

COVARIANCE-PRECONDITIONED ITERATIVE METHODS FOR NONNEGATIVELY CONSTRAINED ASTRONOMICAL IMAGING

JOHNATHAN M. BARDSLEY* AND JAMES G. NAGY†

Abstract. We consider the problem of solving ill-conditioned linear systems $A\mathbf{x} = \mathbf{b}$ subject to the nonnegativity constraint $\mathbf{x} \geq \mathbf{0}$, and in which the vector \mathbf{b} is a realization of a random vector $\hat{\mathbf{b}}$, i.e. \mathbf{b} is noisy. We explore what the statistical literature tells us about solving noisy linear systems; we discuss the effect that a substantial black background in the astronomical object being viewed has on the underlying mathematical and statistical models; and, finally, we present several covariance-based preconditioned iterative methods that incorporate this information. Each of the methods presented can be viewed as an implementation of a preconditioned modified residual-norm steepest descent algorithm with a specific preconditioner, and we show that, in fact, the well-known and often used Richardson-Lucy algorithm is one such method. Ill-conditioning can inhibit the ability to take advantage of *a priori* statistical knowledge, in which case a more traditional preconditioning approach may be appropriate. We briefly discuss this traditional approach as well. Examples from astronomical imaging are used to illustrate concepts and to test and compare algorithms.

Key words. image restoration, linear models, preconditioning, statistical methods, weighted least squares

AMS Subject Classifications: 65F20, 65F30

1. Introduction. We seek solutions of the linear system with nonnegativity constraints

$$A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \quad (1.1)$$

where by $\mathbf{x} \geq \mathbf{0}$ we mean $x_i \geq 0$ for all i . Such models arise naturally in image reconstruction, where \mathbf{x} denotes the measured intensities at various pixels of an object being viewed by an imaging system. Such problems are typically solved via an iterative algorithm. The most common and straightforward approach is to simply ignore the nonnegativity constraint, in which case a large class of powerful methods can be utilized. Effective methods for solving (1.1) when A is an ill-conditioned matrix exist; see, e.g., [1, 3, 21]. In this paper we present preconditioned iterative methods that incorporate the nonnegativity constraint. Furthermore, we note that for many image deblurring problems, and in particular in astronomical imaging, the objects being viewed are often times composed, primarily, of a black background. This affects the mathematical model (1.1) in a nontrivial way. The algorithms that we present in this paper incorporate these changes as well.

An additional complication that is often ignored is that the data vector \mathbf{b} in (1.1) is obtained via measurements that have associated random error, and is therefore a *single* realization of a random vector that we will call $\hat{\mathbf{b}}$. In this paper, we will focus on the case in which the statistical model for $\hat{\mathbf{b}}$ is given by

$$\hat{\mathbf{b}} = A\mathbf{x}_{\text{true}} + \boldsymbol{\eta}, \quad (1.2)$$

where \mathbf{x}_{true} is the discretized object, or true image, and is what we want to estimate; and the random vector $\boldsymbol{\eta}$ characterizes the measurement errors, or noise, in the data collection process, and is assumed to have a known distribution. We assume that

*Department of Mathematical Sciences, University of Montana. E-mail: bardsleyj@mso.umt.edu

†Department of Mathematics and Computer Science, Emory University. E-mail: nagy@mathcs.emory.edu

the exact data $A\mathbf{x}_{\text{true}}$ and the noise $\boldsymbol{\eta}$ are statistically uncorrelated and independent. Each component of \mathbf{b} is then a random variable with distribution

$$\hat{b}_i = [A\mathbf{x}_{\text{true}}]_i + \eta_i, \quad i = 1, \dots, n, \quad (1.3)$$

where η_i is a random variable with a known distribution, and n is the number of pixels in the collected image \mathbf{b} . Our task is to estimate \mathbf{x}_{true} from \mathbf{b} given statistical model (1.2). It is tempting to simply ignore (1.2) and solve (1.1) directly, with or without the nonnegativity constraint. In fact, this is what is usually done. In many instances, ignoring (1.2) has no noticeable effect on the performance of the algorithm used, but as we will see, there are instances in which prior knowledge of noise statistics can be used to accelerate the convergence of the iterative method used.

As a test-bed of examples, we consider a class of ill-posed inverse problems that arise when reconstructing an image of an astronomical object taken by a ground based telescope. In this case, light from an object, \mathbf{x}_{true} , in outer-space travels through a medium with refractive index fluctuations (the earth's atmosphere), which has a blurring effect on the data. In addition, as the light passes through the telescope, diffractive blurring occurs. These blurring effects are assumed to be characterized by a (known) ill-conditioned blurring matrix A . That is, if \mathbf{x}_{true} is the discretized object of interest, $A\mathbf{x}_{\text{true}}$ is what is seen after the light from \mathbf{x}_{true} has travelled through the earth's atmosphere and through the telescope. In order to collect the blurred image data, a charge-coupled-device (CCD) camera is used. A CCD camera is, essentially, an array of sensors that creates a pixelated image by counting the number of photons that hit each sensor. We assume that the random noise that enters the data does so during the collection of the image by the CCD camera. Other information can be included in the formulation of the image reconstruction problem; namely, in our application, the object being viewed, \mathbf{x}_{true} , has nonnegative intensity. This information can be included as a nonnegativity constraint in the reconstruction algorithm.

Finally, we note that in the sequel we will use the notation $N(\mu, \sigma^2)$ to denote a Gaussian random variable with mean μ and variance σ^2 and $\text{Poiss}(\beta)$ to denote a Poisson random variable with mean and variance equal to β .

This paper is outlined as follows. In Section 2, we present the statistical material that will be necessary in order to motivate the covariance-preconditioned iterative methods. The methods for solving (1.1), (1.2) are then presented in Section 3. In Section 4, we present numerical results to illustrate the effectiveness of the preconditioners, and some concluding remarks are given in Section 5.

2. Statistical Considerations. We consider the linear statistical model (1.2). We assume throughout this paper that $\boldsymbol{\eta}$ is a random n -vector with $E(\boldsymbol{\eta}) = \mathbf{0}$, where E denotes the component-wise expected value, or mean, of $\boldsymbol{\eta}$, i.e. $E(\boldsymbol{\eta}) = (E(\eta_1), \dots, E(\eta_n))$. We let $C_{\boldsymbol{\eta}}$ denote the *covariance matrix* of $\boldsymbol{\eta}$ defined by

$$[C_{\boldsymbol{\eta}}]_{ij} = E[\eta_i \eta_j],$$

and assume that it is a known and nonsingular matrix. Given (1.2), we seek the “best” estimate of the object \mathbf{x}_{true} from our data vector \mathbf{b} , but first we must define what we mean by “best” in a statistical sense.

DEFINITION 2.1. *The Best Linear Unbiased Estimator (BLUE) for \mathbf{x}_{true} from \mathbf{b} in (1.2) is the random vector $\hat{\mathbf{x}}_{\text{BLUE}}$, which minimizes*

$$J(\hat{\mathbf{x}}) = E(\|\hat{\mathbf{x}} - \mathbf{x}_{\text{true}}\|_2^2) \quad (2.1)$$

subject to the constraints

$$\hat{\mathbf{x}} = B\mathbf{b}, \quad B \in \mathbb{R}^{n \times n}, \quad (2.2)$$

$$E(\hat{\mathbf{x}}) = \mathbf{x}_{\text{true}}. \quad (2.3)$$

$\hat{\mathbf{x}}_{\text{BLUE}}$ is called linear because of (2.2) and unbiased because of (2.3).

Forgetting, for the moment, ill-conditioning, we assume that $\hat{\mathbf{x}}_{\text{BLUE}}$ is the reconstruction that we seek. The following theorem tells us how to find it.

THEOREM 2.2. (*Gauss-Markov*) *If A has full rank and \mathbf{b} is a realization of the random vector $\hat{\mathbf{b}}$ in (1.2), then*

$$\hat{\mathbf{x}}_{\text{BLUE}} = (A^T C_{\boldsymbol{\eta}}^{-1} A)^{-1} A^T C_{\boldsymbol{\eta}}^{-1} \mathbf{b}. \quad (2.4)$$

Example: Suppose $\boldsymbol{\eta}$ is an independent and identically distributed (i.i.d.) Gaussian random vector, i.e. $\eta_i = N(0, \sigma^2)$ for all i , where $N(0, \sigma^2)$ denotes a Gaussian random variable with mean 0 and variance σ^2 . Then the covariance matrix is given by $C_{\boldsymbol{\eta}} = \sigma^2 I$, and hence, provided A has full rank, $\hat{\mathbf{x}}_{\text{BLUE}}$ is the solution of

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2^2. \quad (2.5)$$

In many of the applications in which the linear model (1.2) arises, the noise is, in fact, i.i.d. Gaussian. Nonetheless, in astronomical imaging the noise does not have this form. We now focus our attention on the noise statistics of CCD camera image formation.

2.1. CCD Camera Noise Statistics. The following statistical model (see Refs. [18, 19]) applies to image data from a CCD detector array:

$$\hat{\mathbf{b}} = \text{Pois}(A\mathbf{x}_{\text{true}}) + \text{Pois}(\beta \cdot \mathbf{1}) + N(\mathbf{0}, \sigma^2 I), \quad (2.6)$$

where $\mathbf{1}$ is a vector of all ones. The relation described by (2.6) means that each element \hat{b}_i of the vector $\hat{\mathbf{b}}$ is a random variable with distribution

$$\hat{b}_i = n_{\text{obj}}(i) + n_0(i) + g(i), \quad i = 1, \dots, n. \quad (2.7)$$

Equation (2.7) can be described as follows:

- $n_{\text{obj}}(i)$ is the number of object dependent photoelectrons measured by the i th detector in the CCD array. It is a Poisson random variable with Poisson parameter $[A\mathbf{x}_{\text{true}}]_i$.
- $n_0(i)$ is the number of background photoelectrons, which arise from both natural and artificial sources, measured by the i th detector in the CCD array. It is a Poisson random variable with a fixed positive Poisson parameter β .
- $g(i)$ is the so-called readout noise, which is due to random errors caused by the CCD electronics and errors in the analog-to-digital conversion of measured voltages. It is a Gaussian random variable with mean 0 and fixed variance σ^2 .

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 $i \neq j$.

As in [18], we consider the case in which the readout noise variance σ^2 is large. Then, according to Feller [6, pp. 190 and 245], the following approximation is accurate:

$$N(\sigma^2, \sigma^2) \approx \text{Pois}(\sigma^2). \quad (2.8)$$

Using the independence properties of the random variables in (2.7) we obtain the following approximation of (2.6):

$$\hat{\mathbf{b}} + \sigma^2 \cdot \mathbf{1} = \text{Pois}(A\mathbf{x}_{\text{true}} + \beta \cdot \mathbf{1} + \sigma^2 \cdot \mathbf{1}). \quad (2.9)$$

This motivates computing a minimizer of the negative log-likelihood function for the statistical model (2.9), which is given by

$$\ell(A\mathbf{x}; \mathbf{b}) = \sum_{i=1}^N ([A\mathbf{x}]_i + \beta + \sigma^2) - \sum_{i=1}^N (b_i + \sigma^2) \log([A\mathbf{x}]_i + \beta + \sigma^2), \quad (2.10)$$

with respect to \mathbf{x} , and subject to the nonnegativity constraint $\mathbf{x} \geq \mathbf{0}$. A popular algorithm that computes such a minimizer is the Richardson-Lucy (RL) iteration [2], which has the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k \odot A^T \left(\frac{\mathbf{b} + \sigma^2 \cdot \mathbf{1}}{A\mathbf{x}_k + \beta \cdot \mathbf{1} + \sigma^2 \cdot \mathbf{1}} \right), \quad (2.11)$$

where \odot denotes component-wise multiplication, and the division operation is also done component-wise. The advantages of RL is that it is very simple to implement, it enforces a nonnegativity constraint, each iteration is relatively inexpensive, and it is statistically valid. A more sophisticated constrained optimization approach for minimizing (2.10) subject to $\mathbf{x} \geq \mathbf{0}$ can be employed (see, e.g., [1]), resulting in methods that are faster to converge, but are more costly per iteration. We do not consider such approaches here.

A different approach, and the one we focus on in this paper, is to invoke the approximation used in (2.8) yet again. First, we note that since (2.8) holds, and since $[A\mathbf{x}_{\text{true}}]_i + \beta + \sigma^2 \geq \sigma^2$ provided that the matrix A has nonnegative entries, which we will assume here, we obtain

$$\text{Pois}([A\mathbf{x}_{\text{true}}]_i + \beta + \sigma^2) \approx N([A\mathbf{x}_{\text{true}}]_i + \beta + \sigma^2, [A\mathbf{x}_{\text{true}}]_i + \beta + \sigma^2), \quad (2.12)$$

again using Feller [6, pp. 190 and 245]. This allows us to express (2.9) as

$$\hat{\mathbf{b}} + \sigma^2 \cdot \mathbf{1} = A\mathbf{x}_{\text{true}} + \beta \cdot \mathbf{1} + \sigma^2 \cdot \mathbf{1} + N(\mathbf{0}, W^{-1}), \quad (2.13)$$

where

$$W^{-1} = \text{diag}(A\mathbf{x}_{\text{true}} + \beta \cdot \mathbf{1} + \sigma^2 \cdot \mathbf{1}); \quad (2.14)$$

or, more succinctly,

$$\hat{\mathbf{b}} - \beta \cdot \mathbf{1} = A\mathbf{x}_{\text{true}} + N(\mathbf{0}, W^{-1}). \quad (2.15)$$

Assuming the linear statistical model (2.15), (2.14), the Gauss-Markov theorem tells us that given the data vector \mathbf{b} , $\hat{\mathbf{x}}_{\text{BLUE}}$ is obtained by solving

$$\min_{\mathbf{x}} \|A\mathbf{x} - (\hat{\mathbf{b}} - \beta \cdot \mathbf{1})\|_W^2. \quad (2.16)$$

3. The Algorithms. We now derive algorithms that incorporate the statistical results from the previous section. In addition, we see that in the case in which the object being viewed has a substantial black background, i.e. is in large part zero, these algorithms implicitly incorporate this information. This is important because when \mathbf{x}_{true} has a large number of zeros, the effect on the mathematical and statistical model can be significant.

3.1. The Effects of Zeros in the Components of \mathbf{x}_{true} . We begin with a simple observation and then analyze its effects on the models (1.1) and (1.2):

If \mathbf{x}_{true} has a large number of zero components, then much of the matrix A is extraneous.

To see this, we first define the *active set* of \mathbf{x}_{true} by

$$\mathcal{A}_{\text{true}} = \{i \mid [\mathbf{x}_{\text{true}}]_i = 0\},$$

and the diagonal matrix D_{true} by

$$[D_{\text{true}}]_{jj} = \begin{cases} 1, & j \notin \mathcal{A}_{\text{true}} \\ 0, & j \in \mathcal{A}_{\text{true}} \end{cases}.$$

We define A_{pos} to be the matrix that results after column i is removed from A for each $i \in \mathcal{A}_{\text{true}}$, and $\mathbf{x}_{\text{true}}^{\text{pos}}$ to be \mathbf{x}_{true} restricted to the indices $i \notin \mathcal{A}_{\text{true}}$. The key observation, then, is

$$A\mathbf{x}_{\text{true}} = (AD_{\text{true}})\mathbf{x}_{\text{true}} = A_{\text{pos}}\mathbf{x}_{\text{true}}^{\text{pos}}, \quad (3.1)$$

which yields the modified linear system

$$A_{\text{pos}}\mathbf{x}_{\text{true}}^{\text{pos}} = \mathbf{b}, \quad (3.2)$$

and corresponding reduced linear statistical model

$$\hat{\mathbf{b}} = A_{\text{pos}}\mathbf{x}_{\text{true}}^{\text{pos}} + \boldsymbol{\eta}. \quad (3.3)$$

We note that for the indices $i \notin \mathcal{A}_{\text{true}}$, (3.2) and (3.3) are equivalent to (1.1) and (1.2) respectively.

By Theorem 2.2, if A has full column rank,

$$\hat{\mathbf{x}}_{\text{BLUE}}^{\text{pos}} = (A_{\text{pos}}^T C_{\boldsymbol{\eta}}^{-1} A_{\text{pos}})^{-1} A_{\text{pos}}^T C_{\boldsymbol{\eta}}^{-1} \mathbf{b}. \quad (3.4)$$

Combining (3.4) with the knowledge that the other indices are known to be zero yields the BLUE for \mathbf{x}_{true} given \mathbf{b} ; specifically,

$$\hat{\mathbf{x}}_{\text{BLUE}} = (D_{\text{true}} A^T C_{\boldsymbol{\eta}}^{-1} A D_{\text{true}})^{\dagger} D_{\text{true}} A^T C_{\boldsymbol{\eta}}^{-1} \mathbf{b}, \quad (3.5)$$

where “ \dagger ” denotes the *pseudo-inverse*. This follows from the fact that (3.5) satisfies (2.1), (2.2), and (2.3), which is straightforward to check.

We conclude with the important observation that if \mathbf{x}_{true} has a large number of zeros, solving the normal equations

$$(A^T C_{\boldsymbol{\eta}}^{-1} A)\mathbf{x} = A^T C_{\boldsymbol{\eta}}^{-1} \mathbf{b} \quad (3.6)$$

using, for example, the conjugate gradient algorithm (CG), may yield a solution that is far from the solution we desire, which is given by (3.5). Unfortunately, we cannot compute (3.5) since in practice we will never know D_{true} .

3.2. Covariance-Preconditioned Iterative Methods.

What we can do, though, is build an iterative method that does not require D_{true} . To do this, we first note that from (3.5) implies that $\hat{\mathbf{x}}_{\text{BLUE}}$ is a solution of the equation

$$D_{\text{true}}A^T C_{\boldsymbol{\eta}}^{-1}(AD_{\text{true}}\hat{\mathbf{x}}_{\text{BLUE}} - \mathbf{b}) = \mathbf{0}. \quad (3.7)$$

But since $\hat{\mathbf{x}}_{\text{BLUE}}$ is zero whenever \mathbf{x}_{true} is zero, we have $D_{\text{true}}\hat{\mathbf{x}}_{\text{BLUE}} = \hat{\mathbf{x}}_{\text{BLUE}}$ and $\hat{\mathbf{x}}_{\text{BLUE}} \odot D_{\text{true}}\mathbf{v} = \hat{\mathbf{x}}_{\text{BLUE}} \odot \mathbf{v}$, where \mathbf{v} is a vector of length n . Here again we use \odot to denote component-wise multiplication. It immediately follows, then, that $\hat{\mathbf{x}}_{\text{BLUE}}$ is a solution of the nonlinear equation

$$\mathbf{x} \odot A^T C_{\boldsymbol{\eta}}^{-1}(A\mathbf{x} - \mathbf{b}) = \mathbf{0}. \quad (3.8)$$

Equation (3.8) motivates the fixed point iteration

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \tau_k \mathbf{x}_k \odot A^T C_{\boldsymbol{\eta}}^{-1}(A\mathbf{x}_k - \mathbf{b}), \quad (3.9)$$

to compute $\hat{\mathbf{x}}_{\text{BLUE}}$, which is nothing other than the Preconditioned Modified Residual Norm Steepest Descent Algorithm (PMRNSD) of [16], with covariance-preconditioner $C_{\boldsymbol{\eta}}^{-1/2}$.

The line search parameter τ_k in (3.9) (see [16] for details) is given by

$$\tau_k = \min\{\tau_{\text{uc}}, \tau_{\text{bd}}\}, \quad (3.10)$$

where, if $\mathbf{v}_k = \mathbf{x}_k \odot \nabla J(\mathbf{x}_k)$,

$$\tau_{\text{uc}} = -\frac{\langle \mathbf{v}_k, \nabla J(\mathbf{x}_k) \rangle}{\langle \mathbf{v}_k, A^T A \mathbf{v}_k \rangle}, \quad (3.11)$$

and

$$\tau_{\text{bd}} = \min\{-[\mathbf{x}_k]_i / [\mathbf{v}_k]_i \mid [\mathbf{v}_k]_i < 0\}. \quad (3.12)$$

We note that (3.10) ensures that the PMRNSD iterates will satisfy the nonnegativity constraint $\mathbf{x}_k \geq \mathbf{0}$ for all k , which is what we wanted.

As was mentioned in the introduction, the nonnegativity constraint $\mathbf{x} \geq \mathbf{0}$ arises naturally as a physical constraint, but it serves another important purpose. Notice that (3.8) does not have a unique solution. Three that we can state immediately are $\mathbf{x} = \mathbf{0}$, $\mathbf{x} = \hat{\mathbf{x}}_{\text{BLUE}}$ and the minimum norm solution of $A\mathbf{x} = \mathbf{b}$ given by $\mathbf{x} = A^\dagger \mathbf{b}$. But the nonnegatively constrained problem

$$\min_{\mathbf{x} \geq \mathbf{0}} \|C_{\boldsymbol{\eta}}^{-1/2} A\mathbf{x} - C_{\boldsymbol{\eta}}^{-1/2} \mathbf{b}\|^2, \quad (3.13)$$

does have a unique solution (assuming A has full column rank and $C_{\boldsymbol{\eta}}$ is positive definite), which is given by

$$\mathbf{x}^* = (D^* A^T C_{\boldsymbol{\eta}}^{-1} A D^*)^\dagger D^* A^T \mathbf{b}. \quad (3.14)$$

Here D^* is a diagonal matrix defined by $[D^*]_{ii} = 1$ if $[\mathbf{x}^*]_i > 0$ and $[D^*]_{ii} = 0$ if $[\mathbf{x}^*]_i = 0$. More importantly, \mathbf{x}^* is also a solution of (3.8). Furthermore, (3.9) can also be obtained by applying the MRNSD algorithm of [16] to the nonnegatively constrained problem (3.13). Thus (3.9) converges, at least in principle, to the unique solution of (3.8) given by (3.14).

We now present algorithms of the form (3.9) that arise from three different choices of $C_{\boldsymbol{\eta}}^{-1}$.

Examples:

1. In the i.i.d. Gaussian noise case, $C_{\boldsymbol{\eta}}^{-1} = \sigma^2 I$, and (3.9) becomes

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \tau_k \mathbf{x}_k \odot A^T (A \mathbf{x}_k - \mathbf{b}), \quad (3.15)$$

which is the MRNSD algorithm of [16] for approximately solving (2.5) subject to $\mathbf{x} \geq \mathbf{0}$.

2. When the noise model is given by (2.15), $C_{\boldsymbol{\eta}}^{-1} = W$, where W is defined in (2.14), then (3.9) becomes

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \tau_k \mathbf{x}_k \odot A^T W (A \mathbf{x}_k - (\mathbf{b} - \beta \cdot \mathbf{1})), \quad (3.16)$$

We will call (3.16) weighted MRNSD (WMRNSD), and it approximately solves (2.16) subject to $\mathbf{x} \geq \mathbf{0}$. Unfortunately, the diagonal weight matrix W defined by (2.14) cannot be obtained exactly since $A \mathbf{x}_{\text{true}}$ is not known, but it can be approximated. For example, in the numerical experiments we will use the approximation

$$W^{-1} = \text{diag}(\mathbf{b} + \sigma^2), \quad (3.17)$$

which is motivated by (2.15). A more sophisticated approach for approximating the weight matrix W is given in [7].

3. Rather than approximate the weight matrix W as is suggested in example 2, one can instead replace it with an iteration dependent weight matrix

$$W(\mathbf{x}_k)^{-1} = \text{diag}(A \mathbf{x}_k + \beta \cdot \mathbf{1} + \sigma^2 \cdot \mathbf{1}). \quad (3.18)$$

Then (3.16) becomes

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - \tau_k \mathbf{x}_k \odot A^T \left(\frac{A \mathbf{x}_k - (\mathbf{b} - \beta \cdot \mathbf{1})}{A \mathbf{x}_k + \beta \cdot \mathbf{1} + \sigma^2 \cdot \mathbf{1}} \right), \\ &= \mathbf{x}_k - \tau_k \mathbf{x}_k \odot A^T \left(\mathbf{1} - \frac{\mathbf{b} + \sigma^2 \cdot \mathbf{1}}{A \mathbf{x}_k + \beta \cdot \mathbf{1} + \sigma^2 \cdot \mathbf{1}} \right), \end{aligned} \quad (3.19)$$

where division is done component-wise. Assuming that $A^T \mathbf{1} = \mathbf{1}$ and setting $\tau_k = 1$ for all k , (3.19) takes the form of the RL algorithm (2.11)! Thus RL can be viewed as an application of algorithm (3.9) in which $C_{\boldsymbol{\eta}}^{-1} = W(\mathbf{x}_k)$. We note that RL iterates have been shown to satisfy $\mathbf{x}_k \geq \mathbf{0}$ for all k . Thus no line search is necessary.

3.3. The Standard Preconditioning Approach and Ill-Conditioning.

The standard approach to preconditioning amounts to constructing a matrix, M , such that $M \approx A$, and then applying the iterative method to the system $M^{-1} A \mathbf{x} = M^{-1} \mathbf{b}$. If M is a good approximation of A in the sense that the singular values of $M^{-1} A$ are clustered around a fixed point away from zero, then the iterative method will converge quickly. However, care must be taken for severely ill-conditioned problems that arise in image deblurring since linear systems of the form $M \mathbf{z} = \mathbf{w}$ must be solved at each iteration. If M is a good approximation of A , then M will be ill-conditioned, and therefore inverting M may cause large inaccuracies during the early iterations.

One preconditioning approach for ill-posed problems, proposed in [11], uses an approximate truncated spectral decomposition preconditioner. Specifically, A is first

approximated by a block circulant matrix with circulant blocks (BCCB), from which it is easy to compute the spectral decomposition

$$M = \mathcal{F}^* \Lambda \mathcal{F},$$

where \mathcal{F} is the unitary two dimensional discrete Fourier transform matrix, and Λ is a diagonal matrix containing the eigenvalues of M . This decomposition can be computed very efficiently using fast Fourier transforms (FFT). With this spectral decomposition, we define a preconditioner

$$M_\tau = \mathcal{F}^* \Lambda_\tau \mathcal{F}, \quad (3.20)$$

where

$$[\Lambda_\tau]_i = \begin{cases} \lambda_i & \text{if } |\lambda_i| \geq \tau \\ 1 & \text{if } |\lambda_i| < \tau \end{cases} \quad (3.21)$$

The *truncation parameter*, τ , can be chosen using a regularization parameter choice method, such as generalized cross validation [5, 12, 21]. To understand why this approach works, we need to understand a little about truncated iteration regularization. It can be shown (see [5, 12, 21]) that the early iterations filter out components of the solution corresponding to the small singular values of the matrix. That is, the early iterations tend to reconstruct lower frequency components of the data, while the higher frequency components of the data, where the noise is primarily contained, are filtered out. It is this part of the iteration that we want to accelerate. At some point, though, the high frequency components start to be reconstructed, and the iterates begin to be corrupted with noise; this part of the iteration we do not want to accelerate.

In the sequel, we denote the subspace comprised of the frequencies containing the noise the *noise subspace*. The subspace comprised of the remaining frequencies will be denoted as the *signal subspace*. For a preconditioned iterative method, it is the singular values of the preconditioned system $M^{-1}A$ that we must consider. By clustering all of the singular values around one, we no longer have the information to distinguish between the signal and noise subspaces. However, if we use the preconditioner M_τ with a good spectral approximation, then it can be shown that the large singular values (i.e., those corresponding to the signal subspace) are clustered at one, and are well separated from the small singular values (those corresponding to the noise subspace); see [10, 11] for further details. Iterative methods applied to such a system will converge quickly to the regularized solution.

The PMRNSD algorithm given in [16], has the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \tau_k \mathbf{x}_k \odot A^T M^{-T} M^{-1} (A \mathbf{x} - \mathbf{b}), \quad (3.22)$$

where the preconditioning matrix M is chosen as described above. Although this approach is not statistically motivated, it is often quite effective. This is due to the fact that if A is an ill-conditioned matrix, which it will usually be in imaging applications, we seek a stable, or regularized, approximation of $\hat{\mathbf{x}}_{\text{BLUE}}$ rather than $\hat{\mathbf{x}}_{\text{BLUE}}$ itself, and the more ill-conditioned A is, the further away from $\hat{\mathbf{x}}_{\text{BLUE}}$ will be the regularized solution. Thus one can expect that in many instances, (3.22) with a good preconditioner will outperform the covariance-preconditioned algorithm (3.9).

Note that (3.22) uses a “left” preconditioning approach since the algorithm can be viewed as applying the (unpreconditioned) MRNSD algorithm to the linear system

$M^{-1}\mathbf{Ax} = M^{-1}\mathbf{b}$. We also mention here that right preconditioning (2.5) or (2.16) does not work well for the algorithms presented above. This is due to the fact that, as was pointed out in [16], MRNSD is already a right-preconditioned algorithm with step-dependent right-preconditioner $X_k = \text{diag}(\mathbf{x}_k)$. Additional right preconditioning modifies the algorithm in such a way that it can no longer be viewed as an iterative method for solving (3.8), which we have argued is the right problem to solve.

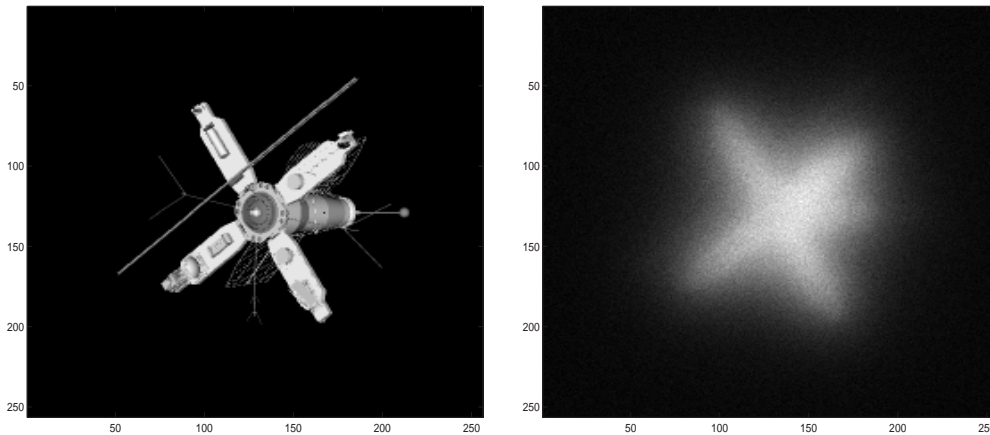


FIG. 4.1. *Satellite: True Image and Blurred, Noisy Data.*

4. Numerical Results. In this section we present numerical results using two image restoration test problems. The first set of data was developed at the US Air Force Phillips Laboratory, Lasers and Imaging Directorate, Kirtland Air Force Base, New Mexico. The image is a computer simulation of a field experiment showing a satellite as taken from a ground based telescope. The true and blurred images have 256×256 pixels, and are shown in Fig. 4.1. We remark that the $65,536 \times 65,536$ blurring matrix A is not constructed explicitly, but is defined implicitly by a so-called *point spread function* (PSF). The data for this test problem, including the true image and PSF, is contained in the *RestoreTools* image restoration package [15]. In addition, this package contains an implementation of MRNSD, as well as functions to efficiently implement matrix-vector multiplications with A (using the PSF), and for construction BCCB preconditioners; see [15] for more details.

The second test problem, which is shown in Fig. 4.2 is a computer simulation of a star cluster, which has been used by astronomers to test and compare image deblurring methods for the Hubble Space Telescope. The data can be obtained from the *Space Telescope Science Institute*. The blurring matrix A is defined by a PSF that was supplied by Dr. Brent Ellerbroek, Adaptive Optics Program Manager at the Gemini Observatory in Hilo, HI. This PSF is a simulation of the type of PSF that results from an adaptive optics telescope system. Such systems have PSFs that are nearly diffraction limited. This is due to the fact that the refractive blurring due to the atmosphere is, for the most part, cancelled out by the deformable mirror of the telescope, and hence, the blur in the image is due almost entirely to the diffractive blurring effects of the finite aperture of the telescope.

In Figs. 4.3 and 4.4, we compare the performance of MRNSD (3.15), WMRNSD (3.16), RL (2.11), PMRNSD (3.22) and the preconditioned conjugate gradient algorithm (PCG). As a measure for the accuracy of the reconstructions, we use the

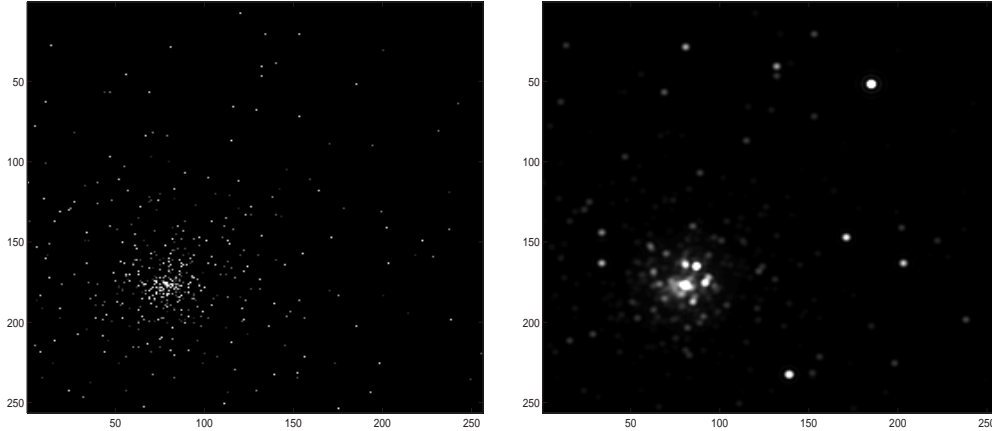


FIG. 4.2. *Star Field: True Image and Blurred, Noisy Data.*

relative error $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|/\|\mathbf{x}_{\text{true}}\|$. We use statistical model (2.6) for data generation. We take the sky, background count parameter to be $\beta = 10$ and the standard deviation of the normal random variable to be $\sigma = 5$. These values are representative of CCD cameras used in astronomy. We compare reconstructions for both test problems at two different noise levels. In particular, we consider the cases in which the signal to noise ratios (SNR) are approximately 100 and 10, which corresponds to a noise power that is 1% and 10% respectively of the signal power. The SNR is defined by $\|A\mathbf{x}_{\text{true}}\|/\sqrt{E(\|\boldsymbol{\eta}\|^2)}$, where, for noise model (2.6), $E(\|\boldsymbol{\eta}\|^2) = n(\sigma^2 + \beta + \beta^2) + \sum_{i=1}^n \hat{b}_i$. For WMRNSD, we approximate W as in (3.17). Note that for moderate to large values of σ^2 , say $\sigma^2 \geq 3^2$, it is extremely unlikely for the Gaussian $N(\sigma^2, \sigma^2)$ to take on negative values. Then since Poisson random variables take on only nonnegative integer values, the random variable $b_i + \sigma^2$ will nearly always be positive. Thus our choice of W will nearly always be positive definite. This was the case in each of our experiments. For PMRNSD and PCG, the BCCB preconditioner (3.20), (3.21) computed by *RestoreTools* [15] was used.

In Fig. 4.3, we see that for the star field data with both 1% and 10% noise, WMRNSD is the most effective algorithm. This is a rather surprising result if one views W as a preconditioner. Then, for this test problem, the statistically motivated, diagonal preconditioned algorithm (3.16) actually has better convergence properties than the structure motivated BCCB preconditioned algorithm (3.22).

In Fig. 4.4, we see that WMRNSD is not as effective when used on the satellite data, and that PMRNSD with the BCCB preconditioner is the fastest to converge for all of the methods and for both 1% and 10% noise levels. Thus, in this case, using the more accurate statistical model is not as critical, and, in fact, can hinder convergence. At first, this may seem disappointing, but in the recent paper [20], the authors make the observation that in the case of statistical model (2.6), reconstructions obtained using the standard (i.e., unweighted and without nonnegativity constraints) least squares approach are comparable, *in terms of relative error*, to reconstructions obtained using approaches that more accurately model noise statistics and that incorporate a nonnegativity constraint provided the object of interest is reasonably smooth with large regions of high intensity. The satellite is an object of this type. On the other hand, for stellar-like objects such as the star field example, it was shown

in [20] that approaches that incorporate a nonnegativity constraint and that more accurately account for CCD noise statistics typically provide better reconstructions, again in terms of relative error. These observations are supported by the convergence graphs in Fig. 4.3 and Fig. 4.4.

The convergence plot on the right in Fig. 4.4 indicates that if the linear system is highly ill-conditioned and there is a large amount of noise in the data, the inversion of noise may happen after just a few iterations, in which case it is important to have reliable stopping criteria.

It is also interesting to note that in Fig. 4.3 the performance of PCG is extremely poor. This has nothing to do with the choice of preconditioner. The poor performance is due to the fact that PCG computes an approximate solution of the normal equations

$$A^T A \mathbf{x} = A^T \mathbf{b}, \quad (4.1)$$

whereas the other algorithms in the comparison seek an approximation of the solution of nonnegatively constrained problems of the form (3.13), which has solution (3.14). It is clear that these two solutions can be quite different. Nonetheless, for extended objects such as the satellite PCG performs quite well (see Fig. 4.4). Thus, in some instances, incorporating nonnegativity constraints provides a significant advantage, but this is not always the case. The effects of incorporating nonnegativity constraints on image reconstruction problems is not well-understood and, the authors believe, is something that needs further study.

An observation that we feel is of particular importance comes with a comparison of the WMRNSD and RL algorithms. We note that the statistical argument has always been the main motivation for using RL. As we have shown in this paper, WMRNSD is also statistically motivated, but, as the results in Fig. 4.3 and Fig. 4.4 show, WMRNSD has superior convergence properties when compared to RL. This suggests that in place of RL, one might consider using WMRNSD. A hybrid of these two algorithms may also be effective.

Finally, we mention that a drawback of using relative error as a measure of reconstruction quality is that the accuracy with which higher frequencies in the object are reconstructed has little effect on its value. This is particularly important given the fact that the imposition of nonnegativity constraints is known to, in many cases, provide more accurate reconstructions of high frequency information. To see that this is the case, in Fig. 4.5 we present the reconstructions given by the PCG and PMRNSD iterates that minimize relative error on the Satellite data with 1% noise. Notice that the nonnegatively constrained reconstruction has better resolution even though the corresponding relative errors are nearly equal.

5. Conclusions. In this paper we presented a class of algorithms that can be used to solve nonnegatively constrained least squares and weighted least squares problems. We provided motivation for using the weighted least squares formulation based on the noise statistics of CCD camera image formation. We saw that in some cases a diagonal preconditioner that incorporates statistical information can be more effective than structure based preconditioners that attempt to approximate the coefficient matrix.

We showed that the RL algorithm, which is often used by application scientists, can be interpreted as a variable preconditioned version of the modified residual norm steepest descent algorithms considered in this paper. This provides an important link that should prove useful when deciding which algorithm should be used for a specific problem.

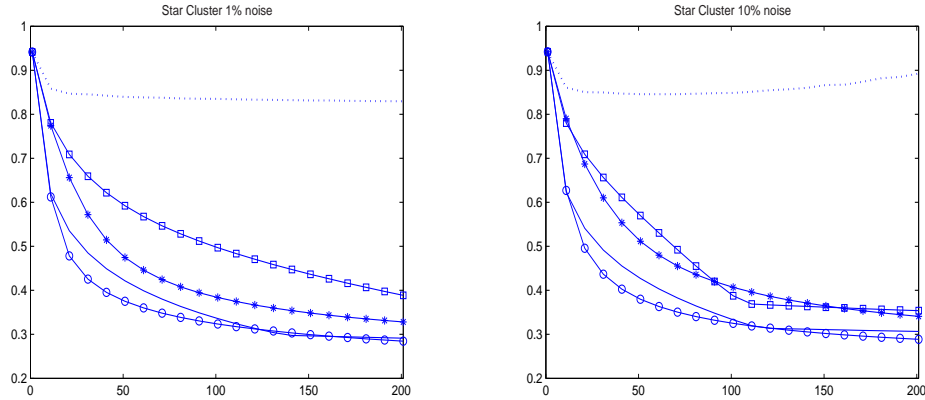


FIG. 4.3. *Star Field Reconstructions: Relative Error $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|/\|\mathbf{x}_{\text{true}}\|$ Versus Iteration Count.* The left and right plots correspond to 1% and 10% error respectively. The dotted line denotes PCG; the solid line with squares denotes MRNSD; the solid line with stars denotes RL; the solid line denotes PRMNSD; and the solid line with circles denotes WMRNSD.

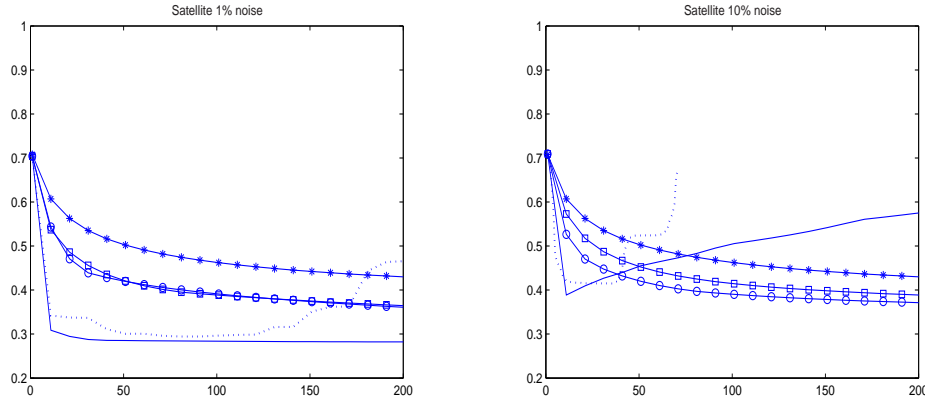


FIG. 4.4. *Satellite Reconstructions: Relative Error $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|/\|\mathbf{x}_{\text{true}}\|$ Versus Iteration Count.* The left and right plots correspond to 1% and 10% error respectively. The dotted line denotes PCG; the solid line with squares denotes MRNSD; the solid line with stars denotes RL; the solid line denotes PRMNSD; and the solid line with circles denotes WMRNSD.

In our numerical experiments, we found that of the two statistically motivated algorithms, WMRNSD and RL, WMRNSD has superior convergence properties. This suggests that in place of RL, application scientists might consider using WMRNSD.

Another important attribute of the algorithms presented in this paper is that they enforce a nonnegativity constraint. The performance of the unconstrained PCG algorithm when compared to the MRNSD algorithms discussed in this paper suggest that the nonnegativity constraint can make a substantial difference in both the accuracy of the reconstructions and in the convergence properties of the algorithms.

REFERENCES

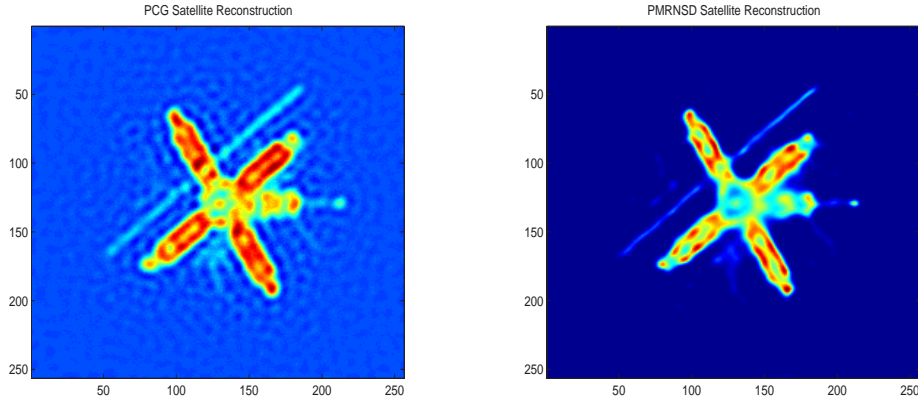


FIG. 4.5. *Satellite Reconstructions for 1% Noise Case. On the left hand side is the PCG reconstruction. On the right hand side is the PMRNSD reconstruction.*

- [1] J. M. Bardsley and C. R. Vogel, *A nonnegatively constrained convex programming method for image reconstruction*, SIAM. J. Sci. Comput., **25**, pp. 1326–1343.
- [2] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*, IOP Publishing Ltd., London, 1998.
- [3] D. Calvetti, B. Lewis, L. Reichel and F. Sgallari *Tikhonov Regularization with Nonnegativity Constraint*, Elec. Trans. Num. Anal., **18** (2004), pp. 153–173.
- [4] R. H. Chan and M. K. Ng, *Conjugate gradient methods for Toeplitz systems*, SIAM Review, **38** (1996), pp. 427–482.
- [5] H. W. Engl, M. Hanke and A. Neubauer, *Regularization of Inverse Problems*, Kluwer Academic Publishers, Dordrecht, 2000.
- [6] W. Feller, *An Introduction to Probability Theory and Its Applications*, Wiley, New York, 1971.
- [7] J. A. Fessler, *Penalized Weighted Least-Squares Image Reconstruction for Positron Emission Tomography*, IEEE Trans. Medical Imaging, **13**(2) (1994), pp. 290-300.
- [8] G. H. Golub and C. Van Loan, *Matrix Computations, third edition*, Johns Hopkins Press, 1996.
- [9] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, 1997.
- [10] M. Hanke and J. G. Nagy, *Restoration of atmospherically blurred images by symmetric indefinite conjugate gradient techniques*, Inverse Problems, **12** (1996), pp. 157–173.
- [11] M. Hanke, J. G. Nagy and R. J. Plemmons, *Preconditioned iterative regularization for ill-posed problems*, in Numerical Linear Algebra, L. Reichel, A. Ruttan and R. S. Varga, eds., pp. 141–163, de Gruyter, Berlin, 1993.
- [12] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, 1997.
- [13] P. C. Hansen, *Regularization tools: A Matlab package for the analysis and solution of discrete ill-posed problems*, Numerical Algorithms, **6** (1994), pp. 1–35.
- [14] L. Kaufman, *Maximum likelihood, least squares, and penalized least squares for PET*, IEEE Trans. Med. Imag., **12** (1993), pp. 200–214.
- [15] J. G. Nagy, K. Palmer, and L. Perrone, *Iterative Methods for Image Restoration: A Matlab Object Oriented Approach*, Numerical Algorithms, **36** (2003), pp. 73–93.
- [16] J. Nagy and Z. Strakoš, *Enforcing nonnegativity in image reconstruction algorithms*, Mathematical Modeling, Estimation, and Imaging, David C. Wilson, et.al., Eds., 4121 (2000), pg. 182–190.
- [17] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.
- [18] 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.
- [19] D. L. Snyder, C. W. Helstrom, A. D. Lanterman, M. Faisal, and R. L. White, *Compensation for readout noise in CCD images*, Journal of the Optical Society of America A, **12** (1995), pp. 272–283.
- [20] R. Vio, J. Bardsley, and W. Wamsteker, *Least-Squares methods with Poissonian noise: an analysis and a comparison with the Richardson-Lucy algorithm*, Astronomy and Astrophysics,

preprint doi:[10.1051/0004-6361:20041997](https://doi.org/10.1051/0004-6361:20041997).

- [21] C. R. Vogel, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, 2002.