Information Sciences 271 (2014) 179–195

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Two soft-thresholding based iterative algorithms for image deblurring ☆

Jie Huang [a], Ting-Zhu Huang [a,*], Xi-Le Zhao [a], Zong-Ben Xu [b], Xiao-Guang Lv [a]

[a] School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China
[b] Institute of Information and System Sciences, Xi'an Jiaotong University, Xi'an, Shaan'xi 710049, China

ARTICLE INFO

Article history:
Received 4 December 2011
Received in revised form 22 August 2013
Accepted 10 February 2014
Available online 20 February 2014

Keywords:
Image deblurring
Regularization
Iterative algorithms
Soft-thresholding
Filter

ABSTRACT

Iterative regularization algorithms, such as the conjugate gradient algorithm for least squares problems (CGLS) and the modified residual norm steepest descent (MRNSD) algorithm, are popular tools for solving large-scale linear systems arising from image deblurring problems. These algorithms, however, are hindered by a semi-convergence behavior, in that the quality of the computed solution first increases and then decreases. In this paper, in order to overcome the semi-convergence behavior, we propose two iterative algorithms based on soft-thresholding for image deblurring problems. One of them combines CGLS with a denoising technique like soft-thresholding at each iteration and another combines MRNSD with soft-thresholding in a similar way. We prove the convergence of MRNSD and soft-thresholding based algorithm. Numerical results show that the proposed algorithms overcome the semi-convergence behavior and the restoration results are slightly better than those of CGLS and MRNSD with their optimal stopping iterations.

1. Introduction

Image deblurring is a process of reconstructing an approximation of an image from an observed but degraded image, which is often modeled as the solution of the linear operator equation

$$Kf + e = g, \tag{1}$$

where $f \in \mathbb{R}^{n^2}$ is a vector representing the true $n \times n$ image we aim to recover, $e \in \mathbb{R}^{n^2}$ represents random noise, $g \in \mathbb{R}^{n^2}$ stands for the observed distorted image, and $K \in \mathbb{R}^{n^2 \times n^2}$ is a blurring matrix with special structures [30,40]. Such problems arise in applications such as astronomy, medical imaging, geophysical applications and many other areas [18–20,23,28,32,39,48,54,55]. Typically Eq. (1) comes from the discretization of an ill-posed continuous model, that is, the matrix K is ill-conditioned and noise in the data may give rise to significant errors in the computed approximate solution. As a result, directly solving the equation $Kf = g$ does not yield an accurate and stable approximate solution and it is necessary to resort to a regularization method. In the literature many regularization techniques can be found, such as Tikhonov regularization [52], truncated singular value decomposition (TSVD), truncated iterative algorithms (e.g. steepest descent and conjugate gradient (CG) iterations) [18,54] and hybrid approaches [8,33,35,36,45].

☆ This research is supported by 973 Program (2013CB329404), NSFC (61370147,61170311), Sichuan Province Sci. & Tech. Research Project (2012GZX0080), and the Fundamental Research Funds for the Central Universities (ZYGX2013J106).

* Corresponding author. Tel.: +86 13688078826.
E-mail addresses: jiehuangjh86@gmail.com (J. Huang), tingzhuhuang@126.com (T.-Z. Huang).

http://dx.doi.org/10.1016/j.ins.2014.02.089
0020-0255/

A suitable value of the regularization parameter is important when incorporated with a regularization approach. For Tikhonov regularization, various parameter-choice techniques can be used, such as the discrepancy principle [41], the L-curve [18,28], generalized cross validation (GCV) [21], the weighted-GCV (W-GCV) [14] and the fixed point algorithm (FP) [3]. These parameter-selection methods have both advantages and disadvantages and it is nontrivial to choose an 'optimal' regularization parameter [28,54].

Iterative regularization algorithms like CG and steepest descent can be favorable alternatives to Tikhonov regularization [10,18,42,49,50,54]. They access the coefficient matrix K only via matrix–vector multiplication with K and/or K^T. It is known that applying these iterative regularization algorithms to solving the linear system $Kf = g$ is often hindered by a *semi-convergence* behavior. That is, the first few iterations produce regularization solutions and, after some 'optimal' iteration, the approximate solutions converge to some other undesired vector. This undesired vector is contaminated by errors and is therefore a poor approximation. In other words, an imprecise estimate of the termination iteration is likely to result in a poor approximate solution and hence it becomes crucial to decide when to stop the iterations. Parameter-selection methods such as the discrepancy principle and the L-curve can be used to choose such a proper termination, but it is also nontrivial as in the case for Tikhonov regularization.

The difficulty in determining the stopping iteration number of iterative regularization algorithms can be partially alleviated by employing hybrid approaches [4,9,12,14,26,27]. Lanczos-hybrid type approaches combine an iterative Lanczos bidiagonalization algorithm with a regularization algorithm such as Tikhonov regularization and TSVD at each iteration. Regularization is therefore achieved by Lanczos bidiagonalization filtering and appropriately selecting a regularization parameter at each iteration. Recently, parameter-selection methods such as W-GCV and FP have been studied for the Lanczos-Tikhonov hybrid approach, see [4,14]. The W-GCV based Lanczos-Tikhonov hybrid approach makes the solution be less sensitive to iteration number. However, it is characterized by the semi-convergence property as the iteration proceeds. The GKB-FP algorithm in [4] combines a partial Golub-Kahan bidiagonalization (GKB) iteration with Tikhonov regularization in the generated Krylov subspace and the regularization parameter for the projected problem is chosen by FP. In some cases, GKB-FP yields more accurate solutions than W-GCV based Lanczos-Tikhonov. But in some cases the two methods perform comparably.

In this paper, we propose two iterative algorithms based on iterative regularization algorithms and soft-thresholding for image deblurring. Recall that it is the noise in the right-hand side g and its propagation with iterations that are the main reason of the semi-convergence behavior of iterative regularization algorithms like CGLS and MRNSD. Therefore, we propose to combine the iterative regularization algorithms with a noise reduction technique like soft-thresholding at each iteration to suppress the propagation of noise and thus overcome the semi-convergence behavior. One of the proposed algorithms combines CGLS with soft-thresholding at each iteration and another combines MRNSD with soft-thresholding in a similar way. The resulting algorithms are stable and very effective in practical applications. We prove the convergence of MRNSD and soft-thresholding based algorithm. Numerical experiments show that the proposed algorithms overcome the semi-convergence behavior and the restoration results are slightly better than those of CGLS and MRNSD with their optimal iterations.

The outline of the paper is as follows. In Section 2.1 we first review the classical CGLS iteration and then give soft-thresholding and CGLS based iterative algorithm, called CGLS-like. A widely used satellite test problem is considered to demonstrate the utility of CGLS-like compared with CGLS. In Section 2.2 we present an MRNSD-like algorithm and prove its convergence. Several numerical examples are given in Section 3 to illustrate the efficacy of the proposed algorithms. Finally Section 4 gives some conclusions.

2. Soft-thresholding based iterative algorithms

We are interested in using iterative regularization algorithms such as CG and the steepest descent to solve the large-scale linear systems $Kf = g$. Since our problems are usually not symmetric, we solve the normal equations $K^TKf = K^Tg$ using the conjugate gradient algorithm for least squares problems (CGLS) [7] and the modified residual norm steepest descent (MRNSD) algorithm [44], respectively. Due to the ill-conditioning of K and the presence of noise in the right-hand side g, these algorithms are characterized by the semi-convergence property: the quality of the computed solution first increases but then after some optimal iteration begins to decrease. In this case, it is hoped to overcome the semi-convergence behavior by combining the iterative algorithms with a denoising technique like soft-thresholding at each iteration, see also [1,24]. Recall that the soft-thresholding operator $\mathbf{S}_\mu$ is defined component-wise by

$$\left(\mathbf{S}_\mu(x)\right)_i = \operatorname{sgn}(x_i)\cdot\max\left\{|x_i| - \frac{\mu}{2}, 0\right\}, \quad i = 1,\ldots,N,\ x \in \mathbb{R}^N. \tag{2}$$

We propose two algorithms based on soft-thresholding in the following.

2.1. CGLS-like algorithm

We first review the classical CGLS iteration [7,25,28]. Express the $(k+1)$-th approximate solution as $f_{k+1} = f_k + \alpha_k p_k$, then the residual vector, i.e. $r_{k+1} = g - Kf_{k+1}$, satisfies $r_{k+1} = r_k - \alpha_k K p_k$, and

$$s_{k+1} = K^T r_{k+1} = K^T r_k - \alpha_k K^T K p_k = s_k - \alpha_k K^T K p_k.$$

Then enforcing s_{k+1} orthogonal to s_k leads to

$$\alpha_k = \frac{\langle s_k, s_k \rangle}{\langle K^T K p_k, s_k \rangle}.$$

Assume the next search direction p_{k+1}, i.e. $p_{k+1} = s_{k+1} + \beta_k p_k$, is orthogonal to $K^T K p_k$. Then

$$\langle K^T K p_k, s_k \rangle = \langle K^T K p_k, p_k - \beta_{k-1} p_{k-1} \rangle = \langle K^T K p_k, p_k \rangle.$$

Thus,

$$\alpha_k = \frac{\langle s_k, s_k \rangle}{\langle K^T K p_k, p_k \rangle} = \frac{\langle s_k, s_k \rangle}{\langle K p_k, K p_k \rangle}.$$

In addition, $\langle p_{k+1}, K^T K p_k \rangle = \langle s_{k+1} + \beta_k p_k, K^T K p_k \rangle = 0$ gives

$$\beta_k = -\frac{\langle s_{k+1}, K^T K p_k \rangle}{\langle K^T K p_k, p_k \rangle} = -\frac{\langle s_{k+1}, -\frac{1}{\alpha_k}(s_{k+1} - s_k) \rangle}{\frac{1}{\alpha_k} \langle s_k, s_k \rangle} = \frac{\langle s_{k+1}, s_{k+1} \rangle}{\langle s_k, s_k \rangle}.$$

These relations give the following algorithm.

Algorithm 1. CGLS

Given: Matrix K, initial guess f_0 and right-hand side g.
Set $f = f_0, r = g - Kf, s = K^T r$, and $\gamma = s^T s$,
for $k = 1, 2, \ldots$, do
  if $(k == 1)$, set $p = s$,
  otherwise compute $\beta = \gamma / \gamma_{old}$ and $p = s + \beta p$,
  $q = Kp$,
  $\alpha = \gamma / q^T q$,
  $f = f + \alpha p$,
  $r = r - \alpha q$,
  $s = K^T r$,
  $\gamma_{old} = \gamma$,
  $\gamma = s^T s$,
  determine if stopping criteria are satisfied
end for

In [31] it was shown that as k increases, $\|r_k\|_2 = \|g - Kf_k\|_2$ will decrease monotonically, $\|f_k\|_2$ will increase monotonically if $f_0 = \mathbf{0}$, but $\|s_k\|_2 = \|K^T r_k\|_2$ will often exhibit large oscillations when K is very ill-conditioned. Moreover, the $(k+1)$-th iterate f_{k+1} minimizes the residual norm $\|g - Kf\|_2$ among all vectors f from the Krylov subspace

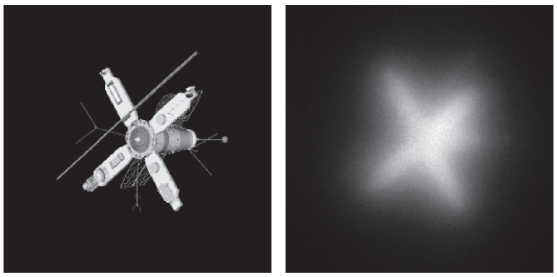
$$\mathcal{K}_{k+1}\left(K^T K, K^T g\right) = \text{span}\left\{K^T g, K^T K K^T g, \ldots, (K^T K)^k K^T g\right\}.$$

Assume f_k and f_{true} are the approximate solution at k-th iteration and the true solution, respectively. Define *relative error* by $\|f_{\text{true}} - f_k\|_2 / \|f_{\text{true}}\|_2$. Then from this definition we see that the lower the relative error is, the closer the computed solution is to the true solution and therefore the higher quality it has. For discrete ill-posed problems, CGLS often exhibits a semi-convergence behavior in terms of relative error because of the presence of noise, and only an early termination of iterations provides an accurate solution. For example, we consider the *satellite* test problem, which has been widely used in the literature for testing algorithms for image deblurring problems [47]. We obtain the test data from image restoration package *RestoreTools*[1] by Nagy and several of his students [43]. The true and blurred noisy images are shown in Fig. 1. The relative error history of CGLS is shown in Fig. 2(a). It is clear that the relative error first decreases and then after about 60 iterations begins to increase. This phenomenon is referred to as semi-convergence.

Here, we propose a CGLS-like algorithm to overcome this semi-convergence behavior by suppressing the propagation of noise at each iteration round of CGLS. Based on the CGLS iteration, we recursively derive that

$$f_{k+1} = f_k + \alpha_k p_k = f_{k-1} + \alpha_{k-1} p_{k-1} + \alpha_k p_k = \cdots = f_0 + \sum_{j=0}^{k} \alpha_j p_j = f_0 + \sum_{j=0}^{k} \frac{\langle s_j, s_j \rangle}{\langle K p_j, K p_j \rangle} p_j,$$

[1] http://www.mathcs.emory.edu/~nagy/RestoreTools/index.html.



(a) Original (b) Blurred and noisy

Fig. 1. Satellite test data.

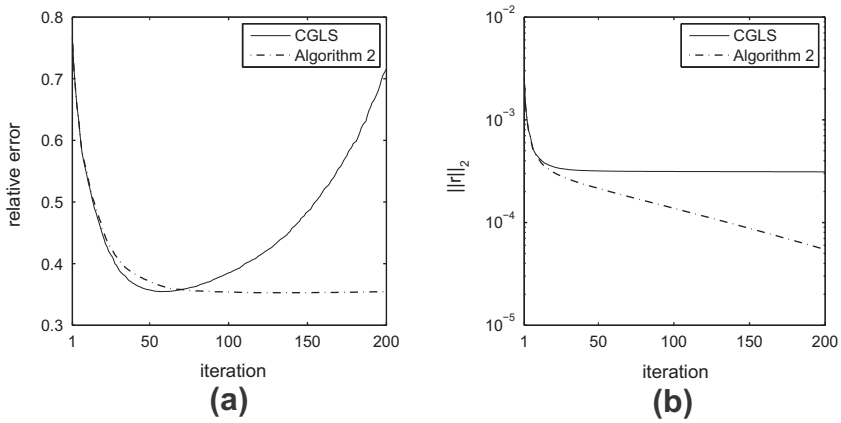


Fig. 2. Convergence behavior of CGLS and Algorithm 2 on satellite test problem. Initial guess for both algorithms is the zero vector.

and in a similar manner,

$$\begin{aligned} r_{k+1} &= g - Kf_0 - \sum_{j=0}^{k} \frac{\langle s_j, s_j\rangle}{\langle Kp_j, Kp_j\rangle} Kp_j, \\ s_{k+1} &= K^T g - K^T K f_0 - \sum_{j=0}^{k} \frac{\langle s_j, s_j\rangle}{\langle Kp_j, Kp_j\rangle} K^T K p_j, \\ p_{k+1} &= s_{k+1} + \sum_{j=0}^{k} \left(\prod_{i=j}^{k} \beta_i \right) s_j = s_{k+1} + \sum_{j=0}^{k} \left(\prod_{i=j}^{k} \frac{\langle s_{j+1}, s_{j+1}\rangle}{\langle s_j, s_j\rangle} \right) s_j. \end{aligned}$$

Notice from components of $f_{k+1}, r_{k+1}, s_{k+1}$ and p_{k+1} that the residual vector r_{k+1} contains directly the noisy right-hand side g, or in other words, the noise in g propagates mainly via the residual vectors, since matrix–vector multiplications Kp_j have suppressed noise in p_j [17,16]. Therefore, we propose to filter the residual vector by a denoising technique like soft-thresholding at each CGLS iteration to suppress noise propagation. To make this more clear, we consider the iteration:

$$\begin{aligned} f_{k+1} &= f_k + \alpha_k p_k, \\ \tilde{r}_{k+1} &= r_k - \alpha_k K p_k, \\ r_{k+1} &= \mathbf{S}_\mu(\tilde{r}_{k+1}), \\ \tilde{s}_{k+1} &= s_k - \alpha_k K^T K p_k, \\ s_{k+1} &= K^T r_{k+1}, \\ p_{k+1} &= s_{k+1} + \beta_k p_k. \end{aligned}$$

Similarly to CGLS, we enforce the following two essential properties:

$$\langle \tilde{s}_{k+1}, s_k\rangle = 0 \qquad \text{and} \qquad \langle p_{k+1}, K^T K p_k\rangle = 0.$$

A direct consequence of the first inner product is

$$\langle \tilde{s}_{k+1}, s_k \rangle = \langle s_k - \alpha_k K^T K p_k, s_k \rangle = 0,$$

and thus

$$\alpha_k = \frac{\langle s_k, s_k \rangle}{\langle K^T K p_k, s_k \rangle}.$$

From the assumption that p_{k+1} is orthogonal to $K^T K p_k$, one obtains that $\langle p_k, K^T K p_{k-1} \rangle = 0$. Substituting $s_k = p_k - \beta_{k-1} p_{k-1}$ in $\langle K^T K p_k, s_k \rangle$ gets

$$\langle K^T K p_k, s_k \rangle = \langle K^T K p_k, p_k - \beta_{k-1} p_{k-1} \rangle = \langle K^T K p_k, p_k \rangle = \langle K p_k, K p_k \rangle.$$

Therefore,

$$\alpha_k = \frac{\langle s_k, s_k \rangle}{\langle K p_k, K p_k \rangle}.$$

In addition, writing $\langle p_{k+1}, K^T K p_k \rangle = 0$, with $p_{k+1} = s_{k+1} + \beta_k p_k$, leads to

$$\langle s_{k+1} + \beta_k p_k, K^T K p_k \rangle = 0,$$

requiring

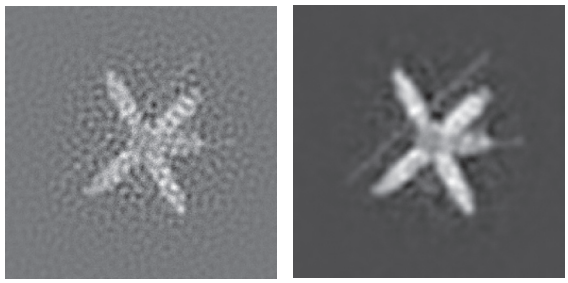
$$\beta_k = \frac{-\langle s_{k+1}, K^T K p_k \rangle}{\langle K p_k, K p_k \rangle}.$$

This yields an iteration described by the following algorithm.

Algorithm 2. CGLS-like

Given: Matrix K, initial guess f_0, right-hand side g, soft-thresholding parameter $\mu > 0$ and threshold value $\tau (0 < \tau < 1)$.
Set $f = f_0, r = g - Kf, s = K^T r, \gamma = s^T s$, and $\sigma = 1$,
for $k = 1, 2, \ldots,$ *do*
  if $(k == 1)$, set $p = s$,
  otherwise compute $\beta = \frac{-\langle s, K^T q \rangle}{\langle q, q \rangle}$ and $p = s + \beta p$,
  $q = Kp$,
  $\alpha = \gamma / q^T q$,
  $f_{old} = f$,
  $f = f_{old} + \alpha p$,
  $r = \mathbf{S}_\mu (r - \alpha q)$,
  $s = K^T r$,
  $\gamma = s^T s$,
  $\sigma = \|f_{old} - f\|_2 / \|f\|_2$,
  if $\sigma < \tau$ or $\|r\|_2 = 0$, *break*;
end for

We remark that applying the soft-thresholding operator on the residual vector at each iteration prevents simplifying β_k in Algorithm 2 to the form in CGLS, i.e. $s_{k+1}^T s_{k+1} / s_k^T s_k$. As a consequence, one additional matrix–vector multiplication is required for β_k in Algorithm 2, compared with the original CGLS algorithm. Recall that applying soft-thresholding on an N-length vector requires $\mathcal{O}(N)$ operations. This cost is clearly less than that of a matrix–vector multiplication involving an $N \times N$ blurring matrix, which is $\mathcal{O}(N \log N)$. In this event, for $K \in \mathbb{R}^{n^2 \times n^2}$, the computational complexity of both Algorithms 1 and 2 at each iteration is $\mathcal{O}(n^2 \log n)$. In fact, the simple soft-thresholding operator makes iterations much stable. For example, we plot the relative error history of Algorithm 2 with $\mu = 4 \times 10^{-8}$ for the satellite test problem in Fig. 2(a). From this plot we see that as the iteration number k increases, the relative error of the approximation first decreases and then remains steady, showing that an accurate stopping iteration number for Algorithm 2 is no longer such vital. In addition, we show the ℓ_2-norm of r_k against iteration k from 1 to 200 in Fig. 2(b). We observe that numerically, the quantity $\|r_k\|_2$ of Algorithm 2 decreases monotonically and moreover, faster than that of CGLS. For further comparison, restored images of CGLS and CGLS-like after 200 iterations are provided in Fig. 3. This figure shows that after 200 iterations, the computed solution of CGLS is very poor but CGLS-like is relatively stable.



(a) CGLS (b) CGLS-like

Fig. 3. Deblurring results at 200th iteration for *satellite* test data.

We finally remark that the proposed algorithm is different from Lanczos-hybrid type approaches in overcoming semi-convergence behavior. Lanczos-hybrid type approaches first project the large-scale problem onto Krylov subspaces of small (but increasing) dimension and then apply any direct regularization algorithm to solve the projected problem. It implies that Lanczos-hybrid type approaches suppress the semi-convergence behavior by filtering the solution of the projected problem with any regularization method like Tikhonov regularization. Algorithm 2, in contrast, is designed to overcome the semi-convergence behavior by filtering the residual vector with a denoising algorithm at each iteration. Recall that it is the noise in the right-hand side g and its propagation with iterations that cause the semi-convergence behavior. Then it may be more effective to filter the residual vector which mainly bears the noise in g rather than filter the solution of the projected problem at each iteration.

2.2. MRNSD-like algorithm

Unconstrained Krylov subspace algorithms like CGLS for computing approximate solutions are fast, but they do not produce approximate solutions preserving nonnegativity. Consider minimizing the regular least squares function

$$J(f) = \frac{1}{2}\|Kf - g\|_2^2 \tag{3}$$

subject to a nonnegativity constraint $f \geqslant \mathbf{0}$. In [44] the authors provided a nonnegatively constrained minimization algorithm, which is a variant of the EMLS algorithm proposed by Kaufman [34]. They parameterized $f = e^z$. Then the constrained minimization problem as in (3) transforms into the problem of minimizing the following unconstrained function

$$\tilde{J}(z) = \frac{1}{2}\|Ke^z - g\|_2^2.$$

Differentiating using the chain rule gives that

$$\mathrm{grad}_z\, \tilde{J}(z) = F\mathrm{grad}_f J(f) = FK^T(Kf - g), \quad F = \mathrm{diag}(f).$$

Setting $\mathrm{grad}_z\tilde{J}(z) = \mathbf{0}$, the authors in [44] obtained the Karush–Kuhn–Tucker conditions for this particular constrained minimization problem in f-space. They assumed the iteration of the form:

$$f_{k+1} = f_k + \alpha_k p_k,$$

where $p_k = F_k K^T(g - Kf_k)$ and the line search parameter α_k is obtained by minimizing the residual norm $\|g - Kf\|_2$ subject to $f \geqslant \mathbf{0}$ at each iteration, i.e.

$$\alpha_k = \underset{\alpha\in\mathbb{R},\ f_k+\alpha p_k\geqslant\mathbf{0}}{\mathrm{argmin}} \|g - K(f_k + \alpha p_k)\|_2 = \underset{\alpha\in\mathbb{R},\ f_k+\alpha p_k\geqslant\mathbf{0}}{\mathrm{argmin}} \|r_k - \alpha K p_k\|_2,$$

in which we have defined the k-th residual vector $r_k = g - Kf_k$. To be more precise, let $s_k = K^T r_k$ and thus $p_k = F_k s_k$, then simple calculation gives that

$$\alpha_k = \min\left(\frac{\langle Kp_k, r_k\rangle}{\langle Kp_k, Kp_k\rangle}, \min_{(p_k)_i<0}(-(f_k)_i/(p_k)_i)\right) = \min\left(\frac{\langle F_k s_k, s_k\rangle}{\langle Kp_k, Kp_k\rangle}, \min_{(p_k)_i<0}(-(f_k)_i/(p_k)_i)\right).$$

Consequently, the $(k+1)$-th residual vector,

$$r_{k+1} = g - Kf_{k+1} = r_k - \alpha_k K p_k,$$

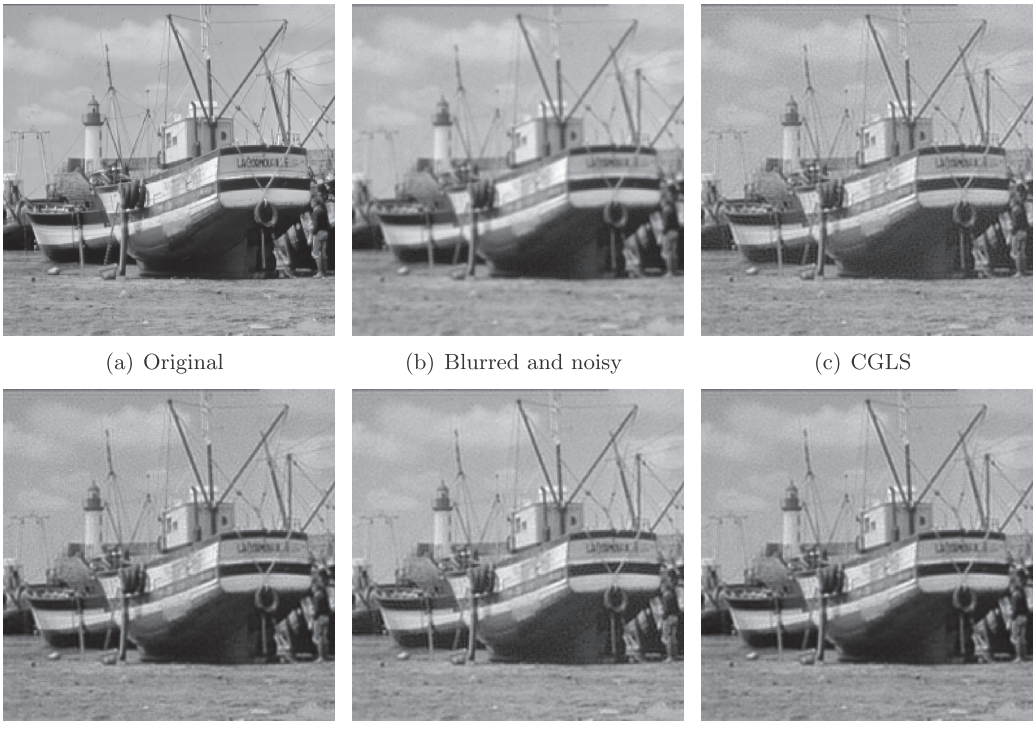
satisfies $\|r_{k+1}\|_2 \leqslant \|r_k\|_2$. This is called the modified residual norm steepest descent (MRNSD) algorithm.

Algorithm 3. MRNSD

Given: Matrix K, initial guess f_0 and right-hand side g.
Set $f = f_0,\ s = K^T(g - Kf),\ F = \mathrm{diag}(f)$, *and* $\gamma = s^T Fs$,
for $k = 1, 2, \ldots,$ *do*
$p = Fs$,
$q = Kp$,
$\alpha = \min(\gamma/q^T q, \min_{p_i<0}(-f_i/p_i))$,
$f = f + \alpha p$,
$F = \mathrm{diag}(f)$,
$z = K^T q$,
$s = s - \alpha z$,
$\gamma = s^T Fs$,
determine if stopping criteria are satisfied
end for

We note that MRNSD has been extended to solve weighted least squares problems by Bardsley and Nagy [2] and the extended algorithm can be viewed as a preconditioned version of MRNSD in which the preconditioning matrix is constructed using a priori knowledge of noise statistics. Preconditioning strategies for both algorithms have been considered as well. These algorithms can provide more accurate approximate solutions and be computationally competitive with unconstrained Krylov subspace algorithms. We remark that preconditioning techniques are beyond the scope of this study and we do not consider them here. We also remark that the authors in [37] used an active set approach to solve nonnegative least squares problems appearing in nonnegative matrix factorization, while recently the authors in [22] use the fast Nesterov's optimal gradient algorithm to solve such nonnegative least squares subproblems. Applying these algorithms to framelet-based image deblurring problems (see e.g. [6,13,51]) is currently under investigation.

It has been shown in [44] that MRNSD can be more stable than unconstrained Krylov subspace algorithms like CGLS. However, MRNSD also exhibits a semi-convergence behavior because of the presence of noise (specific examples are detailed



(a) Original (b) Blurred and noisy (c) CGLS

(d) MRNSD (e) Algorithm 2 (f) Algorithm 4

Fig. 4. Example 3.1: Original, contaminated, and restored images by different algorithms for 256×256 *boat* image convolved by a 3×3 Gaussian kernel and degraded by 2% Gaussian white noise.

in the later section). In this case, we filter the residual vector by the soft-thresholding operator at each iteration to address this undesired behavior as in Algorithm 2. To be more specific, iterate

$$\begin{aligned} f_{k+1} &= f_k + \alpha_k p_k, \\ F_{k+1} &= \mathrm{diag}(f_{k+1}), \\ \tilde{r}_{k+1} &= r_k - \alpha_k K p_k, \\ r_{k+1} &= \mathbf{S}_\mu(\tilde{r}_{k+1}), \\ s_{k+1} &= K^T r_{k+1}, \\ p_{k+1} &= F_{k+1} s_{k+1}. \end{aligned} \tag{4}$$

Although the above r_k is different from that equal to $g - Kf_k$ in MRNSD, we still determine α_k by minimizing $\|r_k - \alpha K p_k\|_2$ subject to a nonnegative solution, i.e.

$$\alpha_k = \underset{\alpha\in\mathbb{R},\ f_k+\alpha p_k \geqslant \mathbf{0}}{\mathrm{argmin}} \|r_k - \alpha K p_k\|_2 = \min\left(\frac{\langle Kp_k, r_k\rangle}{\langle Kp_k, Kp_k\rangle}, \min_{(p_k)_i<0}(-(f_k)_i/(p_k)_i)\right). \tag{5}$$

Observing from the iterate scheme (4) that $s_k = K^T r_k$ and $p_k = F_k s_k$, we derive

$$\alpha_k = \min\left(\frac{\langle F_k s_k, s_k\rangle}{\langle Kp_k, Kp_k\rangle}, \min_{(p_k)_i<0}(-(f_k)_i/(p_k)_i)\right).$$

Putting these relations together results in our algorithm in the following.

Algorithm 4. MRNSD-like

Given: Matrix K, initial guess f_0, right-hand side g, soft-thresholding parameter $\mu > 0$ and threshold value $\tau (0 < \tau < 1)$.
Set $f = f_0,\ r = g - Kf,\ s = K^T r,\ F = \mathrm{diag}(f),\ \gamma = s^T F s$, and $\sigma = 1$,
for $k = 1, 2, \ldots,$ *do*
  $p = Fs$,
  $q = Kp$,
  $\alpha = \min(\gamma/q^T q, \min_{p_i<0}(-f_i/p_i))$,
  $f_{old} = f$,
  $f = f_{old} + \alpha p$,
  $F = \mathrm{diag}(f)$,
  $r = \mathbf{S}_\mu(r - \alpha q)$,
  $s = K^T r$,
  $\gamma = s^T F s$,
  $\sigma = \|f_{old} - f\|_2 / \|f\|_2$,
  if $\sigma < \tau$ or $\|r\|_2 = 0$, *break*;
end for

We see that at each iteration both Algorithms 3 and 4 require two matrix–vector multiplications and Algorithm 4 needs one more soft-thresholding. Recall that the cost of the soft-thresholding operator on an n^2-length vector is $\mathcal{O}(n^2)$, which is much less than $\mathcal{O}(n^2 \log n)$ for matrix–vector multiplications. Thus, both Algorithms 3 and 4 require $\mathcal{O}(n^2 \log n)$ operations at each iteration. We also see that r in Algorithm 4 is different from that in MRNSD. However, the following result shows that the monotonicity of $\|r_k\|_2$ in Algorithm 4 is preserved as that in MRNSD, which consequently guarantees the convergence of Algorithm 4.

Theorem 1. *Let K be any N-by-N matrix and assume $\mathbf{S}_\mu$ as defined in (2). If $\|r_k\|_2 > 0$, then $r_{k+1} = \mathbf{S}_\mu(r_k - \alpha_k K p_k)$ generated by Algorithm 4 satisfies the relation*

$$\|r_{k+1}\|_2 < \|r_k\|_2$$

and Algorithm 4 converges for any initial guess f_0.

Proof. Let $\tilde{r}_{k+1} = r_k - \alpha_k K p_k$ as in (4). We first show that $\|\tilde{r}_{k+1}\|_2 \leqslant \|r_k\|_2$. This inequality holds naturally if $\|Kp_k\|_2 = 0$. For $\|Kp_k\|_2 > 0$, define

$$\Phi(\alpha) = \|r_k - \alpha K p_k\|_2^2$$

and let $\alpha_k^* = \frac{\langle Kp_k, r_k\rangle}{\|Kp_k\|_2^2}$. Simple calculation shows that

$$\Phi(\alpha) = \|r_k\|_2^2 + \|Kp_k\|_2^2\alpha^2 - 2\langle Kp_k, r_k\rangle\alpha = \|r_k\|_2^2 + \|Kp_k\|_2^2(\alpha - \alpha_k^*)^2 - \frac{\langle Kp_k, r_k\rangle^2}{\|Kp_k\|_2^2}.$$

It is clear that $\Phi(\alpha)$ decreases monotonically with respect to α in $[0, \alpha_k^*]$. Note from (5) that $\alpha_k = \min\left(\alpha_k^*, \min_{(p_k)_i<0}(-(f_k)_i/(p_k)_i)\right)$ and as a result $0 \leqslant \alpha_k \leqslant \alpha_k^*$. Thus $\Phi(\alpha_k) \leqslant \Phi(0)$, following

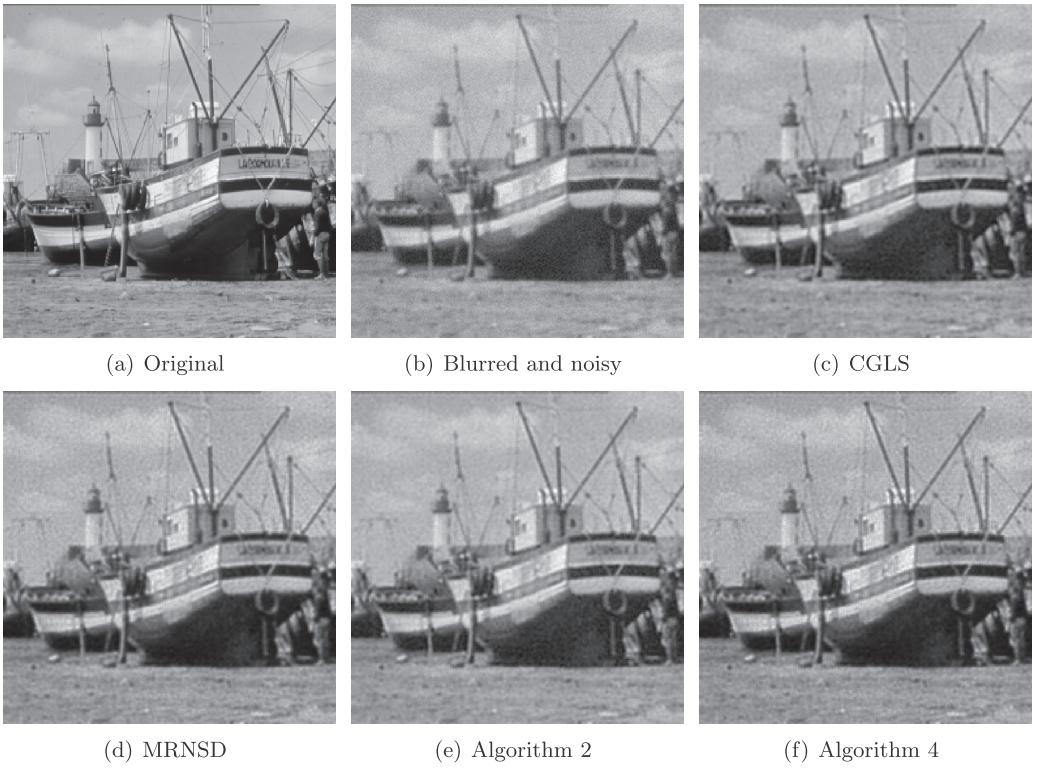
$$\|\tilde{r}_{k+1}\|_2^2 = \Phi(\alpha_k) \leqslant \Phi(0) = \|r_k\|_2^2.$$

We now prove that $\|r_{k+1}\|_2 \leqslant \|\tilde{r}_{k+1}\|_2$. It is clear that $\|r_{k+1}\|_2^2 = \|\mathbf{S}_\mu(\tilde{r}_{k+1})\|_2^2 = \sum_i |(\mathbf{S}_\mu(\tilde{r}_{k+1}))_i|^2$. For each component $(\tilde{r}_{k+1})_i$, the definition of $\mathbf{S}_\mu$ reveals that

$$|(\mathbf{S}_\mu(\tilde{r}_{k+1}))_i| = \begin{cases} |(\tilde{r}_{k+1})_i| - \mu/2 & \text{if } |(\tilde{r}_{k+1})_i| \geqslant \mu/2 \\ 0 & \text{if } |(\tilde{r}_{k+1})_i| < \mu/2 \end{cases} \tag{6}$$

and therefore

$$|(\mathbf{S}_\mu(\tilde{r}_{k+1}))_i| \leqslant |(\tilde{r}_{k+1})_i|. \tag{7}$$



(a) Original (b) Blurred and noisy (c) CGLS

(d) MRNSD (e) Algorithm 2 (f) Algorithm 4

Fig. 5. Example 3.1: Original, contaminated, and restored images by different algorithms for 256×256 *boat* image convolved by a 3×3 Gaussian kernel and degraded by 5% Gaussian white noise.

Table 1
Summary of restoration results of different algorithms for Example 3.1.

Noise level	Method	Itr	ReErr	SNR	CPU
2%	CGLS	5	0.0583	24.68	0.90
	MRNSD	12	0.0587	24.63	1.78
	Algorithm 2	9	0.0568	24.91	2.03
	Algorithm 4	17	0.0584	24.67	2.51
5%	CGLS	2	0.0759	22.40	0.51
	MRNSD	3	0.0762	22.36	0.69
	Algorithm 2	6	0.0745	22.56	1.50
	Algorithm 4	10	0.0762	22.36	1.89

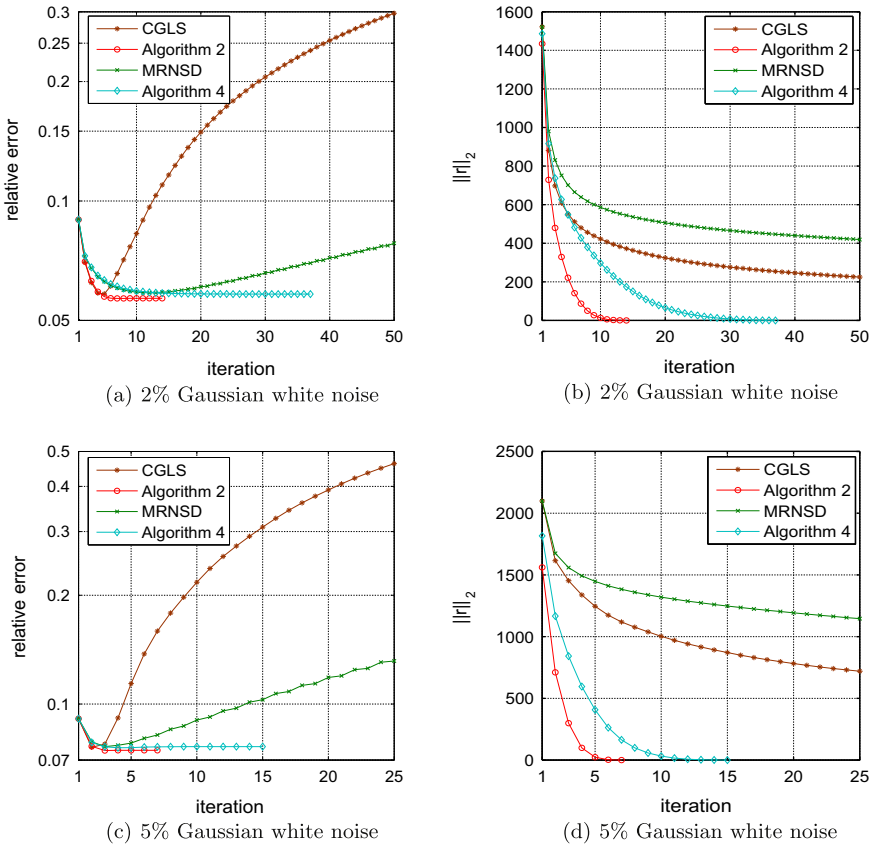
It follows that

$$\|r_{k+1}\|_2^2 = \sum_i |(\mathbf{S}_\mu(\tilde{r}_{k+1}))_i|^2 \leqslant \sum_i |(\tilde{r}_{k+1})_i|^2 = \|\tilde{r}_{k+1}\|_2^2,$$

and as a result,

$$\|r_{k+1}\|_2 \leqslant \|\tilde{r}_{k+1}\|_2 \leqslant \|r_k\|_2.$$

Finally, under the assumption that $\|r_k\|_2 > 0$ we show that $\|r_{k+1}\|_2 < \|r_k\|_2$. Suppose $\|r_{k+1}\|_2 = \|r_k\|_2$ by contradiction and thus $\|r_{k+1}\|_2^2 = \|\tilde{r}_{k+1}\|_2^2 = \|r_k\|_2^2$. Then



(a) 2% Gaussian white noise (b) 2% Gaussian white noise

(c) 5% Gaussian white noise (d) 5% Gaussian white noise

Fig. 6. Example 3.1: Plots of relative error and ℓ_2-norm of vector r against iteration for different algorithms.

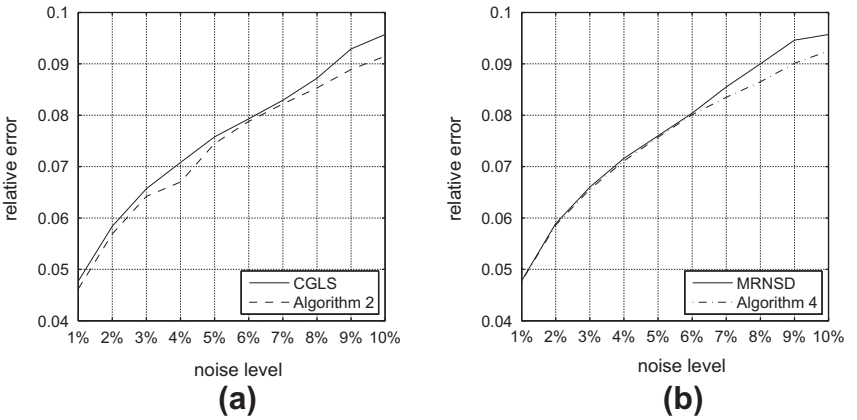


Fig. 7. Example 3.1: Plots of relative error against noise level from 1% to 10% for different Algorithms.

$$\sum_i \left(|(\tilde{r}_{k+1})_i|^2 - |(\mathbf{S}_\mu(\tilde{r}_{k+1}))_i|^2 \right) = 0.$$

For all i, observing from (7) we have $|(\tilde{r}_{k+1})_i|^2 - |(\mathbf{S}_\mu(\tilde{r}_{k+1}))_i|^2 \geqslant 0$, requiring that

$$|(\tilde{r}_{k+1})_i| = |(\mathbf{S}_\mu(\tilde{r}_{k+1}))_i|.$$

This, combining with (6), yields that $(\tilde{r}_{k+1})_i = 0$ for all i and thus $\tilde{r}_{k+1} = \mathbf{0}$. Therefore, $0 = \|\tilde{r}_{k+1}\|_2 = \|r_k\|_2$ and we arrive at a contradiction.

Notice that there is no requirement on f_0 for proving $\|r_{k+1}\|_2 < \|r_k\|_2$, thus the convergence of Algorithm 4 is proved for any initial guess f_0.

Remark 2. In the above proof, we used essentially the property of $\|\mathbf{S}_\mu(x)\|_2 \leqslant \|x\|_2$ for any $x \in \mathbb{R}^N$, and moreover $\|\mathbf{S}_\mu(x)\|_2 = \|x\|_2$ if and only if $x = \mathbf{0}$. We can use this observation to present a slight generalization of Algorithm 4, in which $\mathbf{S}_\mu$ is replaced by any operator $\mathcal{D}$ satisfying

(a) Blurred and noisy (b) CGLS (c) MRNSD

(d) Alg. 2 with the Wiener filter (e) Alg. 2 with soft-thresholding (f) Alg. 2 with median filtering

(g) Alg. 4 with the Wiener filter (h) Alg. 4 with soft-thresholding (i) Alg. 4 with median filtering

Fig. 8. Example 3.2: Contaminated and restored images by different algorithms for 256×256 *house* image convolved by a 3×5 motion kernel and degraded by 8% Gaussian white noise.

$$\|\mathcal{D}(x)\|_2 \leqslant \|x\|_2, \quad x \in \mathbb{R}^N$$

and moreover $\|\mathcal{D}(x)\|_2 = \|x\|_2$ if and only if $x = \mathbf{0}$, and Theorem 1 stands still.

3. Numerical examples

In this section, we give several numerical examples to illustrate the performance of the proposed iterative algorithms: CGLS-like and MRNSD-like, for image deblurring problems. Our tests were done by using MATLAB 7.10.0 (R2010a) on a PC computer with Intel(R) Core(TM)2 Duo CPU 2.93 GHz and 2 GB memory. The floating-point precision is 10^{-16}. The initial guess of each algorithm is set to be a zero vector and the threshold value τ for both Algorithms 2 and 4 is set to be 10^{-3}. The signal-to-noise ratio (SNR) and the relative error (ReErr) are used to measure the quality of a restored image. They are defined by

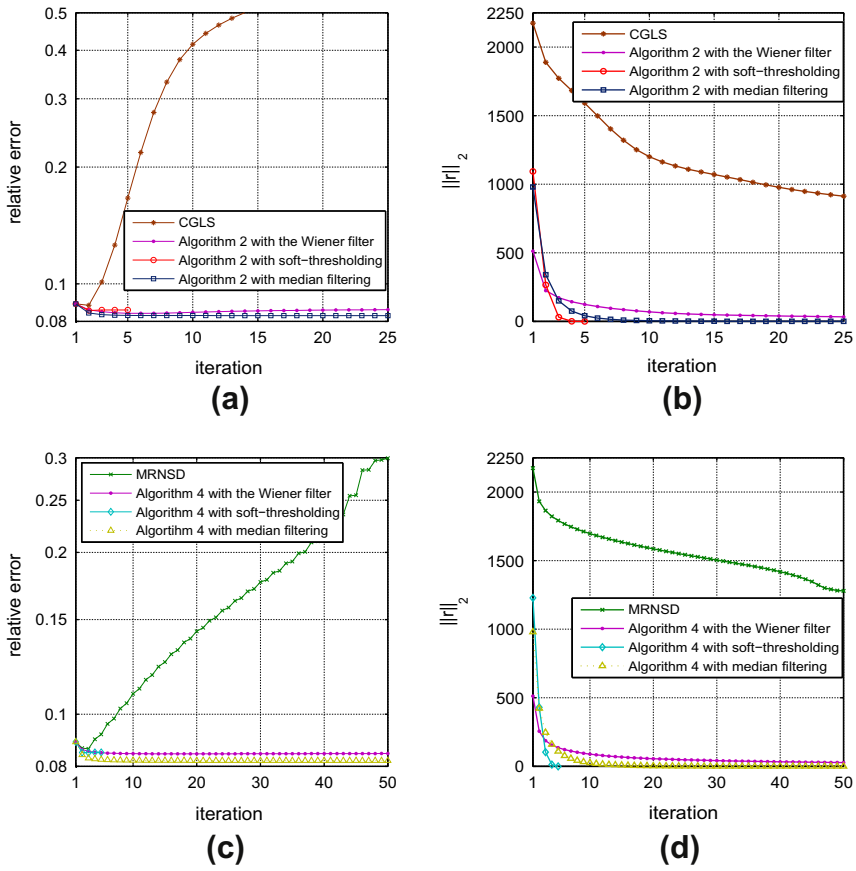


Fig. 9. Example 3.2: Plots of relative error and ℓ_2-norm of vector r against iteration by different algorithms for blurred *house* image degraded by 8% Gaussian white noise.

Table 2
Example 3.2: Summary of restoration results by different algorithms for blurred *house* image degraded by 8% Gaussian white noise.

Method	Itr	ReErr	SNR	CPU
CGLS	2	0.0880	21.11	0.59
MRNSD	3	0.0862	21.26	0.62
Alg. 2 with the Wiener filter	14	0.0851	21.41	3.39
Alg. 2 with soft-thresholding	4	0.0856	21.35	0.97
Alg. 2 with median filtering	7	0.0828	21.64	1.89
Alg. 4 with the Wiener filter	8	0.0845	21.46	1.36
Alg. 4 with soft-thresholding	5	0.0849	21.42	0.90
Alg. 4 with median filtering	9	0.0821	21.71	1.64

$$\mathrm{SNR} = 20\log_{10}\frac{\|f_{\mathrm{true}}\|_2}{\|f_{\mathrm{true}} - f\|_2}, \quad \mathrm{ReErr} = \frac{\|f - f_{\mathrm{true}}\|_2}{\|f_{\mathrm{true}}\|_2},$$

with f_{true} and f being the original and restored images, respectively. The number of iterations (denoted by "Itr") and the elapsed CPU time in seconds (denoted by "CPU") of each algorithm are reported as well. The optimal iterations for Algorithms 1 and 3 are obtained by considering the smallest relative error of the restored images. The optimal parameters μ for Algorithms 2 and 4 are chosen by trial and error.

3.1. Comparison of Algorithms 1–4

We first demonstrate the performance of Algorithms 2 and 4 compared with Algorithm 1 (CGLS) and Algorithm 3 (MRNSD).

Example 3.1. The original 256×256 image *boat*, shown in Fig. 4(a), is first blurred by a 3×3 Gaussian kernel with standard deviation $\sigma_b = 2$ and then degraded by 2% and 5% Gaussian white noises, respectively. Blurred noisy images and the restored images by Algorithms 1–4 are shown in Figs. 4 and 5. It can be seen from the figures that images restored by Algorithms 2 and 4 contain less noise and are smoother than those from original CGLS and MRNSD, respectively. This is not unexpected since we filter the residual vector in Algorithms 2 and 4 at each iteration.

Table 1 reports the restorations determined by Algorithms 1–4 in ReErr, SNR, CPU and Itr. Observing from the table that the ReErr and SNR values of Algorithms 2 and 4 are slightly better than or comparable to those of CGLS and MRNSD, and moreover, Algorithm 2 is shown to be better than Algorithm 4. All algorithms are fast and produce satisfactory results in seconds.

Figs. 6(a) and 6(c) provide convergence histories of Algorithms 1–4 for 2% and 5% Gaussian white noises, respectively. For the two test cases, CGLS again exhibits a clearly semi-convergent behavior. MRNSD is relatively stable but still exhibits the semi-convergence behavior, especially for higher noise level 5%. The proposed Algorithms 2 and 4 successfully overcome the semi-convergence behavior of the original CGLS and MRNSD algorithms, respectively. In addition, we plot the ℓ_2-norm of vector r against iteration number for 2% and 5% noise levels in Figs. 6(b) and 6(d), respectively. The two figures show that $\|r\|_2$ of Algorithms 1–4 decreases monotonically, which is consistent with our theoretical results. It should be noted that for Algorithm 2, after 14 iterations for 2% noise and 7 iterations for 5% noise, $\|r\|_2 = 0$ and hence the iteration stops. Similarly, Algorithm 4 stops after 37 iterations for 2% noise and 15 iterations for 5% noise.

For further comparison, Fig. 7 plots relative errors of all four algorithms for noise level from 1% to 10%. We see that the proposed Algorithms 2 and 4 provide slightly better or comparable results, especially for higher noise levels.

3.2. Comparison of using different filters in Algorithms 2 and 4

In this subsection, we consider using other denoising techniques compared with the soft-thresholding operator in Algorithms 2 and 4. Median filtering and the Wiener filter are used. The median filter is a popular nonlinear noise reduction technique which is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise [38,46,53]. It runs through the signal entry-by-entry, replacing each entry with the median of neighboring

Table 3
Example 3.2: Summary of restoration results of different algorithms for blurred *house* image degraded by 5% salt-and-pepper noise and Poisson noise, respectively.

Noise	Method	Itr	ReErr	SNR	CPU
5% salt & pepper	CGLS	1	0.1312	17.64	0.34
	MRNSD	1	0.1312	17.64	0.27
	Alg. 2 with the Wiener filter	38	0.2268	12.89	8.83
	Alg. 2 with soft-thresholding	1	0.1312	17.64	0.36
	Alg. 2 with median filtering	7	0.1177	18.59	1.81
	Alg. 4 with the Wiener filter	32	0.1507	16.44	5.29
	Alg. 4 with soft-thresholding	1	0.1312	17.64	0.28
	Alg. 4 with median filtering	9	0.1167	18.66	1.59
Poisson noise	CGLS	1	0.0906	20.86	0.39
	MRNSD	1	0.0906	20.86	0.27
	Alg. 2 with the Wiener filter	16	0.0976	20.21	3.90
	Alg. 2 with soft-thresholding	4	0.0891	21.00	1.01
	Alg. 2 with median filtering	7	0.0864	21.27	1.84
	Alg. 4 with the Wiener filter	13	0.0899	20.93	2.20
	Alg. 4 with soft-thresholding	4	0.0898	20.94	0.72
	Alg. 4 with median filtering	9	0.0858	21.33	1.62

entries. For each pixel in an $n \times n$ image, computing the median from a neighborhood of size $m \times m$ requires $\mathcal{O}(m)$ comparisons, and thus applying the median filter to the whole image requires $\mathcal{O}(n^2 m)$ operations. This cost is clearly less than that of matrix–vector multiplications with $K \in \mathbb{R}^{n^2 \times n^2}$, i.e. $\mathcal{O}(n^2 \log n)$. Another noise-removal technique considered to replace soft-thresholding in Algorithms 2 and 4 is the Wiener filter, which is based on statistics estimated from a local neighborhood of each pixel [38]. For a neighborhood of size $m \times m$, estimating the local image mean and standard deviation requires $\mathcal{O}(m^2)$ operations. Therefore, applying the Wiener filter to an $n \times n$ image costs $\mathcal{O}(n^2 m^2)$ operations, again less than $\mathcal{O}(n^2 \log n)$ for matrix–vector products. In summary, both Algorithms 2 and 4 with any of soft-thresholding, the Wiener filter and the median filter require $\mathcal{O}(n^2 \log n)$ operations at each iteration. In our tests, MATLAB build-in functions `medfilt2` with window size 3×3 and `wiener2` with window size 9×9 are used.

Example 3.2. Original 256×256 *house* image is contaminated by a 3×5 motion blur, generated by MATLAB command `fspecial('motion',4,1)`, and 8% Gaussian white noise. Fig. 8(a) shows the blurred and noisy image. The restored images by all algorithms are shown in Fig. 8. From these figures, we see that restorations of our algorithms are smoother than those of CGLS and MRNSD. Moreover, Algorithms 2 and 4 with the median filter produce results comparable to those with soft-

(a) Original (b) Blurred and noisy (c) CGLS

(d) MRNSD (e) RRGMRES (f) ISTA

(g) HyBR (h) Algorithm 2 (i) Algorithm 4

Fig. 10. Example 3.3: Original, contaminated, and restored images by different algorithms for 256×256 *cameraman* image convolved by a 5×5 Moffat blur and degraded by 3% Gaussian white noise.

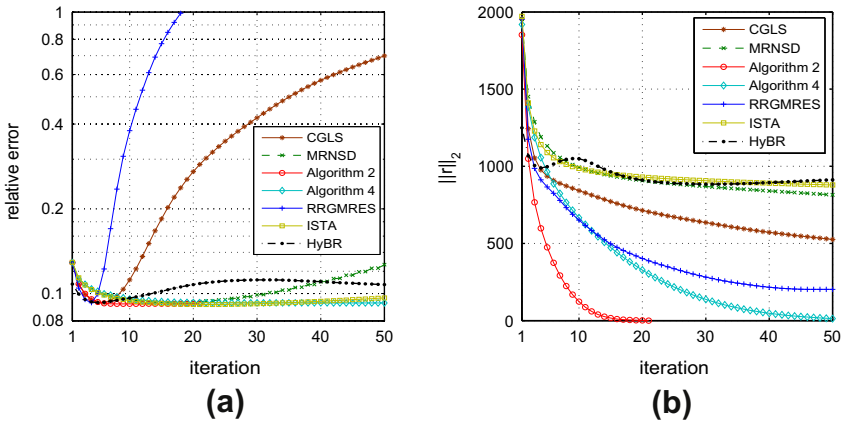


Fig. 11. Example 3.3: Plots of relative error and ℓ_2-norm of vector r against iteration by different algorithms for blurred and noisy *cameraman* image.

thresholding and clearly better than those with the Wiener filter. In addition, we plot convergence histories of these algorithms in Figs. 9(a) and 9(c) and $\|r\|_2$ against iteration number in Figs. 9(b) and 9(d). Indeed, Algorithms 2 and 4 with each denoising technique overcome the semi-convergence behavior and make the resulting approximate solutions be less sensitive to the iterations.

Table 2 lists the corresponding restoration results in ReErr, SNR, Itr and CPU. From this table we observe that Algorithms 2 and 4 with each noise-removal technique produce better ReErr and SNR values and the median filter based algorithms are best.

For further comparison, Table 3 reports the restoration results for the blurred *house* image contaminated by 5% salt-and-pepper noise and Poisson noise, respectively. From the table, we see that Algorithms 2 and 4 based on the median filter are best especially for the blurred image degraded by salt-and-pepper noise. This is not unexpected since median filtering is very effective in removing salt-and-pepper noise [38]. It tells us that we may choose specific noise-removal techniques for specific situations, making the proposed algorithms become more flexible.

3.3. Comparison with other algorithms

We now compare the proposed soft-thresholding based Algorithms 2 and 4 with three popular algorithms: range restricted GMRES (RRGMRES) algorithm from [11], weighted-GCV based Lanczos-Tikhonov hybrid approach (HyBR) from [14] and iterative shrinkage-thresholding algorithm (ISTA) [5,15] given by

$$f_{k+1} = \mathbf{S}_\mu\Big(f_k + K^T(g - Kf_k)\Big), \quad f_0 = \mathbf{0}.$$

Clearly, ISTA is easy to implement. We note that RRGMRES is a variant of the classical Krylov subspace algorithm GMRES and it is more suitable when applied to the solution of linear systems of equations with a singular or nearly singular matrix [11]. Lanczos-hybrid approach combines an iterative Lanczos bidiagonalization with a SVD-based regularization method to stabilize the semi-convergence behavior. The MATLAB implementations of RRGMRES and HyBR are obtained from *Regularization Tools*[2] by Hansen [29] and *RestoreTools* by Nagy et al. [43], respectively. The initial guess for all algorithms is zero and ISTA stops if $\|f_{k-1} - f_k\|_2/\|f_k\|_2 < \tau$. Similarly to CGLS and MRNSD, optimal restorations of RRGMRES are reported according to the smallest relative error of the restored images.

Example 3.3. We seek to restore the 256×256 *cameraman* image, which has been contaminated by a 5×5 Moffat blur with parameters $s_1 = s_2 = 2$, $\rho = 0$ and $\beta = 2$ (see [30] for more details) and 3% Gaussian white noise. The original and contaminated images are shown in Figs. 10(a) and 10(b), respectively.

Figs. 10(c)–10(i) provide a qualitative comparison of images restored by different algorithms. Algorithm 4 can be seen to yield a restoration of higher quality with less noise and artifacts. Corresponding convergence histories of different algorithms and plots of $\|r\|_2$ against iteration are shown in Fig. 11. We remark that at k-th iteration, r_k in CGLS, MRNSD, RRGMRES, ISTA, HyBR represents the residual vector, while r_k in both Algorithms 2 and 4 represents the filtered residual vector given by $r_k = \mathbf{S}_\mu(r_{k-1} - \alpha_{k-1}Kp_{k-1})$. Like ISTA and Lanczos-hybrid, Algorithms 2 and 4 are stable and successfully overcome the

[2] http://www.imm.dtu.dk/~pcha/Regutools/.

Table 4
Example 3.3: Summary of restoration results of different algorithms for blurred and noisy *cameraman* image.

Method	Itr	ReErr	SNR	CPU
CGLS	6	0.0928	20.65	1.05
MRNSD	17	0.0930	20.63	2.61
RRGMRES	4	0.0928	20.65	0.94
ISTA	42	0.0944	20.50	6.38
HyBR	8	0.0947	20.47	2.17
Algorithm 2	12	0.0919	20.73	2.65
Algorithm 4	27	0.0927	20.66	4.20

semi-convergence of the original CGLS and MRNSD algorithms, respectively. Table 4 gives a quantitative comparison of the restorations determined by these algorithms. This table illustrates that Algorithm 2 provides slightly better SNR and ReErr values and as usual, ISTA is relatively slower than other algorithms.

4. Conclusions

In this paper, we have proposed two soft-thresholding based iterative algorithms, CGLS-like and MRNSD-like. The proposed algorithms filter the residual vector at each iteration to overcome the semi-convergence behavior of CGLS and MRNSD, making iterations be more stable. We have proved the convergence of MRNSD-like. We demonstrated through a variety of test problems that our approaches stabilize the iterations, make the solution less sensitive to the number of iterations, and provide slightly better deblurring results than the classical CGLS and MRNSD methods. We have compared the proposed algorithms with Lanczos-hybrid, RRGMRES and ISTA as well. Finally, we note that carefully selecting a noise reduction algorithm for specific situations makes the proposed algorithms become more flexible. The research of computing optimal μ or updating it automatically at each iteration is currently under investigation.

Acknowledgments

The authors would like to express their thanks to the reviewers, the Editor-in-Chief Prof. Witold Pedrycz for their constructive, detailed and helpful advice regarding this paper.

References

[1] D. Alassi, R. Alhajj, Effectiveness of template detection on noise reduction and websites summarization, Inform. Sci. 219 (2013) 41–72.
[2] J.M. Bardsley, J.G. Nagy, Covariance-preconditioned iterative methods for nonnegatively constrained astronomical imaging, SIAM J. Matrix Anal. Appl. 27 (2006) 1184–1198.
[3] F.S. Viloche Bazán, Fixed-point iterations in determining the Tikhonov regularization parameter, Inverse Probl. 24 (2008) 035001.
[4] F.S. Viloche Bazán, L.S. Borges, GKB-FP: an algorithm for large-scale discrete ill-posed problems, BIT 50 (2010) 481–507.
[5] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imag. Sci. 2 (2009) 183–202.
[6] G. Bhatnagar, Q.M. Jonathan Wu, B. Raman, Discrete fractional wavelet transform and its application to multiple encryption, Inform. Sci. 223 (2013) 297–316.
[7] A. Björck, Numerical Methods for Least Squares Problems, SIAM, Philadelphia, 1996.
[8] A. Björck, A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations, BIT 28 (1988) 659–670.
[9] A. Björck, E. Grimme, P. Van Dooren, An implicit shift bidiagonalization algorithm for ill-posed systems of linear equations, BIT 34 (1994) 510–534.
[10] J. Bolz, I. Farmer, E. Grinspun, P. Schröder, Sparse matrix solvers on the GPU: conjugate gradients and multigrid, ACM Trans. Graph. 22 (2003) 917–924.
[11] D. Calvetti, B. Lewis, L. Reichel, GMRES-type methods for inconsistent systems, Linear Algebra Appl. 316 (2000) 157–169.
[12] D. Calvetti, L. Reichel, Tikhonov regularization of large linear problems, BIT 43 (2003) 263–283.
[13] D.Q. Chen, L.Z. Cheng, Deconvolving Poissonian images by a novel hybrid variational model, J. Visual Commun. Image Rep. 22 (2011) 643–652.
[14] J. Chung, J.G. Nagy, D.P. O'Leary, A weighted-GCV method for Lanczos-hybrid regularization, Electron. Trans. Numer. Anal. 28 (2008) 149–167.
[15] I. Daubechies, M. Defrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Commun. Pure Appl. Math. 57 (2004) 1413–1457.
[16] M. Donatelli, C. Estatico, A. Martinelli, S. Serra-Capizzano, Improved image deblurring with anti-reflective boundary conditions and re-blurring, Inverse Probl. 22 (2006) 2035–2053.
[17] M. Donatelli, S. Serra-Capizzano, Anti-reflective boundary conditions and re-blurring, Inverse Probl. 21 (2005) 169–182.
[18] H.W. Engl, M. Hanke, A. Neubauer, Regularization of Inverse Problems, Kluwer Academic Publishers, Dordrecht, 2000.
[19] X.B. Gao, Q. Wang, X.L. Li, D.C. Tao, K.B. Zhang, Zernike-moment-based image super resolution, IEEE Trans. Image Process. 20 (2011) 2738–2747.
[20] X.B. Gao, K.B. Zhang, D.C. Tao, X.L. Li, Joint learning for single-image super-resolution via a coupled constraint, IEEE Trans. Image Process. 21 (2012) 469–480.
[21] G.H. Golub, M. Heath, G. Wahba, Generalized cross-validation as a method for choosing a good ridge parameter, Technometrics 21 (1979) 215–222.
[22] N.Y. Guan, D.C. Tao, Z.G. Luo, B. Yuan, NeNMF: an optimal gradient method for nonnegative matrix factorization, IEEE Trans. Signal Process. 60 (2012) 2882–2898.
[23] Q. Guo, S.Y. Chen, H. Leung, S.T. Liu, Covariance intersection based image fusion technique with application to pansharpening in remote sensing, Inform. Sci. 180 (2010) 3434–3443.
[24] X.H. Han, X.M. Chang, An intelligent noise reduction method for chaotic signals based on genetic algorithms and lifting wavelet transforms, Inform. Sci. 218 (2013) 103–118.
[25] M. Hanke, Conjugate Gradient Type Methods for Ill-Posed Problems, Longman Scientific & Technical, Harlow, UK, 1995.
[26] M. Hanke, On Lanczos based methods for the regularization of discrete ill-posed problems, BIT 41 (2001) 1008–1018.
[27] M. Hanke, P.C. Hansen, Regularization methods for large-scale problems, Surveys Math. Indust. 3 (1993) 253–315.

[28] P.C. Hansen, Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion, SIAM, Philadelphia, 1998.
[29] P.C. Hansen, Regularization tools version 4.0 for Matlab 7.3, Numer. Algor. 46 (2007) 189–194.
[30] P.C. Hansen, J.G. Nagy, D.P. O'Leary, Deblurring Images: Matrices, Spectra, and Filtering, SIAM, Philadelphia, 2006.
[31] M.R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. Nat. Bur. Standards 49 (1952) 409–436.
[32] G. Jeon, M. Anisetti, V. Bellandi, E. Damiani, J. Jeong, Designing of a type-2 fuzzy logic filter for improving edge-preserving restoration of interlaced-to-progressive conversion, Inform. Sci. 179 (2009) 2194–2207.
[33] M.F. Jiang, L. Xia, G.F. Shou, F. Liu, S. Crozier, Two hybrid regularization frameworks for solving the electrocardiography inverse problem, Phys. Med. Biol. 53 (2008) 5151–5164.
[34] L. Kaufman, Maximum likelihood, least squares, and penalized least squares for PET, IEEE Trans. Med. Imag. 12 (1993) 200–214.
[35] M.E. Kilmer, P.C. Hansen, M.I. Espanol, A projected-based approach to general-form Tikhonov regularization, SIAM J. Sci. Comput. 29 (2007) 315–330.
[36] M.E. Kilmer, D.P. O'Leary, Choosing regularization parameters in iterative methods for ill-posed problems, SIAM J. Matrix Anal. Appl. 22 (2001) 1204–1221.
[37] H. Kim, H. Park, Nonnegative matrix factorization based on alternating non-negativity constrained least squares and active set method, SIAM J. Matrix Anal. Appl. 30 (2008) 713–730.
[38] J.S. Lim, Two-Dimensional Signal and Image Processing, Prentice-Hall, New Jersey, 1990.
[39] T.C. Lin, C.M. Lin, Wavelet-based copyright-protection scheme for digital images based on local features, Inform. Sci. 179 (2009) 3349–3358.
[40] X.-G. Lv, T.-Z. Huang, Z. Xu, X.-L. Zhao, Kronecker product approximations for image restoration with whole-sample symmetric boundary conditions, Inform. Sci. 186 (2012) 150–163.
[41] V.A. Morozov, Regularization Methods for Solving Incorrectly Posed Problems, Springer-Verlag, New York, 1984.
[42] J.G. Nagy, K.M. Palmer, Steepest descent, CG, and iterative regularization of ill-posed problems, BIT 43 (2003) 1003–1017.
[43] J.G. Nagy, K.M. Palmer, L. Perrone, Iterative methods for image deblurring: a Matlab object-oriented approach, Numer. Algor. 36 (2004) 73–93.
[44] J.G. Nagy, Z. Strakos, Enforcing nonnegativity in image reconstruction algorithms, in: David C. Wilson et al. (Eds.), Mathematical Modeling, Estimation, and Imaging, vol. 4121, 2000, pp. 182–190.
[45] D.P. O'Leary, J.A. Simmons, A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems, SIAM J. Sci. Stat. Comput. 2 (1981) 474–489.
[46] W.K. Pratt, Digital Image Processing, fourth ed., John Wiley & Sons, Hoboken, 2007.
[47] M.C. Roggemann, B. Welsh, Imaging Through Turbulence, CRC Press, Boca Raton, Florida, 1996.
[48] D.D. Sha, J.P. Sutton, Towards automated enhancement, segmentation and classification of digital brain images using networks of networks, Inform. Sci. 138 (2001) 45–77.
[49] M.L. Song, D.C. Tao, C. Chen, J.J. Bu, J.B. Luo, C.Q. Zhang, Probabilistic exposure fusion, IEEE Trans. Image Process. 21 (2012) 341–357.
[50] M.L. Song, D.C. Tao, C. Chen, X.L. Li, C.W. Chen, Color to gray: visual cue preservation, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 1537–1552.
[51] J.G. Sun, C. Fyfe, M. Crowe, Extending Sammon mapping with Bregman divergences, Inform. Sci. 187 (2012) 72–92.
[52] A.N. Tikhonov, Solution of incorrectly formulated problems and the regularization method, Soviet Math. Dokl. 4 (1963) 1035–1038.
[53] J.W. Tukey, Exploratory Data Analysis, Addison-Wesley, Reading, Massachusetts, 1977.
[54] C.R. Vogel, Computational Methods for Inverse Problems, SIAM, Philadelphia, 2002.
[55] C.Y. Wee, R. Paramesran, Measure of image sharpness using eigenvalues, Inform. Sci. 177 (2007) 2533–2552.