A NONNEGATIVELY CONSTRAINED CONVEX PROGRAMMING
METHOD FOR IMAGE RECONSTRUCTION *

JOHNATHAN M. BARDISLEY† AND CURTIS R. VOGEL‡

Abstract. We consider a large-scale convex minimization problem with nonnegativity constraints which 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 we take 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.

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 which arises in astronomical imaging. Data for this problem takes the form

\[ d = Sf_{\text{true}} + \eta, \]

where \( S \) is a large, non-sparse, block Toeplitz, highly 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 \([18]\)) and incorporate prior knowledge of the statistics of the noise and the nonnegativity of \( f_{\text{true}} \). This yields an optimization problem

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

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

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

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.

---

*This work was supported by DEPSCoR/AFOSR under grant F49620-02-1-0297
†Center for Research in Scientific Computation, Box 205, North Carolina State University, Raleigh, NC 27695-8205 (bardisley@math.montana.edu)
‡Department of Mathematical Sciences, Montana State University, Bozeman, MT 59717-2400 (vogel@math.montana.edu)
In many imaging applications, the data can be more accurately modeled as a sum of independent Poisson processes. We will consider such a case in which the appropriate fit-to-data functional has Poisson log likelihood form,

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

(1.5)

where $\beta$ and $\sigma$ are positive parameters. The corresponding regularized Poisson likelihood functional (1.5), (1.3) is not quadratic, but it is convex.

Our goal is the efficient minimization of the regularized Poisson likelihood functional under the constraint $f \geq 0$. A variety of methods can be applied towards this goal, including the gradient projection method [1], active set methods like the projected Newton method [2, 9], a bound-constrained variant of the limited memory BFGS method [4], bound-constrained trust region methods [10], and interior point methods [3]. In this paper we introduce an efficient hybrid gradient projection-reduced Newton method. By “reduced Newton” we mean we take 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 [12] for large-scale bound-constrained quadratic minimization. We also introduce in this paper a highly effective sparse matrix preconditioner to speed up CG convergence.

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. 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

$$
d_{ij} = \int \int s(x_i - x, y_j - y)f_{\text{true}}(x, y) dx dy + \eta_{ij},
$$

(2.1)

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

$$
d_{ij} = \sum_{i'=i}^{i'} \sum_{j'=j}^{j'} s_{i',j'} [f_{\text{true}}(x, y) + \eta_{i,j}],
$$

(2.2)

We refer to the array with components $s_{i,j}$ as the discrete PSF. We assume that the quadrature error is negligible.
With lexicographical ordering of unknowns, equation (2.2) can be rewritten as equation (1.1), where \( d, f_{\text{true}}, \) and \( \eta \) 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) [18, Chapter 5].

In typical imaging applications \( S \) is highly ill-conditioned, and in many cases it is also a singular matrix. Consequently, the pseudo-inverse estimate \( S^d d \) 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 [18, Chapter 4]). This requires the minimization of a functional of the form

\[
J(f) = \ell(Sf;d) + \frac{\alpha}{2} f^T L f. 
\]  
(2.3)

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 (2.3), 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 semi-definite and is usually sparse.

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

The \( \ell \) in (2.3) 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 CCD camera. The following statistical model (see [16, 17]) applies to image data from a CCD detector array:

\[
d_i = n_{\text{obj}}(i) + n_0(i) + g(i), \quad i = 1, \ldots, N. 
\]  
(2.4)

Here \( d_i \) is the \( i^{\text{th}} \) component of the vector \( d \) and is the data acquired by a readout 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 \( [S f_{\text{true}}]_i \); \( n_0(i) \) 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

\[
\eta_i = n_{\text{obj}} - [S f_{\text{true}}]_i + n_0(i) + g(i).
\]

The log-likelihood for the mixed Poisson-Gaussian model (2.4) has an infinite series representation (see [16, equation (10)]) which is computationally intractable. To approximate it we observe as in [16, 17] 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 \( \sigma^2 \) [5]. Then \( d_i + \sigma^2 \) is well-approximated by a Poisson random variable with Poisson parameter
\[
\lambda_i = [Sf]_i + \beta + \sigma^2. \quad \text{From this we obtain the corresponding negative Poisson log likelihood functional}
\]
\[
\ell(Sf; d) = \sum_{i=1}^{N}([Sf]_i + \beta + \sigma^2) - \sum_{i=1}^{N}(d_i + \sigma^2) \log([Sf]_i + \beta + \sigma^2). \quad (2.5)
\]

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 (2.3) and (2.5) is infinitely differentiable for \(f > 0\). Its gradient is
\[
\text{grad } J(f) = S^T \left(\frac{(Sf + \beta - d)}{(Sf + \beta + \sigma^2)}\right) + \alpha Lf, \quad (2.6)
\]
where \(./\) denotes component-wise division. Its Hessian is
\[
\text{Hess } J(f) = S^T W(f) S + \alpha L, \quad (2.7)
\]
where \(W(f) = \text{diag}(w)\) and the components of \(w\) are given by
\[
w_i = \frac{d_i + \sigma^2}{([Sf]_i + \beta + \sigma^2)^2}. \quad (2.8)
\]

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

**Theorem 2.1.** Let \(J\) denote the regularized Poisson likelihood functional defined by (2.3) and (2.5). Assume that \(d_i + \sigma^2 \geq 0\) and that \(L\) is positive definite. 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 (2.5). 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 since \(S\) and \(f\) both have nonnegative entries and each \(d_i + \sigma^2 \geq 0\), from (2.8) the diagonal entries of \(W(f)\) are bounded below by some \(w_{\min} \geq 0\). Then by (2.7), for any \(v \in \mathbb{R}^N\),
\[
v^T \text{Hess } J(f) v \geq v^T \left(\omega_{\min} S^T S + \frac{\alpha}{2} L\right) v
\]
\[
\geq \frac{\alpha \omega_{\min}(L)}{2} \|v\|^2. \quad (2.9)
\]
Hence \(\text{Hess } J(f)\) is strictly positive for \(f \geq 0\), 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 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 is comprised of two stages. The first stage consists of projected gradient iterations to identify the active set, while the second stage uses conjugate gradient 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

$$\Omega = \{f \in \mathbb{R}^N \mid f \geq 0\}. \quad (3.1)$$

The projection of a vector $f \in \mathbb{R}^N$ onto the feasible set is given by

$$P_\Omega(f) \overset{\text{def}}{=} \arg \min_{v \in \Omega} ||v - f|| = \max\{f, 0\},$$

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

$$[\text{grad}_P J(f)]_i = \begin{cases} \frac{\partial J(f)}{\partial f_i}, & i \in \mathcal{I}(f) \text{ or } (i \in \mathcal{A}(f) \text{ and } \frac{\partial J(f)}{\partial f_i} < 0) \\ 0, & \text{otherwise.} \end{cases} \quad (3.2)$$

The reduced gradient of $J$ at $f \in \Omega$ is given by

$$[\text{grad}_R J(f)]_i = \begin{cases} \frac{\partial J(f)}{\partial f_i}, & i \in \mathcal{I}(f) \\ 0, & i \in \mathcal{A}(f) \end{cases}$$

and the reduced Hessian is given by

$$[\text{Hess}_R J(f)]_{ij} = \begin{cases} \frac{\partial^2 J(f)}{\partial f_i \partial f_j}, & \text{if } i \in \mathcal{I}(f) \text{ or } j \in \mathcal{I}(f) \\ \delta_{ij}, & \text{otherwise.} \end{cases}$$

Let $D_\mathcal{I}$ denote the diagonal matrix with components

$$[D_\mathcal{I}(f)]_{ii} = \begin{cases} 1, & i \in \mathcal{I}(f) \\ 0, & i \in \mathcal{A}(f). \end{cases} \quad (3.3)$$

Then

$$\text{grad}_R J(f) = D_\mathcal{I}(f) \text{ grad } J(f), \quad (3.4)$$

$$\text{Hess}_R J(f) = D_\mathcal{I}(f) \text{ Hess } J(f) \ D_\mathcal{I}(f) + D_\mathcal{A}(f), \quad (3.5)$$

where $D_\mathcal{A}(f) = I - D_\mathcal{I}(f)$.

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 $\{f_k\} \subseteq \Omega$ via the following iteration:

$$p_k = - \nabla J(f_k)$$

$$\lambda_k = \arg\min_{\lambda \geq 0} J(P_{\Omega}(f_k + \lambda p_k))$$

$$f_{k+1} = P_{\Omega}(f_k + \lambda_k p_k)$$

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,

$$\lambda_k^0 = \frac{\|p_k\|^2}{\langle \text{Hess} \ J(f_k) p_k, p_k \rangle}.$$  \hfill (3.9)

This is the minimizer of the quadratic $\tilde{q}(\lambda) = q(f_k + \lambda p_k)$, where

$$q(f + s) = J(f) + \langle \nabla J(f), s \rangle + \frac{1}{2} \langle \text{Hess} \ J(f) s, s \rangle.$$  \hfill (3.10)

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 [12] can also be used. We stop at the first $m$ for which the sufficient decrease condition

$$J(f_k(\lambda_k^m)) \leq J(f_k) - \frac{\mu}{\lambda_k^m} \|f_k - f_k(\lambda_k^m)\|^2$$  \hfill (3.11)

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

$$f_k(\lambda) = P_{\Omega}(f_k + \lambda p_k).$$  \hfill (3.12)

Theorem 3.1. Let $\{f_k\}$ be a sequence generated by the gradient projection iteration as discussed above. Then, if $\overline{f}$ is a limit point of $\{f_k\}$, grad$_{p} \ J(\overline{f}) = 0$. 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 [13, 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 (2.3) and (2.5); assume $d_i + \sigma_i \geq 0$ for each $i$, and assume that $L$ is positive definite. Then the gradient projection iterates $f_k$ converge 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 $\overline{f} \geq J(f^*)$. Since $J$ is coercive, $\{f_k\}$ is bounded, and hence there exists a subsequence $\{f_{k_j}\}$ converging to some $\overline{f}$. By Theorem 3.1, grad$_{p} \ (\overline{f}) = 0$, and since $J$ is strictly convex, $\overline{f} = f^*$. Thus $\{J(f_k)\}$ converges to $J(f^*)$. Then using Taylor’s Theorem, equation (2.9), and the fact that $f^*$ is the global constrained minimizer,

$$J(f_k) - J(f^*) \geq \frac{\alpha_{\text{min}}(L)}{2} \|f_k - f^*\|^2.$$  \hfill (3.13)

Consequently, $\{f_k\}$ converges to $f^*$. \hfill $\Box$

The asymptotic convergence rate for the gradient projection iteration is linear and can be quite slow when 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, 9], one replaces the \( p_k \) in (3.6) with the solution to

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

With a properly implemented line search, this yields a quadratically convergent scheme [2, 9]. However, it lacks robustness if the reduced Hessian is ill-conditioned and system (3.13) is solved inexact. For this reason, we advocate interspersing gradient projection iterations with reduced Newton steps, in which we (approximately) solve

\[
\text{Hess}_R J(f_k) \ p = -\text{grad}_R J(f_k). \tag{3.14}
\]

(Note the replacement of the gradient on the right-hand-side of (3.13) by the reduced gradient.) Equivalently, we approximately minimize the quadratic

\[
q_k(p) = J(f_k) + \langle \text{grad}_R J(f_k), p \rangle + \frac{1}{2} \langle \text{Hess}_R J(f_k), p, p \rangle. \tag{3.15}
\]

This can be viewed as 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

\[
\mathcal{F}_k = \{ f \in \Omega \mid f_i = 0 \ \text{whenever} \ i \in \mathcal{A}(f_k) \}.
\]

When CG iteration is used to approximately solve (3.14) we obtain a scheme which resembles the gradient projection-CG (GPCG) algorithm of Moré and Toraldo [12] 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 a much less stringent acceptance criteria than (3.11), namely

\[
J(f_k(\lambda^m_k)) < J(f_k). \tag{3.16}
\]

3.3.1. CG Iteration and Stopping Criteria. Since \( \text{Hess}_R J(f_k) \) 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 unknowns to 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 [9, 13]) 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 Moré and Toraldo [12] to be much more effective:

\[
q_k(p^{j-1}) - q_k(p^j) \leq \gamma_{CG} \max\{q_k(p^{i-1}) - q_k(p^i) \mid i = 1, \ldots, j\}, \tag{3.17}
\]

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 BTBT matrix $\hat{S}$ corresponding to such an approximate PSF is then a sparse approximation to the blurring matrix $S$. We select $\hat{S}$ to have entries

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

(3.18)

where

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

Motivated by the forms of the Hessian (2.7) and the reduced Hessian (3.5), we take the preconditioning matrix at outer iteration $k$ to be

$$
M_k = D_k^b \hat{S} \hat{W}(f_k) \hat{S} D_k^b + \alpha D_k^b \hat{L} D_k^b + D_k^A,
$$

(3.20)

where $D_k^b = D_2(f_k)$ and $D_k^A = I - D_2(f_k)$. Note that $D_k^b$, $D_k^A$, and $W(f_k)$ are each diagonal matrices. In addition, the regularization matrix $\hat{L}$ is sparse. Hence, if $\hat{S}$ is sparse, then so is each $M_k$. Moreover, if the size of the active set increases with $k$, the number of nonzero (diagonal) entries in $D_k^b$ 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 [12], we stop when

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

(3.21)

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 $\lambda_k^m$ which satisfies (3.16), and return $f_{k+1} = f_k(\lambda_k^m)$, 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 the algorithm of Section 3 was applied to data that was 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 [15] used in astronomical imaging. This PSF is displayed in Figure 4.1. Note that the PSF is quite localized in space and that the blurring matrix \(S\) has many zero eigenvalues and is ill-conditioned in the sense that the ratio of the largest eigenvalue to the smallest nonzero eigenvalue is quite large.

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

Noisy data was generated by taking a pseudo-random realization from the statistical model (2.4). We selected parameter values \(\beta = 10\) for sky background and \(\sigma = 5\) for read noise. These values are representative of CCD cameras used in astronomy. The system size \(N = 128^2 = 16,384\).

Reconstructions of \(f_{\text{true}}\) were generated by nonnegatively constrained minimum
tion of the regularized Poisson likelihood cost functional $J(f)$ defined by (2.3) and (2.5). We also generated reconstructions by minimizing (again with constraint $f \geq 0$) the regularized weighted least squares cost functional

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

using Moré and Toraldo’s GPCG algorithm [12]. The weight matrix was taken to be diagonal with diagonal entries taken to be the inverse of the variance of the noise in model (2.4),

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

In both cases the regularization matrix $L$ was taken to be the identity. We selected the regularization parameter $\alpha$ to minimize $||f_\alpha - f_{\text{true}}||$, where $f_\alpha$ denotes the constrained minimizer of the cost functional with $\alpha$ fixed. For the regularized Poisson likelihood and the regularized weighted least squares cost functionals, $\alpha = 2 \times 10^{-8}$. Figure 4.3 presents a comparison of the reconstructions obtained with these two different cost functionals. From this figure, it can be seen that the Poisson likelihood reconstruction is superior to least squares in that it has less background clutter at low light intensities.

We applied a variety of 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, which does not allow for preconditioning; 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 with FORTRAN77 source code made publicly available by the authors of [4, 19]. KNITRO was implemented in MATLAB via a mex interface with FORTRAN77 binary code made available to us by the authors of [3].

For GPRNCG we took $\gamma_{CG} = 0.25$ for CG stopping criterion (3.17), and the line search parameter sufficient decrease parameter in (3.11) was taken to be $\mu = 0.5$. We found that it was effective to take only one gradient projection iteration per outer iteration. When preconditioning was not used, we found it effective to set an upper limit of 30 CG iterations in Step 2 (1) of the GPRNCG algorithm. With sparse preconditioning, no such limit was needed. The truncation parameter in the sparse preconditioner (3.18)-(3.19) was taken to be $r = 0.1$. Then only 0.13 percent of the entries of $\hat{S}$ were nonzero.

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 equations (2.5), (2.6), and (2.7). Each BTTB matrix-vector multiplication was carried out by block circulant embedding [18, 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 4.4. For the L-BFGS-B results presented in this figure, we saved 10 vectors in the L-BFGS recursion, see [13, p. 224].

We also recorded total CPU time for all algorithms except KNITRO, which failed to converge for our application. CPU time comparisons are presented in Table 4.1. FFT time refers to the amount of time spent computing BTTB matrix-vector products using FFTs. L-BFGS-B(m) refers to L-BFGS-B with m saved vectors. In all
Fig. 4.3. Object Reconstructions. At the top is a mesh plot of the lower left $32 \times 32$ pixels of the true image. On the lower left is a reconstruction obtained using regularized, weighted least squares. On the lower right is a regularized Poisson likelihood reconstruction.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>FFT time</th>
<th>Total CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPRNCG without preconditioning</td>
<td>620</td>
<td>720</td>
</tr>
<tr>
<td>L-BFGS-B(10)</td>
<td>283</td>
<td>647</td>
</tr>
<tr>
<td>L-BFGS-B(5)</td>
<td>305</td>
<td>554</td>
</tr>
<tr>
<td>GPRNCG with preconditioning</td>
<td>105</td>
<td>304</td>
</tr>
</tbody>
</table>

implementations, the termination criterion $|\text{grad}_f J(f_0)|_\infty < 10^{-7} |\text{grad}_f J(f_0)|_\infty$ was used. Computations were carried out on a SUN Microsystems SunBlade 2000 workstation.

4.1. Conclusions. Based on our astronomical imaging test problem, we conclude that when measuring computational cost in terms of FFTs, GPRNCG with sparse preconditioning is the most efficient algorithm. CPU time comparisons yield the same conclusion. When GPRNCG is used without preconditioning, both the number of FFTs and the total CPU time increase dramatically.
FIG. 4.4. 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 solid line with circles at the top represents KNITRO. The circles denote KNITRO outer iterates. The dash-dot line, second from the top, represents GPRNC with sparse preconditioning. The solid line, third from the top, denotes L-BFGS-B iterates. The dashed line at the bottom denotes GPRNC with sparse preconditioning.

L-BFGS-B is also quite effective. Our experience with non-convex 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.

A direct comparison between L-BFGS-B and GPRNC is problematic due to the fact that GPRNC is coded in MATLAB, while L-BFGS-B is coded in FORTRAN. In this case, L-BFGS-B is about twice as expensive as GPRNC with sparse preconditioning in terms of CPU time and about three times as expensive in terms of FFTs. It is possible that a FORTRAN implementation could reduce preconditioned GPRNC CPU time relative to L-BFGS-B. No such relative time reduction is likely for GPRNC without preconditioning, since most of the time cost was incurred in FFT computations.

The performance of KNITRO 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 circles in Figure 4.4 denote outer KNITRO iterates. The gap between these circles indicate that several hundred CG iterations were often taken per outer iteration. It is possible that preconditioning could dramatically decrease CG costs for KNITRO.
We also tested a preconditioner based on the limited memory BFGS recursion [14] 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.

5. Acknowledgments. We wish to thank Dr. Brent Ellerbroek, Adaptive Optics Program Manager at the Gemini Observatory in Hilo, HI, for supplying us with the PSF’s 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 would like to thank our Program Officer at the AFOSR, Dr. William Hilburn, for his support and encouragement.

REFERENCES