

Research Article

Two New Efficient Iterative Regularization Methods for Image Restoration Problems

Chao Zhao, Ting-Zhu Huang, Xi-Le Zhao, and Liang-Jian Deng

School of Mathematical Sciences/Institute of Computational Science, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China

Correspondence should be addressed to Ting-Zhu Huang; tingzhu Huang@126.com

Received 4 March 2013; Revised 2 June 2013; Accepted 10 June 2013

Academic Editor: Marco Donatelli

Copyright © 2013 Chao Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Iterative regularization methods are efficient regularization tools for image restoration problems. The IDR(s) and LSMR methods are state-of-the-arts iterative methods for solving large linear systems. Recently, they have attracted considerable attention. Little is known of them as iterative regularization methods for image restoration. In this paper, we study the regularization properties of the IDR(s) and LSMR methods for image restoration problems. Comparative numerical experiments show that IDR(s) can give a satisfactory solution with much less computational cost in some situations than the classic method LSQR when the discrepancy principle is used as a stopping criterion. Compared to LSQR, LSMR usually produces a more accurate solution by using the L -curve method to choose the regularization parameter.

1. Introduction

Image deblurring is one of the most classic linear inverse problems. Blurring in images can arise from many sources, such as limitations of the optical system, camera and object motion, astigmatism, and environmental effects. The blurring process of an image can be formulated as a Fredholm integral equation of the first kind which has the following classic form:

$$\int_{\mathbb{R}_2} K(s-s', t-t') f(s', t') ds' dt' = g(s, t), \quad (1)$$
$$(s, t) \in \mathbb{R}^2,$$

where f and g are the original image and the blurred image, respectively. K is a given point spread function (PSF). PSF describes the blurring and the resulting image of the point source. The degree of blurring of the point object is a measure for the quality of an imaging system. More information about different PSFs can be found in [1].

Equation (1) can be discretized to form a linear system

$$Ax_{\text{true}} = b_{\text{true}}, \quad A \in \mathbb{R}^{m^2 \times m^2}, \quad x, b \in \mathbb{R}^{m^2}, \quad (2)$$

where the matrix A is ill-conditioned since it has many singular values close to zero [1]. For simplicity, we assume that A in this paper is nonsingular.

However, the right-hand side b is not available in many practical applications of image restoration because of the contamination of noise, so the linear system (2) can be reformulated as

$$Ax = b, \quad b = b_{\text{true}} + e, \quad e \in \mathbb{R}^{m^2}, \quad (3)$$

where e represents the noise.

Our goal is to obtain a good approximation of the original image x_{true} by solving the system (3) instead of the system (2) since b_{true} is not known. However, the solution of (3) is not a good approximation of the solution of (2) because of ill-conditioned A . In fact, there is a quite remarkable disparity among the corresponding solutions of (3) and (2) even if the norm of e is small.

The singular value decomposition (SVD) is a powerful tool for the analysis of discrete ill-posed problems. The SVD takes the form

$$A = U \Sigma V^T = \sum_{i=1}^{m^2} u_i \sigma_i v_i^T, \quad (4)$$

where Σ is a diagonal matrix with the singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{m^2} \geq 0$. For most image deblurring problems, the discrete Picard condition is satisfied: for all singular values larger than the norm of noise, the corresponding coefficients $|u_i^T b|$ decay faster than the σ_i .

Regularization methods are used to produce better solution. One of the most famous regularization methods is truncated SVD (TSVD) [2], which yields solution $x_k = \sum_{i=1}^k \sigma_i^{-1} (u_i^T b) v_i$. Another regularization method is Tikhonov regularization which consists of solving the minimization problem

$$\min_{x \in \mathbb{R}^{m^2}} \{ \|Ax - b\|_2^2 + \lambda \|Lx\|_2^2 \}, \quad (5)$$

where L is a regularization operator, which is often chosen to be the identity matrix ([3, Chapter 5]) and λ is the regularization parameter that controls the weighting between the two terms. However, the previously mentioned techniques are usually computationally expensive for large-scale problems like image deblurring. Iterative methods, especially Krylov subspace iterative methods, are used to solve these problems due to their inexpensive computational cost and easy implementation (see [4] for more details about Krylov subspace methods).

When we apply a Krylov subspace method to solve (3), the semiconvergence property is observed. Iterative methods can produce a sequence of iterates $\{x_1, x_2, \dots, x_k, \dots\}$ that initially tends to get closer to the exact solution but diverges again from the exact solution in later stages because the influence of the noise starts to dominate the solution. In this situation, the iteration number k can be considered as a regularization parameter. We also introduce different methods for choosing an effective k in this paper.

The conjugate gradient (CG) method is a classical Krylov subspace iterative method for solving the linear systems with symmetric positive definite (SPD) matrix, and its regularizing effects are well known (see [5] and [3, Chapter 6] for more details). When A is not SPD, the method can be applied to the normal equations $A^T A x = A^T b$. CGLS is the most stable way to implementation of CG algorithm for the normal equations ([3, Chapter 6], [6, Chapter 6]). Other Krylov subspace methods also have been used to solve image restoration problems. The regularizing properties of GMRES have been studied in [7]. The regularizing properties of BiCG and QMR have been tested in [8], which shows that BiCG and QMR are much faster than CGLS. The comparison of the regularizing properties of Krylov subspace methods CGLS, GMRES, CGS, BiCG, QMR, and BiCGSTAB has been investigated in [9]: CGS, BiCG, and BiCGSTAB are not sufficiently consistent with the discrepancy principle (we say that a method is consistent with the discrepancy principle if the residual norm with discrepancy principle as the stopping rule is linear dependence to the noise norm [9]. See Section 3.1 for details of the discrepancy principle); CGLS method is both efficient and consistent with the discrepancy principle, but it needs a large number of iterations; GMRES and QMR show intermediate features.

Recently, IDR(s) [10] and LSMR [11] iterative methods have been proposed for solving large linear system. IDR(s) is a new family of iterative methods which is fast and requires low memory. IDR(s) has attracted considerable attention, and different variants have been proposed in [12, 13]. LSMR is similar to LSQR by applying the MINERS method to the normal equation. The regularizing effects of MINERS have been analysed in [14, 15]. It has been proved that both $\|Ar_k\|$ and $\|r_k\|$ decrease monotonically for LSMR, while only $\|r_k\|$ is monotonic for LSQR. So, it is safer to terminate LSMR early.

Many methods have been proposed for choosing the regularization parameter. But a robust method appropriate for different situations is to be found. Two methods are used in this paper: the discrepancy principle (DP) and the discrete L -curve criterion. DP is the most famous and frequently used method. It has distinct advantages and disadvantages: it is simple and easy to be utilized, but it needs to know the norm of the noise in advance. The discrete L -curve criterion is used to choose the regularization parameter by using the curve of $(\log \|r_k\|, \log \|x_k\|)$. For more information about methods for choosing the regularization parameter, we suggest [6, Chapter 5] and references therein.

Inspired by the advantages of IDR(s) and LSMR, we apply them to solve the image restoration problems. We compare the performance of the IDR(s) method with that of the LSQR method for solving (3) with the iteration terminated by the DP method. We also make the comparison between the LSMR method and the LSQR method when the discrete L -curve criterion is used to choose the regularization parameter.

The paper is organized as follows. In Section 2, we review the IDR(s) and LSMR algorithms briefly and present analysis of the regularizing properties of the IDR(s) and LSMR algorithms. In Section 3, methods for choosing the regularization parameter especially the DP method and the discrete L -curve criterion are introduced. IDR(s) and LSMR methods are applied to solve image restoration problems and compared with with LSQR in Section 4. Section 5 summarizes conclusions.

2. IDR(s) and LSMR

2.1. The IDR(s) Method. The IDR(s) methods are a new family of iterative methods based on the induced dimension reduction (IDR) method proposed by Sonneveld in 1980. IDR(s) algorithm can generate a sequence of nested subspaces that corresponding residuals are forced to be in. The dimension of these nested subspaces is monotonically decreasing, and so in exact arithmetic, the solution can be obtained within a not very large number of iterations.

The nested subspaces are defined by

$$\mathcal{E}_j = (I - \omega_j A) (\mathcal{E}_{j-1} \cap \mathcal{S}), \quad (6)$$

where \mathcal{S} is a fixed proper subspace of \mathbb{R}^N and ω_j are nonzero scalars. The space \mathcal{S} is usually assumed as the left null space of $P \in \mathbb{R}^{N \times s}$:

$$P = (p_1 \ p_2 \ \dots \ p_s), \quad \mathcal{S} = \mathcal{N}(P^H). \quad (7)$$

The columns of P are chosen as orthogonalization of a set of random vectors.

Suppose that after n iterations we have s vectors $g_i \in \mathcal{G}_j$, $i = n - 1, \dots, n - s$ and s corresponding vectors u_i with $g_i = Au_i$. Define the matrices

$$\begin{aligned} G_n &= (g_{n-s} \ g_{n-s+1} \ \cdots \ g_{n-1}), \\ U_n &= (u_{n-s} \ u_{n-s+1} \ \cdots \ u_{n-1}). \end{aligned} \tag{8}$$

The algorithm produces $s + 1$ residuals in \mathcal{G}_j ; then the next residual can be found in \mathcal{G}_{j+1} . Two steps are needed to obtain the $s + 1$ residuals.

Step 1

Compute the first residual in \mathcal{G}_{j+1} .
 Calculate c from $(P^H G_n)c = P^H r_n$
 $v_n = r_n - G_n c$
 $r_{n+1} = r_n - \omega_{j+1} v_n - G_n c$
 $x_{n+1} = x_n + \omega_{j+1} A v_n + U_n c$.

Step 2

Compute the remaining residuals in \mathcal{G}_{j+1} , for $k = 1, \dots, s$.
 Calculate c from $(P^H G_{n+k-1})c = P^H r_{n+k}$
 $v_{n+k} = r_{n+k} - G_{n+k-1} c$
 $u_{n+k} = \omega_{j+1} v_{n+k} + U_{n+k-1} c$
 $g_{n+k} = A u_{n+k}$
 select α_i and β_i
 $g_{n+k} = g_{n+k} - \sum_{i=1}^{k-1} \alpha_i g_{n+i}$; $u_{n+k} = u_{n+k} - \sum_{i=1}^{k-1} \alpha_i u_{n+i}$;
 $r_{n+k+1} = r_{n+k} - \sum_{i=1}^k \beta_i g_{n+i}$; $x_{n+k+1} = x_{n+k} + \sum_{i=1}^k \beta_i u_{n+i}$;
 update $G_{:,k} = g_{n+k}$ to get G_{n+k+1} ; $U_{:,k} = u_{n+k}$ to get U_{n+k+1} .

Different choices of α_i and β_i would result in different variants of IDR(s). Take the same notations as in [12]: IDR(s)-proto represents the algorithm in [10] and IDR(s)-biortho represents the variant of the IDR(s) in [12]. IDR(s)-proto just computes

$$r_{n+k+1} = r_{n+k} - g_{n+k}, \quad x_{n+k+1} = x_{n+k} + u_{n+k}, \tag{9}$$

that is, $\alpha_i = 0$ ($i = 1, \dots, k - 1$), $\beta_i = 0$ ($i = 1, \dots, k - 1$), and $\beta_i = 1$ ($i = k$).

IDR(s)-biortho selects α_i and β_i to construct vectors that satisfy the biorthogonality conditions:

$$\begin{aligned} g_{n+k} \perp p_i, \quad i = 1, \dots, k - 1, \quad k = 2, \dots, s, \\ r_{n+k+1} \perp p_i, \quad i = 1, \dots, k, \quad k = 1, \dots, s. \end{aligned} \tag{10}$$

The IDR(s)-biortho algorithm is more stable and more accurate than the IDR(s)-proto algorithm, so we apply the IDR(s)-biortho algorithm to solve the problems in this paper.

As described in [10, 16], we have the following expression after n iterations:

$$r_n = \Phi_n(A) r_0, \tag{11}$$

$\Phi_n(A)$ is a polynomial of degree n . $\Phi_n(A)$ can be explicitly written as a product of two polynomials

$$\Phi_n(A) = \Omega_i(A) \Psi_{n-i}(A), \tag{12}$$

where $\Omega_i(t) = \prod_{k=1}^i (1 - \omega_k t)$, $\Psi_{n-i}(t) = \prod_{j=0}^{n-i} c_j t^j$. The factor $\Omega_i(A)$ is called damping factor and $\Psi_{n-i}(A)$ is called Lanczos factor. The damping factor is the residual polynomial of damped Richardson iterative method. Damped Richardson iterative method is used as an iterative regularizing method in [17] and sometimes referred to as Van Citter method in the literature; compare [18]. According to the conclusion of [16], the Lanczos factor is determined completely by the choice of P , whereas the damping factor depends on the choice of the parameters ω_i . Actually, in our experiments, different choices of parameter s (which controls the dimension of P) did not show notable influence to the performance in solving image restoration problems. So, the damping factor plays a role of iterative regularizing method.

2.2. The LSMR Method. The LSMR algorithm is equivalent to the MINRES applied to the normal equation. The algorithm is implemented by the Lanczos bidiagonalization algorithm. One form of the bidiagonalization procedure is the Golub-Kahan process:

$$\begin{aligned} \beta_1 u_1 = b, \alpha_1 v_1 = A^T u_1 \quad \text{for } k = 1, 2, \dots, \\ \beta_{k+1} u_{k+1} = A v_k - \alpha_k u_k, \quad \alpha_{k+1} v_{k+1} = A^T u_k - \beta_{k+1} v_k. \end{aligned} \tag{13}$$

The scalars $\alpha_k \geq 0$ and $\beta_k \geq 0$ are chosen so that the norms of u_k and v_k are both equal to one.

We define that $V_k = (v_1 \ v_2 \ \cdots \ v_k)$, $U_k = (u_1 \ u_2 \ \cdots \ u_k)$, and

$$B_k = \begin{pmatrix} \alpha_1 & & & & \\ \beta_2 & \alpha_2 & & & \\ & \ddots & \ddots & & \\ & & \beta_k & \alpha_k & \\ & & & & \beta_{k+1} \end{pmatrix}. \tag{14}$$

Then, the following relations hold:

$$\begin{aligned} U_{k+1} (\beta_1 e_1) &= b, \\ A V_k &= U_{k+1} B_k, \\ A^T U_{k+1} &= V_{k+1} \begin{pmatrix} B_k^T \\ \alpha_{k+1} e_{k+1}^T \end{pmatrix}, \end{aligned}$$

$$\begin{aligned}
A^T AV_k &= A^T U_{k+1} B_k = V_{k+1} \begin{pmatrix} B_k^T \\ \alpha_{k+1} e_{k+1}^T \end{pmatrix} B_k \\
&= V_{k+1} \begin{pmatrix} B_k^T B_k \\ \alpha_{k+1} \beta_{k+1} e_{k+1}^T \end{pmatrix}.
\end{aligned} \tag{15}$$

While LSQR computes at each iteration the vector y_k which minimizes $\|r_k\|$, LSMR chooses y_k differently, trying to minimize $\|A^T r_k\|$. Since

$$\begin{aligned}
r_k &= b - AV_k y_k = \beta_1 u_1 - U_{k+1} B_k y_k = U_{k+1} (\beta_1 e_1 - B_k y_k), \\
A^T r_k &= A^T b - A^T A x_k = \beta_1 \alpha_1 v_1 - A^T AV_k y_k \\
&= \beta_1 \alpha_1 v_1 - V_{k+1} \begin{pmatrix} B_k^T B_k \\ \alpha_{k+1} \beta_{k+1} e_{k+1}^T \end{pmatrix} y_k \\
&= V_{k+1} \left(\beta_1 \alpha_1 e_1 - \begin{pmatrix} B_k^T B_k \\ \alpha_{k+1} \beta_{k+1} e_{k+1}^T \end{pmatrix} y_k \right).
\end{aligned} \tag{16}$$

So, we have the following two least squares problems:

$$\begin{aligned}
&\min_{y_k} \|\beta_1 e_1 - B_k y_k\|, \quad \text{for LSQR} \\
&\min_{y_k} \left\| \left(\beta_1 \alpha_1 e_1 - \begin{pmatrix} B_k^T B_k \\ \alpha_{k+1} \beta_{k+1} e_{k+1}^T \end{pmatrix} y_k \right) \right\|, \quad \text{for LSMR.}
\end{aligned} \tag{17}$$

Both $\|r_k\|$ and $\|A^T r_k\|$ decrease monotonically for LSMR algorithm, while only $\|r_k\|$ is monotonic for LSQR algorithm. So, compared to LSQR, it is safer to terminate LSMR early.

LSMR works on the normal equation, so a simple analysis for the regularizing properties of the LSMR algorithm can be done, which is similar to the analysis of the CGLS method in [15]. Indeed LSMR and LSQR share the same Krylov subspace:

$$\begin{aligned}
&\mathcal{K} \\
&= \text{span} \left\{ A^T b, (A^T A) A^T b, (A^T A)^2 A^T b, \dots, (A^T A)^{k-1} A^T b \right\}.
\end{aligned} \tag{18}$$

We insert the SVD of A into the expansion

$$\mathcal{K} = \text{span} \{ V \Sigma U^T b, V \Sigma^3 U^T b, \dots, V \Sigma^{2k-1} U^T b \}. \tag{19}$$

The Krylov vectors for LSMR are $V \Sigma^{2i-1} U^T b$, $i = 1, 2, \dots, k$. The diagonal elements of Σ and the coefficients in $U^T b$ decay, due to the discrete Picard condition. So, the Krylov subspace

\mathcal{K} can be considered as an approximation to the subspace spanned by the first right singular vectors. Hence, x_k can be considered as an approximation to the TSVD solution.

From another perspective, LSMR is equivalent to the MINRES method applied to the normal equation. The regularizing properties of MINRES have been analysed in [14, 15]. So, LSMR has the regularizing properties apparently.

3. Choosing the Regularization Parameter

A reliable, efficient, and robust method for choosing the regularization parameter has not yet been found. Many techniques have been proposed to choose a suitable regularization parameter that works well under certain conditions. New methods are being developed too.

Several well-known methods, including the DP [19], the L -curve [20], the generalized cross validation (GCV) [21], and the NCP method, are described in Hansen's book [6], and a comparison of these methods is involved. DP is often used to choose the regularization parameter of Krylov subspace methods. When $\|x_k\|$ and $\|r_k\|$ show monotonic behavior, the discrete L -curve criterion can be utilized. Hnětynková et al. recently proposed to use the information from the Golub-Kahan iterative bidiagonalization to estimate the norm of the noise, which can be used to construct efficient stopping criteria [22]. For more information about choosing the regularization parameter, we suggest [23] and references therein.

Two frequently used methods are utilized to choose the regularization parameter in this paper: the DP and the discrete L -curve criterion.

3.1. The Discrepancy Principle. One simple and frequently used method for choosing the regularization parameter is the DP ([14, Chapter 3.3], [6, Chapter 5.2]). It suggests that we should choose parameter k such that the residual norm equals the discrepancy, which is the norm of the noise e . Generally, we terminate the iteration as soon as we have a solution x_k , such that

$$\|b - Ax_k\| \leq \eta, \quad \|b - Ax_{k-1}\| > \eta, \tag{20}$$

where η is the norm of e .

The main disadvantage of the DP method is that η is often not available in advance. Because the choice of regularization parameter is sensitive to the accuracy of the estimate for η , it is not easy to get a good parameter k if we use an estimate of the η .

3.2. The Discrete L -Curve Criterion. The discrete L -curve criterion is more complicated than DP. The L -curve tries to get the regularization parameter by the analysis of the norm of regularized solution $\|x_k\|$ and corresponding residual norm $\|b - Ax_k\|$. The L -curve is a log-log plot of $\|x_k\|$ versus $\|b - Ax_k\|$, and it has two different parts, an approximately horizontal part and an approximately vertical part.

(1) When k is small,

$$\begin{aligned} \|x_k\| &\approx \|x_{\text{true}}\| \text{ approximates a constant,} \\ \|b - Ax_k\| &\text{ decreases as } k \text{ increases.} \end{aligned} \tag{21}$$

(2) When k is large enough, the solution is dominated by the error caused by noise,

$$\begin{aligned} \|x_k\| &\text{ increases as } k \text{ increases,} \\ \|b - Ax_k\| &\approx \|e\| \text{ approximates a constant.} \end{aligned} \tag{22}$$

This also gives an explanation why DP is effective to choose the regularization parameter.

The effective regularization parameter can be determined by the ‘‘corner’’ that separates the two parts. The corner is the point with maximum curvature for continuous L -curves. It is similar to determine the corner for discrete L -curves. However, it is not straightforward to find the corner, because the discrete L -curve may have many local corners. In this paper, we adopt the adaptive pruning algorithm in [24] for the discrete L -curve criterion, which uses a sequence of pruned L -curves at different scales to get the global corner. The MATLAB implementations of this algorithm are available at the website (<http://www.mathworks.com/matlabcentral/fileexchange/52-regtools/content/regu/corner.m>).

The discrete L -curve criterion does not need information of the norm of noise, which is an advantage in comparison with the DP. The L -curve also has its limitations [25–27]: it will fail when the solution is very smooth; the regularization parameter computed by the L -curve may not behave consistently with the problem size increasing.

The discrete L -curve criterion has a requirement that the norm of x_k increases monotonically as k increases and the norm of r_k decreases monotonically as k increases. CGLS and LSQR are mathematically equivalent, and both of them can use the L -curve criterion as a stopping rule. The monotonicities of $\|r_k\|$ in CGLS and LSQR are apparent. Starting with the zero vector, the monotonicities of $\|x_k\|$ in CGLS and LSQR have been proved in [3, Theorem 6.3.1] and [28, Theorem 3.3.1]. The monotonicities of $\|r_k\|$ and $\|x_k\|$ in LSMR have been proved in [28, Theorem 3.3.11 and Theorem 3.3.6] accordingly. So, L -curve is allowed to choose the regularization parameter when LSMR is used as an iterative regularization method.

In practical implementations, we need stopping criterions to terminate the iteration (after the corner appears). Then, we can get the regularization parameter using the information we have. The iteration for choosing the regularization parameter needs effective stopping criterion. The frequently-used stopping criterion $\|r_k\| < \text{tol}$ ($0 < \text{tol} < 1$) is not a good one. The iteration may be terminated before the ‘‘corner’’ of L -curve appears for a bigger tol, while the number of iterations may be very large for a smaller tol because of the ill-condition of A . We choose

$$\|x_k\| > 2 \|b\|, \quad \|x_{k-1}\| \leq 2 \|b\| \tag{23}$$

as a stopping criterion thanks to the monotonicity of $\|x_k\|$. This stopping criterion is used with other stopping criterions together:

$$\|r_k\| < \text{tol} \quad (0 < \text{tol} < 1), \quad \text{for a small tol,} \tag{24}$$

and $k \geq \text{Maxitn}$, where Maxitn is an explicit limit on the iteration number.

4. Numerical Experiments

In this section we present some numerical examples to compare the IDR(s) and LSMR methods with the LSQR method. All examples are problems of the restoration of blurred and noisy images. The blurring operator A is constructed by *blur.m* in regularization tools package by Hansen [29], which is a block Toeplitz with Toeplitz block matrix that models the blurring of an image by a Gaussian PSF. This section includes two subsections: the first part is the numerical examples for comparison of the IDR(s) method and LSQR method where the DP is used to terminate the iteration; the second part is the numerical examples for comparison of the LSMR method and the LSQR method where the regularization parameters are chosen by the discrete L -curve criterion. The relative error norm (REN) is defined by

$$\text{REN} = \frac{\|x_{\text{true}} - x_k\|_2}{\|x_{\text{true}}\|_2}. \tag{25}$$

All experiments presented in this section are implemented on a computer with an Intel(R) Pentium(R) D CPU (3.00 GHz) and 1 GB of RAM using MATLAB R2010a.

4.1. The Comparison of IDR(s) and LSQR. In this section, the IDR(s)-biortho method is used due to its stability and accuracy.

Example 1. The example is the restoration of a noisy and blurred image ‘‘Lena.’’ The true image is shown in Figure 1(a). The image has 128×128 pixels. The parameters of *blur.m* are set as the bandwidth $\text{band} = 10$, $\sigma = 1$. The parameter σ controls the width of the Gaussian PSF, and the larger the σ is, the more ill-conditioned the problem is. Different Gaussian white noises with noise levels $\delta = \eta/\|b\| = 10^{-2}, 10^{-3}, 10^{-4}$ are added to the noise-free image in our experiments.

The IDR(s) method is applied to solve this problem, while the popular iterative regularization method LSQR is used for comparison. The parameter s in IDR(s) is chosen as $s = 1, 2, 4$, and 8 . The DP method is used to terminate the iteration.

The cost for each IDR(s) step is dominated by one matrix-vector product, while LSQR needs two matrix-vector products. Actually, IDS(s) need more memory and computation as s increases. More information about the cost of IDR(s) can be found in [12].

The data in Table 1 indicates that IDR(s) can produce a satisfying solution with much less iterations than LSQR (‘‘Iter’’ is the number of iterations and ‘‘MV’’ is the number of matrix-vector products). Moreover, the parameter s has

TABLE 1: Image deblurring using LSQR and IDR(s) for “Lena” ($\sigma = 1$).

Method		LSQR	IDR(1)	IDR(2)	IDR(4)	IDR(8)
$\delta = 10^{-2}$	Iter	8	4	6	5	9
	MV	16	4	6	5	9
	REN	0.0625	0.0705	0.1352	0.0799	0.1832
$\delta = 10^{-3}$	Iter	32	10	10	9	13
	MV	64	10	10	9	13
	REN	0.0347	0.0323	0.0314	0.0313	0.0319
$\delta = 10^{-4}$	Iter	145	32	33	30	35
	MV	290	32	33	30	35
	REN	0.0170	0.0135	0.0175	0.0140	0.0177

TABLE 2: Image deblurring using LSQR and IDR(s) for “Lena” ($\sigma = 1.5$).

	$\delta = 10^{-2}$			$\delta = 10^{-3}$			$\delta = 10^{-4}$		
	Iter	MV	REN	Iter	MV	REN	Iter	MV	REN
LSQR	12	24	0.0919	58	116	0.0670	329	658	0.0486
IDR(s)	6	6	0.1344	36	36	0.7551	40	40	0.0488

a small influence on the convergence and the number of iterations. This suggests that we should use IDR(1) to solve the problem rather than IDR(s) ($s > 1$) for saving storage and computational load.

However, IDR(s) does not give a desirable solution when the noise level is high. Observing the relative error norm when $\delta = 10^{-2}$, IDR(s) cannot give a good enough solution compared to LSQR. The performance of IDR(s) is worse if the noise level is higher. Another shortcoming of IDR(s) is that it cannot obtain a desirable solution or even cannot converge for very ill-condition problems which can be obtained with a larger σ . The data in Table 2 shows the performance of LSQR and IDR(1) for Example 1 with $\sigma = 1.5$ (the condition number of A is about 10^9). Although IDR(s) can obtain the solution much faster than LSQR, we should use these methods carefully. In our experiments, IDR(s) can always give a satisfactory solution when A is not very ill-conditioned and the noise level is low.

Figure 1 shows the true image and blurred and noisy image with $\delta = 10^{-3}$, $\sigma = 1$ and corresponding images restored by LSQR and IDR(1).

4.2. The Comparison of LSMR and LSQR

Example 2. The true image is constructed by MATLAB command `phantom(m)` which can create a head phantom image with $m \times m$ pixels. In this experiment, we set the size $m = 64$ and the bandwidth band = 5, $\sigma = 2$. L -curve is fit for choosing the regularization parameter in this situation. The Gaussian white noise is added to the true image with noise level $\delta = \eta/\|b\| = 10^{-2}$. The true image is Figure 3(a), and the blurred and noisy image is Figure 3(b).

Example 3. The true image is obtained from regularization tools package by Hansen [29]. The remaining parameters are the same as the parameters of Example 2. The true image and

the blurred and noisy image are shown in Figures 5(a) and 5(b).

The LSMR method is applied to solve this problem, while the LSQR method is used for comparison. The discrete L -curve criterion is employed to choose the regularization parameter. We generate 100 instances of Gaussian white noise with noise level $\delta = 10^{-2}$. For each instance, we use the discrete L -curve criterion to get the regularization parameters of LSMR and LSQR. Then, we obtain the solutions by running corresponding LSMR algorithm and LSQR algorithm. The comparison of the relative error norm is shown in Figure 2 (Example 2) and Figure 4 (Example 3). It is evident that LSMR more likely produces a better solution than LSQR. Figure 3 shows one instance of Example 2 including the true image, blurred and noisy image, and images restored by LSQR and LSMR. Similarly, Figure 5 shows one instance of Example 3. Furthermore, if we compare the optimal solution produced by LSMR and LSQR, the corresponding REN of the solution produced by LSMR is usually the smaller one.

The L -curve works well for large noise level. Actually, we found that the regularization parameter obtained from the L -curve is almost optimal when the level of noise is 5×10^{-2} . But the L -curve may fail or need a large number of iterations to find the corner for small noise level.

Besides, we have done the comparison of LSMR and LSQR with the regularization parameter chosen by DP. LSMR method can get nearly the same REN as LSQR method, while LSQR method could be terminated somewhat sooner. The memory and computational costs of every step in LSMR are also slightly more expensive than that in LSQR [11].

5. Conclusion

In this paper, we briefly introduced the IDR(s) method and the LSMR method, and then we applied the IDR(s) and



FIGURE 1: (a) True image, (b) blurred and noisy image with $\sigma = 1$, (c) restored image with LSQR, and (d) restored image with IDR(1).

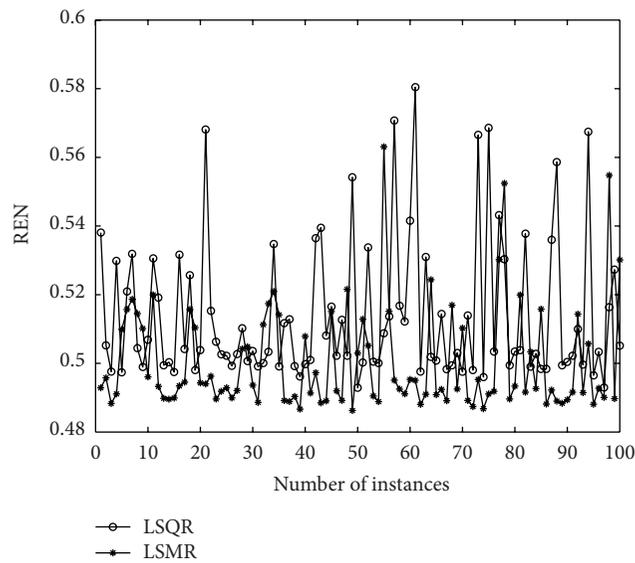


FIGURE 2: Comparison of the REN produced by LSQR and LSMR.

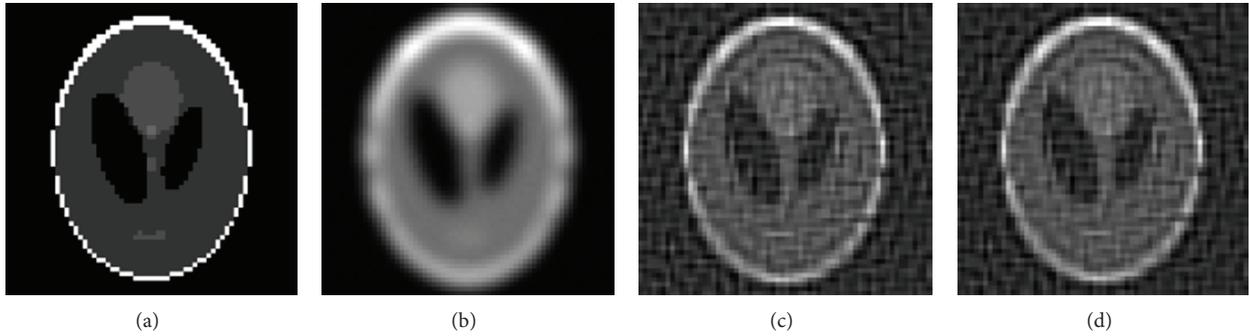


FIGURE 3: (a) True image, (b) blurred and noisy image, (c) restored image with LSQR, Iter = 104, and REN = 0.5004, and (d) restored image with LSMR, Iter = 128, and REN = 0.4919.

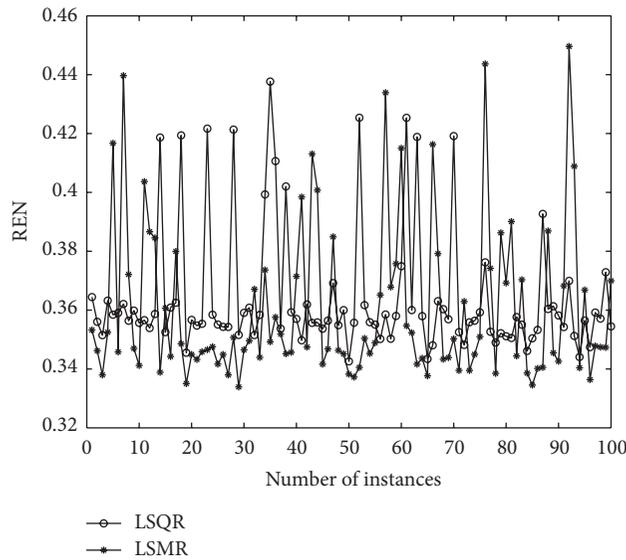


FIGURE 4: Comparison of the REN produced by LSQR and LSMR.

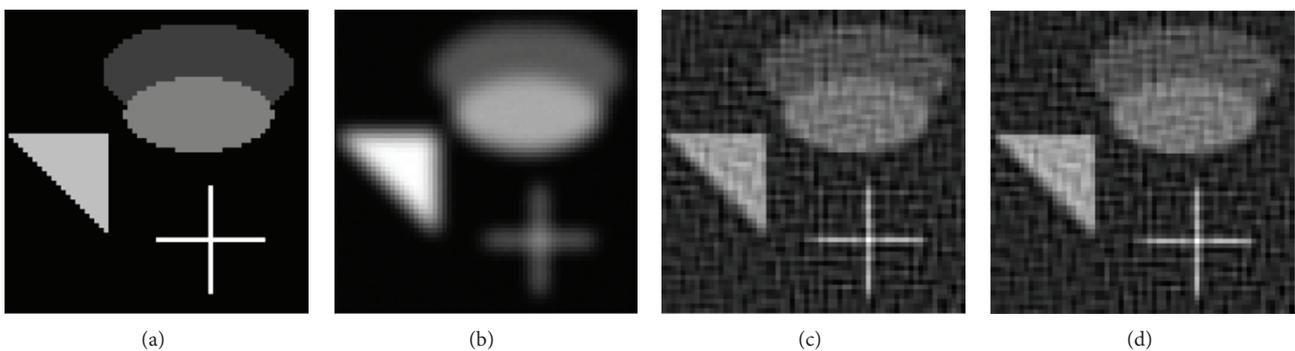


FIGURE 5: (a) True image, (b) blurred and noisy image, (c) restored image with LSQR, Iter = 95, and REN = 0.3519, and (d) restored image with LSMR, Iter = 129, and REN = 0.3417.

LSMR methods to solve image deblurring problems. These methods show some superiorities when compared to the classic iterative regularization method LSQR.

When DP is used to terminate the iteration, IDR(s) can give a satisfactory solution with a low computational cost when A is not very ill-condition and the noise level is low. The parameter s has little influence on the performance of IDR(s),

so we can use IDR(1) for saving storage and computational cost.

LSMR performs as well as LSQR but needs more computational cost when DP is used to terminate the iteration. When the L -curve method is utilized to choose the regularization parameter, LSMR more likely produces a more accurate solution than LSQR.

Acknowledgments

This research is supported by 973 Program (2013CB329404), NSFC (61170311), Chinese Universities Specialized Research Fund for the Doctoral Program (20110185110020), and Sichuan Province Sci. & Tech. Research Project (2012GZX0080). The authors would like to express their thankfulness to the referees and the editor for their much helpful suggestions for revising this paper.

References

- [1] P. C. Hansen, J. G. Nagy, and D. P. O’Leary, *Deblurring Images: Matrices, Spectra and Filtering*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2006.
- [2] R. J. Hanson, “A numerical method for solving Fredholm integral equations of the first kind using singular values,” *SIAM Journal on Numerical Analysis*, vol. 8, pp. 616–622, 1971.
- [3] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 1998.
- [4] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2nd edition, 2003.
- [5] R. Plato, “Optimal algorithms for linear ill-posed problems yield regularization methods,” *Numerical Functional Analysis and Optimization*, vol. 11, no. 1-2, pp. 111–118, 1990.
- [6] P. C. Hansen, *Discrete Inverse Problems: Insight and Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2010.
- [7] D. Calvetti, B. Lewis, and L. Reichel, “On the regularizing properties of the GMRES method,” *Numerische Mathematik*, vol. 91, no. 4, pp. 605–625, 2002.
- [8] D. Calvetti, B. Lewis, and L. Reichel, “Krylov subspace iterative methods for nonsymmetric discrete ill-posed problems in image restoration,” in *Advanced Signal Processing: Algorithms, Architectures and Implementations XI*, vol. 4474 of *Proceedings of SPIE*, pp. 224–233, The International Society for Optical Engineering, Bellingham, Wash, USA, August 2001.
- [9] P. Brianzi, P. Favati, O. Menchi, and F. Romani, “A framework for studying the regularizing properties of Krylov subspace methods,” *Inverse Problems*, vol. 22, no. 3, pp. 1007–1021, 2006.
- [10] P. Sonneveld and M. B. van Gijzen, “IDR(s): a family of simple and fast algorithms for solving large nonsymmetric systems of linear equations,” *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 1035–1062, 2008.
- [11] D. C.-L. Fong and M. Saunders, “LSMR: an iterative algorithm for sparse least-squares problems,” *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2950–2971, 2011.
- [12] M. B. van Gijzen and P. Sonneveld, “An elegant IDR(s) variant that efficiently exploits bi-orthogonality properties,” Tech. Rep. 08-21, Department of Applied Mathematical Analysis, Delft University of Technology, Delft, The Netherlands, 2008.
- [13] V. Simoncini and D. B. Szyld, “Interpreting IDR as a Petrov-Galerkin method,” *SIAM Journal on Scientific Computing*, vol. 32, no. 4, pp. 1898–1912, 2010.
- [14] M. Hanke, *Conjugate Gradient Type Methods for Ill-Posed Problems*, vol. 327 of *Pitman Research Notes in Mathematics Series*, Longman, Harlow, UK, 1995.
- [15] T. K. Jensen and P. C. Hansen, “Iterative regularization with minimum-residual methods,” *BIT*, vol. 47, no. 1, pp. 103–120, 2007.
- [16] P. Sonneveld, “On the convergence behaviour of IDR(s),” Tech. Rep. 10-08, Department of Applied Mathematical Analysis, Delft University of Technology, Delft, The Netherlands, 2010.
- [17] M. Donatelli and S. Serra-Capizzano, “On the regularizing power of multigrid-type algorithms,” *SIAM Journal on Scientific Computing*, vol. 27, no. 6, pp. 2053–2076, 2006.
- [18] M. Donatelli and S. Serra-Capizzano, “Filter factor analysis of an iterative multilevel regularizing method,” *Electronic Transactions on Numerical Analysis*, vol. 29, pp. 163–177, 2007/08.
- [19] V. A. Morozov, “On the solution of functional equations by the method of regularization,” *Soviet Mathematics. Doklady*, vol. 7, pp. 414–417, 1966.
- [20] P. C. Hansen, “Analysis of discrete ill-posed problems by means of the L -curve,” *SIAM Review*, vol. 34, pp. 658–672, 1992.
- [21] G. Wahba, “Practical approximate solutions to linear operator equations when the data are noisy,” *SIAM Journal on Numerical Analysis*, vol. 14, no. 4, pp. 651–667, 1977.
- [22] I. Hnětynková, M. Plešinger, and Z. Strakoš, “The regularizing effect of the Golub-Kahan iterative bidiagonalization and revealing the noise level in the data,” *BIT*, vol. 49, no. 4, pp. 669–696, 2009.
- [23] L. Reichel and G. Rodriguez, “Old and new parameter choice rules for discrete ill-posed problems,” *Numerical Algorithms*, vol. 63, no. 1, pp. 65–87, 2013.
- [24] P. C. Hansen, T. K. Jensen, and G. Rodriguez, “An adaptive pruning algorithm for the discrete L -curve criterion,” *Journal of Computational and Applied Mathematics*, vol. 198, no. 2, pp. 483–492, 2007.
- [25] P. C. Hansen, “The L -curve and its use in the numerical treatment of inverse problems,” in *Computational Inverse Problems in Electrocardiology*, P. Johnston, Ed., pp. 119–142, WIT Press, Southampton, Ceremonial, 2001.
- [26] C. R. Vogel, “Non-convergence of the L -curve regularization parameter selection method,” *Inverse Problems*, vol. 12, no. 4, pp. 535–547, 1996.
- [27] M. Hanke, “Limitations of the L -curve method in ill-posed problems,” *BIT*, vol. 36, no. 2, pp. 287–301, 1996.
- [28] D. C. L. Fong, *Minimum-residual methods for sparse least-squares using golub-kahan bidiagonalization [Ph.D. thesis]*, Stanford University, Stanford, Calif, USA, 2011.
- [29] P. C. Hansen, “Regularization tools: a Matlab package for analysis and solution of discrete ill-posed problems,” *Numerical Algorithms*, vol. 6, no. 1-2, pp. 1–35, 1994.