A NONNEGATIVELY CONSTRAINED CONVEX PROGRAMMING METHOD FOR IMAGE RECONSTRUCTION

JOHNATHAN M. BARDLEY† AND CURTIS R. VOGEL‡

Abstract. We consider a large-scale convex minimization problem with nonnegativity constraints that arises in astronomical imaging. We develop a cost functional which incorporates the statistics of the noise in the image data and Tikhonov regularization to induce stability. We introduce an efficient hybrid gradient projection-reduced Newton (active set) method. By “reduced Newton” we mean taking Newton steps only in the inactive variables. Due to the large size of our problem, we compute approximate reduced Newton steps using conjugate gradient (CG) iteration. We also introduce a highly effective sparse preconditioner that dramatically speeds up CG convergence. A numerical comparison between our method and other standard large-scale constrained minimization algorithms is presented.

Key words. constrained optimization, astronomical imaging

AMS subject classifications. 65K10, 65F22

DOI. 10.1137/S1064827502410451

1. Introduction. Image reconstruction gives rise to some challenging large-scale constrained optimization problems. In this paper we consider a convex minimization problem with nonnegativity constraints that arises in astronomical imaging. Data for this problem take the form

(1.1) \[ d = S f_{\text{true}} + \eta, \]

where \( S \) is a large, nonsparse, block Toeplitz, ill-conditioned matrix, and \( \eta \) represents noise. To accurately reconstruct \( f_{\text{true}} \), we apply Tikhonov regularization (i.e., stabilization with an additive penalty term [24]) and incorporate prior knowledge of the statistics of the noise and the nonnegativity of \( f_{\text{true}} \). This yields an optimization problem

(1.2) \[ \min_{f \geq 0} J(f), \]

where \( f \geq 0 \) means \( f_i \geq 0 \) for each \( i \), and the cost functional \( J : \mathbb{R}^N \to \mathbb{R} \) has the form

(1.3) \[ J(f) = \ell(Sf; d) + \frac{\alpha}{2} f^T L f. \]

Here \( L \) is a symmetric positive definite (SPD), sparse regularization matrix; \( \ell \) is a fit-to-data functional; and the regularization parameter \( \alpha > 0 \) quantifies the trade-off between data fidelity and stability.
For computational convenience, \( \ell \) can be taken to have least squares form,

\[
\ell(Sf, d) = \frac{1}{2} \|Sf - d\|^2.
\]

This is appropriate when the noise in the data is additive Gaussian with independent components and common variance. Combining (1.4) with (1.3) results in a regularized least squares functional which is quadratic.

In many imaging applications, the data can be more accurately modeled as a sum of independent Poisson random variables. We will consider such a case in which the appropriate fit-to-data functional has (negative) Poisson log likelihood form,

\[
\ell(Sf, d) = \sum_{i=1}^{N} (|Sf|_i + \beta + \sigma^2) - \sum_{i=1}^{N} (d_i + \sigma^2) \log(|Sf|_i + \beta + \sigma^2),
\]

where \( \beta \) and \( \sigma \) are positive parameters. The corresponding regularized Poisson likelihood functional (1.3) is not quadratic, but it is convex.

Our goal is the efficient minimization of the regularized Poisson likelihood functional under the constraint \( f \geq 0 \). We motivate the need for constrained minimization algorithms by noting that, in many astronomical image processing problems, most of the variables (pixels) will be at their lower bound of zero (black). This minimization problem is quite challenging. It is large-scale, nonquadratic, and somewhat ill-conditioned. Moreover, it is nearly degenerate. Recall that a nonnegatively constrained minimizer \( f^* \) of a functional \( J \) is nondegenerate if we have the strict complementarity condition,

\[
\frac{\partial J}{\partial f_i}(f^*) > 0 \text{ whenever } f_i^* = 0.
\]

A variety of large-scale constrained minimization methods can be applied, including the gradient projection method [1], active set methods such as the projected Newton-type methods [2, 10], a bound-constrained variant of the limited memory BFGS method [4], bound-constrained trust region methods [11], and interior point methods [3]. In this paper we introduce an efficient hybrid gradient projection-reduced Newton active set method. By “reduced Newton” we mean taking Newton steps only in the inactive variables. Due to the large size of our problem, we compute approximate reduced Newton steps using conjugate gradient (CG) iteration. The resulting algorithm resembles the gradient projection CG (GPCG) algorithm of Moré and Toraldo [15] for large-scale bound-constrained quadratic minimization. GPCG can be viewed as an enhancement of the active set/CG methods of O’Leary [18, 19]; it improves efficiency by allowing the active set to change dramatically at each iteration. By applying ideas from GPCG to our nonquadratic problem, we obtain convergence and active set identification properties superior to those of the algorithms found in [2, 10]. This is particularly true when the minimizer of the cost function \( J \) is a nearly degenerate stationary point, which is the case in our application. We also introduce in this paper a highly effective sparse matrix preconditioner to speed up CG convergence.

A fundamentally different approach to image reconstruction is to apply a “few” steps of an iterative method to the unregularized fit-to-data functional (either least squares (1.4) or Poisson (1.5)). The number of iterations then plays the role of the regularization parameter; see [24]. In the Poisson case a variant of the expectation maximization (EM) algorithm [13, 9] known as the Richardson–Lucy iteration [20, 12]
is appropriate because it maintains nonnegativity and monotonically decreases the negative Poisson log likelihood functional. This can sometimes be a viable alternative to the application of constrained optimization techniques to an explicitly regularized functional of the form (1.3).

This paper is organized as follows. In section 2 we present an integral equation used to model image formation. We also discuss discretization of this equation as well as the statistics of the noise in the data, and we derive the fit-to-data functional (1.5) based on these statistics. Section 3 contains background material related to constrained optimization. Also in this section we present our computational algorithm and our sparse matrix preconditioner. Numerical results appear in section 4. A realistic test problem from astronomical imaging is used to compare our algorithm with standard methods, including Richardson–Lucy. Conclusions are also presented in this section.

2. The mathematical model. A model for pixelated data obtained from a linear, monochromatic, translation-invariant optical imaging system [7] is

\[
d_{i,j} = \int \int s(x_i - x, y_j - y)f_{\text{true}}(x, y)dx\,dy + \eta_{i,j}
\]

for \(i = 1, \ldots, n_x, j = 1, \ldots, n_y\). Here \(f_{\text{true}}\) denotes the light source, or object. We will assume that the object is incoherent [6]. Then the object represents an energy density, or photon density, and hence is nonnegative. The \(s\) in (2.1) is called the point spread function (PSF) and is the response of the imaging system to a point light source. With an incoherent object, the PSF is also nonnegative. The two-dimensional array \(d\) in (2.1) is called the (discrete, noisy, blurred) image, and \(\eta\) represents noise processes in the formation of the image.

For computational purposes, we discretize the integral in (2.1), e.g., using midpoint quadrature, to obtain

\[
d_{i,j} = \sum_{i'} \sum_{j'} s_{i'-j',j'-j'}[f_{\text{true}}]_{i',j'} + \eta_{i,j}, \quad 1 \leq i \leq n_x, 1 \leq j \leq n_y.
\]

We refer to the array with components \(s_{i,j}\) as the discrete PSF. We assume that the quadrature error is negligible and that the components \(s_{i,j}\) are nonnegative.

With lexicographical ordering of unknowns, (2.2) can be rewritten as (1.1), where \(d, f_{\text{true}}, \eta\) are \(N \times 1\) vectors with \(N = n_x \cdot n_y\), and \(S\) is an \(N \times N\) matrix. We refer to \(S\) as the blurring matrix. Because of (2.2), \(S\) is block Toeplitz with Toeplitz blocks (BTTB) [24, Chapter 5]. Since \(s_{i,j} \geq 0\),

\[
Sf \geq 0 \quad \text{whenever} \quad f \geq 0.
\]

In typical imaging applications \(S\) is highly ill-conditioned, and in many cases it is also a singular matrix. Consequently, the pseudo-inverse estimate \(S^{\dagger}d\) for \(f_{\text{true}}\) is unstable with respect to perturbations in the data vector \(d\). In order to obtain stable, accurate approximations to \(f_{\text{true}}\) from noisy data, we employ an approach known to mathematicians as Tikhonov regularization and to statisticians as maximum a posteriori estimation (see [24, Chapter 4]). This requires the minimization of a functional of the form

\[
J(f) = \ell(Sf; d) + \frac{\alpha}{2}f^T Lf.
\]
The quadratic $\frac{1}{2} f^T L f$ is known as the regularization functional, or the prior. In addition to having a stabilizing effect on the minimization of (2.4), this functional allows the incorporation of prior smoothness information about the unknown $f$. $L$ is known as the regularization matrix; it is symmetric and positive semidefinite and is usually sparse.

The additional prior information that the source is nonnegative gives rise to the constraint $f \geq 0$.

The $\ell$ in (2.4) is called the fit-to-data functional. By choosing $-\ell$ to be a log-likelihood functional, one can incorporate prior statistical information about noise in the data. The nonnegative real number $\alpha$ is known as the regularization parameter, and it quantifies the trade-off between fit-to-data and stability.

2.1. The data noise model. Astronomical image data is typically collected with a device known as a charged-coupled-device (CCD) camera. The following statistical model (see [22, 23]) applies to image data from a CCD detector array:

\begin{equation}
\label{eq:2.5}
d_i = n_{\text{obj}}(i) + n_0(i) + g(i), \quad i = 1, \ldots, N.
\end{equation}

Here $d_i$ is the $i$th component of the vector $d$ and is the data acquired by a read-out of pixel $i$ of the CCD detector array; $n_{\text{obj}}(i)$ is the number of object-dependent photoelectrons; $n_0(i)$ is the number of background electrons; and $g(i)$ is the readout noise. The random variables $n_{\text{obj}}(i)$, $n_0(i)$, and $g(i)$ are assumed to be independent of one another and of $n_{\text{obj}}(j)$, $n_0(j)$, and $g(j)$ for $i \neq j$. The random variable $n_{\text{obj}}(i)$ has a Poisson distribution with Poisson parameter $[Sf_{\text{true}}]_i$; $n_0(i)$ is a Poisson random variable with a fixed positive Poisson parameter $\beta$; and $g(i)$ is a Gaussian random variable with mean 0 and fixed variance $\sigma^2$. Then the error term $\eta$ in the model equation (1.1) has components

\begin{equation}
\eta_i = n_{\text{obj}}(i) - [Sf_{\text{true}}]_i + n_0(i) + g(i).
\end{equation}

The log likelihood for the mixed Poisson–Gaussian model (2.5) has an infinite series representation (see [22, equation (10)]) which is computationally intractable. To approximate it we observe as in [22, 23] that the random variable $g(i) + \sigma^2$ has a Gaussian distribution with mean and variance both equal to $\sigma^2$. For large $\sigma^2$, this is well approximated by a Poisson random variable with Poisson parameter $\lambda_i = [Sf]_i + \beta + \sigma^2$. Then $d_i + \sigma^2$ is well approximated by a Poisson random variable with Poisson parameter $\lambda_i$. From this we obtain the corresponding negative Poisson log-likelihood functional

\begin{equation}
\ell(Sf; d) = \sum_{i=1}^{N} ([Sf]_i + \beta + \sigma^2) - \sum_{i=1}^{N} (d_i + \sigma^2) \log([Sf]_i + \beta + \sigma^2).
\end{equation}

Note that for moderate to large values of $\sigma^2$, say $\sigma^2 \geq 3^2$, it is extremely unlikely for the Gaussian $g(i) + \sigma^2$ to take on negative values. Then since Poisson random variables take on only nonnegative integer values, the random variable $d_i + \sigma^2$ is also highly unlikely to take on negative values.

Since $\beta$ and $\sigma^2$ are both positive parameters and $Sf \geq 0$ whenever $f \geq 0$, the regularized Poisson likelihood functional $J$ defined by (2.4) and (2.6) is infinitely differentiable for $f \geq 0$. Its gradient is

\begin{equation}
\text{grad } J(f) = S^T \left( (Sf + \beta - d) / (Sf + \beta + \sigma^2) \right) + \alpha L f.
\end{equation}
where ./ denotes componentwise division. Its Hessian is
\[
\text{Hess } J(f) = S^T W(f) S + \alpha L,
\]
where \(W(f) = \text{diag}(w)\) and the components of \(w\) are given by
\[
w_i = \frac{d_i + \sigma^2}{(|Sf|_i + \beta + \sigma^2)^2}.
\]

Recall that a functional \(J\), defined on an unbounded set \(\Omega \subset \mathbb{R}^N\), is coercive if
\[
J(f) \to +\infty \quad \text{whenever} \quad f \in \Omega \quad \text{and} \quad ||f|| \to +\infty.
\]

**Theorem 2.1.** Let \(J\) denote the regularized Poisson likelihood functional defined by (2.4) and (2.6). Assume that \(d_i + \sigma^2 \geq 0\) and that \(L\) is SPD. Then \(J\) is strictly convex and coercive for \(f \geq 0\), and hence the nonnegatively constrained problem (1.2) has a unique solution.

**Proof.** The coercivity of \(J\) follows from the fact that for large \(||f||\), the quadratic regularization term \(\alpha f^T L f / 2\) dominates the linear and logarithmic terms in the fit-to-data functional (2.6). Also, since \(\alpha > 0\) and \(L\) is SPD with minimum eigenvalue \(\lambda_{\min}(L) > 0\), \(\alpha f^T L f / 2 \geq \alpha \lambda_{\min}(L)||f||^2 / 2 \to +\infty\) as \(||f|| \to +\infty\).

To prove that \(J\) is strictly convex for \(f \geq 0\), note that by (2.3) the denominator of \(w_i\) in (2.9) is bounded below by \((\beta + \sigma^2)^2\). Since \(d_i + \sigma^2 \geq 0\) for each \(i\), this implies that the diagonal entries of \(W(f)\) are bounded below by some \(w_{\min} \geq 0\). Then by (2.8), for any \(v \in \mathbb{R}^N\),
\[
\begin{align*}
v^T \text{Hess } J(f) v &\geq v^T \left( w_{\min} S^T S + \frac{\alpha}{2} L \right) v \\
&\geq \frac{\alpha \lambda_{\min}(L)}{2} ||v||^2.
\end{align*}
\]
Hence \(\text{Hess } J(f)\) is positive definite for each \(f \geq 0\), and \(J\) is strictly convex.

Existence and uniqueness for problem (1.2) now follow from the fact that the constraint set \(\{f \in \mathbb{R}^N \mid f \geq 0\}\) is closed and convex. \(\square\)

This theorem also holds under the weaker assumption that \(L\) is symmetric positive semidefinite and \(L\) and \(S\) share only the zero vector in their null spaces. In our computations we take \(L\) to be the identity, so this additional generality is not needed.

**3. A gradient projection-reduced Newton-CG method.** In this section we introduce a nested iterative scheme for nonnegatively constrained convex minimization. Each outer iteration consists of two stages. The first stage consists of projected gradient iterations to identify the active set, while the second stage uses CG iterations to compute a Newton step on the inactive variables.

The cost functional \(J : \mathbb{R}^N \to \mathbb{R}\) is given in the previous section, with properties enumerated in Theorem 2.1. In particular, \(J(f)\) is smooth for \(f \geq 0\), strictly convex and coercive, and problem (1.2) has a unique solution.

**3.1. Preliminaries.** We first define the feasible set
\[
\Omega = \{f \in \mathbb{R}^N \mid f \geq 0\}.
\]
The projection of a vector \(f \in \mathbb{R}^N\) onto the feasible set is given by
\[
\mathcal{P}_\Omega(f) \overset{\text{def}}{=} \arg \min_{v \in \Omega} ||v - f|| = \max\{f, 0\},
\]
where \( \max\{f, 0\} \) is the vector whose \( i \)-th component is \( \max\{f_i, 0\} \). The active set for a vector \( f \in \Omega \) is given by
\[
\mathcal{A}(f) = \{ i \mid f_i = 0 \}.
\]
The complementary set of indices is called the inactive set and is denoted by \( \mathcal{I}(f) \). The inactive, or free, variables consist of the components \( f_i \) for which the index \( i \) is in the inactive set. Thus \( f_i > 0 \) whenever \( f \in \Omega \) and \( i \in \mathcal{I}(f) \).

Given \( J : \mathbb{R}^N \to \mathbb{R} \), the projected gradient of \( J \) at \( f \in \Omega \) is the \( N \)-vector with components
\[
(3.2) \quad [\text{grad}_P J(f)]_i = \begin{cases} \frac{\partial J(f)}{\partial f_i}, & i \in \mathcal{I}(f) \text{ or } \left( i \in \mathcal{A}(f) \text{ and } \frac{\partial J(f)}{\partial f_i} < 0 \right), \\ 0, & \text{otherwise.} \end{cases}
\]
The reduced gradient of \( J \) at \( f \in \Omega \) is given by
\[
(3.3) \quad [\text{grad}_R J(f)]_i = \begin{cases} \frac{\partial J(f)}{\partial f_i}, & i \in \mathcal{I}(f), \\ 0, & i \in \mathcal{A}(f), \end{cases}
\]
and the reduced Hessian is given by
\[
(3.4) \quad [\text{Hess}_R J(f)]_{ij} = \begin{cases} \frac{\partial^2 J(f)}{\partial f_i \partial f_j}, & \text{if } i \in \mathcal{I}(f) \text{ or } j \in \mathcal{I}(f), \\ \delta_{ij}, & \text{otherwise.} \end{cases}
\]

Let \( D_\mathcal{I} \) denote the diagonal matrix with components
\[
(3.5) \quad [D_\mathcal{I}(f)]_{ii} = \begin{cases} 1, & i \in \mathcal{I}(f), \\ 0, & i \in \mathcal{A}(f). \end{cases}
\]
Then
\[
(3.6) \quad \text{grad}_R J(f) = D_\mathcal{I}(f) \text{ grad } J(f),
\]
\[
(3.7) \quad \text{Hess}_R J(f) = D_\mathcal{I}(f) \text{ Hess } J(f) D_\mathcal{I}(f) + D_A(f),
\]
where \( D_A(f) = I - D_\mathcal{I}(f) \).

A vector \( f \in \Omega \) is the solution of problem (1.2) if and only if \( \text{grad}_P J(f) = 0 \). Our goal is to compute a sequence \( \{f_k\} \subset \Omega \) such that \( \text{grad}_P J(f_k) \to 0 \). Then \( \{f_k\} \) is guaranteed to converge to the unique global solution \( f^* \) of problem (1.2).

### 3.2. Gradient projection iteration
In principle, gradient projection generates a sequence of approximate minimizers \( \{f_k\} \subset \Omega \) via the following iteration:

\[
(3.8) \quad f_{k+1} = P_\Omega(f_k + \lambda_k p_k).
\]

In practice, subproblem (3.7) is solved inexactly using a projected backtracking line search. We take the initial step length parameter to be the so-called Cauchy point,
\[
(3.9) \quad \lambda_k^0 = \frac{||p_k||^2}{(\text{Hess } J(f_k)p_k, p_k)}.
\]
This is the minimizer of the quadratic \( \tilde{q}(\lambda) = q(f_k + \lambda p_k) \), where

\[
(3.10) \quad q(f + s) = J(f) + \langle \text{grad } J(f), s \rangle + \frac{1}{2} \langle \text{Hess } J(f), s, s \rangle.
\]

As in [1], step length reduction can be accomplished by taking \( \lambda_k^m = \beta^m \lambda_k^0 \), \( m = 0, 1, \ldots \), for some \( \beta \in (0, 1) \). A quadratic interpolation scheme as found in [15] can also be used. We stop at the first \( m \) for which the sufficient decrease condition

\[
(3.11) \quad J(f_k(\lambda_k^m)) \leq J(f_k) - \frac{\mu}{\lambda_k^m} ||f_k - f_k(\lambda_k^m)||^2
\]

holds, where \( \mu \in (0, 1) \) and

\[
(3.12) \quad f_k(\lambda) = P_I(f_k + \lambda p_k).
\]

**Theorem 3.1.** Let \( \{f_k\} \) be a sequence generated by the gradient projection iteration as discussed above. Then, if \( \bar{f} \) is a limit point of \( \{f_k\} \), \( \text{grad}_p J(\bar{f}) = 0 \). Furthermore, the optimal active set is identified in finitely many iterations. More precisely, there exists an integer \( m_0 \) such that for all \( k \geq m_0 \), \( A(f_k) = A(f^*) \).

The first half of this theorem is proved in [1] and the second half is proved in [8, Theorem 4.1]. Note that a nondegeneracy condition (see [16, p. 455]) is not required.

Under the hypotheses of Theorem 2.1 the regularized Poisson functional \( J \) is strictly convex and coercive, and we can prove a much stronger convergence result.

**Theorem 3.2.** Let \( J \) denote the functional defined by (2.4) and (2.6); assume \( d_i + \sigma^2 \geq 0 \) for each \( i \), and assume that \( L \) is positive definite. Then the gradient projection iteration \( f_k \) converges to the (unique) solution \( f^* \) to (1.2).

**Proof.** \( \{J(f_k)\} \) is a decreasing sequence which is bounded below by \( J(f^*) \). Hence it converges to some \( J \geq J(f^*) \). Since \( J \) is coercive, \( \{f_k\} \) is bounded, and hence there exists a subsequence \( \{f_{k_i}\} \) converging to some \( \bar{f} \). By Theorem 3.1, \( \text{grad}_p (\bar{f}) = 0 \), and since \( J \) is strictly convex, \( \bar{f} = f^* \). Thus \( \{J(f_k)\} \) converges to \( J(f^*) \). Then using Taylor’s theorem, (2.10), and the fact that \( f^* \) is the global constrained minimizer,

\[
J(f_k) - J(f^*) \geq \frac{\alpha \lambda_{\min}(L)}{2} ||f_k - f^*||^2.
\]

Consequently, \( \{f_k\} \) converges to \( f^* \). \( \Box \)

The asymptotic convergence rate for the gradient projection iteration is linear and can be quite slow when \( \text{Hess } J(f^*) \) is poorly conditioned. This motivates taking projected Newton steps.

**3.3. The reduced Newton step.** To obtain the standard projected Newton iteration [2, 10], one replaces the \( p_k \) in (3.6) with the solution to

\[
(3.13) \quad \text{Hess}_R J(f_k) p = -\text{grad}_R J(f_k).
\]

With a properly implemented line search, this yields a quadratically convergent scheme [2, 10]. However, it lacks robustness if the reduced Hessian is ill-conditioned and system (3.13) is solved inexactly. For this reason, we advocate interspersing gradient projection iterations with reduced Newton steps, in which we (approximately) solve

\[
(3.14) \quad \text{Hess}_R J(f_k) p = -\text{grad}_R J(f_k).
\]
(Note the replacement of the gradient on the right-hand side of (3.13) by the reduced gradient.) Equivalently, we approximately minimize the quadratic

$$q_k(p) = J(f_k) + \langle \text{grad}_{R} J(f_k), p \rangle + \frac{1}{2} \langle \text{Hess}_{R} J(f_k) p, p \rangle.$$  

(3.15)

This can be viewed as a minimization of the quadratic obtained from the three-term Taylor approximation to $J(f_k + p)$ over the face associated with $f_k$, defined by

$$F_k = \{ f \in \Omega \mid f_i = 0 \text{ whenever } i \in A(f_k) \}.$$  

When CG iteration is used to approximately solve (3.14) we obtain a scheme which resembles the GPCG algorithm of More and Toraldo [15] for bound-constrained quadratic minimization.

Given a reduced Newton search direction $p_k$, we again apply a projected backtracking line search. In this case the initial step length parameter (see (3.9)) will always be 1. We apply much less stringent acceptance criteria than (3.11), namely

$$J(f_k(\lambda_k^n)) < J(f_k).$$  

(3.16)

### 3.3.1. CG iteration and stopping criteria

Since Hess$_R$ $J(f_k)$ is symmetric and positive definite, CG iteration is guaranteed to converge to the unique minimizer $p_k$ of (3.15) in at most $N$ iterations. For notational simplicity, we drop the outer iteration index $k$ and let $p_j$ denote the approximation to $p_k$ obtained after $j$ CG iterations. By taking the initial guess $p_0 = 0$, we effectively reduce the number of inactive variables. This can speed up CG convergence significantly if the active set is relatively large. Preconditioning can further speed up convergence (see section 3.4 below).

Even with rapid CG convergence, it is important to choose effective stopping criteria to reduce overall computational cost. One often finds in the literature (see [10, 16]) a stopping criterion like

$$||g^j|| \leq \gamma ||g^0||^p,$$

where $g^j = \text{grad} q_k(p^j)$ and $\gamma$ and $p$ are parameters with $0 < \gamma < 1, p \geq 1$. We have found the following stopping criterion from More and Toraldo [15] to be much more effective:

$$q_k(p^{j-1}) - q_k(p^j) \leq \gamma_{CG} \max \{ q_k(p^{i-1}) - q_k(p^i) \mid i = 1, \ldots, j - 1 \},$$  

(3.17)

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

### 3.4. A sparse preconditioner

In many imaging applications, the discrete PSF is 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 4.1. Such a PSF can be accurately approximated by zeroing out the smaller components. The BTTB matrix $\hat{S}$ corresponding to such an approximate PSF is then a sparse approximation to the blurring matrix $S$. We select the truncated PSF $\hat{S}$ to be the array with entries

$$\hat{s}_{ij} = \begin{cases} 0, & s_{ij} \geq \tau, \\ s_{ij}, & \text{otherwise}, \end{cases}$$  

(3.18)
where the truncation parameter

\[ \tau = r \max_{i,j} s_{ij}, \quad 0 < r < 1. \]

We then take \( \hat{S} \) to be the BTTB matrix generated by \( \hat{s} \).

With lexicographical ordering of the unknowns, \( \hat{S} \) is a banded, sparse matrix. The bandwidth decreases and the sparsity increases either as the PSF becomes more concentrated about its central peak, or as the truncation parameter \( \tau \) increases.

Motivated by the forms of the Hessian (2.8) and the reduced Hessian (3.5), we take the preconditioning matrix at outer iteration \( k \) to be

\[ M_k = D^k_T \hat{S}^T W(f_k) \hat{S} D^k_T + \alpha D^k_T L D^k_T + D^k_A, \]

where \( D^k_T = D_T(f_k) \) and \( D^k_A = I - D_T(f_k) \). Note that \( D^k_T, D^k_A, W(f_k), \) and \( L \) are each diagonal matrices. Hence, due to the banded, sparse structure of \( \hat{S} \), each \( M_k \) is a banded, sparse matrix. Moreover, if the size of the active set increases with \( k \), the number of nonzero (diagonal) entries in \( D^k_T \) decreases, and \( M_k \) becomes even more sparse.

3.5. The numerical algorithm. In the first stage of our algorithm we need stopping criteria for the gradient projection iterations. Borrowing from Moré and Toraldo [15], we stop when

\[ J(f_{k-1}) - J(f_k) \leq \gamma_{GP} \max \{ J(f_{i-1}) - J(f_i) \mid i = 1, \ldots, k-1 \}, \]

where \( 0 < \gamma_{GP} < 1 \).

**Gradient projection-reduced Newton-cg (gprncg) algorithm.**

**Step 0:** Select initial guess \( f_0 \), and set \( k = 0 \).

**Step 1:** Given \( f_k \):

(1) Take gradient projection steps until (3.21) is satisfied. Return updated \( f_k \).

**Step 2:** Given \( f_k \):

(1) Do CG iterations to approximately minimize the quadratic (3.15) until (3.17) is satisfied. Return \( p_k \).

(2) Find \( \lambda^m_0 \) which satisfies (3.16), and return \( f_{k+1} = f_k(\lambda^m_0) \); see (3.12).

(3) Update \( k := k + 1 \) and return to Step 1.

The iterates \( \{ f_k \} \) generated by GPRNCG are guaranteed to converge to the unique solution \( f^* \) of problem (1.2). This follows from Theorem 3.2 since, at each outer GPRNCG iteration, at least one gradient projection iteration is taken and condition (3.16) holds.

4. Numerical results. In this section we present results obtained when various algorithms were applied to simulated data that were generated according to the model of section 2. The PSF \( s \) in (2.1) simulates the time-averaged point response of a ground-based adaptive optics system [21] used in astronomical imaging. This PSF is displayed in Figure 4.1. Note that it is quite localized in space and has a hint of the distinctive Airy pattern [7], which is characteristic of a diffraction-limited optical imaging system with a circular aperture. Note also that the blurring matrix \( S \) has many zero eigenvalues (this is also a consequence of diffraction limiting) and is ill-conditioned in the sense that the nonzero eigenvalues cluster at zero and the ratio of the largest eigenvalue to the smallest nonzero eigenvalue is quite large.
Fig. 4.1. PSF plots. On the left is a plot of the central 32 \times 32 pixels of the discrete PSF, s. To the right is a grayscale plot of the logarithmic-scaled power spectrum (squared magnitude of the Fourier transform) of the PSF.

Fig. 4.2. Astronomical object and image data. To the left is a grayscale plot of the log of the object (light source) f_{true}. On the right is the log of the blurred, noisy image data, d.

The simulated object f_{true} represents a star cluster. Both the object and the simulated blurred, noisy data are shown in Figure 4.2.

Noise in the data was generated by taking a pseudorandom realization from the statistical model (2.5). We selected parameter values \( \beta = 10 \) for sky background and \( \sigma = 5 \) for read-out noise. These values are representative of CCD cameras used in astronomy. The system size \( N = 128^2 = 16,384 \).

Before discussing algorithm performance, we will examine the reconstructions obtained using various optimization approaches. In each case, the regularization matrix \( L \) is taken to be the identity. We compare (i) unconstrained regularized ordinary least squares reconstructions,

\[
f^{\text{OLS}}_\alpha = (S^T S + \alpha L)^{-1} S^T d,
\]

which minimize (1.3)–(1.4); (ii) nonnegatively constrained, regularized weighted least
squares reconstructions $f_{\alpha}^{WLS}$, which minimize

$$J_{WLS}(f) = \frac{1}{2}(Sf - d)^T W (Sf - d) + \frac{\alpha}{2} f^T Lf;$$

and (iii) nonnegatively constrained, regularized Poisson likelihood reconstructions $f_{\alpha}^{LHD}$, which minimize (2.4), (2.6).

Weighted least squares reconstructions were computed using Moré and Toraldo’s GPCG algorithm [15]. The weight matrix in (4.1) is diagonal with diagonal entries taken to be the inverse of the variance of the noise in model (2.5),

$$w_{ii} = 1/([Sf_{true}] + \beta + \sigma^2).$$

Figure 4.3 shows the relative reconstruction error

$$E_{\alpha} = \|f_{\alpha} - f_{true}\|/\|f_{true}\|$$

for the three optimization approaches. The unconstrained ordinary least squares error curve is qualitatively different from the other two. It has a well-defined minimizer at $\alpha^* \approx 10^{-9}$. For smaller values of $\alpha$, the reconstructions deteriorate because of noise amplification associated with small singular values of the blurring matrix $S$. For $\alpha$ significantly larger than $\alpha^*$, the ordinary least squares reconstructions become too smooth as larger singular values of $S$ are damped out by the regularization. See the error analysis in [24].

The reconstruction error for constrained, weighted least squares decreases with $\alpha$ until $\alpha \approx 10^{-10}$. For values of $\alpha$ smaller than this, the error is essentially constant. The reconstruction error for constrained Poisson likelihood has this same behavior. We attribute this behavior to the fact that the object is a collection of “point-like” features with small support. Once the support of these features has been identified, their intensities appear to be relatively insensitive to the noise in the data.

Note that the Poisson likelihood reconstruction errors are somewhat smaller than for weighted least squares. This occurs because at relatively low photon counts (and hence, small Poisson parameters), the Poisson distribution differs significantly from a Gaussian distribution, and hence, it models the noise more accurately.

Figure 4.4 compares the reconstructions obtained with these three different approaches. Rather than viewing reconstructions of the entire object (see left subplot in Figure 4.2) at low resolution, we look at only the lower left corner, but at a much higher resolution. For the constrained weighted least squares and Poisson likelihood reconstructions, we picked reconstructions for which the regularization parameter $\alpha = 10^{-10}$. In the unconstrained ordinary least squares case, we selected the minimizer of the reconstruction error, $\alpha = 10^{-9}$. From this figure, we see that explicitly imposing nonnegativity constraints in the weighted regularized least squares minimization, rather than simply computing the unconstrained minimizer and then setting negative pixels to zero, results in a higher quality reconstruction. Moreover, there is a significant improvement in reconstruction quality when the weighted least squares fit-to-data functional is replaced with the Poisson likelihood.

We applied a variety of constrained optimization methods to minimize the regularized Poisson likelihood cost functional, including (i) GPRNCG without preconditioning; (ii) GPRNCG with the sparse preconditioner described in section 3.4; (iii) the bound-constrained limited memory BFGS (L-BFGS-B) [4] method; and (iv) the trust region interior point method KNITRO [3]. Both versions of GPRNCG were implemented in MATLAB. L-BFGS-B was implemented in MATLAB via a mex interface
Fig. 4.3. Relative reconstruction error versus regularization parameter $\alpha$. The dash-dotted line connected by asterisks (top graph) corresponds to unconstrained, regularized ordinary least squares. The dashed line connected by x’s (middle graph) corresponds to nonnegatively constrained, regularized, weighted least squares. The solid line connected by circles (bottom graph) corresponds to nonnegatively constrained, regularized Poisson likelihood.

with FORTRAN77 source code made publicly available by the authors of [4, 25], KNITRO was implemented in MATLAB via a mex interface with FORTRAN77 binary code made available to us by the authors of [3].

In the following comparison of optimization methods, we again selected the regularization parameter $\alpha = 10^{-10}$. For GPRNCG we found that it was cost effective to take only one gradient projection iteration per outer iteration. The truncation parameter in the sparse preconditioner (3.18)-(3.19) was taken to be $r = 0.1$. Then only 0.13% of the entries of $\hat{S}$ were nonzero.

At each iteration $k$ the inversion of the preconditioner $M_k$ (see (3.20)) was carried out using a sparse Choleski reordering, followed by a sparse Choleski decomposition, followed by sparse forward eliminations and back substitutions. While this was expensive in the early iterations, the cost of preconditioning decreased significantly as the active set was identified, and hence as the number of off-diagonal nonzero entries of $M_k$ decreased.

To obtain a rough comparison of convergence behavior as a function of computational cost for the various methods, we plotted the norm of the projected gradient (3.2) versus the cumulative number of two-dimensional fast Fourier transforms (FFTs). Function evaluations, gradient evaluations, and Hessian matrix-vector multiplications each required matrix-vector multiplications involving the BTTB matrix $S$ and/or the BTTB matrix $S^T$. See (2.6), (2.7), and (2.8). Each BTTB matrix-vector multiplication was carried out by block circulant embedding [24, Chapter 5] and required a pair of FFTs on a $2N \times 2N$ grid. Hence FFTs constituted a significant cost
in our implementations.

The results of this comparison are given in Figure 4.5. For the L-BFGS-B results presented in this figure, we saved 10 vectors in the L-BFGS recursion; see [16, p. 224]. GPRNCG with sparse preconditioning outperformed the other methods; it achieved the stopping criterion \( \| \text{grad}_P J(f_k) \|_{\infty} < 10^{-7} \| \text{grad}_P J(f_0) \|_{\infty} \) in about half the FFTs required for L-BFGS-B. KNITRO not only didn’t converge, but it failed to significantly decrease the projected gradient norm. GPRNCG without preconditioning also failed to converge within \( 5 \times 10^4 \) FFTs, but it did significantly reduce the projected gradient norm.

Some CPU time comparisons are presented in Table 4.1. Computations were carried out on a SUN Microsystems SunBlade 2000 workstation. FFT time refers to the amount of time spent computing BTB matrix-vector products using FFTs. L-BFGS-B(\( m \)) refers to L-BFGS-B with \( m \) saved vectors. The termination criterion \( \| \text{grad}_P J(f_k) \|_{\infty} < 10^{-7} \| \text{grad}_P J(f_0) \|_{\infty} \) was again used.

Increasing the number \( m \) of saved vectors for L-BFGS-B decreased the number of FFTs required to reach the stopping tolerance. However, it also increased other over-
Nonnegatively Constrained Image Reconstruction

Fig. 4.5. Performance of methods to minimize the regularized Poisson likelihood functional. The horizontal axis represents cumulative FFTs. The vertical axis shows the norm of the projected gradient on a logarithmic scale. The line connected by x’s (top graph) denotes KNITRO. The line connected by asterisks (second graph from top) denotes GPRNCG without preconditioning. The solid line (third from the top) denotes L-BFGS-B iterates. The line connected by circles (bottom graph) denotes GPRNCG with sparse preconditioning.

Table 4.1
CPU time comparisons.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>FFT time</th>
<th>Total CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>L-BFGS-B(10)</td>
<td>514 sec</td>
<td>1147 sec</td>
</tr>
<tr>
<td>L-BFGS-B(5)</td>
<td>580 sec</td>
<td>1067 sec</td>
</tr>
<tr>
<td>GPRNCG with preconditioning</td>
<td>258 sec</td>
<td>538 sec</td>
</tr>
</tbody>
</table>

head costs. For our problem, the overall cost for L-BFGS-B was relatively insensitive to the number of saved vectors.

Half of the total CPU time for preconditioned GPRNCG was spent computing FFTs. Further timing studies revealed that most of the remaining CPU time was spent in preconditioning. We found that we could reduce this cost by preconditioning only after a certain fraction of the pixel indices were in the active set (i.e., the pixel values were set to zero). By taking this fraction to be 2/3, we were able to reduce the total CPU time for preconditioned GPRNCG by about 20%—to 423 seconds—for this test problem.

4.1. Richardson–Lucy iteration. We also applied the Richardson–Lucy iteration to our test problem. Corresponding to the (negative) Poisson log likelihood
fit-to-data functional (1.5), this iteration takes the form
\begin{equation}
 f_{k}^{\text{RL}} = f_{k-1}^{\text{RL}} ./ (S^T 1) .* S^T ((d + \sigma^2 1) ./ [Sf_{k-1}^{\text{RL}} + (\beta + \sigma^2) 1]), \quad k = 1, 2, \ldots,
\end{equation}
where ./ and .* denote componentwise multiplication and division, respectively; 1 is the vector whose components are all equal to one; and \( \sigma^2 \) and \( \beta \) are, respectively, the Gaussian noise variance and the Poisson background noise parameter. The initial guess \( f_0 \) was taken to be a vector with positive constant entries.

Unlike the previously discussed optimization-based approaches, the functional (1.5) to which Richardson–Lucy iteration (4.3) is applied contains no explicit regularization term. (Note that the regularization term of \( f^T L f / 2 \) in (2.4) is missing here.) Richardson–Lucy appears to fall into the class of iterative regularization methods. This means that the iteration count \( k \) plays a role analogous to that of the regularization parameter \( \alpha \) in (2.4).

A cost comparison between optimization-based regularization methods and iterative regularization methods is difficult to carry out for several reasons. First, the two approaches typically yield somewhat different reconstructions. Second, quantifying the cost of an iterative regularization method requires a means of determining how many iterations are needed to obtain a reconstruction that is “good enough.” For an optimization-based method, an optimality criterion such as the norm of the (projected) gradient quantifies how well one has solved the optimization problem. (Of course, stopping tolerances add a subjective element. Moreover, one must somehow choose a regularization parameter like the \( \alpha \) in (2.4).) Complicating these issues is the fact that there may be no metric to clearly delineate a “good” reconstruction from a “bad” one.

Figure 4.6 provides information with which one might conduct a cost comparison between the Richardson–Lucy and the preconditioned GPRNCG algorithms. Cost is measured in terms of the cumulative number of FFTs. Reconstruction quality is measured in terms of the relative reconstruction error. For Richardson–Lucy, this is
\begin{equation}
 E_{k}^{\text{RL}} = \|f_{k}^{\text{RL}} - f_{\text{true}}\| / \|f_{\text{true}}\|,
\end{equation}
where \( f_{k}^{\text{RL}} \) is given in (4.3). For GPRNCG we take
\begin{equation}
 E_{k}^{\alpha} = \|f_{k}^{\alpha} - f_{\text{true}}\| / \|f_{\text{true}}\|,
\end{equation}
where \( f_{k}^{\alpha} \) is the \( k \)th iterate from the GPRNCG algorithm applied to minimize (2.4), (2.6) with regularization parameter \( \alpha \). As above, we select the regularization parameter \( \alpha = 10^{-10} \).

Until about 200 FFTs, Richardson–Lucy produces a better reconstruction than GPRNCG. After 300 iterations, the GPRNCG reconstruction is better. The difference between the GPRNCG reconstruction at iteration \( k \) and the true object can be decomposed as
\begin{equation}
 (f_{k}^{\alpha} - f^{\alpha}) + (f^{\alpha} - f_{\text{true}}).
\end{equation}
The first term quantifies how well GPRNCG solves the constrained minimization problem. This term vanishes as GPRNCG converges (after about 9,000 FFTs). Thereafter, the GPRNCG reconstruction error depends only on the effectiveness of the regularization and does not change with iteration count.
The Richardson–Lucy reconstruction error decreases monotonically with iteration count $k$, which plays the role of the regularization parameter $\alpha$ in the GPRNCG reconstruction. This decrease is initially fairly rapid. However, after about 10,000 FFTs, the rate of decrease in the Richardson–Lucy reconstruction error slows dramatically. After a million FFTs, its reconstruction error of 35.1% is still slightly larger than that achieved by GPRNCG (34.9%) after about 9,000 FFTs. On the other hand, it should be noted that the Richardson–Lucy reconstructions after several thousand FFTs are hard to distinguish visually from the reconstruction obtained with GPRNCG, which is displayed in the lower right subplot of Figure 4.4.

4.2. Conclusions. A variety of constrained optimization methods were applied to a large-scale, ill-conditioned, nonnegatively constrained convex programming problem arising in image deblurring. For this test problem, GPRNCG with the sparse preconditioner introduced in this paper is very robust and is the most efficient algorithm, both in terms of FFTs and in total CPU time. Preconditioning is relatively expensive, especially in the early iterations, before the active set has been accurately identified. An ad hoc strategy of employing preconditioning only after a certain portion of the indices are active can decrease the computational cost by about 20% for this test problem. When GPRNCG is used with no preconditioning, robustness is lost and computational cost increases dramatically.

L-BFGS-B is also quite effective for this problem. It is robust and has a computational cost about twice that of preconditioned GPRNCG. Our experience with
nonconvex problems indicates that performance suffers when L-BFGS updates are skipped. For this test problem the cost functional is strictly convex, so no L-BFGS updates were skipped. As the number of saved vectors increases, the number of FFTs decreases, but other overhead costs increase, so the overall cost seems to be relatively insensitive to the number of saved vectors.

The performance of KNITRO, a trust region interior point method, was quite disappointing. Not only did this method fail to converge, but the initial decrease in the projected gradient norm was extremely slow compared to the other methods. The x’s in Figure 4.5 denote outer KNITRO iterates. The gaps between these x’s indicate that several hundred CG iterations were often taken per outer iteration. It is possible that preconditioning could significantly decrease CG costs and improve robustness for KNITRO.

We also tested a preconditioner based on the limited memory BFGS recursion [17] in the implementation of GPRNCG. Although we had success with this preconditioner on less difficult test problems, for this particular data set and PSF, this preconditioner was not found to be effective.

Finally, we attempted a comparison between Richardson–Lucy, an iterative regularization method for image deblurring, and preconditioned GPRNCG. By the measures of performance used in this comparison, GPRNCG was more efficient than Richardson–Lucy iteration. However, one could argue that this was an apples-to-oranges comparison and that the performance measures were not appropriate.

Acknowledgments. We wish to thank Dr. Brent Ellerbroek, Adaptive Optics Program Manager at the Gemini Observatory in Hilo, HI, for supplying us with the PSFs used in our numerical comparisons. We also wish to thank Dr. Julian Christou of the Center for Adaptive Optics at the University of Santa Cruz, CA, for supplying us with information on CCD noise modeling and parameters. Richard Waltz and Jorge Nocedal at Northwestern University deserve thanks for their help in implementing the KNITRO algorithm. Finally, we thank our Program Officer at the AFOSR, Dr. William Hilbun, for his support and encouragement, and we thank the anonymous referees for their helpful comments.

REFERENCES


