Technical Report

TR-2004-010

Preconditioning Strategies for a Nonnegatively Constrained Steepest Descent Algorithm

by

Jonathan Bardsley, James Nagy

MATHEMATICS AND COMPUTER SCIENCE

EMORY UNIVERSITY

PRECONDITIONING STRATEGIES FOR A NONNEGATIVELY CONSTRAINED STEEPEST DESCENT ALGORITHM

JONATHAN M. BARDSLEY* AND JAMES G. NAGY†

Abstract. A class of ill-posed inverse problems that arises in astronomical imaging is considered. An iterative steepest descent method for regular least squares problems which constrains the solution to be nonnegative is presented. A careful consideration of the noise statistics that arise from the use of a CCD camera for data generation motivates the extension of this algorithm for use on weighted least squares problems. Preconditioning strategies are examined for both algorithms, and it is shown that in order to preserve noise statistics, preconditioners must be highly structured. Examples from astronomical imaging are used to illustrate behavior of the methods.

Key words. astronomical imaging, image restoration, least squares, nonnegativity constraint, preconditioning, weighted least squares

AMS Subject Classifications: 65F20, 65F30

1. Introduction.

The basic linear system

$$A\mathbf{x} = \mathbf{b} \tag{1.1}$$

is ubiquitous in science and engineering applications; given the matrix A and vector $\mathbf{b}$, the goal is to compute the vector $\mathbf{x}$. For moderately sized problems, direct methods based on matrix factorizations can be used [7], while for large problems it is more common to use iterative methods [8, 16].

An interesting and difficult variation of (1.1) arises from discretization of ill-posed inverse problems, where $\mathbf{b}$ is not known exactly, and the matrix A is ill-conditioned. That is, consider the problem

$$A\mathbf{x} + \boldsymbol{\eta} = \mathbf{b} \tag{1.2}$$

where $\mathbf{b}$ and A are known, and the goal is to compute an approximation of $\mathbf{x}$. The vector $\boldsymbol{\eta}$ represents errors or noise in the measured data, and is generally not known. If $\boldsymbol{\eta}$ is small, it is tempting to simply ignore it, and use standard approaches to solve (1.1). However, if A is ill-conditioned, the resulting *inverse solution* is likely to be a very poor approximation of the true vector, $\mathbf{x}$. In order to compute a decent approximation of $\mathbf{x}$, some form of *regularization* must be employed [4, 11, 20]. In addition, if the effects of the noise are significant, *a priori* information regarding noise statistics should be incorporated, if possible, into the regularization method that is chosen.

In this paper, we consider a class of ill-posed inverse problems of the form (1.2) that arise when reconstructing an image of an astronomical object taken by a ground based telescope. In this case, light from an object, $\mathbf{x}$, 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 $\mathbf{x}$ is the discretized object of interest, $A\mathbf{x}$ is what is seen after the light from $\mathbf{x}$ has traveled through the refractive index fluctuations and through the telescope. In order to collect the blurred image

*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

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. Noise enters the data during this process, and thus the discrete mathematical model representing the image formation process is given by (1.2).

Other *a priori* information can be included in the formulation of the image reconstruction problem; namely, in our application of interest, the object being viewed, $\mathbf{x}$, has nonnegative intensity. This information can be formulated in the reconstruction algorithm by, say, solving a nonnegatively constrained least squares problem. In addition, incorporating *a-priori* statistical information can produce more accurate object reconstructions.

In this paper we consider a constrained least squares image reconstruction algorithm proposed by Kaufman [13], which can be interpreted as a *modified residual norm steepest descent* (MRNSD) method [15]. An advantage of the MRNSD formulation is that preconditioning can often be used to improve the rate of convergence [15]. We show how to extend MRNSD to solve constrained weighted least squares problems. In addition, we consider preconditioning strategies for MRNSD and weighted MRNSD (WMRNSD). In particular, with *a priori* statistical information in hand, we show that not all preconditioners preserve noise statistics. For regular least-squares problems a class of well-known preconditioners, namely circulant preconditioners, do preserve noise statistics, while for the weighted least-squares problems the class of such preconditioners is very restrictive.

This paper is outlined as follows. In section 2 we present the statistical model that characterizes the noise, $\boldsymbol{\eta}$, from CCD array data collection, as well as the approximate model used to motivate the weighted least squares formulation of the problem. In section 3 we describe the MRNSD algorithm, and show how it can be extended to solve weighted least squares problems. Preconditioning strategies will be discussed in section 4, and in section 5, we show that in cases of practical interest the weighted least-squares approach, which more accurately incorporates *a priori* statistical information, results in more accurate object reconstructions.

2. Statistical Models. The following statistical model (see Refs. [17, 18]) applies to image data from a CCD detector array:

$$b_i = \mathrm{n_{obj}}(i) + \mathrm{n_0}(i) + g(i), \quad i = 1, \ldots, N. \tag{2.1}$$

Here b_i is the i^{th} component of the vector $\mathbf{b}$ and is the data acquired by a readout of pixel i of the CCD detector array; $\mathrm{n_{obj}}(i)$ is the number of object dependent photoelectrons; $\mathrm{n_0}(i)$ is the number of background electrons; and $g(i)$ is the readout noise. The random variables $\mathrm{n_{obj}}(i)$, $\mathrm{n_0}(i)$, and $g(i)$ are assumed to be independent of one another and of $\mathrm{n_{obj}}(j)$, $\mathrm{n_0}(j)$, and $g(j)$ for $i \neq j$. The random variable $\mathrm{n_{obj}}(i)$ has a Poisson distribution with Poisson parameter $[A\mathbf{x}]_i$, where $\mathbf{x}$ is the true image, or object; $\mathrm{n_0}(i)$ is a Poisson random variable with a fixed positive Poisson parameter β; and $g(i)$ is a Gaussian random variable with mean 0 and fixed variance σ^2.

In the sequel, we will use the following notation to denote (2.1):

$$b_i = \mathrm{Poiss}([A\mathbf{x}]_i) + \mathrm{Poiss}(\beta) + N(0, \sigma^2), \qquad i = 1, \ldots, n. \tag{2.2}$$

As in [17], we consider the case in which the readout noise variance σ^2 is large. Then, according to Feller [5, pp. 190 and 245], the following approximation is accurate:

$$N(\sigma^2, \sigma^2) \approx \mathrm{Poiss}(\sigma^2). \tag{2.3}$$

Our own numerical experiments suggest that this approximation is accurate for $\sigma^2 > 40$. Using the independence properties of the random variables in (2.1) we obtain the following approximation of (2.2):

$$b_i + \sigma^2 = \text{Poiss}([A\mathbf{x}]_i + \beta + \sigma^2). \tag{2.4}$$

This motivates computing a minimizer of the corresponding negative log-likelihood function

$$\ell(A\mathbf{x}; \mathbf{b}) = \sum_{i=1}^{N}([A\mathbf{x}]_i + \beta + \sigma^2) - \sum_{i=1}^{N}(b_i + \sigma^2)\log([A\mathbf{x}]_i + \beta + \sigma^2). \tag{2.5}$$

with respect to $\mathbf{x}$, and subject to the nonnegativity constraint $\mathbf{x} \geq \mathbf{0}$ (i.e., $x_i \geq 0$ for all i). A popular algorithm that computes such a minimizer is the expectation-maximization (EM) algorithm (also known as the Richardson-Lucy method in the image processing literature) [2]. An advantage of the EM algorithm is that it is very simple to implement, and each iteration is relatively inexpensive, but it can be very slow to converge. More sophisticated optimization approaches can be employed (see, e.g., [1]), resulting in methods that are faster to converge, but are more costly per iteration.

A different approach is to invoke the approximation used in (2.3) yet again with σ^2 replaced by $[A\mathbf{x}]_i + \beta + \sigma^2$. We assume here that the elements of A are positive. Then, since the true image $\mathbf{x}$ is nonnegative, $[A\mathbf{x}]_i$ will be nonnegative for all i. Thus, due to our assumption that σ^2 is large, one can accurately approximate the Poisson random variable in (2.4) by a Gaussian random variable to obtain

$$b_i - \beta = [A\mathbf{x}]_i + N(0, [A\mathbf{x}]_i + \beta + \sigma^2). \tag{2.6}$$

Minimizing the negative log-likelihood function corresponding to (2.6) is equivalent to minimizing

$$J(\mathbf{x}) = \frac{1}{2}||A\mathbf{x} - \mathbf{b}_\beta||_C^2. \tag{2.7}$$

Here $\mathbf{b}_\beta = \mathbf{b} - \beta$, and $||\mathbf{v}||_C^2 = \mathbf{v}^T C\mathbf{v}$, where C is the covariance matrix, which is diagonal with diagonal entries $C_{ii} = ([A\mathbf{x}]_i + \beta + \sigma^2)^{-1}$. Typically, one does not know the values of $[A\mathbf{x}]_i$, but they can be approximated in order to obtain an approximation of C. For example, motivated by (2.6), one could take $C_{ii} = (b_i + \sigma^2)^{-1}$. A more sophisticated approach for approximating C is given in [6]. For regions of $A\mathbf{x}$ with an approximately constant intensity profile, the corresponding diagonal elements of C are approximately constant. In fact, if all of the diagonal elements of C are constant, minimizing (2.7) is equivalent to minimizing the regular least squares function, which is given by (2.7) with $C = I$ and $\mathbf{b}_\beta = \mathbf{b}$.

3. Iterative Algorithms. In this section we begin by presenting a simple iterative algorithm for minimizing the regular least squares function

$$J(\mathbf{x}) = \frac{1}{2}||A\mathbf{x} - \mathbf{b}||^2 \tag{3.1}$$

subject to a nonnegativity constraint $\mathbf{x} \geq \mathbf{0}$. (Note that (3.1) corresponds to (2.7) with $\mathbf{b}_\beta = \mathbf{b}$ and $C = I$.) The algorithm we derive is equivalent to the EMLS algorithm proposed by Kaufman [13]. However, we provide an alternative derivation that shows the approach is very closely related to the steepest descent method. Our derivation also shows how to modify the algorithm for use on (2.7).

We begin by transforming the constrained minimization problem into an unconstrained problem using the parameterization

$$\mathbf{x} = e^{\mathbf{z}}$$

(i.e., $x_i = e^{z_i}$) in (2.7) with $C = I$, which yields

$$\tilde{J}(\mathbf{z}) = \frac{1}{2}||Ae^{\mathbf{z}} - \mathbf{b}||^2.$$

Then, differentiating using the chain rule, we obtain

$$\begin{aligned}\nabla\tilde{J}(\mathbf{z}) &= e^{\mathbf{z}} \odot [A^T(Ae^{\mathbf{z}} - \mathbf{b})] \\ &= \mathbf{x} \odot [A^T(A\mathbf{x} - \mathbf{b})] \\ &= \mathbf{x} \odot \nabla J(\mathbf{x}),\end{aligned}$$

where "$\odot$" denotes Hadamard (component-wise) multiplication. Thus for $\mathbf{x} \geq \mathbf{0}$ to be a local nonnegatively constrained minimizer of J, it is necessary for

$$\mathbf{x} \odot \nabla J(\mathbf{x}) = \mathbf{0}. \tag{3.2}$$

It is interesting to note that (3.2) is a Karush-Kuhn-Tucker (KKT) condition for the nonnegatively constrained minimization problem of interest (see [20] for details).

Equation (3.2) is equivalent to

$$\mathbf{x} = \mathbf{x} - \tau\mathbf{x} \odot \nabla J(\mathbf{x}),$$

which yields the corresponding fixed point iteration

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \tau_k\mathbf{v}_k \quad \text{where} \quad \mathbf{v}_k = -\mathbf{x}_k \odot \nabla J(\mathbf{x}_k) \tag{3.3}$$

Notice the similarity with the steepest descent algorithm, in which $\mathbf{v}_k$ in (3.3) is replaced by $-\nabla J(\mathbf{x}_k)$, and where a line search is used to minimize the residual norm, $||\mathbf{b} - A\mathbf{x}_k||$, at each iteration.

Thus, we refer to the basic iteration (3.3) as *modified residual norm steepest descent* (MRNSD). However, in MRNSD, τ_k is given by

$$\tau_k = \min\{\tau_{\text{uc}}, \tau_{\text{bd}}\},$$

where τ_{uc} corresponds to the unconstrained minimizer of J in the direction $\mathbf{v}_k$, i.e.,

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

and τ_{bd} is the minimum $\tau > 0$ such that $\mathbf{x}_k + \tau\mathbf{v}_k \geq \mathbf{0}$, i.e.,

$$\tau_{\text{bd}} = \min\left\{-[\mathbf{x}_k]_i/[\mathbf{v}_k]_i \mid [\mathbf{v}_k]_i < 0\right\}.$$

We can now express MRNSD in algorithmic form.

MRNSD: To minimize $J(\mathbf{x}) = \frac{1}{2}||A\mathbf{x} - \mathbf{b}||^2$ subject to $\mathbf{x} \geq \mathbf{0}$.

$k = 0$;
$\mathbf{x}_0$ = nonnegative initial guess;
begin iterations
$\mathbf{v}_k = -\mathbf{x}_k \odot \nabla J(\mathbf{x}_k)$;
$\tau_k = \min\{\tau_{\mathrm{uc}}, \tau_{\mathrm{bd}}\}$;
$\mathbf{x}_{k+1} = \mathbf{x}_k + \tau_k \mathbf{v}_k$;
end iterations

In order to extend MRNSD to weighted least squares problems of the form (2.7), we note that J in (2.7) is also given by

$$J(\mathbf{x}) = \frac{1}{2}||\hat{A}\mathbf{x} - \hat{\mathbf{b}}||^2, \tag{3.4}$$

where $\hat{A} = C^{1/2}A$ and $\hat{b} = C^{1/2}\mathbf{b}_\beta$. The weighted MRNSD (WMRNSD) algorithm is then given by applying MRNSD to minimizing (3.4) subject to $\mathbf{x} \geq \mathbf{0}$.

4. Preconditioning. One of the disadvantages of using an iterative method such as MRNSD or WMRNSD is that they are often very slow to converge. As a result, preconditioning strategies for improving convergence properties are important. In general, for most applications, an effective preconditioner is a matrix M that approximates the inverse of A; that is, it satisfies $MA \approx I$. A better approximation leads to faster convergence to the inverse solution. This, however, is not a desirable property for noisy ill-posed problems because the inverse solution is corrupted with noise. Instead, an effective preconditioner should provide fast convergence to a regularized solution [10, 9, 14]. In addition, it should be relatively inexpensive to construct M, and relatively inexpensive to perform matrix-vector multiplies with M. In this section, we discuss preconditioning strategies for the MRNSD and WMRNSD algorithms.

4.1. The MRNSD Algorithm. For MRNSD, preconditioning was discussed in [15], but we will reconsider the approach that is advocated in light of the knowledge that in many cases the cost function that is chosen relates directly to the underlying statistical noise model. Consequently, if one wants to use preconditioning, it is important to consider what effect, if any, it has on the noise statistics. In order to illustrate, consider the linear system (1.2) and suppose that $\boldsymbol{\eta}$ is a stationary, Gaussian random vector, i.e., that $\eta_i \sim N(0, \sigma^2)$ for all i. In this case, the maximum likelihood estimator can be gotten by minimizing the regular least squares function (3.1). The preconditioning strategy advocated in [15] for minimizing (3.1) is to replace (1.2) by a preconditioned system

$$MA\mathbf{x} + M\boldsymbol{\eta} = M\mathbf{b}\,, \tag{4.1}$$

where M is a preconditioning matrix. MRNSD is then used to minimize

$$J_{\mathrm{prec}}(\mathbf{x}) = \frac{1}{2}||MA\mathbf{x} - M\mathbf{b}||^2, \tag{4.2}$$

which corresponds to the assumption that $M\boldsymbol{\eta}$ is also a stationary, Gaussian random vector. This will not always be the case. In particular, since

$$
\begin{aligned}
[M\boldsymbol{\eta}]_i &= \sum_{j=1}^{n} m_{ij}\eta_j \\
&\sim \sum_{j=1}^{n} m_{ij} N(0,\sigma^2) \\
&= N(0, \sigma^2 \sum_{j=1}^{n} m_{ij}^2),
\end{aligned}
$$

we see that $M\boldsymbol{\eta}$ will be stationary only if

$$\sum_{j=1}^{n} m_{ij}^2 = \sum_{j=1}^{n} m_{kj}^2 \tag{4.3}$$

for $i, k = 1, \ldots, n$; that is, if the row sums of the matrix

$$M \odot M\,, \quad \text{where} \quad [M \odot M]_{ij} = m_{ij}^2 \tag{4.4}$$

are equal to the same constant.

Condition (4.3) seems rather restrictive at first, but it is satisfied by a class of matrices that can be highly effective when used as preconditioners for astronomical imaging. In this application, A is often a block Toeplitz matrix with Toeplitz blocks (BTTB), which can be approximated well by block circulant matrices with circulant blocks (BCCB) [3, 14, 20]. Circulant matrices have the property that each row (and column) is a circular shift of its previous row (column), and BCCB matrices are the natural extension to two dimensional problems. It is evident that such matrices satisfy the condition given by (4.3). We remark that there exist efficient approaches for constructing BCCB preconditioners; see [3], and in the context of image deblurring, see [10, 9, 14, 20]. Section 5 provides numerical experiments that illustrate the effectiveness of BCCB preconditioners for MRNSD.

4.2. The WMRNSD Algorithm. We now turn our attention to the problem of incorporating preconditioning into the WMRNSD algorithm. This is motivated by our desire to improve the efficiency of WMRNSD for use in approximately minimizing (2.7).

An approach analogous to that discussed in Section 4.1 is to replace (1.2) by (4.1), where M is a preconditioner for A, then solve a least-squares problem of the form (3.4) with $\hat{A} = C^{1/2}MA$ and $\hat{b} = C^{1/2}M\mathbf{b}$, where C is the covariance matrix of $M\boldsymbol{\eta}$. This approach has shown itself to be ineffective in numerical experiments, which is not surprising due to the fact that even if MA is well-conditioned, unless $C \approx I$, the same may not be true for $C^{1/2}MA$.

Another approach, which circumvents the difficulties discussed in the previous paragraph, is to replace the linear system (1.2) first by

$$\hat{A}\mathbf{x} + N(0, I) = \hat{\mathbf{b}}, \tag{4.5}$$

where $\hat{A} = C^{1/2}A$, $\hat{\mathbf{b}} = C^{1/2}\mathbf{b}$ and C is the covariance matrix associated with $\boldsymbol{\eta}$, and then obtain a preconditioned linear system of the form

$$M\hat{\mathbf{b}} = M\hat{A}\mathbf{x} + M\,N(\mathbf{0}, I)\,, \tag{4.6}$$

As was discussed above, if M is BCCB the noise statistics remain stationary Gaussian. The difficulty with this approach is the computation of the BCCB preconditioner M. As was mentioned above, if A is BTTB, one can compute M rapidly. However, in the case of nonstationary, independent Gaussian noise statistics, e.g., statistical model (2.6), C is a diagonal matrix with a non-constant diagonal, and hence, even if A is a BTTB matrix, $\hat{A}$ won't be, in which case it is unlikely to be well approximated by a BCCB matrix. One could also choose as a preconditioner $\hat{M} = MC^{-1/2}$, where M is the BCCB approximation for the BTTB matrix A, but then noise statistics are not preserved unless $\boldsymbol{\eta}$ is stationary, i.e. that $C = c \cdot I$, where c is a constant, which is not the case.

Before continuing, we make the observation that in both of the above approaches, it is the presence of the weighting (covariance) matrix C that inhibits the effectiveness of preconditioning. On the other hand, when viewed as an application of MRNSD to (4.5) (see Section 3), WMRNSD itself can be viewed as a special case of the preconditioned MRNSD algorithm (compare (4.1) and (4.5)) in which the preconditioning matrix M is replaced by the weighting matrix C. This is similar to the idea of using preconditioning to ensure that the iteration vector lies in the "correct" subspace, as is discussed in Hansen [11] (for an implementation, see [12]).

A final possibility is to use *right preconditioning* instead of the *left preconditioning* approach that is represented by equation (4.1). In this approach, (1.2) is replaced by

$$\hat{A}\hat{\mathbf{x}} + \boldsymbol{\eta} = \mathbf{b}, \tag{4.7}$$

where $\hat{A} = AM^{1/2}$ and $\hat{\mathbf{x}} = M^{1/2}\mathbf{x}$. The WMRNSD algorithm is then applied to minimizing (2.7) with A replaced by $\hat{A}$ and $\mathbf{x}$ by $\hat{\mathbf{x}}$. To obtain $\mathbf{x}$ once iterations have stopped, we then solve the equation $M^{1/2}\mathbf{x} = \hat{\mathbf{x}}$. At first, it seems that this approach will work, but numerical results show that, in fact, the resulting algorithm often performs worse that the unpreconditioned version. This is due in part to the difficulties that arise due to the presence of the weighting matrix C, but it also stems from the form of the MRNSD algorithm itself. To see this, we consider using *right preconditioning* together with the MRNSD algorithm. Our arguments and conclusions extend in a straightforward manner to the WMRNSD algorithm. First, we note that once a component of $\mathbf{x}_k$ is zero in the MRNSD algorithm, the same component is zero for $\mathbf{x}_j$ for all $j > k$. Consequently, after iteration k, the function J is restricted to the nonzero indices in the iterations that follow. More specifically, if we define

$$[D_k]_{jj} = \begin{cases} 1, & [\mathbf{x}_k]_j = 0 \\ 0, & [\mathbf{x}_k]_j > 0 \end{cases}, \tag{4.8}$$

then the $k + 1$st iteration of MRNSD can also be obtained by applying one iteration of MRNSD to

$$AD_k\mathbf{x} + \boldsymbol{\eta} = \mathbf{b} \tag{4.9}$$

with initial guess $\mathbf{x}_k$. Thus, we can see that the underlying linear system actually changes with each iteration, and so, an effective preconditioning strategy will likely also require a different preconditioner at each iteration. It is not clear how to compute an appropriate preconditioner for AD_k, and more importantly, how the preconditioner could be efficiently updated at each iteration.

5. 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 Figure 5.1. 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 [14]. 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 [14] for more details.

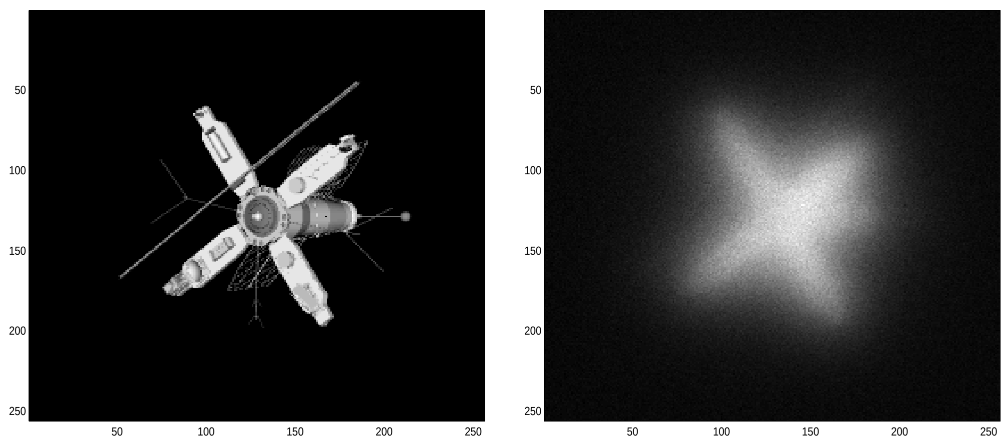

FIG. 5.1. *Satellite: True Image and Blurred, Noisy Data.*

The second test problem is a simulated 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.

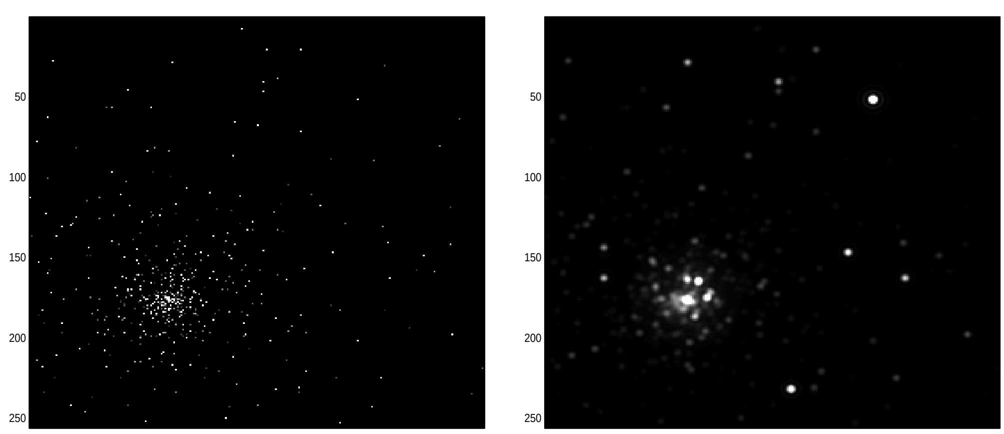

FIG. 5.2. *Star Field: True Image and Blurred, Noisy Data.*

In our first numerical experiment, we compare the reconstructions obtained by WMRNSD and MRNSD when the statistical model (2.2) is used 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. We compare the performance of the MRNSD and WMRNSD algorithms to the reconstruction problems. In particular, we apply MRNSD to the problem of minimizing the regular least squares function (3.1), and WMRNSD to the problem of minimizing (2.7). For WMRNSD, the diagonal weighting matrix C is defined by $C_{ii} = (b_i + \sigma^2)^{-1}$. 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 C will nearly always be positive definite. This was the case in each of the experiments below.

Figure 5.3 shows comparisons for the four separate test problems. First, we

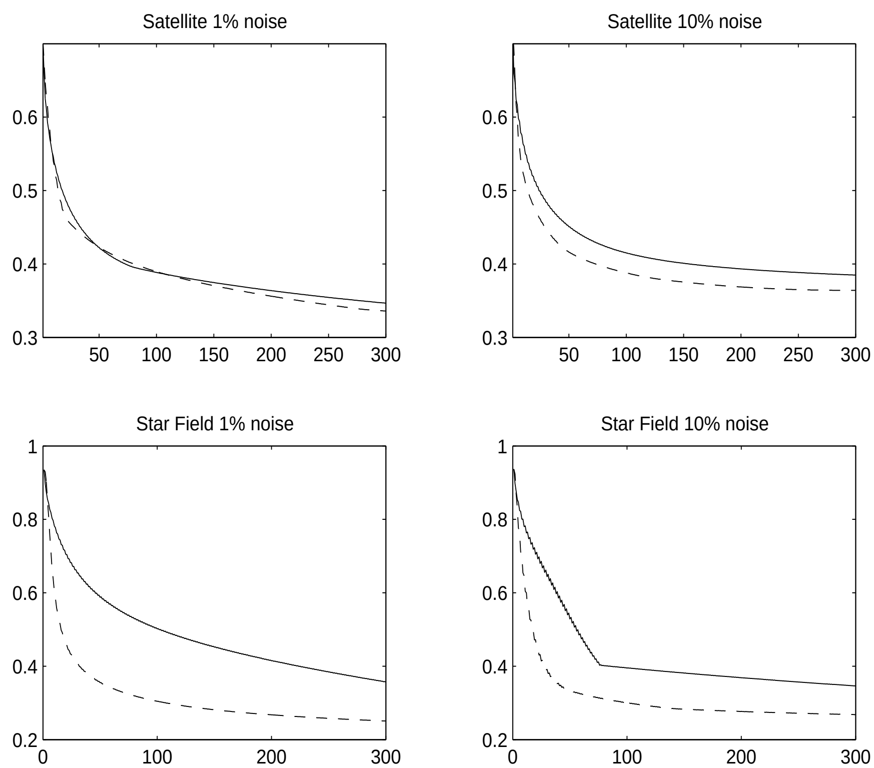

FIG. 5.3. *Relative error $||\mathbf{x}_k - \mathbf{x}_{\text{true}}||/||\mathbf{x}_{\text{true}}||$ versus iteration count for MRNSD and WMRNSD. The upper plots correspond to the satellite test problem, with 1% and 10% on the left and right respectively. The lower plots correspond to the star field test problem, with 1% and 10% on the left and right respectively. The solid line denotes MRNSD. The dashed line denotes WMRNSD.*

note that the weighted least squares approach provides only a small increase in the resolution (measured by the relative error) of the reconstructions of the satellite data. Furthermore, if we take into account that preconditioning can be used to enhance the performance of MRNSD, WMRNSD seems to be a poor choice of algorithms for this problem. At first, this may seem disappointing, but in the recent paper [19], the authors make the observation that in the case of statistical model (2.2), the standard (i.e., unweighted) least squares approach provides object reconstructions that are roughly on par with approaches that more accurately model noise statistics 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 star-like objects such as the star field example, it is noted in [19] that approaches that more accurately account for CCD noise statistics can provide substantially better reconstructions. The observations are supported by the convergence graphs in Figure 5.3.

We now show that using BCCB preconditioners with MRNSD can be very effective. This is clearly shown in Figure 5.4. The figure in the upper right-hand corner

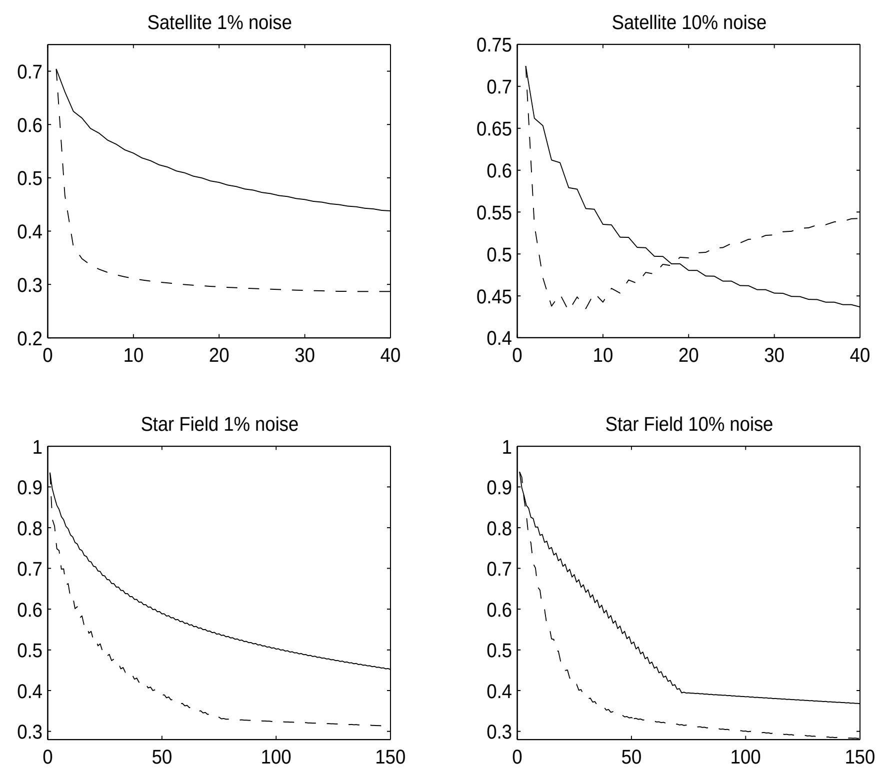

FIG. 5.4. *Relative error* $||\mathbf{x}_k - \mathbf{x}_{\text{true}}||/||\mathbf{x}_{\text{true}}||$ *versus iteration count for MRNSD and preconditioned MRNSD (PMRNSD). The upper plots correspond to the satellite test problem, with 1% and 10% on the left and right respectively. The lower plots correspond to the star field test problem, with 1% and 10% on the left and right respectively. The solid line denotes MRNSD. The dashed line denotes PMRNSD.*

does indicate, though, 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.

6. Conclusions. In this paper we have shown that the MRNSD algorithm, which is an iterative method that solves a nonnegativity constrained least squares problems, can be extended to weighted least squares problems. We denote this algorithm WMRNSD and note that it can be viewed as a preconditioned version of MRNSD in which the preconditioning matrix is constructed using *a priori* knowledge of noise statistics. In addition, we have examined preconditioning strategies for these approaches, and found that effective preconditioning of the MRNSD algorithm is possible. However, in order to preserve noise statistics, we advocate the use of circulant, or BCCB, preconditioners. For the WMRNSD algorithm, on the other hand, effective and computationally efficient preconditioners are much more difficult to construct. The difficulties in choosing such preconditioners stem both from the presence of the weighting (covariance) matrix associated with the weighted least squares problem of interest, as well as from the way in which the enforcement of the nonnegativity constraints results in a modified underlying linear system at each iteration.

REFERENCES

[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] R. H. Chan and M. K. Ng, *Conjugate gradient methods for Toeplitz systems*, SIAM Review, **38** (1996), pp. 427–482.
[4] H. W. Engl, M. Hanke and A. Neubauer, *Regularization of Inverse Problems*, Kluwer Academic Publishers, Dordrecht, 2000.
[5] W. Feller, *An Introduction to Probability Theory and Its Applications*, Wiley, New York, 1971.
[6] J. A. Fessler, *Penalized Weighted Least-Squares Image Reconstruction for Positron Emission Tomography*, IEEE Trans. Medical Imaging, **13(2)** (1994), pp. 290-300.
[7] G. H. Golub and C. Van Loan, *Matrix Computations, third edition*, Johns Hopkins Press, 1996.
[8] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, 1997.
[9] M. Hanke and J. G. Nagy, *Restoration of atmospherically blurred images by symmetric indefinite conjugate gradient techniques*, Inverse Problems, **12** (1996), pp. 157–173.
[10] 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.
[11] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, 1997.
[12] 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.
[13] L. Kaufman, *Maximum likelihood, least squares, and penalized least squares for PET*, IEEE Trans. Med. Imag., **12** (1993), pp. 200–214.
[14] 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.
[15] 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.
[16] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.
[17] 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.
[18] 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.
[19] 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*, submitted. Copies are available at `http://web.math.umt.edu/bardsley/`.
[20] C. R. Vogel, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, 2002.