Technical Report

TR-2008-014

Efficient Iterative Approach for Large Scale Separable Nonlinear Inverse Problems.

by

Julianne Chung, James G. Nagy

Mathematics and Computer Science

EMORY UNIVERSITY

EFFICIENT ITERATIVE APPROACH FOR LARGE SCALE SEPARABLE NONLINEAR INVERSE PROBLEMS

JULIANNE CHUNG[*] AND JAMES G. NAGY[†]

Abstract. This paper considers an efficient iterative approach to solve separable nonlinear least squares problems that arise in large scale inverse problems. A variable projection Gauss-Newton method is used to solve the nonlinear least squares problem, and Tikhonov regularization is incorporated using an iterative Lanczos hybrid scheme. Regularization parameters are chosen automatically using a weighted generalized cross validation method, thus providing a nonlinear solver that requires very little input from the user. An application from image deblurring illustrates the effectiveness of the resulting numerical scheme.

Key words. Gauss-Newton method, ill-posed inverse problems, iterative methods, Lanczos bidiagonalization, hybrid method, Tikhonov regularization

AMS Subject Classifications: 65F20, 65F30

1. Introduction.

Ill-posed inverse problems arise in many important applications, including astrophysics, astronomy, medical imaging, geophysics, parameter identification, and inverse scattering; see, for example, [8, 16, 17, 41] and the references therein. In this paper we consider large scale inverse problems of the form

$$\boldsymbol{b} = \boldsymbol{A}(\boldsymbol{y}_{\text{true}})\,\boldsymbol{x}_{\text{true}} + \boldsymbol{\varepsilon}\,, \tag{1.1}$$

where $\boldsymbol{b}$ is a known (measured data) vector, $\boldsymbol{A}(\boldsymbol{y}_{\text{true}})$ is a severely ill-conditioned matrix defined by a parameter vector $\boldsymbol{y}_{\text{true}}$, and $\boldsymbol{\varepsilon}$ is unknown additive noise. It is often assumed that the matrix $\boldsymbol{A}(\boldsymbol{y}_{\text{true}})$ is known, and the goal then is to compute an approximation of $\boldsymbol{x}_{\text{true}}$. However, in realistic applications we may only know the parametric form of $\boldsymbol{A}(\boldsymbol{y})$, and $\boldsymbol{y}_{\text{true}}$ must be approximated through additional measurements or device calibration.

For example, in image reconstruction the vector $\boldsymbol{b}$ is data measured by an imaging device (such as a camera, telescope, microscope, or medical imaging scanner), and $\boldsymbol{A}(\boldsymbol{y})$ is an operator that models how the image is captured. The vector $\boldsymbol{y}$ is obtained through a calibration process, for example by collecting images of known objects, and thus is only an approximation of the true parameters, $\boldsymbol{y}_{\text{true}}$.

Equation (1.1) arises from discretization of ill-posed inverse problems, such as a first kind Fredholm integral equation, and thus regularization must be used to compute approximations of $\boldsymbol{x}_{\text{true}}$ [8, 16, 41]. In this paper we consider standard Tikhonov regularization and develop an efficient iterative approach to solve the nonlinear least squares problem

$$\min_{\boldsymbol{x},\boldsymbol{y}} \frac{1}{2}\left\{\|\boldsymbol{A}(\boldsymbol{y})\,\boldsymbol{x}-\boldsymbol{b}\|_2^2 + \lambda^2\|\boldsymbol{x}\|_2^2\right\} = \min_{\boldsymbol{x},\boldsymbol{y}} \frac{1}{2}\left\|\begin{bmatrix}\boldsymbol{A}(\boldsymbol{y})\\ \lambda\boldsymbol{I}\end{bmatrix}\boldsymbol{x} - \begin{bmatrix}\boldsymbol{b}\\ \boldsymbol{0}\end{bmatrix}\right\|_2^2. \tag{1.2}$$

We remark that other regularization methods, such as generalized Tikhonov regularization (which dampens the norm of an appropriate derivative of $\boldsymbol{x}$), total variation, ℓ_p-norm constraints, or even bound constraints [8, 16, 34, 41] may be preferable in

[*] Department of Mathematics and Computer Science, Emory University. Email: jmchung@mathcs.emory.edu. Research supported in part by DOE Computational Sciences Graduate Research Fellowship.

[†] Department of Mathematics and Computer Science, Emory University. Email: nagy@mathcs.emory.edu. Research supported in part by the NSF under grant DMS-0811031.

some applications. However, we focus on standard Tikhonov regularization because it is widely used in applications. Moreover, we show that for standard Tikhonov regularization it is possible to obtain a very efficient iterative solver in which it is not necessary to specify the regularization parameter λ; our approach can determine an appropriate regularization parameter from the data. We also remark that in some applications it may be necessary to also include a regularization term for $\boldsymbol{y}$, but in the applications we consider this is generally not necessary. It would not be difficult to extend the iterative methods we use for this case, but in order to keep the discussion focused, we do not consider such extensions in this paper.

This paper is outlined as follows. In Section 2 we consider the linear problem, where we assume the matrix $\boldsymbol{A}$ is known exactly. Specifically, we show that a Lanczos-Tikhonov hybrid iterative method can efficiently solve the linear problem with Tikhonov regularization. Moreover, we show that a weighted generalized cross validation scheme can reliably choose appropriate regularization parameters directly from the data. The nonlinear problem (1.2) is then considered in Section 3, where we use a variable projection method [10, 11, 19, 35] to exploit separability of the nonlinear least squares problem. The approach essentially reduces to using a Gauss-Newton scheme to minimize a reduced cost functional that depends only on $\boldsymbol{y}$, where at the k-th iteration it is necessary to solve a linear least squares problem of the form (1.2) for a fixed $\boldsymbol{y}_k$. For this problem we use the hybrid approach described in Section 2. An example from image deblurring is given in Section 4, and concluding remarks are given in Section 5.

2. Hybrid Iterative Scheme for the Linear Problem.

In this section we describe a hybrid Lanczos-Tikhonov iterative method to solve the linear least squares problem

$$\min_{\boldsymbol{x}} \frac{1}{2} \left\| \begin{bmatrix} \boldsymbol{A} \\ \lambda \boldsymbol{I} \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \right\|_2^2 . \tag{2.1}$$

The data vector $\boldsymbol{b} = \boldsymbol{b}_{\text{true}} + \boldsymbol{\varepsilon}$ is given, where $\boldsymbol{b}_{\text{true}} = \boldsymbol{A}\boldsymbol{x}_{\text{true}}$, and we assume in this section that $\boldsymbol{A}$ is known exactly. We first use the singular value decomposition (SVD) to review regularization by filtering. Since the SVD cannot be used for large scale problems, we then describe how filtering can be done through an iterative Lanczos bidiagonalization method, and finally we consider ways to stabilize the iterations by incorporating Tikhonov regularization.

2.1. Regularization by Filtering.

If $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the SVD of $\boldsymbol{A}$, and if $\boldsymbol{A}$ is invertible, we may be tempted to ignore the noise component $\boldsymbol{\varepsilon}$ and compute an inverse solution,

$$\boldsymbol{x}_{\text{inv}} = \boldsymbol{A}^{-1}\boldsymbol{b} = \sum_{i=1}^{n} \frac{\boldsymbol{u}_i^T\boldsymbol{b}}{\sigma_i}\boldsymbol{v}_i = \underbrace{\sum_{i=1}^{n} \frac{\boldsymbol{u}_i^T\boldsymbol{b}_{\text{true}}}{\sigma_i}\boldsymbol{v}_i}_{\boldsymbol{x}_{\text{true}}} + \underbrace{\sum_{i=1}^{n} \frac{\boldsymbol{u}_i^T\boldsymbol{\varepsilon}}{\sigma_i}\boldsymbol{v}_i}_{\text{error}}$$

where $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are the singular vectors of $\boldsymbol{A}$ (that is, the columns of the orthogonal matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively), and $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_n > 0$ are the singular values of $\boldsymbol{A}$ (that is, the elements of diagonal matrix $\boldsymbol{\Sigma}$). As indicated above, the inverse solution is comprised of two components: $\boldsymbol{x}_{\text{true}}$, which we would like to approximate, and an error term. Before discussing algorithms to compute approximations of $\boldsymbol{x}_{\text{true}}$ it is useful to study the error term.

For matrices arising from ill-posed inverse problems, we have the following properties:

P1. The matrix $\boldsymbol{A}$ is severely ill-conditioned, with the singular values σ_i decaying to zero without a significant gap to indicate numerical rank.

P2. The singular vectors corresponding to the small singular values tend to oscillate more (i.e, have higher frequency) than singular vectors corresponding to large singular values.

P3. The components $|\boldsymbol{u}_i^T \boldsymbol{b}_{\text{true}}|$ decay on average faster than the singular values σ_i. This is referred to as the *discrete Picard condition* [16].

From the first two properties we see that the high frequency components of the error term are highly magnified by division of small singular values. The computed inverse solution is dominated by these high frequency components, and is in general a very poor approximation of $\boldsymbol{x}_{\text{true}}$. However, the third property suggests that there is hope of reconstructing some information about $\boldsymbol{x}_{\text{true}}$; that is, an approximate solution can be obtained by reconstructing components corresponding to the large singular values, and filtering out components corresponding to small singular values. A filtered solution can be computed as

$$\boldsymbol{x}_{\text{filt}} = \sum_{i=1}^{n} \phi_i \frac{\boldsymbol{u}_i^T \boldsymbol{b}}{\sigma_i} \boldsymbol{v}_i \,, \tag{2.2}$$

where the *filter factors*, ϕ_i, satisfy $\phi_i \approx 1$ for large σ_i, and $\phi_i \approx 0$ for small σ_i. That is, the large singular value components of the solution are reconstructed, while the components corresponding to the small singular values are filtered out. Different choices of filter factors lead to different methods; for example the solution using Tikhonov regularization (2.1) can be written in filtered form as [16]

$$\boldsymbol{x}_{\lambda} = \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} \frac{\boldsymbol{u}_i^T \boldsymbol{b}}{\sigma_i} \boldsymbol{v}_i \,.$$

The filtering approach extends naturally to the case when $\boldsymbol{A}$ is singular (i.e., some of the singular values are exactly 0), and the often encountered case when $\boldsymbol{A}$ is $m \times n$, $m > n$.

Selecting an appropriate regularization parameter λ is crucial. If λ is too small, we risk amplifying noise, but if λ is too large we put too much weight on the regularization term, and computed solutions are over smoothed. Several approaches exist for guiding the choice of an appropriate value of λ, such as the discrepancy principle, L-curve, and generalized cross validation (GCV) [8, 16, 41]. In this paper we consider GCV, which is a predictive statistics-based method that does not require *a priori* estimates of the error norm. The basic idea of GCV is that a good choice of λ should predict missing values of the data. That is, if an arbitrary element of the observed data is left out, then the corresponding regularized solution should be able to predict the missing observation fairly well. We leave out each data value b_j in turn and seek the value of λ that minimizes the prediction errors, measured by the GCV function

$$G_{\scriptscriptstyle A,\,b}(\lambda) = \frac{n\|(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}_\lambda^\dagger)\boldsymbol{b}\|_2^2}{\left(\text{trace}(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}_\lambda^\dagger)\right)^2} \,, \tag{2.3}$$

where $\boldsymbol{A}_\lambda^\dagger = (\boldsymbol{A}^T\boldsymbol{A} + \lambda^2 \boldsymbol{I})^{-1}\boldsymbol{A}^T$, and $\boldsymbol{x}_\lambda = \boldsymbol{A}_\lambda^\dagger \boldsymbol{b}$. In the general case when $\boldsymbol{A}$ is $m \times n$, using the SVD we see that equation (2.3) can be rewritten as

$$G_{_{A,\,b}}(\lambda) = \frac{n\left(\displaystyle\sum_{i=1}^{n}\left(\frac{\lambda^2 \boldsymbol{u}_i^T \boldsymbol{b}}{\sigma_i^2+\lambda^2}\right)^2 + \sum_{i=n+1}^{m}(\boldsymbol{u}_i^T \boldsymbol{b})^2\right)}{\left((m-n)+\displaystyle\sum_{i=1}^{n}\frac{\lambda^2}{\sigma_i^2+\lambda^2}\right)^2}\,, \tag{2.4}$$

which is a computationally convenient form to evaluate, thus making GCV easily used with standard minimization algorithms.

2.2. Lanczos Bidiagonalization and LSQR. Computing the SVD can be very expensive for large scale problems, and unfortunately algorithms cannot easily exploit sparsity or structure of $\boldsymbol{A}$ if it is necessary to compute many singular values and singular vectors. The difficulty is that even if $\boldsymbol{A}$ is sparse or structured, it is highly unlikely that the same will be true of the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ containing the singular vectors. Thus other approaches must be considered for large scale inverse problems.

An alternative to SVD based filtering is to use an iterative method, such as LSQR [31], to solve the least squares problem (2.1). The drawback of this approach is that it is necessary to choose the regularization parameter λ, which is difficult to do without the SVD or additional information about the problem, such as a very good bound on the norm of $\boldsymbol{\varepsilon}$ or $\boldsymbol{x}_{\text{true}}$ [32, 33].

Another alternative to regularizing large scale problems is to apply an iterative method, again such as LSQR, directly to the least squares problem $\displaystyle\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2$, and incorporate regularization within the iteration process. To understand how this can be done, we briefly describe how the LSQR iterates are computed.

LSQR is based on the Lanczos bidiagonalization (LBD) process. Given an $m \times n$ matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$, the k-th iteration of Lanczos bidiagonalization ($k = 1, \ldots, n$) computes an $m \times (k+1)$ matrix $\boldsymbol{W}_k$, an $n \times k$ matrix $\boldsymbol{Y}_k$, an $n \times 1$ vector $\boldsymbol{y}_{k+1}$, and a $(k+1) \times k$ bidiagonal matrix $\boldsymbol{B}_k$ such that

$$\boldsymbol{A}^T\boldsymbol{W}_k = \boldsymbol{Y}_k\boldsymbol{B}_k^T + \gamma_{k+1}\boldsymbol{y}_{k+1}\boldsymbol{e}_{k+1}^T \tag{2.5}$$

$$\boldsymbol{A}\,\boldsymbol{Y}_k = \boldsymbol{W}_k\boldsymbol{B}_k, \tag{2.6}$$

where $\boldsymbol{e}_{k+1}$ denotes the $(k+1)$st standard unit vector and $\boldsymbol{B}_k$ has the form

$$\boldsymbol{B}_k = \begin{bmatrix} \gamma_1 & & & \\ \beta_2 & \gamma_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \gamma_k \\ & & & \beta_{k+1} \end{bmatrix}. \tag{2.7}$$

Matrices $\boldsymbol{W}_k$ and $\boldsymbol{Y}_k$ have orthonormal columns, and the first column of $\boldsymbol{W}_k$ is $\boldsymbol{b}/\|\boldsymbol{b}\|_2$. Given these relations, an approximate solution $\boldsymbol{x}_k$ can be computed from the *projected* least squares problem

$$\min_{\boldsymbol{x}\in R(\boldsymbol{Y}_k)} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2 = \min_{\hat{\boldsymbol{x}}} \frac{1}{2}\|\boldsymbol{B}_k\hat{\boldsymbol{x}}-\boldsymbol{W}_k^T\boldsymbol{b}\|_2^2 = \min_{\hat{\boldsymbol{x}}} \frac{1}{2}\|\boldsymbol{B}_j\hat{\boldsymbol{x}}-\beta\boldsymbol{e}_1\|_2^2 \tag{2.8}$$

where $\beta = \|\boldsymbol{b}\|_2$, and $\boldsymbol{x}_k = \boldsymbol{Y}_k\hat{\boldsymbol{x}}$. An efficient implementation of LSQR uses a QR updating scheme to compute $\hat{\boldsymbol{x}}$ at each iteration; see [31] for details.

An important property of LBD is that for small values of k the singular values of the matrix $\boldsymbol{B}_k$ approximate very well certain singular values of $\boldsymbol{A}$, with the quality of the approximation depending on the relative spread of the singular values; specifically, the larger the relative spread, the better the approximation [2, 12, 36]. For ill-posed inverse problems the singular values decay to and cluster at zero, such as $\sigma_i = O(i^{-c})$ where $c > 1$, or $\sigma_i = O(c^i)$, where $0 < c < 1$ and $i = 1, 2, \ldots, n$ [38, 40]. Thus the relative gap between large singular values is generally much larger than the relative gap between small singular values. We therefore expect that if we apply the LBD iteration to a linear system arising from discretization of an ill-posed inverse problem, then the singular values of $\boldsymbol{B}_k$ converge very quickly to the largest singular values of $\boldsymbol{A}$. The following example illustrates this situation.

Example. Consider an inverse heat equation generated by the function `heat` in the MATLAB package *Regularization Tools* [15]. This function constructs an $n \times n$ matrix $\boldsymbol{A}$, true solution vector $\boldsymbol{x}_{\text{true}}$, and (noise free) observation vector $\boldsymbol{b}_{\text{true}}$, using the Volterra integral equation of the first kind on $[0, 1]$ with kernel $a(s, t) = k(s - t)$, where

$$k(t) = \frac{t^{-3/2}}{2\sqrt{\pi}} \exp\left(-\frac{1}{4t}\right).$$

The vector $\boldsymbol{x}_{\text{true}}$ does not have a simple functional representation, but rather is constructed directly as a discrete vector; see [15] for details. The right-hand side $\boldsymbol{b}$ is produced as $\boldsymbol{b}_{\text{true}} = \boldsymbol{A}\boldsymbol{x}_{\text{true}}$. In this example we take $n = 256$.

We are interested in the nonzero singular values of $\boldsymbol{A}$ and their approximations computed from the LBD algorithm. In Figure 2.1 we show a plot of the singular values of $\boldsymbol{A}$ and their relative spread; that is,

$$\frac{\sigma_i(\boldsymbol{A}) - \sigma_{i+1}(\boldsymbol{A})}{\sigma_i(\boldsymbol{A})},$$

where we use the notation $\sigma_i(\boldsymbol{A})$ to denote the ith largest singular value of $\boldsymbol{A}$.

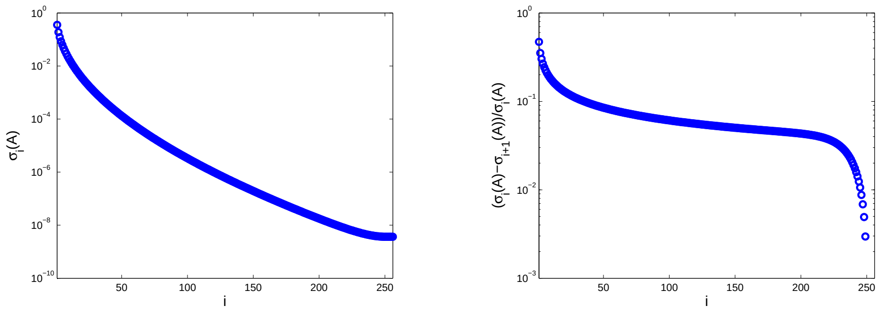


FIG. 2.1. *This figure shows plots of the singular values of* $\mathbf{A}$*, denoted as* $\sigma_i(\mathbf{A})$ *(left plot), and the relative spread of* $\mathbf{A}$*'s singular values (right plot).*

Figure 2.1 clearly illustrates the properties of ill-posed inverse problems; the singular values of $\boldsymbol{A}$ decay to and cluster at 0. Moreover, we clearly see that in general the relative gap of the singular values is larger for large singular values and smaller for the smaller singular values. Thus for small values of k, we expect to observe that the singular values of $\boldsymbol{B}_k$ converge quickly to the large singular values of $\boldsymbol{A}$. This can be seen in Figure 2.2, which compares the singular values of $\boldsymbol{A}$ with those of the bidiagonal matrix $\boldsymbol{B}_k$ for $k = 10, 20, 50$. □

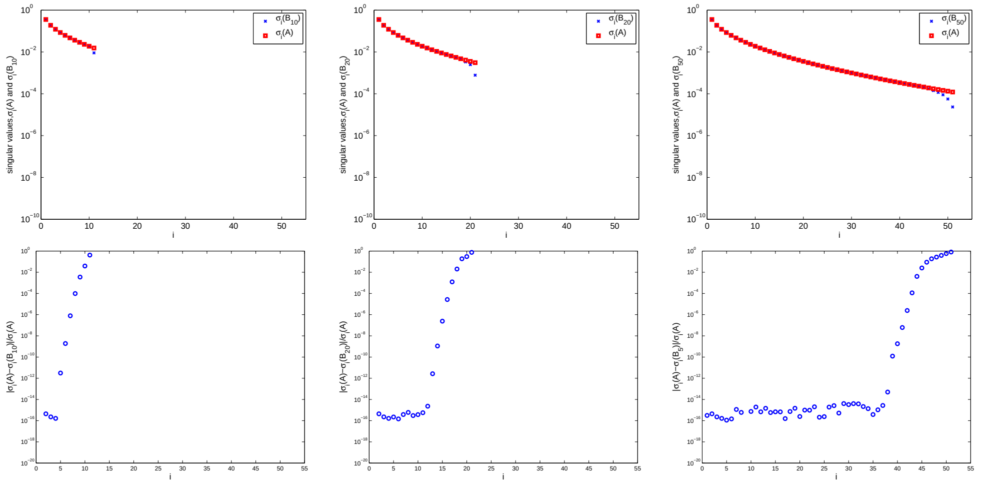


FIG. 2.2. *The plots in the top row of this figure show the singular values of* $\mathbf{A}$*, denoted as* $\sigma_i(\mathbf{A})$*, along with the singular values of* $\mathbf{B}_k$*, denoted as* $\sigma_i(\mathbf{B}_k)$*, for* $k = 10, 20, 50$*. The plots in the bottom row show the relative difference,* $\dfrac{|\sigma_i(\mathbf{A}) - \sigma_i(\mathbf{B}_k)|}{\sigma_i(\mathbf{A})}$.

The above example implies that if LSQR is applied to the least squares problem $\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$, then at early iterations the approximate solutions $\boldsymbol{x}_k$ will be in a subspace that approximates a subspace spanned by the large singular components of $\boldsymbol{A}$. Thus for $k \ll n$, $\boldsymbol{x}_k$ is a regularized solution. However, eventually $\boldsymbol{x}_k$ should converge to the inverse solution, which is corrupted with noise (recall the discussion in the previous subsection). This means that the iteration index k plays the role of a regularization parameter; if k is too small, then the computed approximation $\boldsymbol{x}_k$ is an over smoothed solution, while if k is too large, $\boldsymbol{x}_k$ is corrupted with noise. More extensive theoretical arguments of this semi-convergence behavior of conjugate gradient methods can be found elsewhere; see [13] and the references therein.

Unfortunately determining appropriate stopping criteria for ill-posed inverse problems is very difficult. The next subsection describes a hybrid approach that combines Lanczos bidiagonalization with Tikhonov regularization. Our implementation is able to choose regularization parameters and stopping criteria from the given data, and is very efficient for large scale problems.

2.3. Hybrid Lanczos-Tikhonov Method. Hybrid approaches that enforce regularization at each iteration of the Lanczos bidiagonalization method were first proposed by O'Leary and Simmons in 1981 [28], and later by Björck in 1988 [1]. The basic idea is to regularize the projected least squares problem (2.8) involving $\boldsymbol{B}_k$, which can be done very cheaply because of the smaller size of $\boldsymbol{B}_k$. Hybrid methods have the following benefits:

- Powerful regularization parameter choice methods can be implemented efficiently on the projected problem.
- Semi-convergence behavior of the relative errors observed in LSQR is avoided, so an imprecise (over) estimate of the stopping iteration does not have a

deleterious effect on the computed solution.

Realizing these benefits in practice, though, is nontrivial. Thus, various authors have considered computational and implementation issues, such as robust approaches to choose regularization parameters and stopping iterations; see for example, [3, 4, 5, 14, 21, 22, 24, 28]. In this subsection we focus on the implementation of Chung, Nagy and O'Leary [5], called HyBR[1], which has the advantage that it can refine the choice of the regularization parameter during the iteration process, and which can estimate the stopping iteration.

Because the singular values of $\boldsymbol{B}_k$ approximate those of $\boldsymbol{A}$, as the Lanczos iteration proceeds, the matrix $\boldsymbol{B}_k$ becomes more ill-conditioned. The iteration can be stabilized by including Tikhonov regularization in the projected least square problem (2.8), to obtain

$$\min_{\hat{\boldsymbol{x}}} \left\{ \|\boldsymbol{B}_k\hat{\boldsymbol{x}} - \beta\boldsymbol{e}_1\|_2^2 + \lambda^2\|\hat{\boldsymbol{x}}\|_2^2 \right\} \tag{2.9}$$

where again $\beta = \|\boldsymbol{b}\|_2$ and $\boldsymbol{x}_k = \boldsymbol{Y}_k\hat{\boldsymbol{x}}$. Thus we can build an iterative method where at each iteration we solve a regularized least squares problem involving a bidiagonal matrix $\boldsymbol{B}_k$. Notice that since the dimension of $\boldsymbol{B}_k$ is very small compared to $\boldsymbol{A}$, it is much easier to solve for $\hat{\boldsymbol{x}}$ in equation (2.9) than it is to solve for $\boldsymbol{x}$ in the full Tikhonov regularized problem (2.1). More importantly, when solving equation (2.9) we can use sophisticated parameter choice methods to find a suitable λ at each iteration.

Recall that for an ill-posed inverse problem the singular values of $\boldsymbol{B}_k$ converge very quickly to the largest singular values of $\boldsymbol{A}$. This means that very little regularization should be needed in (2.9) when the iteration index, k, is small, but eventually small singular values will converge, and more regularization will be needed in later iterations. Unfortunately standard regularization parameter choice methods do not work well in this dynamic situation, especially at the early iterations.

Recently Chung, Nagy and O'Leary [5] proposed using a weighted GCV (W-GCV) method, which had been used previously in certain statistical applications [6, 9, 23, 27, 39]. The W-GCV function, applied to the regularized projected least squares problem (2.9), has the form

$$G_{B_k,\,\beta e_1}(\omega,\lambda) = \frac{k\|(\boldsymbol{I} - \boldsymbol{B}_k\boldsymbol{B}_{k,\lambda}^{\dagger})\beta\boldsymbol{e}_1\|_2^2}{\left(\text{trace}(\boldsymbol{I} - \omega\boldsymbol{B}_k\boldsymbol{B}_{k,\lambda}^{\dagger})\right)^2}$$

Note the additional weighting parameter ω. Choosing $\omega = 1$ gives the standard GCV function for the matrix $\boldsymbol{B}_k$ and vector $\beta\boldsymbol{e}_1$. Choosing $\omega > 1$ results in smoother solutions, while $\omega < 1$ results in less smooth solutions. For smoothing spline applications, Kim and Gu [23] empirically found that standard GCV consistently produced regularization parameters that were too small, and choosing ω in the range of 1.2-1.4 worked well. However, when used on the projected least squares problem (2.8), it was shown in [5] that the standard GCV regularization parameter is chosen too large, requiring ω to be in the range $0 < \omega \leq 1$. It was also shown in [5] how to find an appropriate value for ω during the iteration process, and how to use the W-GCV information to determine an appropriate stopping iteration. The next example illustrates the benefits of using a hybrid approach such as HyBR.

[1] A MATLAB implementation of HyBR, including several examples, can be obtained from http://www.mathcs.emory.edu/~nagy/WGCV.

Example. Consider the inverse heat equation described in the previous example. To simulate noisy measured data, we construct a vector $\boldsymbol{b} = \boldsymbol{b}_{\text{true}} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is a vector containing pseudo-random entries chosen from a normal distribution, with zero mean and standard deviation equal to one, and scaled so that $\|\boldsymbol{\varepsilon}\|_2/\|\boldsymbol{b}_{\text{true}}\|_2 = 0.01$. The purpose of this example is illustrate how the HyBR method of [5] behaves on this problem, compared to using LSQR. Note that the LSQR iteration is equivalent to solving the projected problem involving $\boldsymbol{B}_k$ without any regularization. The convergence history for these two approaches are shown in Figure 2.3.

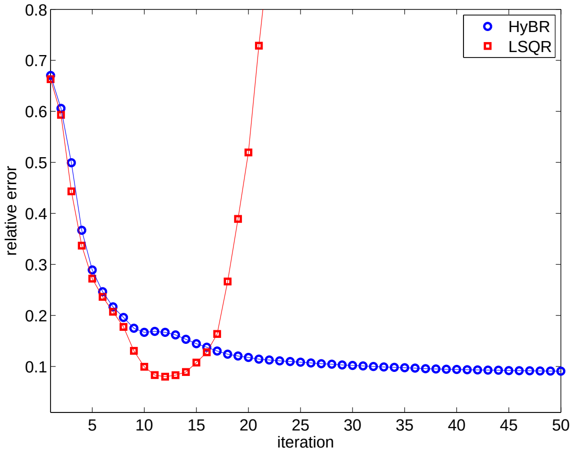


Fig. 2.3. *This figure compares the convergence history of* HyBR *and LSQR. Convergence history is measured by the relative errors at each iteration,* $\dfrac{\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|_2}{\|\mathbf{x}_{\text{true}}\|_2}$.

Notice that the convergence behavior of HyBR and LSQR are similar in the early iterations, indicating that no regularization is needed at this stage, and that HyBR does a good job in recognizing this. However, as the iterations proceed, if the projected problem is not regularized as in LSQR, then the noise begins to corrupt the iterates, $\boldsymbol{x}_k$, and we observe an increase in the LSQR errors. If we can recognize when the LSQR error is minimized and stop the iterations at that point, then we can obtain a good regularized solution. However, it is very difficult to recognize this optimal stopping point without knowledge of the true solution. Methods for estimating the stopping iteration based only on measured data are imprecise (usually over estimating), and as we can see from Figure 2.3, being off by a few iterations in LSQR can result in a poor approximation of the true solution.

Now observe the stable convergence behavior of HyBR, which recognizes that regularization should be incorporated on the projected problem. Although the method tends to over regularize in the early iterations, resulting in slower convergence, HyBR does eventually determine an appropriate amount of regularization that should be used. Finally we mention that although we show the error for 50 HyBR iterations, the method actually suggested stopping at the 38-th iteration, which as we can see from Figure 2.3 results in a reasonably accurate approximation of $\boldsymbol{x}_{\text{true}}$. □

HyBR is a very efficient and robust method for large scale linear inverse problems. In the next section we see how it is a critical component in our approach to solve the large scale nonlinear least squares problem (1.2).

3. Regularized Variable Projection for Nonlinear Problem. To simplify notation, we rewrite the nonlinear least squares problem given in equation (1.2) as

$$\min_{\mathbf{z}} \psi(\mathbf{z}) = \min_{\mathbf{z}} \frac{1}{2}\|\boldsymbol{f}(\mathbf{z})\|_2^2\,, \tag{3.1}$$

where $\mathbf{z}^T = [\ \boldsymbol{x}^T \quad \boldsymbol{y}^T\ ]$ and

$$\boldsymbol{f}(\mathbf{z}) = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y}) = \begin{bmatrix} \boldsymbol{A}(\boldsymbol{y}) \\ \lambda \boldsymbol{I} \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix}.$$

The nonlinear least squares problem (3.1) can be solved using a Gauss-Newton method [7, 20, 26, 29], which is an iterative algorithm that computes

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \boldsymbol{d}_k\,, \quad k = 0, 1, 2, \ldots$$

where

$$\boldsymbol{d}_k = -(\widehat{\psi}''(\mathbf{z}_k))^{-1}\psi'(\mathbf{z}_k)\,,$$

$\mathbf{z}_0$ is an initial guess, and $\widehat{\psi}''$ is an approximation of ψ''. It is not difficult to show that $\psi' = \boldsymbol{J}_\psi^T \boldsymbol{f}$, and an approximation of ψ'' is given by $\widehat{\psi}'' = \boldsymbol{J}_\psi^T \boldsymbol{J}_\psi$, where $\boldsymbol{J}_\psi$ the Jacobian matrix; for our problem it can be written as

$$\boldsymbol{J}_\psi = [\ \boldsymbol{f}_x \quad \boldsymbol{f}_y\ ] = \begin{bmatrix} \dfrac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{x}} & \dfrac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{y}} \end{bmatrix}.$$

If we define $\boldsymbol{r} = -\boldsymbol{f} = \boldsymbol{b} - \boldsymbol{A}(\boldsymbol{y})\,\boldsymbol{x}$, then the computation to update the search direction $\boldsymbol{d}_k$ at each Gauss-Newton iteration is equivalent to solving a least squares problem of the form

$$\min_{\boldsymbol{d}} \frac{1}{2}\|\boldsymbol{J}_\psi \boldsymbol{d} - \boldsymbol{r}\|_2^2\,.$$

To summarize, then, a Gauss-Newton method applied to (3.1) has the basic form:

General Gauss-Newton Algorithm

```
choose initial $\mathbf{z}_0 = \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{y}_0 \end{bmatrix}$
for $k = 0, 1, 2, \ldots$
    $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}(\boldsymbol{y}_k)\,\boldsymbol{x}_k$
    $\boldsymbol{d}_k = \arg\min_{\boldsymbol{d}} \|\boldsymbol{J}_\psi \boldsymbol{d} - \boldsymbol{r}_k\|_2$
    $\mathbf{z}_{k+1} = \mathbf{z}_k + \boldsymbol{d}_k$
end
```

This general Gauss-Newton approach can work well, but constructing and solving linear systems with $\boldsymbol{J}_\psi$ can be very expensive. Moreover, it requires specifying *a priori* the regularization parameter λ. We would like to develop an approach where the regularization parameter can be estimated by the algorithm. To do this, we adapt the variable projection method [10, 11, 19, 30, 35] for separable nonlinear least squares problems to large scale inverse problems.

The variable projection method exploits structure in the nonlinear least squares problem (1.2). The approach exploits the fact that $\psi(\boldsymbol{x}, \boldsymbol{y})$ is linear in $\boldsymbol{x}$, and that $\boldsymbol{y}$ contains relatively few parameters compared to $\boldsymbol{x}$. However, rather than explicitly separating variables $\boldsymbol{x}$ and $\boldsymbol{y}$ as in coordinate descent, variable projection implicitly eliminates the linear parameters $\boldsymbol{x}$, obtaining a reduced cost functional that depends only on $\boldsymbol{y}$. We then apply a Gauss Newton method to the reduced cost functional. Specifically, consider

$$\rho(\boldsymbol{y}) \equiv \psi(\boldsymbol{x}(\boldsymbol{y}), \boldsymbol{y})$$

where $\boldsymbol{x}(\boldsymbol{y})$ is a solution of

$$\min_{\boldsymbol{x}} \psi(\boldsymbol{x}, \boldsymbol{y}) = \min_{\boldsymbol{x}} \left\| \begin{bmatrix} \boldsymbol{A}(\boldsymbol{y}) \\ \lambda \boldsymbol{I} \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \right\|_2^2 . \tag{3.2}$$

To use the Gauss-Newton algorithm to minimize the reduced cost functional $\rho(\boldsymbol{y})$, we need to compute $\rho'(\boldsymbol{y})$. Note that because $\boldsymbol{x}$ solves (2.1), it follows that $\psi_{\boldsymbol{x}} = 0$, and thus

$$\rho'(\boldsymbol{y}) = \frac{d\boldsymbol{x}}{d\boldsymbol{y}} \psi_{\boldsymbol{x}} + \psi_{\boldsymbol{y}} = \psi_{\boldsymbol{y}} = \boldsymbol{f}_y^T \boldsymbol{f},$$

where the Jacobian of the reduced cost functional is given by $\boldsymbol{J}_\rho = \boldsymbol{f}_y = \boldsymbol{A}'(\boldsymbol{y})\boldsymbol{x}$. Thus, a Gauss-Newton method applied to the reduced cost functional has the basic form:

Reduced Gauss-Newton Algorithm

```
choose initial $\boldsymbol{y}_0$
for $k = 0, 1, 2, \dots$
    $\boldsymbol{x}_k = \arg\min_{\boldsymbol{x}} \left\| \begin{bmatrix} \boldsymbol{A}(\boldsymbol{y}_k) \\ \lambda \boldsymbol{I} \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \right\|_2$
    $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}(\boldsymbol{y}_k)\, \boldsymbol{x}_k$
    $\boldsymbol{d}_k = \arg\min_{\boldsymbol{d}} \|\boldsymbol{J}_\rho \boldsymbol{d} - \boldsymbol{r}_k\|_2$
    $\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + \boldsymbol{d}_k$
end
```

Although computing $\boldsymbol{J}_\rho$ is nontrivial, it is often much more tractable than constructing $\boldsymbol{J}_\psi$. However, the big advantage of using the variable projection method for large scale inverse problems is that we can use the hybrid approach discussed in the previous section to simultaneously estimate an appropriate regularization parameter and to compute $\boldsymbol{x}_k$ at each iteration. That is, in the above algorithm we replace the statement for $\boldsymbol{x}_k$ with

$$\boldsymbol{x}_k = \mathrm{HyBR}(\boldsymbol{A}(\boldsymbol{y}_k), \boldsymbol{b}) .$$

We conclude this section with a few remarks on computational issues. First, it may be necessary to include a line search in the Gauss Newton method [20], such as an Armijo rule. A finite difference approach can be used to numerically approximate the Jacobian if it cannot be computed analytically. Moreover, the least squares problem involving the Jacobian is generally not very difficult to solve, at least for the applications we have in mind where the number of parameters in $\boldsymbol{y}$ is significantly less than the number of parameters in $\boldsymbol{x}$. In this case, $\boldsymbol{J}_\rho$ has only a few columns (corresponding to the number of parameters in $\boldsymbol{y}$), and is generally well conditioned.

4. Numerical Examples. In this section we illustrate the use of our approach on an image deblurring problem, when the blurring operator is not known exactly. This problem is often referred to as *blind deconvolution* in the image processing literature. The image formation process is modeled as equation (1.1), where $\boldsymbol{b}$ is an observed, blurred and noisy image, and $\boldsymbol{x}$ is the image we want to reconstruct. The matrix $\boldsymbol{A}(\boldsymbol{y})$ models the blurring operation, and can be written as

$$\boldsymbol{A}(\boldsymbol{y}) = \boldsymbol{A}(\boldsymbol{P}(\boldsymbol{y}))$$

where $\boldsymbol{P}(\boldsymbol{y})$ is a *point spread function* (PSF). In many applications the blur is assumed to be spatially invariant, which means $\boldsymbol{P}(\boldsymbol{y})$ is an image of a point source object, and $\boldsymbol{A}(\boldsymbol{P}(\boldsymbol{y}))$ is structured. The precise structure depends on the imposed boundary conditions, but it is usually a combination of Toeplitz and Hankel matrices; see [17] for more details.

In any blind deconvolution problem it is necessary to make some assumptions about the blur. Here we assume a general Gaussian blur, where the PSF, centered at (k, l), has the form

$$\begin{aligned} p_{ij} &= \exp\left(-\frac{1}{2} \begin{bmatrix} i-k \\ j-l \end{bmatrix}^T \begin{bmatrix} s_1^2 & r^2 \\ r^2 & s_2^2 \end{bmatrix}^{-1} \begin{bmatrix} i-k \\ j-l \end{bmatrix} \right) \\ &= \exp\left(\frac{-(i-k)^2 s_2^2 - (j-l)^2 s_1^2 + 2(i-k)(j-l)r^2}{2s_1^2 s_2^2 - 2r^4} \right). \end{aligned}$$

The parameters s_1, s_2 and r determine the spread and orientation of the PSF, and oftentimes a scaling factor is introduced so that the PSF entries sum to 1.

Let

$$\boldsymbol{y} = \begin{bmatrix} s_1 \\ s_2 \\ r \end{bmatrix}$$

and let $\boldsymbol{A}(\boldsymbol{y}) = \boldsymbol{A}(\boldsymbol{P}(\boldsymbol{y}))$ be the matrix defined by the Gaussian PSF with parameters $\boldsymbol{y}$. We then use the regularized reduced Gauss-Newton method described in the previous section to jointly estimate the image $\boldsymbol{x}$ and blur parameters $\boldsymbol{y}$.

The Jacobian of the reduced cost functional can be computed using the chain rule,

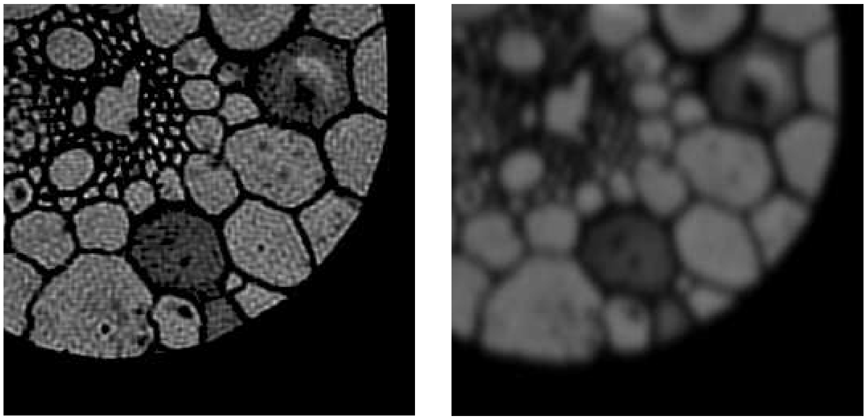
$$\begin{aligned} \boldsymbol{J}_\rho &= \frac{\partial}{\partial \boldsymbol{y}} \{ \boldsymbol{A}(\, \boldsymbol{P}(\boldsymbol{y})\,)\, \boldsymbol{x} \} \\ &= \frac{\partial}{\partial \boldsymbol{P}} \{ \boldsymbol{A}(\, \boldsymbol{P}(\boldsymbol{y})\,)\, \boldsymbol{x} \} \cdot \frac{\partial}{\partial \boldsymbol{y}} \{ \boldsymbol{P}(\boldsymbol{y}) \} \\ &= \boldsymbol{A}(\boldsymbol{X}) \cdot \frac{\partial}{\partial \boldsymbol{y}} \{ \boldsymbol{P}(\boldsymbol{y}) \} \end{aligned} \tag{4.1}$$

where $\boldsymbol{x} = \mathrm{vec}(\boldsymbol{X})$. The last equality is due to the special structure of a spatially invariant blurring operator and the commutative property of convolution.

As mentioned in Section 3, finite differences can be used to approximate the Jacobian, so it is not necessary to have an analytical expression for $\boldsymbol{J}_\rho$. However, if one chooses to compute $\frac{\partial}{\partial \boldsymbol{y}} \{ \boldsymbol{P}(\boldsymbol{y}) \}$ analytically, we remark that disregarding scaling factors which depend on the blur parameters can lead to erroneous derivative calculations, and careful calculation is needed to compute the correct values.

For efficient implementation we use the MATLAB package, *RestoreTools* [25], which can be obtained from `www.mathcs.emory.edu/∼nagy/RestoreTools`. In particular, we use the function `psfMatrix` to construct $\boldsymbol{A}(\boldsymbol{P}(\boldsymbol{y}))$ and $\boldsymbol{A}(\boldsymbol{X})$, and the associated routines for efficiently computing matrix-vector multiplications. The object oriented approach with operator overloading used in RestoreTools makes it very easy to perform these operations.

For the particular results presented in this paper, we assume that our goal is to reconstruct the image shown in Figure 4.1(a), given the blurred image in Figure 4.1(b).



(a) True Image (b) Observed Image

FIG. 4.1. *Test image of a piece of grain. The goal is to reconstruct an approximation of the true image, given the blurred and noisy observed image.*

The images presented here are 256 × 256, however the blurred image was created by convolving a larger 512 × 512 image with a Gaussian PSF defined by parameters $\boldsymbol{y}_{\text{true}} = [3, 4, .5]$, and then cropping the center of the image. In this way, we construct a realistic example in which boundary conditions and artifacts may play a role. We also added 1% Gaussian white noise to the image.

For the proposed Gauss Newton algorithm, we use an initial guess of $\boldsymbol{y}_0 = [5, 6, 1]$, and define the relative error for the blur parameters as

$$\Delta \boldsymbol{y} = \frac{||\boldsymbol{y}_k - \boldsymbol{y}_{\text{true}}||_2}{||\boldsymbol{y}_{\text{true}}||_2} \,.$$

Convergence results for the relative objective function value, the relative gradient value, the error in the blur parameters and the computed regularization parameters are presented in Table 4.1. It is impressive to see the accuracy of the blur parameters improve by more than an order of magnitude after only 7 Gauss Newton iterations.

Table 4.1
Convergence of iterations for reduced Gauss-Newton approach with HyBR. (1% noise)

iteration	relative objective	relative gradient	$\Delta \boldsymbol{y}$	HyBR computed λ
1	1.0000	1.0000	0.5716	0.1684
2	0.3309	0.6411	0.3347	0.1224
3	0.1493	0.4342	0.2192	0.0988
4	0.0762	0.2953	0.1469	0.0807
5	0.0479	0.2262	0.0998	0.0718
6	0.0355	0.1862	0.0639	0.0678
7	0.0288	0.1614	0.0345	0.0661
8	0.0242	0.1419	0.0139	0.0651
9	0.0213	0.1265	0.0240	0.0645
10	0.0190	0.1151	0.0449	0.0645
11	0.0171	0.1042	0.0656	0.0641
12	0.0159	0.0947	0.0853	0.0638

Based on the values of $\Delta \boldsymbol{y}$, a good stopping point is at 8 iterations. The improved reconstructed image is shown in Figure 4.2, along with the true image and the initial reconstruction using $\boldsymbol{y}_0$. We also notice that with too many iterations, the error in the blur parameters eventually begins to increase. Since $\boldsymbol{y}_{\text{true}}$ is not known in practice, alternate methods for selecting a stopping iteration still need to be investigated.

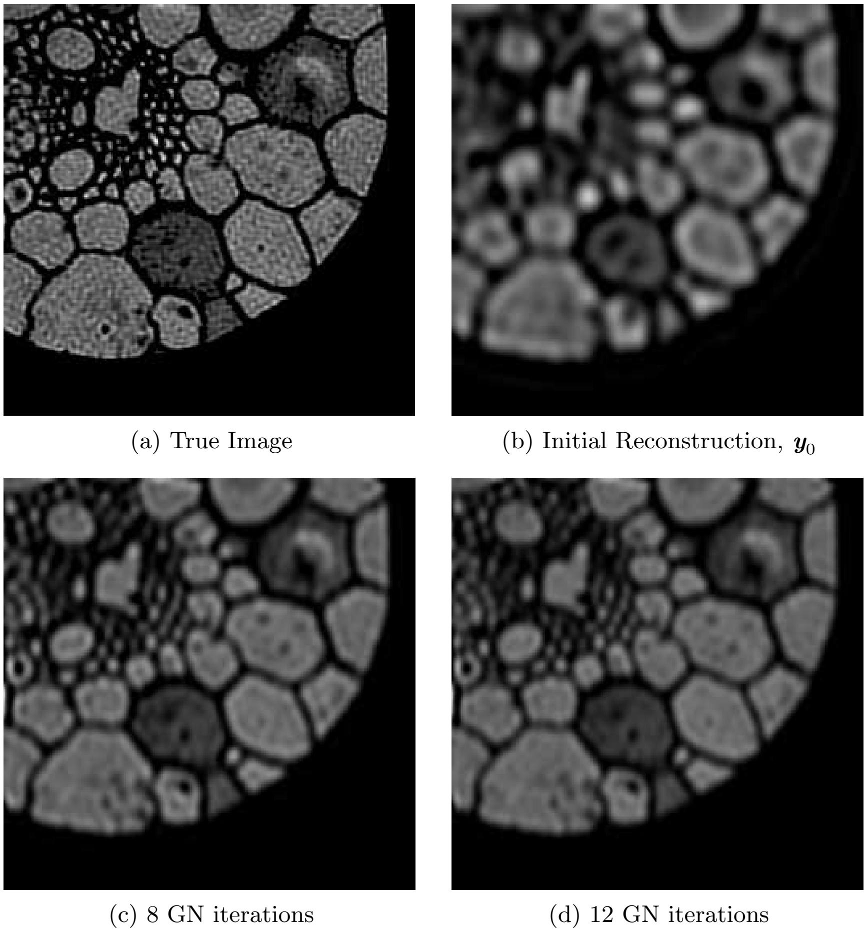
A well known property of the blind deconvolution problem is the lack of a unique solution. Previously proposed blind deconvolution algorithms have addressed this issue by requiring a significant amount of additional constraints on the problem, and including many parameters for the user to tune and select [18, 42]. Though we have not directly addressed the non-uniqueness problem, we have reduced the number of user-defined parameters to one, specifically, the stopping iteration. Furthermore, from the reconstructed image after 12 Gauss-Newton iterations, shown in Figure 4.2(d), we can see that an advantage of this algorithm is that a slight overestimate of the stopping iteration does not severely affect the image.

In this section, we have shown that a reduced Gauss Newton approach combined with an efficient linear solver with regularization can improve blind deconvolution algorithms. We mention here that this approach can be easily extended to a related, but more complex multi-frame blind (MFB) deconvolution problem [37]. In MFB deconvolution, multiple blurred images are collected, each with a different point spread function and noise realization. Following the notation from Equation (1.1), we have

$$\boldsymbol{b}^{(i)} = \boldsymbol{A}(\boldsymbol{y}_{\text{true}}^{(i)})\,\boldsymbol{x}_{\text{true}} + \boldsymbol{\varepsilon}^{(i)}\,,$$

for $i = 1, 2, ..., K$. The goal once again is to update the blur parameters and reconstruct the image simultaneously, and the regularized variable projection framework from Section 3 can be used. The blurred images and blur parameters should be concatenated to form one long vector so that all of the blur parameters for all of the images can be updated at the same time. The key difference is that the Jacobian for the MFB deconvolution problem will be a block diagonal matrix of the form

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}^{(1)} & & \\ & \ddots & \\ & & \boldsymbol{J}^{(K)} \end{bmatrix}$$



(a) True Image

(b) Initial Reconstruction, $\boldsymbol{y}_0$

(c) 8 GN iterations

(d) 12 GN iterations

FIG. 4.2. *A Comparison of Reconstructed Images corresponding to the* 1% *noise case.*

where $\boldsymbol{J}^{(i)}$ is the Jacobian defined in Equation (4.1) corresponding to image i. In our experiments, we found that the Gauss Newton algorithm with HyBR applied to the MFB deconvolution problem produced similar results to the blind deconvolution problem presented above.

5. Concluding Remarks. In this paper, we have described an efficient iterative approach for solving separable nonlinear inverse problems. Many researchers have studied the separable nonlinear least squares problem, but few have specifically applied it to large-scale ill-posed inverse problems. We have addressed this problem and shown that by combining a Gauss Newton approach for minimizing a reduced cost functional with a sophisticated iterative solver for computing Tikhonov regularized solutions for linear ill-posed inverse problems, one can efficiently solve large-scale nonlinear inverse problems, with relatively little user input required.

Nonlinear inverse problems of this form arise in many applications, and we have provided two examples from image deblurring in which the proposed algorithm can successfully update both the image and the blur parameters simultaneously. Future work includes understanding how the additional regularization term affects the theoretical convergence properties of this algorithm when applied to ill-posed problems, and developing an automated way to select the stopping iteration.

REFERENCES

[1] Å. Björck, *A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations*, BIT, 28 (1988), pp. 659–670.

[2] ———, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, 1996.

[3] Å. Björck, E. Grimme, and P. van Dooren, *An implicit shift bidiagonalization algorithm for ill-posed systems of linear equations*, BIT, 34 (1994), pp. 510–534.

[4] D. Calvetti and L. Reichel, *Tikhonov regularization of large scale problems*, BIT, 43 (2003), pp. 263–283.

[5] J. Chung, J. G. Nagy, and D. P. O'Leary, *A weighted GCV method for Lanczos hybrid regularization*, Elec. Trans. Numer. Anal., 28 (2008), pp. 149–167.

[6] D. Cummins, T. Filloon, and D. Nychka, *Confidence intervlas for non-parametric curve estimates: Toward more uniform pointwise coverage*, J. Am. Stat. Assoc., 96 (2001), pp. 233–246.

[7] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996.

[8] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Kluwer Academic Publishers, Dordrecht, 2000.

[9] J. Friedman and B. Silverman, *Flexible parsimonious smoothing and additive modeling*, Technometrics, 31 (1989), pp. 3–21.

[10] G. Golub and V. Pereyra, *The differentiation of pseudo-inverses and nonlinear least squares whose variables separate*, SIAM J. Numer. Anal., 10 (1973), pp. 413–432.

[11] ———, *Separable nonlinear least squares: the variable projection method and its applications*, Inverse Problems, 19 (2003), pp. R1–R26.

[12] G. H. Golub, F. T. Luk, and M. L. Overton, *A block Lanczos method for computing the singular values and corresponding singular vectors of a matrix*, ACM Trans. Math Soft., 7 (1981), pp. 149–169.

[13] M. Hanke, *Conjugate Gradient Type Methods for Ill-Posed Problems*, Pitman Research Notes in Mathematics, Longman Scientific & Technical, Harlow, Essex, 1995.

[14] ———, *On lanczos based methods for the regularization of discrete ill-posed problems*, BIT, 41 (2001), pp. 1008–1018.

[15] P. C. Hansen, *Regularization tools: A MATLAB package for analysis and solution of discrete ill-posed problems*, Numerical Algorithms, 6 (1994), pp. 1–35.

[16] ———, *Rank-deficient and discrete ill-posed problems*, SIAM, Philadelphia, PA, 1997.

[17] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra and Filtering*, SIAM, Philadelphia, PA, 2006.

[18] L. He, A. Marquina, and S. Osher, *Blind deconvolution using tv regularization and bregman iteration*, International Journal of Imaging Systems and Technology, 15 (2005), pp. 74–83.

[19] L. Kaufman, *A variable projection method for solving separable nonlinear least squares problems*, BIT, 15 (1975), pp. 49–57.

[20] C. T. Kelley, *Iterative Methods for Optimization*, SIAM, Philadelphia, 1999.

[21] M. E. Kilmer, P. C. Hansen, and M. I. Español, *A projection-based approach to general-form Tikhonov regularization*, SISC, submitted (2006).

[22] M. E. Kilmer and D. P. O'Leary, *Choosing regularization parameters in iterative methods for ill-posed problems*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1204–1221.

[23] Y. Kim and C. Gu, *Smoothing spline Gaussian regression: More scalable computation via efficient approximation*, J. Roy. Stat. Soc., 66 (2004), pp. 337–356.

[24] R. M. Larsen, *Lanczos bidiagonalization with partial reorthogonalization*, (1998).

[25] J. Nagy, K. Palmer, and L. Perrone, *Iterative methods for image deblurring: A matlab object oriented approach*, Numerical Algorithms, 36 (2004), pp. 73–93.

[26] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, New York, 1999.

[27] D. Nychka, B. Bailey, S. Ellner, P. Haaland, and M. O'Connel, *FUNFITS: Data analysis and statistical tools for estimating functions*, in Case Studies in Environmental Statistics, Springer, New York, 1998, pp. 159–179.

[28] D. P. O'Leary and J. A. Simmons, *A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems*, SIAM J. Sci. Stat. Comp., 2 (1981), pp. 474–489.

[29] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables (2nd edition)*, SIAM, Philadelphia, 2003.

[30] M. R. Osborne, *Separable least squares, variable projection, and the Gauss-Newton algorithm*, Elec. Trans. Numer. Anal., 28 (2007), pp. 1–15.

[31] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Soft., 8 (1982), pp. 43–71.

[32] M. Rojas and D. Sorensen, *A trust-region approach to the regularization of large-scale discrete forms of ill-posed problems*, SIAM J. Sci. Comput., 23 (2002), pp. 1843–1861.

[33] M. Rojas and T. Steihaug, *An interior point trust-region based method for large, non-negative regularization*, Inverse Problems, 18 (2002), pp. 1291–1307.

[34] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D, 60 (1992), pp. 259–268.

[35] A. Ruhe and P. Wedin, *Algorithms for separable nonlinear least squares problems*, SIAM Review, 22 (1980), pp. 318–337.

[36] Y. Saad, *On the rates of convergence of the lanczos and the block-lanczos methods*, SIAM J. Numer. Anal., 17 (1980), pp. 687–706.

[37] T. Schulz, *Multiframe blind deconvolution of astronomical images*, J. Opt. Soc. Am. A, 10 (1993), pp. 1064–1073.

[38] J. M. Varah, *Pitfalls in the numerical solution of linear ill-posed problems*, SIAM J. Sci. Stat. Comp., 4 (1983), pp. 164–176.

[39] R. Vio, P. Ma, W. Zhong, J. Nagy, L. Tenorio, and W. Wamsteker, *Estimation of regularization parameters in multiple-image deblurring*, Astronomy and Astrophysics, 423 (2004), pp. 1179–1186.

[40] C. R. Vogel, *Optimal choice of a truncation level for the truncated SVD solution of linear first kind integral equations when data are noisy*, SIAM J. Numer. Anal., 23 (1986), pp. 109–117.

[41] ———, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, PA, 2002.

[42] C. R. Vogel, T. Chan, and R. Plemmons, *Fast algorithms for phase-diversity-based blind deconvolution*, Proc. SPIE, 3353 (1998), pp. 994–1005.