstract. In numerous applications of image processing, e.g. astronomical and medical imaging, data-noise is well-modeled by a Poisson distribution. This motivates the use of the negative-log Poisson likelihood function for data fitting. (The fact that application scientists in both astronomical and medical imaging regularly choose this function for data fitting provides further motivation.) However difficulties arise when the negative-log Poisson likelihood is used. Chief among them are the facts that it is non-quadratic and is defined only for vectors with nonnegative values. The nonnegatively constrained, convex optimization problems that arise when the negative-log Poisson likelihood is used are therefore more challenging than when least squares is the fit-to-data function.

Edge preserving deblurring and denoising has long been a problem of keen interest in the image processing community. While total variation regularization is the gold standard for such problems, its use yields computationally intensive optimization problems. This motivates the desire to develop regularization functions that are edge preserving, but are less difficult to use. We present one such regularization function here. This function is quadratic, and can be viewed as the discretization of a diffusion operator with a diffusion function that is approximately 1 in smooth regions of the true image and is less than 1 (but still positive) at or near an edge.

Combining the negative-log Poisson likelihood function with this quadratic, edge preserving regularization function yields a strictly convex, nonnegatively constrained optimization problem. A large portion of this paper is dedicated to the presentation of and convergence proof for an algorithm designed for this problem. Finally, we apply the algorithm to synthetically generated data in order to test the methodology.

Key words. edge-preserving regularization, inverse problems, nonnegatively constrained optimization, Bayesian statistical methods.

AMS Subject Classifications: 15A29, 65K10, 65F22.

1. Introduction. We begin with the discrete linear equation

$$\mathbf{z} = \mathbf{A}\mathbf{u}, \tag{1.1}$$

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 $\sqrt{N} \times \sqrt{N}$. The matrix \mathbf{A} is assumed to be the discretization of a convolution operator in the case of deblurring, in which case, (1.1) is an ill-conditioned linear system.

An image \mathbf{z} collected by a CCD camera is the realization of a random vector [14]

$$\hat{\mathbf{z}} = \text{Poiss}(\mathbf{A}\mathbf{u}) + \text{Poiss}(\gamma \cdot \mathbf{1}) + N(\mathbf{0}, \sigma^2 \mathbf{I}), \tag{1.2}$$

where $\mathbf{1}$ is the $N \times 1$ vector of all ones. By (1.2) 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 $\sigma^2 \mathbf{I}$. In many instances, (1.2) can be well-approximated by [14]

$$\hat{\mathbf{z}} + \sigma^2 \cdot \mathbf{1} = \text{Poiss}(\mathbf{A}\mathbf{u} + (\gamma + \sigma^2) \cdot \mathbf{1}), \tag{1.3}$$

which has probability density function

$$p_{\mathbf{z}}(\mathbf{z}; \mathbf{u}) := \prod_{i=1}^N \frac{([\mathbf{A}\mathbf{u}]_i + \gamma + \sigma^2)^{z_i + \sigma^2} \exp[-([\mathbf{A}\mathbf{u}]_i + \gamma + \sigma^2)]}{(z_i + \sigma^2)!}. \tag{1.4}$$

[†]Department of Mathematical Sciences, University of Montana, Missoula, MT, 59812-0864. Email: bardsleyj@mso.umn.edu, jgoldes@mso.umn.edu.

[‡]This work was partially supported by the NSF under grant DMS-0504325.

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

$$p_{\mathbf{u}}(\mathbf{u}; \mathbf{z}) := \frac{p_{\mathbf{z}}(\mathbf{z}; \mathbf{u})p_{\mathbf{u}}(\mathbf{u})}{p_{\mathbf{z}}(\mathbf{z})}, \quad (1.5)$$

given by Bayes' Law, is maximized with respect to \mathbf{u} . The maximizer of the posterior density function $p_{\mathbf{u}}(\mathbf{u}; \mathbf{z})$ is called the maximum a posteriori (MAP) estimator. We note that maximizing (1.5) with respect to \mathbf{u} is equivalent to minimizing

$$T(\mathbf{u}) = \sum_{i=1}^N \left\{ ([\mathbf{A}\mathbf{u}]_i + \gamma + \sigma^2) - (z_i + \sigma^2) \ln([\mathbf{A}\mathbf{u}]_i + \gamma + \sigma^2) \right\} - \ln p_{\mathbf{u}}(\mathbf{u}). \quad (1.6)$$

Once the function $-\ln p_{\mathbf{u}}(\mathbf{u})$ in (1.6) has been defined, our task will be to solve the nonnegatively constrained problem

$$\arg \min_{\mathbf{u} \geq \mathbf{0}} T(\mathbf{u}). \quad (1.7)$$

The nonnegativity constraint arises from the prior knowledge that the true intensity vector \mathbf{u} has nonnegative components (intensities).

The function $-\ln p_{\mathbf{u}}(\mathbf{u})$ is the regularization term from classical inverse problems. Thus we see that in using the Bayesian formulation above a statistically rigorous interpretation of regularization follows. Note, in particular, that $p_{\mathbf{u}}(\mathbf{u})$ is the probability density function – known as the *prior* – from which the unknown \mathbf{u} is assumed to arise. This motivation of regularization in the context of large-scale, ill-posed inverse problems is relatively recent and is the focus of [10].

In this paper, our goal is to define a quadratic regularization function of the form

$$-\ln p_{\mathbf{u}}(\mathbf{u}) = \frac{\alpha}{2} \langle \mathbf{C}\mathbf{u}, \mathbf{u} \rangle, \quad (1.8)$$

with $\mathbf{C} \in \mathbb{R}^{N \times N}$ symmetric positive semi-definite, that allows for edges in reconstructed images. Equation (1.8) corresponds to the statistical assumption that the prior probability density $p_{\mathbf{u}}(\mathbf{u})$ is a degenerate Gaussian with mean $\mathbf{0}$ and covariance matrix $\alpha^{-1}\mathbf{C}^\dagger$, where “ \dagger ” denotes psuedo-inverse.

Quadratic, edge-preserving regularization functions have been studied in the context of penalized least squares problems in [6, 7, 9]. Here we will test the effectiveness of edge-preserving, quadratic regularization in the Poisson imaging case. We note that such an approach was used in [16], though the implementation of their method involved a separate stage involving the estimation of edges with the image via the solution of a complicated level set subproblem. Our approach is much simpler.

The general approach of penalized negative-log Poisson likelihood estimation is important due to the fact that it provides the most convenient means of incorporating prior knowledge about the true image. When the true image is known to have jumps, or edges, an edge preserving regularization method is desirable. The motivation for using a quadratic, edge-preserving regularization function stems from the fact that the resulting computational problem is more tractable than when total variation regularization (the classic approach to edge preserving regularization) is used. Finally,

we note that in previous work of the first author [3], a theoretical justification of this approach in the context of Poisson imaging is given that implies, via arguments in [2], that it defines a regularization scheme.

The algorithm that we present for solving (1.7) is a direct extension of the non-negatively constrained convex programming algorithm of [4], a variant of which was applied to the total variation regularization problem in [1]. 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. We prove global convergence of the method here, which has not been done previously, and we present the method and its proof for use on general, nonnegatively constrained, strictly convex minimization problems.

The paper is organized as follows. We begin in Section 2 by explicitly defining $-\ln p_{\mathbf{u}}(\mathbf{u})$. We then analyze the resulting posterior density and prove the existence of a unique minimizer in Section 3. The computational method used for solving (1.7) is then presented, and convergence of the algorithm is proved, in Section 4. Finally, numerical tests are made in Section 5, and we end with conclusions in Section 6.

2. The Regularization Function. As was stated above, the form of $-\ln p_{\mathbf{u}}(\mathbf{u})$ in (1.8) should be chosen to incorporate prior information about the object to be estimated. Here, we suppose that the location of edges, or discontinuities, in the true image is known (at least approximately) and that elsewhere the true image is known to be smooth. This motivates choosing $-\ln p_{\mathbf{u}}(\mathbf{u}) = \alpha J(\mathbf{u})$, where $\alpha > 0$ and

$$J(\mathbf{u}) = \frac{1}{2} \left\| \begin{bmatrix} \Lambda^{1/2} \Gamma_x \mathbf{u} \\ \Lambda^{1/2} \Gamma_y \mathbf{u} \end{bmatrix} \right\|^2 = \frac{1}{2} \mathbf{u}^T (\Gamma_x^T \Lambda \Gamma_x + \Gamma_y^T \Lambda \Gamma_y) \mathbf{u}, \quad (2.1)$$

with Λ a diagonal matrix with entries near 1 corresponding to pixels away from an edge and less than 1 for pixels near an edge. We see that then $-\ln p_{\mathbf{u}}(\mathbf{u})$ has the form of (1.8) with

$$\mathbf{C} = \Gamma_x^T \Lambda \Gamma_x + \Gamma_y^T \Lambda \Gamma_y, \quad (2.2)$$

which corresponds to the assumption that the unknown true object \mathbf{u} arises from a degenerate Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\alpha^{-1} \mathbf{C}^\dagger$, where “ \dagger ” denotes the psuedo-inverse. It remains to define the matrix Λ . We will discuss this problem when we present our numerical experiment in Section 5.

Before continuing, we note that in the functional setting, J in (2.1) takes the form

$$J(u) = \int_{\Omega} \|\lambda(x) \nabla u(x)\|^2 dx,$$

with $\Omega \subset \mathbb{R}^2$ the computational domain, $\lambda : \Omega \rightarrow \mathbb{R}$ continuously differentiable, and “ ∇ ” denoting the gradient. In the most general case, $\lambda : \Omega \rightarrow \mathbb{R}^{2 \times 2}$, in which case the corresponding diffusion operator is allowed to be anisotropic. (An anisotropic diffusion regularization function is presented in [9].) This functional formulation establishes the connection between our discussion here and the theoretical results presented in [2, 3].

3. Analysis of the Cost Function T . The problem that we must solve in order to obtain the MAP estimator is

$$\arg \min_{\mathbf{u} \in \Omega} T(\mathbf{A}\mathbf{u}; \mathbf{z}), \quad (3.1)$$

where $\Omega = \{\mathbf{u} \in \mathbb{R}^N \mid \mathbf{u} \geq \mathbf{0}\}$, and

$$T(\mathbf{A}\mathbf{u}; \mathbf{z}) = \ell(\mathbf{A}\mathbf{u}; \mathbf{z}) + \alpha \mathbf{u}^T \mathbf{C} \mathbf{u}, \quad (3.2)$$

with $\alpha > 0$ the regularization parameter, $\mathbf{C} \in \mathbb{R}^{N \times N}$ a positive semi-definite matrix, and

$$\ell(\mathbf{A}\mathbf{u}; \mathbf{z}) := \sum_{i=1}^N \left\{ ([\mathbf{A}\mathbf{u}]_i + \gamma + \sigma^2) - (z_i + \sigma^2) \ln([\mathbf{A}\mathbf{u}]_i + \gamma + \sigma^2) \right\}. \quad (3.3)$$

In the remainder of this section, we devote our time to proving results regarding T that will be useful later on.

The gradient and Hessian of $T(\mathbf{A}\mathbf{u}; \mathbf{z})$ with respect to \mathbf{u} are then given, respectively, by

$$\nabla T(\mathbf{A}\mathbf{u}; \mathbf{z}) = \mathbf{A}^T \left(\frac{\mathbf{A}\mathbf{u} - (\mathbf{z} - \gamma)}{\mathbf{A}\mathbf{u} + \gamma + \sigma^2} \right) + \alpha \mathbf{C} \mathbf{u}, \quad (3.4)$$

$$\nabla^2 T(\mathbf{A}\mathbf{u}; \mathbf{z}) = \mathbf{A}^T \text{diag} \left(\frac{\mathbf{z} + \sigma^2}{(\mathbf{A}\mathbf{u} + \gamma + \sigma^2)^2} \right) \mathbf{A} + \alpha \mathbf{C}, \quad (3.5)$$

where $\text{diag}(\mathbf{v})$ is the diagonal matrix with \mathbf{v} as its diagonal. Here we 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 to denote the Hadamard square.

Note that for moderate-to-large values of σ^2 , say $\sigma^2 \geq 3^2$, it is extremely unlikely for $z_i + \sigma^2$ 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. Motivated by this, we make the assumption that $\mathbf{z} + \sigma^2 \mathbf{1} > \mathbf{0}$. Furthermore, we assume that $\mathbf{A}\mathbf{u} \geq \mathbf{0}$ whenever $\mathbf{u} \geq \mathbf{0}$ and that the null spaces of \mathbf{A} and \mathbf{C} don't intersect. Taken all together, this gives us that

$$\begin{aligned} \langle \nabla^2 T(\mathbf{A}\mathbf{u}; \mathbf{z}) \mathbf{u}, \mathbf{u} \rangle &= \left\langle \left(\mathbf{A} \text{diag} \left(\frac{\mathbf{z} + \sigma^2}{(\mathbf{A}\mathbf{u} + \gamma + \sigma^2)^2} \right) \mathbf{A} + \mathbf{C} \right) \mathbf{u}, \mathbf{u} \right\rangle \\ &\geq \min \left\{ \min_{i=1, \dots, N} \left\{ \frac{z_i + \sigma^2}{([\mathbf{A}\mathbf{u}]_i + \gamma + \sigma^2)^2} \right\} \sigma_{\min}(\mathbf{A}^T \mathbf{A}), \sigma_{\min}(\mathbf{C}) \right\} \|\mathbf{u}\|_2^2, \\ &> 0, \end{aligned}$$

where σ_{\min} denotes the smallest positive eigenvalue. Thus $\nabla^2 T(\mathbf{A}\mathbf{u}; \mathbf{z})$ is positive definite for all $\mathbf{u} \geq \mathbf{0}$, which implies that T is a strictly convex function on $\Omega = \{\mathbf{u} \in \mathbb{R}^N \mid \mathbf{u} \geq \mathbf{0}\}$.

We can now prove that T has a unique minimizer on Ω . This follows if in addition to being strictly convex, T is also coercive on Ω [15, Chapter 2]; that is,

$$\|\mathbf{u}\|_2 \rightarrow \infty \quad \text{implies} \quad T(\mathbf{A}\mathbf{u}; \mathbf{z}) \rightarrow \infty \quad \text{for all} \quad \mathbf{u} \geq \mathbf{0}.$$

THEOREM 3.1. *Given the assumptions made above, T is strictly convex and coercive on Ω and hence has a unique nonnegative minimizer.*

Proof. First, we argued above that T is strictly convex. The coercivity of T is proved using the following application of Jensen's inequality:

$$T(\mathbf{A}\mathbf{u}; \mathbf{z}) \geq \|\mathbf{A}\mathbf{u} + (\gamma + \sigma^2)\mathbf{1}\|_1 - \|\mathbf{z} + \sigma^2\|_\infty \ln \|\mathbf{A}\mathbf{u} + (\gamma + \sigma^2)\mathbf{1}\|_1 + \mathbf{u}^T \mathbf{C} \mathbf{u} \quad (3.6)$$

for $\mathbf{u} \geq \mathbf{0}$. Now, if $\|\mathbf{u}\|_2 \rightarrow +\infty$ then $\sup\{\|\mathbf{A}\mathbf{u}\|_1, \mathbf{u}^T \mathbf{C}\mathbf{u}\} \rightarrow +\infty$, since \mathbf{A} and \mathbf{C} have non-intersecting null spaces, and hence $T(\mathbf{A}\mathbf{u}; \mathbf{z}) \rightarrow +\infty$, establishing coercivity.

The existence of a unique solution then follows from the fact that Ω is a convex set. \square

Also, note that (3.6) implies that $T(\mathbf{A}\mathbf{u}; \mathbf{z})$ is bounded below.

Finally, we show that T is Lipschitz continuous:

$$\begin{aligned} \|\nabla T(\mathbf{u}) - \nabla T(\mathbf{v})\|_2 &= \left\| \mathbf{A}^T \left(\frac{\mathbf{A}\mathbf{u} + \gamma - \mathbf{z}}{\mathbf{A}\mathbf{u} + \gamma + \sigma^2} - \frac{\mathbf{A}\mathbf{v} + \gamma - \mathbf{z}}{\mathbf{A}\mathbf{v} + \gamma + \sigma^2} \right) + \alpha(\mathbf{C}(\mathbf{u} - \mathbf{v})) \right\|_2 \\ &\leq \|\mathbf{A}\|_2 F(\mathbf{u}, \mathbf{v}) + \alpha \sigma_{\max}(\mathbf{C}) \|\mathbf{u} - \mathbf{v}\|_2, \end{aligned}$$

where $\sigma_{\max}(\mathbf{C})$ is the maximum eigenvalue of \mathbf{C} , and

$$\begin{aligned} F(\mathbf{u}, \mathbf{v}) &= \left\| \frac{(\mathbf{A}(\mathbf{u} - \mathbf{v})) \odot (\mathbf{z} + \sigma^2)}{(\mathbf{A}\mathbf{u} + \gamma + \sigma^2) \odot (\mathbf{A}\mathbf{v} + \gamma + \sigma^2)} \right\|_2 \\ &\leq \|\mathbf{A}\|_2 \left\| \frac{\mathbf{z} + \sigma^2}{(\gamma + \sigma^2)^2} \right\|_2 \|\mathbf{u} - \mathbf{v}\|_2, \end{aligned}$$

and hence,

$$\|\nabla T(\mathbf{u}) - \nabla T(\mathbf{v})\|_2 \leq \left(\|\mathbf{A}\|_2^2 \left\| \frac{\mathbf{z} + \sigma^2}{(\gamma + \sigma^2)^2} \right\|_2 + \alpha \lambda_{\max}(\mathbf{C}) \right) \|\mathbf{u} - \mathbf{v}\|_2,$$

which gives us the result.

4. A Nonnegatively Constrained Convex Programming Method. In this section, we outline a computationally efficient method for solving problems of the form

$$\arg \min_{\mathbf{u} \in \Omega} T(\mathbf{u}) \tag{4.1}$$

where T is assumed to satisfy the following properties:

Assumption 1: T is strictly convex, coercive, and twice continuously differentiable;

Assumption 2: ∇T is Lipschitz continuous with Lipschitz constant L .

Here ∇T and $\nabla^2 T$ denote the gradient and Hessian, respectively, of T . Note that given our discussion in Section 3, the function T defined in (3.2) satisfies Assumptions 1 and 2.

The method we present is that of [4]. However there, the cost function T had a specific form, and an incomplete proof of convergence was given. Here we consider the general problem (4.1) together with Assumptions 1 and 2 and give a rigorous proof of its convergence.

Before continuing, we note that the discussion in this section is independent of the rest of the paper, and the results apply to (4.1) for any cost function T satisfying Assumptions 1 and 2.

4.1. Preliminaries. The projection of a vector $\mathbf{u} \in \mathbb{R}^N$ onto the feasible set Ω 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

$$\mathcal{A}(\mathbf{u}) = \{i \mid u_i = 0\}, \tag{4.2}$$

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

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

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

the projected gradient of T by

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

and the reduced Hessian by

$$[\nabla_{\text{red}}^2T(\mathbf{u})]_{ij} = \begin{cases} [\nabla^2T(\mathbf{u})]_{i,j}, & \text{if } i \in \mathcal{I}(\mathbf{u}) \text{ and } j \in \mathcal{I}(\mathbf{u}) \\ \delta_{ij}, & \text{otherwise.} \end{cases} \quad (4.3)$$

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

$$[\mathbf{D}_{\mathcal{I}}(\mathbf{u})]_{ii} = \begin{cases} 1, & i \in \mathcal{I}(\mathbf{u}) \\ 0, & i \in \mathcal{A}(\mathbf{u}), \end{cases} \quad (4.4)$$

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

$$\nabla_{\text{red}}T(\mathbf{u}) = \mathbf{D}_{\mathcal{I}}(\mathbf{u}) \nabla T(\mathbf{u}), \quad (4.5)$$

$$\nabla_{\text{red}}^2T(\mathbf{u}) = \mathbf{D}_{\mathcal{I}}(\mathbf{u}) \nabla^2T(\mathbf{u}) \mathbf{D}_{\mathcal{I}}(\mathbf{u}) + \mathbf{D}_{\mathcal{A}}(\mathbf{u}), \quad (4.6)$$

4.2. The Gradient Projection Iteration. A key component of the iterative method introduced in [4], and that is the subject of this paper, is the gradient projection iteration [5, 11], which we present now: given $\mathbf{u}_k \geq \mathbf{0}$, we compute \mathbf{u}_{k+1} via

$$\mathbf{p}_k = -\nabla T(\mathbf{u}_k), \quad (4.7)$$

$$\lambda_k = \arg \min_{\lambda > 0} T(\mathcal{P}(\mathbf{u}_k + \lambda \mathbf{p}_k)), \quad (4.8)$$

$$\mathbf{u}_{k+1} = \mathcal{P}(\mathbf{u}_k + \lambda_k \mathbf{p}_k). \quad (4.9)$$

In practice, subproblem (4.8) is solved inexactly using a projected backtracking line search. In this approach, a sequence $\{\lambda_k^j\}_{j=0}^m$ is generated using a line search algorithm, and iterations are stopped once

$$T(\mathbf{u}_k(\lambda_k^j)) \leq T(\mathbf{u}_k) - \frac{\mu}{\lambda_k^j} \|\mathbf{u}_k - \mathbf{u}_k(\lambda_k^j)\|^2 \quad (4.10)$$

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

$$\mathbf{u}_k(\lambda) = \mathcal{P}_{\Omega}(\mathbf{u}_k + \lambda \mathbf{p}_k). \quad (4.11)$$

Then $\lambda_k \stackrel{\text{def}}{=} \lambda_k^m$ in (4.8).

In our implementation of the line search algorithm, the finite sequence $\{\lambda_k^j\}_{j=0}^m$ is generated as follows. The initial step length parameter is taken to be

$$\lambda_k^0 = \frac{\|\mathbf{p}_k\|^2}{\langle \nabla^2T(\mathbf{u}_k) \mathbf{p}_k, \mathbf{p}_k \rangle}, \quad (4.12)$$

and at the j th line search iteration, if λ_k^{j-1} does not satisfy (4.10), the minimizer of the quadratic function $q_k(\lambda)$ satisfying $q_k(0) = T(\mathbf{u}_k)$, $q'_k(0) = -\|\mathbf{p}_k\|^2$, and $q_k(\lambda_k^{j-1}) = T(\mathbf{u}_k(\lambda_k^{j-1}))$ is computed. We denote this minimizer by $\hat{\lambda}_k^j$ and define the new line search parameter by

$$\lambda_k^j = \text{median} \left[\lambda_k^{i-1}/100, \hat{\lambda}_k^i, \lambda_k^{i-1}/2 \right]. \quad (4.13)$$

We note that these details will be needed in the proof of convergence of the complete method, which we have yet to present.

That gradient projection iterations are well-defined and convergent for problems of the form (4.1) is a consequence of results found in [11, Section 5.4].

4.3. The Reduced Newton 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

$$\nabla_{\text{red}}^2 T(\mathbf{u}_k) \mathbf{p} = -\nabla_{\text{red}} T(\mathbf{u}_k). \quad (4.14)$$

This is the approach taken in [4].

Approximate solutions of (4.14) can be efficiently obtained using conjugate gradient iteration (CG) [13] applied to the problem of minimizing

$$q_k(\mathbf{p}) = T(\mathbf{u}_k) + \langle \nabla_{\text{red}} T(\mathbf{u}_k), \mathbf{p} \rangle + \frac{1}{2} \langle \nabla_{\text{red}}^2 T(\mathbf{u}_k) \mathbf{p}, \mathbf{p} \rangle. \quad (4.15)$$

The result is a sequence $\{\mathbf{p}_k^j\}$ that converges to the minimizer of (4.15). 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:

$$q_k(\mathbf{p}_k^{j-1}) - q_k(\mathbf{p}_k^j) \leq \gamma_{CG} \max\{q_k(\mathbf{p}_k^{i-1}) - q_k(\mathbf{p}_k^i) \mid i = 1, \dots, j-1\}, \quad (4.16)$$

where $0 < \gamma_{CG} < 1$. Then the approximate solution of (4.15) is taken to be the $\mathbf{p}_k^{m_{CG}}$ where m_{CG} is the smallest integer such that (4.16) 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

$$T(\mathbf{u}_k(\lambda_k^m)) < T(\mathbf{u}_k). \quad (4.17)$$

4.4. The Numerical Algorithm. In the first stage of our algorithm, we need a stopping criteria for the gradient projection iterations. For the ill-posed imaging problems that we consider here and in [4], no advantage was gained by taking more than a single gradient projection step in each outer iteration. However, it may be advantageous in other applications to use the stopping rule for the similar algorithm of Moré and Toraldo [12]; that is, stop gradient projection iteration when

$$T(\mathbf{u}_{k-1}) - T(\mathbf{u}_k) \leq \gamma_{GP} \max\{T(\mathbf{u}_{i-1}) - T(\mathbf{u}_i) \mid i = 1, \dots, k-1\}, \quad (4.18)$$

where $0 < \gamma_{GP} < 1$. In [12], γ_{GP} is taken to be 0.1.

Gradient Projection-Reduced Newton (GPRN) Iteration

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

Step 1: Given \mathbf{u}_k .

- (1) Compute gradient projection iterations $\{\mathbf{u}_{k,j}\}_{j=0}^{j_k}$, with $\mathbf{u}_{k,0} \stackrel{\text{def}}{=} \mathbf{u}_k$, until either (4.18) is satisfied or GP_{\max} iterations have been computed.

Step 2: Given $\mathbf{u}_k \stackrel{\text{def}}{=} \mathbf{u}_{k,j_k}$.

- (1) Do CG iterations to approximately minimize the quadratic (4.15) until either (4.16) is satisfied or CG_{\max} iterations have been computed. Return $\mathbf{p}_k = \mathbf{p}_k^{m_{\text{CG}}}$.
- (2) Find λ_k^m that satisfies (4.17), and return $\mathbf{u}_{k+1} = \mathbf{u}_k(\lambda_k^m)$.
- (3) Update $k := k + 1$ and return to Step 1.

4.5. Proof of Convergence. We now prove that GPRN applied to (4.1), with T satisfying Assumption 1 and 2 stated at the beginning of Section 4, is convergent. We begin with a lemma that provides a bound on the line search parameters used within the gradient projection iterations in Stage 1 of the GPRN iterations. We will use the notation $\mathbf{u}(\lambda) = \mathcal{P}(\mathbf{u} - \lambda \nabla T(\mathbf{u}))$.

LEMMA 4.1. *Let $\{\{\lambda_{k,j}\}_{j=0}^{j_k-1}\}_{k=1}^{\infty}$ be the set of line search parameters generated by the gradient projection algorithm within Step 1 of the GPRN algorithm (i.e., $\mathbf{u}_{k+1,j} = \mathbf{u}_{k,j}(\lambda_{k,j})$). Then, there exists constants β and M such that*

$$0 < \beta < \lambda_{k,j} < M \text{ for all } j \text{ and } k.$$

Proof. First, note that (4.10) holds if (c.f. [11, Theorem 5.4.5])

$$0 \leq \lambda_{k,j} \leq \frac{2(1-\mu)}{L},$$

where L is the Lipschitz constant for ∇T given by Assumption 2. Using (4.5) together with (4.12) and (4.13), it can be shown that

$$\min \left\{ \sigma_{\max,k,j}^{-1}, (1-\mu)/(50 \cdot L) \right\} \leq \lambda_{k,j} \leq \sigma_{\min,k,j}^{-1},$$

where $\sigma_{\max,k,j}$ and $\sigma_{\min,k,j}$ are the maximum and minimum eigenvalues, respectively, of $\nabla^2 T(\mathbf{u}_{k,j})$. Now, we note that with each gradient projection iteration, and also in Step 2, the value of T decreases. Thus, since T is coercive, the set of all gradient projection iterates computed within Stage 1 of GPRN, i.e. $\{\{\mathbf{u}_{k,j}\}_{j=0}^{j_k}\}_{k=1}^{\infty}$, is bounded. Given that T is convex and twice continuously differentiable on Ω , this implies that the set $\{\sigma_{\max,k,j}\}$ is positive and bounded above and the set $\{\sigma_{\min,k,j}\}$ is positive and bounded away from zero. This establishes the result. \square

Next, we state a lemma, whose proof can be found in [11, Section 5.4].

LEMMA 4.2. *For every $\mathbf{u}, \mathbf{y} \in \Omega$ and $\lambda \geq 0$,*

$$\lambda \langle \nabla T(\mathbf{u}), \mathbf{y} - \mathbf{u}(\lambda) \rangle \geq \langle \mathbf{u} - \mathbf{u}(\lambda), \mathbf{y} - \mathbf{u}(\lambda) \rangle. \quad (4.19)$$

The definition of stationary point will be needed in our main result.

DEFINITION 4.3. *A stationary point for problem (4.1) is a vector $\bar{\mathbf{u}} \geq \mathbf{0}$ satisfying $\langle \nabla T(\bar{\mathbf{u}}), \bar{\mathbf{u}} - \mathbf{y} \rangle \leq 0$, for all $\mathbf{y} \in \Omega$.*

REMARKS: We note that if $\bar{\mathbf{u}}$ is a stationary point of (4.1) then $\nabla_{\text{proj}} T(\bar{\mathbf{u}}) = \mathbf{0}$. Moreover, since T is strictly convex $\bar{\mathbf{u}}$ is a solution of (4.1) if and only if it is a stationary point.

THEOREM 4.4. *The iterates $\{\mathbf{u}_k\}$ generated by GPRN converge to the unique solution of problem (4.1) for any initial guess $\mathbf{u}_0 \geq \mathbf{0}$.*

Proof. Since $\{T(\mathbf{u}_k)\}$ is monotone decreasing and bounded below, the coercivity of T implies that \mathbf{u}_k is a bounded set. Hence it has a convergent subsequence. Suppose $\{\mathbf{u}_{k_\ell}\}$ is the convergent sequence and $\bar{\mathbf{u}}$ its limit. Then $T(\mathbf{u}_k) \rightarrow T(\bar{\mathbf{u}})$.

Note that $\mathbf{u}_{k_\ell,0} = \mathbf{u}_{k_\ell}$ and let $\mathbf{u}_{k_\ell,1}$ be defined as in Stage 1 of the GPRN iteration above. Then $T(\mathbf{u}_{k_\ell,0}) > T(\mathbf{u}_{k_\ell,1})$, and hence $T(\mathbf{u}_{k_\ell,1}) \rightarrow T(\bar{\mathbf{u}})$. Moreover, by (4.10),

$$\|\mathbf{u}_{k_\ell,0} - \mathbf{u}_{k_\ell,1}\|^2 \leq (\lambda_{k_\ell,0}/\mu)[T(\mathbf{u}_{k_\ell,0}) - T(\mathbf{u}_{k_\ell,1})],$$

which converges to zero, since by Lemma 4.1, the $\lambda_{k,0}$'s are bounded above. Thus

$$\|\mathbf{u}_{k_\ell,0} - \mathbf{u}_{k_\ell,1}\| \rightarrow 0. \quad (4.20)$$

Now, using Lemma 4.2, for all $\mathbf{y} \in \Omega$,

$$\begin{aligned} \langle \nabla T(\mathbf{u}_{k_\ell}), \mathbf{u}_{k_\ell} - \mathbf{y} \rangle &= \langle \nabla T(\mathbf{u}_{k_\ell,0}), \mathbf{u}_{k_\ell,1} - \mathbf{y} \rangle + \langle \nabla T(\mathbf{u}_{k_\ell,0}), \mathbf{u}_{k_\ell,0} - \mathbf{u}_{k_\ell,1} \rangle \\ &\leq \frac{1}{\lambda_{k_\ell,0}} \langle \mathbf{u}_{k_\ell,0} - \mathbf{u}_{k_\ell,1}, \mathbf{u}_{k_\ell,1} - \mathbf{y} \rangle + \langle \nabla T(\mathbf{u}_{k_\ell,0}), \mathbf{u}_{k_\ell,0} - \mathbf{u}_{k_\ell,1} \rangle \\ &\leq \|\mathbf{u}_{k_\ell,0} - \mathbf{u}_{k_\ell,1}\| \cdot \left\| \frac{\mathbf{u}_{k_\ell,1} - \mathbf{y}}{\lambda_{k_\ell,0}} + \nabla T(\mathbf{u}_{k_\ell,0}) \right\|. \end{aligned}$$

Lemma 4.1 tells us that the $\lambda_{k_\ell,0}$'s are bounded below. Also, since ∇T is Lipschitz continuous, it is bounded on $\{\mathbf{u}_{k_\ell}\}$. Hence, taking limits as $k_\ell \rightarrow \infty$ and using (4.20), we have $\langle \nabla T(\bar{\mathbf{u}}), \bar{\mathbf{u}} - \mathbf{y} \rangle \leq 0$, for all $\mathbf{y} \in \Omega$. Thus $\bar{\mathbf{u}}$ is a stationary point, which is also the unique solution of (4.1), since T is strictly convex.

Now we must show that $\mathbf{u}_k \rightarrow \bar{\mathbf{u}}$. By Taylor's Theorem, given $\mathbf{p} \in \mathbb{R}^N$ such that $\bar{\mathbf{u}} + \mathbf{p} \in \Omega$,

$$T(\bar{\mathbf{u}} + \mathbf{p}) = T(\bar{\mathbf{u}}) + \langle \nabla T(\bar{\mathbf{u}}), \mathbf{p} \rangle + \frac{1}{2} \langle \nabla^2 T(\bar{\mathbf{u}} + t\mathbf{p})\mathbf{p}, \mathbf{p} \rangle,$$

for some $t \in (0, 1)$. Letting $\mathbf{p} = \mathbf{u}_k - \bar{\mathbf{u}}$ yields

$$\begin{aligned} T(\mathbf{u}_k) - T(\bar{\mathbf{u}}) &= \langle \nabla T(\bar{\mathbf{u}}), \mathbf{u}_k - \bar{\mathbf{u}} \rangle + \frac{1}{2} \langle \nabla^2 T(\bar{\mathbf{u}} + t(\mathbf{u}_k - \bar{\mathbf{u}}))(\mathbf{u}_k - \bar{\mathbf{u}}), \mathbf{u}_k - \bar{\mathbf{u}} \rangle \\ &\geq \frac{1}{2} \langle \nabla^2 T(\bar{\mathbf{u}} + t(\mathbf{u}_k - \bar{\mathbf{u}}))(\mathbf{u}_k - \bar{\mathbf{u}}), \mathbf{u}_k - \bar{\mathbf{u}} \rangle \\ &\geq \frac{\sigma_{min}}{2} \|\mathbf{u}_k - \bar{\mathbf{u}}\|^2, \end{aligned} \quad (4.21)$$

where $\sigma_{min} \stackrel{\text{def}}{=} \min\{\sigma_{\min,k}\}$ (see the proof of Lemma 4.1 for the definition of $\sigma_{\min,k}$). The first inequality follows from Definition 4.3 and the second from the convexity of T and the boundedness of $\{\mathbf{u}_k\}$. Finally, since $\{T(\mathbf{u}_k)\}$ converges to $T(\bar{\mathbf{u}})$ and $\sigma_{\min} > 0$, (4.21) implies that $\mathbf{u}_k \rightarrow \bar{\mathbf{u}}$. \square

5. Numerical Experiments. In this section, we present an example that illustrates the effectiveness of using the regularization function presented in Section 2 together with the negative-log Poisson likelihood. For this, we choose the standard simulated satellite seen on the left side in Figure 5.1. We will denote this image by $\mathbf{u}_{\text{exact}}$. Generating corresponding blurred noisy data requires a discrete PSF \mathbf{a} , which we compute using the Fourier optics PSF model

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

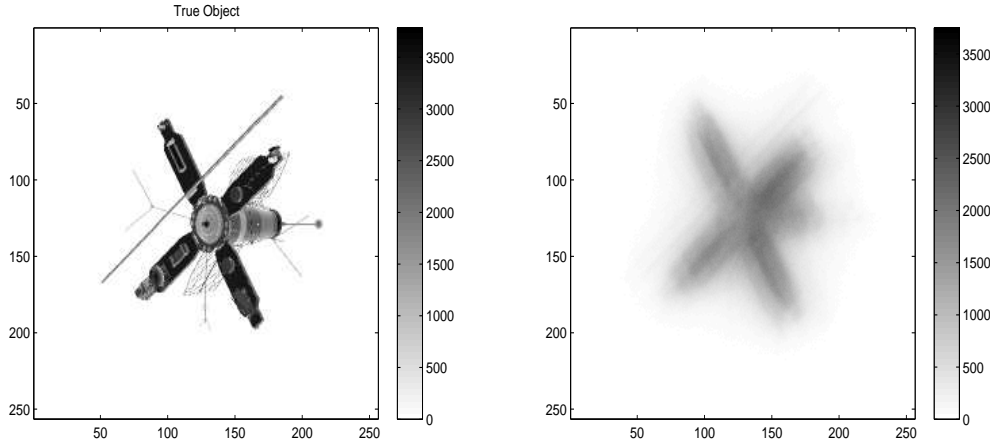


FIG. 5.1. On the left is the true object. On the right, is the blurred, noisy image \mathbf{z} .

where \mathbf{p} is the $\sqrt{N} \times \sqrt{N}$ indicator array for the telescopes pupil; ϕ is the $\sqrt{N} \times \sqrt{N}$ 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 $256^2 \times 256^2$ blurring matrix \mathbf{A} obtained from \mathbf{a} is block circulant with circulant blocks (BCCB) [15]—we have assumed periodic boundary conditions—which allows for efficient storage of and multiplication by \mathbf{A} via diagonalization by the discrete Fourier transform [15]. Data \mathbf{z} with a signal-to-noise ratio of approximately 30 is then generated using (1.2) with $\sigma^2 = 25$ and $\gamma = 10$, which are 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 right in Figure 5.1.

As was mentioned in Section 2, our regularization function will be of the form (2.1). We define $\mathbf{\Lambda}$ as follows. First, let

$$\mathbf{v} = [\mathbf{\Gamma}_x \mathbf{u}_{\text{approx}}]^2 + [\mathbf{\Gamma}_y \mathbf{u}_{\text{approx}}]^2, \quad (5.1)$$

where $\mathbf{u}_{\text{approx}}$ is an approximate solution of (3.1) and $\mathbf{\Gamma}_x$ and $\mathbf{\Gamma}_y$ are discrete x and y gradient matrices. (We used MATLAB’s `gradient` function for these computations.) We then define

$$\mathbf{v}_\epsilon = \max\{\mathbf{v}, \epsilon * \mathbf{v}\}, \quad (5.2)$$

where $0 < \epsilon < 1$ (we chose $\epsilon = 0.01$ for our experiments). $\mathbf{\Lambda}$ is then given by

$$\mathbf{\Lambda} = \text{diag} \left(\max \left\{ \frac{\mathbf{1}}{\mathbf{1} + \mathbf{v}_\epsilon}, \frac{1}{10} \right\} \right), \quad (5.3)$$

where division is computed component-wise. Note that when $[\mathbf{v}_\epsilon]_i$ is large, i.e. at or near an edge, $\mathbf{\Lambda}$ has the effective of de-emphasizing the regularization by an order of magnitude, whereas when $[\mathbf{v}_\epsilon]_i \approx 0$ the regularization parameter remains approximately the same.

In order to compute $\mathbf{\Lambda}$, we need an approximation $\mathbf{u}_{\text{approx}}$, which we take to be the solution of (3.1) with standard Laplacian regularization (i.e. $\mathbf{\Lambda} = \mathbf{I}$). The

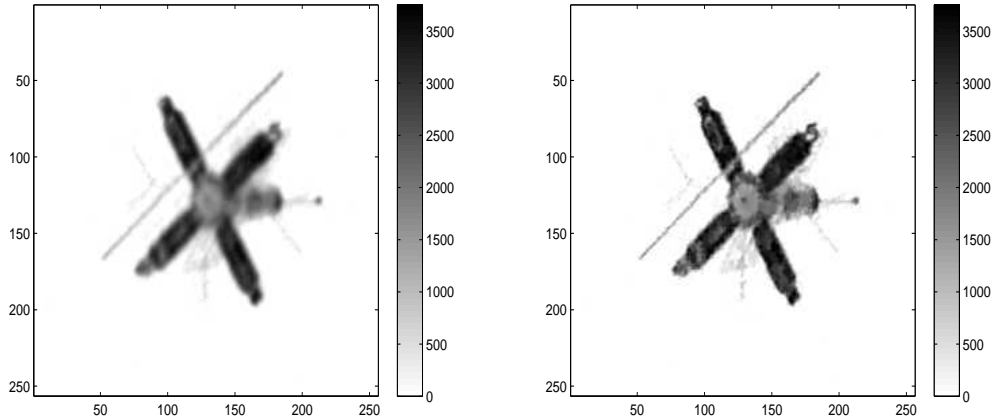


FIG. 5.2. On the left is the reconstruction obtained using Laplacian regularization with $\alpha = 10^{-5}$. On the right, is the reconstruction obtained using the regularization function defined by (2.1), (5.1)-(5.3) with $\alpha = 10^{-5}$.

regularization parameter α was taken to be 10^{-5} , which approximately minimizes $\|\mathbf{u}_\alpha - \mathbf{u}_{\text{exact}}\|_2$ with \mathbf{u}_α defined to be the regularized reconstruction, and is plotted on the left-hand side in Figure 5.2. Setting $\mathbf{u}_{\text{approx}} = \mathbf{u}_\alpha$, we then compute Λ via (5.1)-(5.3) and solve (3.1) again. The resulting reconstruction, which is clearly of a higher resolution, is given on the right in Figure 5.2.

Problem (3.1) is solved, in all cases, using GPRN with initial guess $\mathbf{u}_0 = \mathbf{1}$ and optimization parameters $\text{GP}_{\text{max}} = 5$, $\gamma_{\text{GP}} = 0.1$, $\gamma_{\text{CG}} = 0.1$, and $\text{CG}_{\text{max}} = 40$. We stop iterations once

$$\|\nabla_{\text{proj}} T_\alpha(\mathbf{u}_k)\|_2 / \|\nabla_{\text{proj}} T_\alpha(\mathbf{u}_0)\|_2 < \text{GradTol}, \quad (5.4)$$

where $\text{GradTol} = 10^{-6}$. The method converged rapidly in all of our tests; in fact, the CPU time required to solve (3.1) for standard Laplacian regularization (6.6 sec) hardly increased when diffusion regularization (2.1), (5.1)-(5.3) was used (8.8 sec). (All computations were done on a Dell Desktop with a 2.6 GHz processor and 2 GB RAM.) Due to the computational efficiency of the method, preconditioning was unnecessary, though it is possible using the approach presented in [1, 4].

One can further refine reconstructions by taking $\mathbf{u}_{\text{approx}}$ to be the most recent reconstruction obtained using the diffusion regularization function (2.1), (5.1)-(5.3), leaving α the same in all computations. This suggests the following iteration.

Iteratively Updated Diffusion Regularization

Step 0: Fix $\alpha > 0$, and set $\Lambda = \mathbf{I}$.

Step 1: Compute the solution \mathbf{u}_α of (3.1) with regularization function (2.1).

Step 2: Set $\mathbf{u}_{\text{approx}} = \mathbf{u}_\alpha$ and update Λ using (5.1)-(5.3), then return to Step 1.

In Figure 5.3, we give reconstructions obtained after 4 and 6 iterations of this algorithm with $\alpha = 10^{-5}$; note that the reconstructions in Figure 5.2 are those obtained after 1 and 2 iterations. There is a noticeable improvement in quality going from the second to the 4th iteration, however, it is difficult to see much difference between the

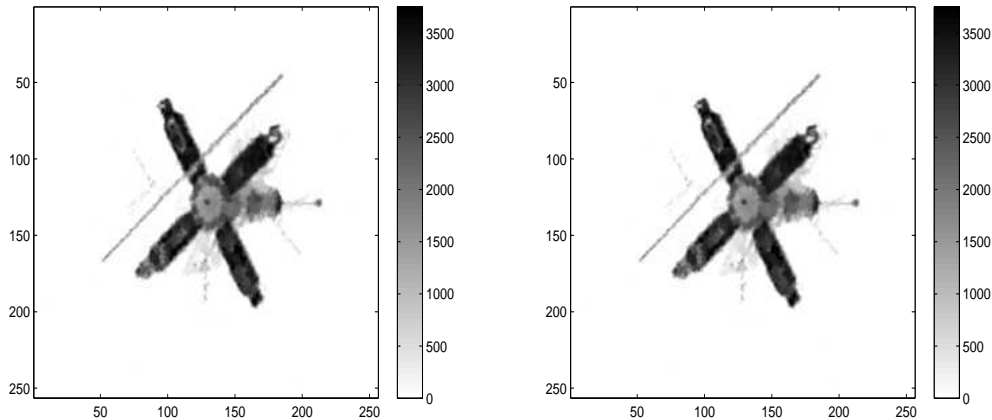


FIG. 5.3. On the left is the reconstruction obtained after 4 iterations of iteratively updated diffusion regularization, and on the right, is the reconstruction obtained after 6 iterations with $\alpha = 10^{-5}$.

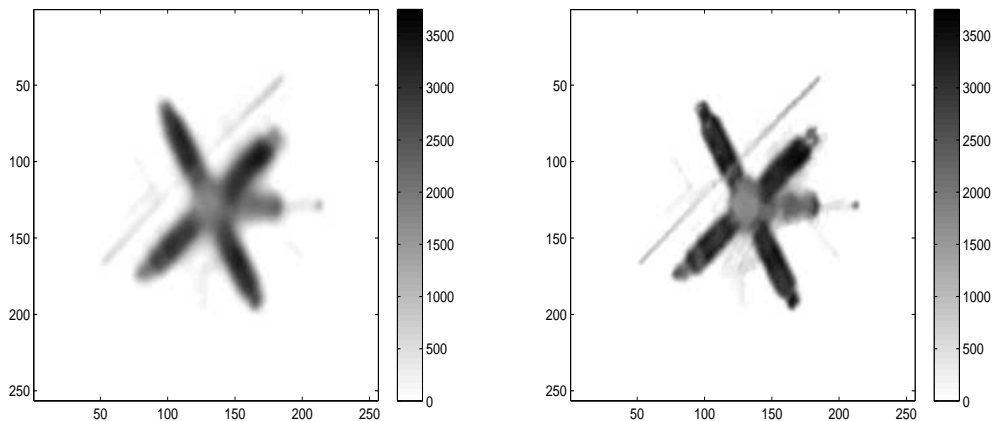


FIG. 5.4. On the left is the reconstruction obtained after 1 iterations of iteratively updated diffusion regularization, and on the right, is the reconstruction obtained after 2 iterations with $\alpha = 10^{-4}$.

reconstructions obtained after 4 and 6 iterations.

This iterative approach is also effective in the event that the regularization parameter has been over-estimated. For example, suppose that we have chosen $\alpha = 10^{-4}$ in Step 0 of the algorithm. Although the first reconstruction is quite blurry (see the left-hand image in Figure 5.4), we see that higher resolution reconstructions can be obtained with the iterative approach (see the right-hand image in Figure 5.4).

We note that a statistically rigorous updating scheme for Λ is given in [7]. There the diagonal of Λ is assumed to arise from a hyper-prior distribution. Using this structure, the update of Λ is put on firm footing. We hope to explore this approach in future work.

6. Conclusions. We have presented a simple edge preserving, quadratic regularization function for use on regularized, negative-log Poisson likelihood (or MAP) estimation problems. In order to use this regularization function, however, an effective iterative method is needed. Such a method was presented in [4], however there, a convergence proof of the method was not given. We have given such a proof here in the context of general nonnegatively constrained, strictly convex minimization problems. For this reason, we have presented the method in detail.

The effectiveness of the edge preserving regularization function that we have introduced was demonstrated in our numerical experiments. A scheme for iteratively updating the regularization function was also presented. The numerical results indicate that our approach—the regularization and computational methods combined—is promising. In particular, the resolution of reconstructions noticeably improves when the edge preserving regularization function is used, and the computational method is extremely efficient. Moreover, both are straightforward to implement.

REFERENCES

- [1] Johnathan M. Bardsley, *An Efficient Computational Method for Total Variation-Penalized Poisson Likelihood Estimation*, Inverse Problems and Imaging, vol. 2, no. 2, 2008, pp. 167 - 185.
- [2] Johnathan M. Bardsley, *A Theoretical Framework for the Regularization of Poisson Likelihood Estimation Problems*, University of Montana, Department of Mathematical Sciences, Technical Report #25, 2008.
- [3] Johnathan M. Bardsley and N'djekornom Laobeul, *An Analysis of Regularization by Diffusion for Ill-Posed Poisson Likelihood Estimation*, accepted in Inverse Problems in Science and Engineering, University of Montana, Department of Mathematical Sciences, Technical Report #23, 2008.
- [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] D. P. Bertsekas, *On the Goldstein-Levitin-Poljak Gradient Projection Method*, IEEE Transactions on Automatic Control, **21** (1976), pp. 174–184.
- [6] Daniela Calvetti and Erkki Somersalo, *A Gaussian Hypermodel for Recovering Blocky Objects*, Inverse Problems 23, 2007, pp. 733-754.
- [7] Daniela Calvetti and Erkki Somersalo, , Inverse Problems ??, 2008, pp. ??.
- [8] J. W. Goodman, *Introduction to Fourier Optics, 2nd Edition*, McGraw-Hill, 1996.
- [9] J. P. Kaipio, V. Kolehmainen, M. Vauhkonen, and E. Somersalo, *Inverse Problems with Structural Prior Information*, Inverse Problems, 15, 1999, pp.713-729.
- [10] Jari Kaipio and Erkki Somersalo, *Satistical 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] 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.
- [15] C. R. Vogel, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, 2002.
- [16] Daniel F. Yu and Jeffrey A. Fessler, *Edge-Preserving Tomographic Reconstruction with Nonlocal Regularization*, IEEE Trans. on Medical Imaging, 21(2), 2002, pp. 159-173.