On the regularizing behavior of recent gradient methods in the solution of linear ill-posed problems *

Roberta De Asmundis†, Daniela di Serafino‡, Germana Landi§

May 12, 2014

Abstract

We analyze the regularization properties of two recently proposed gradient methods applied to discrete linear inverse problems. By studying their filter factors, we show that the tendency of these methods to eliminate first the eigencomponents of the gradient corresponding to large singular values allows to reconstruct the most significant part of the solution, thus yielding a useful filtering effect. This behavior is confirmed by numerical experiments performed on some image restoration problems. Furthermore, the experiments show that, for severely ill-conditioned problems and high noise levels, the two methods can be competitive with the Conjugate Gradient (CG) method, since they are slightly slower than CG, but exhibit a better semiconvergence behavior.

Keywords: discrete linear inverse problems, least squares problems, iterative regularization, gradient methods.

AMS subject classifications: 65F22, 65K10, 90C20.

1 Introduction

We consider discrete linear inverse problems of the form

$$\mathbf{b} = A\mathbf{x} + \mathbf{n}, \tag{1}$$

where $A \in \mathbb{R}^{p\times n}$ and $\mathbf{b} \in R^p$ ($p \geq n$) are known data, $\mathbf{n} \in \mathbb{R}^p$ is unknown and represents perturbations in the data, and $\mathbf{x} \in \mathbb{R}^n$ represents an object to be recovered. We assume that A is ill-conditioned, with singular values decaying to zero; we also assume that A is full rank. Such problems often arise from the discretization of Fredholm integral equations of the first kind, which are used, e.g., to model instrument distortions in the measure of unknown functions in a variety of application fields, including statistical inference, geophysics, inverse scattering, and image processing [18]. For example, in image processing, A may model the blurring effect produced by the image acquisition process, as in image deblurring, or it may represent the discretization of a tomographic linear

*Work partially supported by INdAM-GNCS, under the 2013 Project *Numerical methods and software for large-scale optimization with applications to image processing* and the 2014 Project *First-order optimization methods for image restoration and analysis*.

†Dipartimento di Ingegneria Informatica Automatica e Gestionale "Antonio Ruberti", Sapienza Università di Roma, Via Ariosto 25, 00185 Roma, Italy, email: roberta.deasmundis@uniroma1.it.

‡Dipartimento di Matematica e Fisica, Seconda Università degli Studi di Napoli, Viale A. Lincoln 5, 81100 Caserta, Italy, email: daniela.diserafino@unina2.it.

§Dipartimento di Matematica, Università degli Studi di Bologna, Piazza di Porta S. Donato 5, 40127 Bologna, Italy, email: germana.landi@unibo.it.

operator, as in computed tomography, or a partial Fourier transform, as in magnetic resonance imaging.

Because of the ill-conditioning of A, computing the solution of the least squares problem[1]

$$\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{b} - A\mathbf{x}\|^2 \tag{2}$$

does not provide a meaningful solution of (1), since it amplifies the noise contained in the data. Therefore, a regularization method is applied to compute a reasonable approximation to the exact solution. Roughly speaking, a regularization method replaces the original problem with a family of "close" better-conditioned (regularized) problems, depending on a parameter, such that, for an appropriate choice of the parameter, the solution of the corresponding regularized problem converges to the exact solution when the noise tends to zero [18]. The regularized problems can be obtained by adding to the objective function in (2) a penalty term based on some norm or seminorm of the solution, such as in the Tikhonov and l_1 regularizations, or by exploting the truncated SVD and GSVD decompositions, or by applying iterative methods (for more details see, e.g., [6, 18, 19, 29] and the references therein).

As observed in [6], iterative regularization methods for the solution of (2) are very flexible (e.g., they can be efficiently applied to both spatially variant and invariant blurs), allow easy integration of other regularization techniques, and easy treatment of constraints such as non-negativity. These methods generally show a semiconvergence behavior of the relative error, i.e., this error decreases in the early iterations and then begins to increase. Therefore a suitable early stop of the iterations is needed to obtain a good approximation to the solution. The choice of the iteration index where the method has to be stopped plays a fundamental role and it is based on further information on the problem. For example, the Morozov's discrepancy principle [34] requires terminating the iterations as soon as

$$\|A\mathbf{x}_k - \mathbf{b}\| \leq \tau\delta,$$

where δ is an available estimate of the noise norm $\|\mathbf{n}\|$ and $\tau > 1$. Other popular criteria for stopping the iterations are the L-curve method [28] and the Generalized Cross Validation method [23], which do not require any a priori estimates of δ.

The regularizing properties of the classical Landweber, steepest descent (SD) and conjugate gradient (CG) methods have been widely investigated (see, e.g., [24, 18, 36]). In particular, it is well known that the Landweber and SD methods generally exhibit very slow convergence and thus they are rarely used in practice, despite their "stable" convergence behavior, unless they are coupled with ad hoc preconditioners (see, e.g., [36]). Conversely, CG methods such as CGLS and LSQR rapidly compute a good approximation to the solution and for this reason are usually preferred in practical applications. However, as pointed out in [36], the fast convergence of CG methods makes them very sensitive to the stopping criterion, and an early or late stopping may give a low-quality approximate solution. On the other hand, starting from the innovative Barzilai-Borwein approach [3], several new gradient methods have been developed that use suitable steplengths to achieve a significant speedup over SD [22, 13, 14, 12, 39, 40, 21, 20, 17, 16]. This has motivated the interest toward the possible use of these gradient methods as regularization methods, and recent work has been devoted to understand the behavior of some of them in the solution of discrete inverse problems [2, 10].

In this work we analyze the regularization properties of two gradient methods recently proposed in [17] and [16], named SDA and SDC, which have shown to be highly competitive with the currently available fastest gradient methods. Both SDA and SDC share the idea of fostering a selective elimination of the components of the gradient along

[1] Here and henceforth $\|\cdot\|$ denotes the Euclidean norm.

the eigenvectors of the Hessian matrix, thus pushing the search in subspaces of smaller dimensions and speeding up the convergence of the method. This is achieved by using suitable steplength selection rules, which alternate, in a cyclic way, some Cauchy steplengths with some constant steplengths containing spectral information on the Hessian. Following [36], we perform a filter factor analysis of the two methods applied to problem (2). In particular, we show that the above-mentioned tendency to selective elimination of the gradient components corresponds to a progressive approximation of the components of the solution along the right singular vectors of the matrix A, starting from those associated with the largest singular values. Therefore, SDA and SDC have filtering properties. This behavior is confirmed by numerical experiments performed on some image restoration problems. The experiments also confirm that SDA and SDC realize a good tradeoff between convergence speed and regularization properties, and thus they can be competitive with CG, especially for large noise and severely ill-conditioned problems.

This article is organized as follows. In Section 2 we briefly describe the SDA and SDC gradient methods for problem (2), highlighting their main features. In Section 3 we study the filter factors of the two methods. We also compare these filter factors with those of other gradient methods and CG on a test problem from Hansen's Regularization Tools [25]. In Section 4 we discuss the results of numerical experiments concerning the application of the SDA and SDC methods to image restoration problems widely used to test regularization methods. In particular, we compare SDA and SDC with other gradient methods and CG, in terms of speed and relative error behavior. Finally, we draw some conclusions in Section 5.

2 The SDA and SDC methods

Gradient methods for problem (2) generate a sequence of iterates $\{\mathbf{x}_k\}$ as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k, \quad k = 0, 1, 2, \ldots, \tag{3}$$

where

$$\mathbf{g}_k = A^T(A\mathbf{x}_k - \mathbf{b}) \tag{4}$$

is the gradient at $\mathbf{x}_k$ of the objective function in (2) and $\alpha_k > 0$ is a steplength computed by applying a suitable rule.

In order to analyze the behavior of some gradient methods, we consider the singular value decomposition of A,

$$A = U\Sigma V^T, \tag{5}$$

where $U = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p] \in \mathbb{R}^{p\times p}$, $V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n] \in \mathbb{R}^{n\times n}$, and $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n) \in \mathbb{R}^{p\times n}$. Note that the squares of the singular values σ_i are the eigenvalues of the Hessian matrix of the objective function in (2), and the right singular vectors $\mathbf{v}_i$ are a set of associated orthonormal eigenvectors. By using (3) and (4), it is easy to verify that if $\mathbf{g}_0 = \sum_{i=1}^n \mu_i^0 \mathbf{v}_i$, then

$$\mathbf{g}_k = \sum_{i=1}^n \mu_i^k \mathbf{v}_i, \qquad \mu_i^k = \mu_i^0 \prod_{j=0}^{k-1} (1 - \alpha_j \sigma_i^2). \tag{6}$$

It follows that if at the k-th iteration $\mu_i^k = 0$ for some i, then for $l > k$ it will be $\mu_i^l = 0$, i.e., the component of the gradient along $\mathbf{v}_i$ will be zero at all subsequent iterations. The condition $\mu_i^k = 0$ holds if and only if $\mu_i^0 = 0$ or $\alpha_j = 1/\sigma_i^2$ for some $j < k$. In the following, without loss of generality (see, e.g., [16, Section 1]) we assume that

$$\sigma_1 > \sigma_2 > \cdots > \sigma_n \tag{7}$$

and

$$\mu_1^0 \neq 0, \quad \mu_n^0 \neq 0. \tag{8}$$

It is well known that the SD method, which uses the Cauchy steplength

$$\alpha_k^{SD} = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_k^T A \mathbf{g}_k},$$

generally shows very slow convergence. Specifically, SD eventually performs its search in the two-dimensional subspace generated by $\mathbf{v}_1$ and $\mathbf{v}_n$, producing a zigzag pattern which is the main reason for its convergence behavior [1, 37].

As mentioned in Section 1, different steplength strategies have been developed to overcome this difficulty, leading to gradient methods that may be competitive with CG, especially when low accuracy in the solution is required. Here we focus on two gradient methods recently proposed in [17] and[16], called SDA and SDC, respectively (the meaning of these names is explained later in this section). In both methods the choice of the steplength can be described as follows:

$$\alpha_k = \begin{cases} \alpha_k^{SD} & \text{if } \mathrm{mod}(k, h+m) < h, \\ \bar{\alpha}_s & \text{otherwise, with } s = \max\{i \leq k \ : \ \mathrm{mod}(i, h+m) = h\}, \end{cases} \tag{9}$$

where $h \geq 2$ and $\bar{\alpha}_s$ is a "special" steplength computed at a certain iteration s by exploiting information from previous SD steps. In other words, the methods make h consecutive exact line searches and then compute a different steplength, which is kept constant and applied in m consecutive gradient iterations.

In [17] the new steplength

$$\widetilde{\alpha}_s = \left(\frac{1}{\alpha_{s-1}^{SD}} + \frac{1}{\alpha_s^{SD}} \right)^{-1} \tag{10}$$

is proposed and used in (9) as $\bar{\alpha}_s$, obtaining the SDA method. Actually, the original version of SDA performs a dynamical choice of the number h of steps where α_k^{SD} is applied, by exploiting a so-called switch condition; however, in this work we use fixed values of h, as explained later. The steplength (10) is related to the largest and smallest singular values of A as shown next.

Proposition 2.1 *Let $\{\mathbf{x}_k\}$ be the sequence of iterates generated by the SD method applied to problem (2), starting from any point $\mathbf{x}_0$, and suppose that (7) and (8) hold. Then*

$$\lim_{k \to \infty} \widetilde{\alpha}_k = \frac{1}{\sigma_1^2 + \sigma_n^2}. \tag{11}$$

Proof. The thesis follows straightforwardly from Proposition 3.1 in [17]. □

As discussed in [17], the previous result and the properties of the SD method suggest that the steplength (9), with $\bar{\alpha}_s$ defined by (10), combines the tendency of the SD method to choose its search direction in the two-dimensional space spanned by $\mathbf{v}_1$ and $\mathbf{v}_n$ with the tendency of the gradient method with constant steplength $1/(\sigma_1^2 + \sigma_n^2)$ to align the search direction with $\mathbf{v}_n$, i.e., to eliminate the components of the gradient along the other vectors $\mathbf{v}_i$. This yields a significant improvement of convergence speed over the SD method, as shown by the numerical experiments reported in [17]. We note that the name SDA stands for "Steepest Descent with Alignment", i.e., it refers to the alignment property mentioned above.

The SDC method, proposed in [16], uses as $\bar{\alpha}_s$ the Yuan steplength [39]

$$\alpha_s^Y = 2\left(\sqrt{\left(\frac{1}{\alpha_{s-1}^{SD}} - \frac{1}{\alpha_s^{SD}}\right)^2 + 4\frac{\|\mathbf{g}_s\|^2}{\left(\alpha_{s-1}^{SD}\|\mathbf{g}_{s-1}\|\right)^2}} + \frac{1}{\alpha_{s-1}^{SD}} + \frac{1}{\alpha_s^{SD}}\right)^{-1}. \tag{12}$$

We observe that SDC is different from the Dai-Yuan (DY) method that combines Cauchy steps with steps using (12) as specified in [14, formulas (5.2) and (5.3)], since in the latter case α_s^Y is recomputed every time it is used. We also note that the name SDC was chosen in [16] to remind that SD steplengths are alternated with (Yuan) Constant steplengths (more generally, it could be applied to any gradient method using (9)). In order to explain the effect of setting $\bar{\alpha}_s = \alpha_s^Y$, we give a result showing the asymptotic behavior of the Yuan steplength.

Proposition 2.2 *Let $\{\mathbf{x}_k\}$ be the sequence of iterates generated by the SD method applied to problem (2), starting from any point $\mathbf{x}_0$, and suppose that (7) and (8) hold. Then*

$$\lim_{k\to\infty} \alpha_k^Y = \frac{1}{\sigma_1^2}. \tag{13}$$

Proof. The thesis follows straightforwardly from Theorem 3.3 in [16]. □

By using (13) and (6), we can conclude that the better the approximation of $1/\sigma_1^2$ provided by α_k^{SD}, the smaller the component along $\mathbf{v}_1$ of the gradient computed by using that steplength. The SDC method is based on the idea of using a finite sequence of Cauchy steps to force the search in a two dimensional space and to get a suitable approximation of $1/\sigma_1^2$ by computing α_s^Y. According to (6), a multiple application of this step is performed to drive toward zero the component of the gradient along the right singular vector $\mathbf{v}_1$, i.e., μ_1^k. In the ideal case where the component along $\mathbf{v}_1$ is completely removed, problem (2) reduces to a $(n-1)$-dimensional problem, and a new sequence of Cauchy steps followed by some steps with a fixed value of α_s^Y can drive toward zero the component along $\mathbf{v}_2$. This procedure can be repeated with the aim of eliminating the components of the gradient according to the decreasing order of the singular values of A. The effectiveness of this approach is confirmed by the numerical experiments reported in [16].

It is worth noting that if problem (2) is ill-conditioned, then

$$\frac{1}{\sigma_1^2 + \sigma_n^2} \approx \frac{1}{\sigma_1^2};$$

in this case, SDA tends to eliminate first the component of the gradient along $\mathbf{v}_1$, similarly to SDC. More generally, SDA fosters the elimination of the components of the gradient corresponding to $\sigma_i \gg \sigma_n$ according to the decreasing size of the singular values.

We note that the steplength selection rule (9), with $\bar{\alpha}_s$ defined by (10) or (12), does not guarantee monotonicity of the gradient method. A way to enforce monotonicity is using $\min\left\{\bar{\alpha}_s, \alpha_k^{SD}\right\}$ instead of $\bar{\alpha}_s$, as it is done in the original version of SDA presented in [17] and in a variant of SDC described in [16], but this may slightly slow down the methods. However, numerical experiments reported in [16] have shown that for small values of m, such as $m = 2$ or $m = 4$, the SDC method shows monotonicity in practice if the constant steplengths provide fairly good approximations of the inverses of the squared singular values, i.e., if h is sufficiently large. On the other hand, too large values of h slow down the method because of the low efficiency of the Cauchy steps. Furthermore, the choice of h is also related to the accuracy requirement, and we have verified that small values of h are effective when very low accuracy is required, as for the problems

considered in this work. Finally, additional numerical experiments have shown that SDA behaves similarly to SDC. Based on the previous considerations, we use $h = 2$ or $h = 3$, and $m = 2$, for all the test problems discussed in the next sections. For these problems we rarely got non-monotonic SDA and SDC iterations.

3 Filter factor analysis

In order to analyze the behavior of SDA and SDC as regularization methods, we express the solution of the least squares problem (2) by using the SVD decomposition (5):

$$\mathbf{x}^\dagger = A^\dagger \mathbf{b} = \sum_{i=1}^{n} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i = \mathbf{x}_{true} + \sum_{i=1}^{n} \frac{\mathbf{u}_i^T \mathbf{n}}{\sigma_i} \mathbf{v}_i, \tag{14}$$

where $\mathbf{x}_{true}$ is the "true" solution of (1). Since the singular values of A decay to zero, the division by small singular values amplifies the corresponding noise components, and the solution $\mathbf{x}^\dagger$ results useless. A regularized solution can be obtained by modifying the least squares solution (14) as

$$\mathbf{x}_{reg} = \sum_{i=1}^{n} \phi_i \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i, \tag{15}$$

where the scalars ϕ_i, called *filter factors*, are such that the components of the solution corresponding to large singular values are preserved ($\phi_i \approx 1$) and those corresponding to small singular values are filtered out ($\phi_i \approx 0$) [29].

The following proposition shows that the iterates of any gradient method can be written in the form (15). We note that expression (17) given in the proposition has been reported also in [10].

Proposition 3.1 *Let $\{\mathbf{x}_k\}$ be the sequence of iterates generated by the SD method applied to problem (2) starting from $\mathbf{x}_0 = 0$. Then*

$$\mathbf{x}_{k+1} = \sum_{i=1}^{n} \phi_i^{k+1} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i, \tag{16}$$

where

$$\phi_i^{k+1} = 1 - \prod_{l=0}^{k} \left(1 - \alpha_l \sigma_i^2\right), \quad i = 1, \ldots, n. \tag{17}$$

Proof. The proof is by induction. By (3), (4) and (5) we get

$$\begin{aligned} \mathbf{x}_1 &= \alpha_0 A^T \mathbf{b} = \alpha_0 V \Sigma^T U^T \mathbf{b} \\ &= \sum_{i=1}^{n} \alpha_0 \sigma_i^2 \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i = \sum_{i=1}^{n} \left(1 - \left(1 - \alpha_0 \sigma_i^2\right)\right) \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i; \end{aligned}$$

thus (16) and (17) hold for $k = 0$. Now we assume that the thesis holds for $k > 0$. By using again (3), (4) and (5), we have

$$
\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{x}_k - \alpha_k A^T(A\mathbf{x}_k - \mathbf{b}) = (I - \alpha_k A^T A)\mathbf{x}_k + \alpha_k A^T \mathbf{b} \\
&= (I - \alpha_k A^T A) \sum_{i=1}^{n} \left(1 - \prod_{l=0}^{k-1} \left(1 - \alpha_l \sigma_i^2\right)\right) \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i + \alpha_k \sum_{i=1}^{n} \sigma_i^2 \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i \\
&= \sum_{i=1}^{n} \left(1 - \prod_{l=0}^{k-1} \left(1 - \alpha_l \sigma_i^2\right)\right) \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \left(1 - \alpha_k \sigma_i^2\right) \mathbf{v}_i + \sum_{i=1}^{n} \alpha_k \sigma_i^2 \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i \\
&= \sum_{i=1}^{n} \left(1 - \prod_{l=0}^{k} \left(1 - \alpha_l \sigma_i^2\right)\right) \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i,
\end{aligned}
$$

i.e., (16) and (17) hold for $k+1$. This completes the proof. □

Proposition 3.1 shows that the scalars (17) play the role of filter factors associated with the $(k+1)$-th iterate. From (17) it follows that if $\alpha_l = 1/\sigma_i^2$ for some $l \leq k$, then $\phi_i^{k+1} = 1$ at the $(k+1)$-th iteration and all subsequent ones. More generally, the better α_l approximates $1/\sigma_i^2$ the closer ϕ_i^{k+1} will be to 1; furthermore, multiple values of α_l close to $1/\sigma_i^2$ push ϕ_i^{k+1} to go toward 1 quickly. We also note that $1/\alpha_l \gg \sigma_i^2$ implies $\phi_i^k \approx 0$. Therefore, the SDA and SDC methods, thanks to the use of their own constant steplengths, are expected to (approximately) reconstruct the components of the pseudoinverse solution (14) according to the decreasing order of the associated singular values. In other words, the tendency of the two methods to push toward zero the components of the gradient, following the decreasing order of the singular values, translates into the approximation of the most significant components of the solution, thus yielding a useful regularization effect.

In order to illustrate this behavior, in Figure 1 we plot the filter factors of SDA and SDC at the k-th iteration, with $k = 5, 10, 20, 40$, for the `heat` test problem from Hansen's Regularization Tools [25]. We consider the ill-conditioned instance of the problem for $p = n = 64$ (its condition number is about 10^{28}) and add Gaussian random noise, scaled to get noise level $nl = \|\mathbf{n}\|/\|A\mathbf{x}_{true}\| = 0.01$. In SDA and SDC we set $h = 3$ and $m = 2$. For comparison purposes, we plot also the filter factors of the CGLS method (henceforth called simply CG) and other gradient methods, i.e., the SD method, the Barzilai-Borwein (BB) method [3] with steplength $\alpha_k^{BB} = \alpha_{k-1}^{SD}$, and the most efficient DY method [14], which uses the steplength

$$
\alpha_k^{DY} = \begin{cases} \alpha_k^{SD} & \text{if } \mathrm{mod}(k,4) = 0, 1, \\ \alpha_k^{Y} & \text{otherwise,} \end{cases}
$$

where, as already observed, α_k^Y is recomputed every time it is applied. BB and DY are included in the comparison because they are among the fastest gradient methods for quadratic programing problems. For the sake of readability, for each value of k we do not represent all the filter factors in the same picture, but we plot on the left the filter factors associated with SDA, SDC, SD and CG, and on the right those associated with SDA, SDC, DY and BB. Furthermore, in the pictures on the right we zoom on the first 20 filter factors, to better see the differences among the methods considered there.

We see that the filter factors of SDA and SDC behave as expected, i.e., as the number of iterations increases, there is an increasing number of filter factors that are about 1. Both methods are faster than SD in generating filter factors close to 1, i.e., they are faster in reconstructing the significant components of the solution, and SDC is slightly faster than SDA. The filter factors of DY and BB are similar to those of SDA and SDC, but show small oscillations; furthermore, as k grows, SDC appears to generate a larger

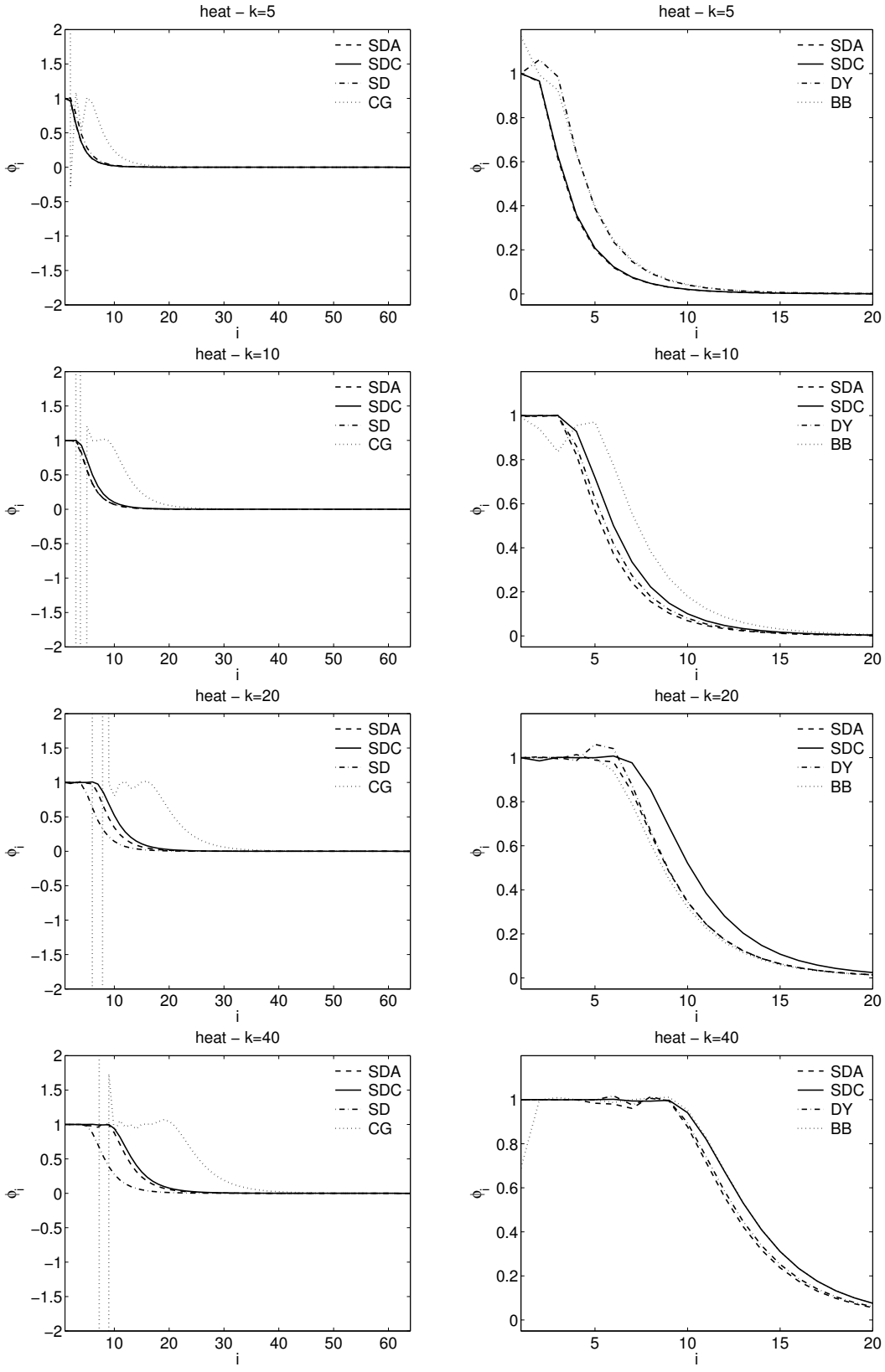

Figure 1: Filter factors of gradient and CG methods applied to `heat` problem, at iterations 5, 10, 20 and 40. Left: comparison of SDA, SDC, SD and CG; right: comparison of SDA, SDC, DY and BB.

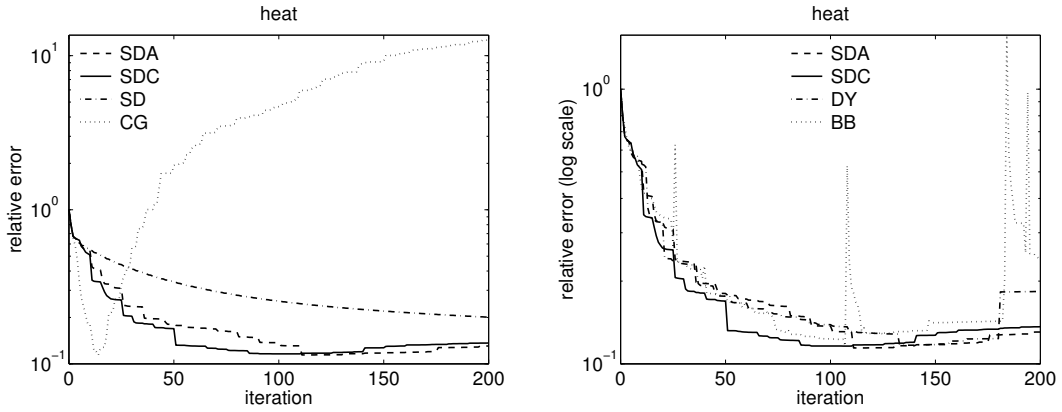

Figure 2: Relative errors of gradient and CG methods applied to `heat` problem. Left: comparison of SDA, SDC, SD and CG; right: comparison of SDA, SDC, DY and BB.

number of filter factors close to 1 than the other methods. As well known, the filter factors of CG become soon oscillating.

We also show the history of the relative error $e_k = \|\mathbf{x}_k - \mathbf{x}_{true}\|/\|\mathbf{x}_{true}\|$ for all the previous methods (Figure 2). As expected, SDC, SDA, DY and BB are much faster than SD in reducing the error. Of course, the relative error of CG achieves its minimum faster than the gradient methods and then rapidly increases, according to the well known CG semiconvergence behavior. Conversely, all the gradient methods but BB exhibit a much more stable convergence behavior. The error increase observed for BB at some iterations is due to the non-monotonicity of the method; as expected, SDA and SDC show in practice a monotonic behavior like DY, which is a monotone method. The errors of SDA, SDC and DY are close each other; however, SDC appears to be slightly faster in decreasing the error. For SDA and SDC, the "staircase" effect in the error curve is due to the alternate use of SD steps and constant steps of type (10) and (12). A similar behavior of the error curve is also observed for DY. By taking into account the previous analysis, in the numerical experiments discussed in the next section we do not consider BB for its strong non-monotonic behavior.

4 Numerical experiments

As pointed out in [6, 26], iterative regularization is particularly attractive for large-scale ill-posed problems such as, for example, image restoration problems. Therefore, we carried out numerical experiments by applying SDA and SDC to image restoration problems of the form (1), that are widely used as benchmark problems. The esperiments were aimed at analyzing the behavior of SDA and SDC as regularization methods, as well as comparing them with CG and SD, whose regularizing properties have been deeply investigated, and with DY, which, to the best of our knowledge, has not been analyzed as a regularizer yet. Here we present the results concerning three problems, which can be considered representative of the general behavior of the different methods.

All the tests were performed by using the Matlab environment. For each problem, the perturbed vector $\mathbf{b}$ was obtained by adding Gaussian white noise to $A\mathbf{x}_{true}$, with noise level $nl = 0.01, 0.025, 0.05, 0.075, 0.1$. For each value of nl, 20 realizations of $\mathbf{n}$ were generated by using the Matlab `randn` function. The gradient and CG methods were implemented in Matlab and executed by using the zero vector as starting guess; the iterations were stopped when 500 iterations were achieved. The parameters in SDA

and SDC were set as follows: $h = 2$ or $h = 3$ (depending on the problem, as specified next) and $m = 2$. The experiments were run using Matlab v. 8.0.0.783 (R2012b) on a Macintosh computer with a dual-core Intel Core i5 processor (1.7 GHz), 4 GB of RAM, 3 MB of cache memory, and the OS X 10.8.5 operating system.

For each experiment we computed the relative error, e_{dp}, at the first iteration where $\|A\mathbf{x}_k - \mathbf{b}\| \leq \|\mathbf{n}\|$, and the minimum relative error, e_{min}, achieved within the maximum number of iterations. Furthermore, in order to provide some measure of the "stability" of the convergence behavior, we considered the cardinality $|\Omega|$ of the following set:

$$\Omega = \{k : 0 \leq k \leq iters \text{ and } e_k \leq e_{dp}\},$$

where $iters = 500$. The rationale behind this choice is that the larger the value of $|\Omega|$ the "flatter" the error curve should be. Note that if the method does not reach e_{dp} within the maximum number of iterations, then $|\Omega| = 0$ and it cannot provide any information on the semiconvergence behavior.

The first set of results concerns the `blur` problem from Hansen's Regularization Tools, which is a benchmark for digital image deblurring. We chose a 64×64 image, corresponding to a square matrix A of dimension $n = 64^2$, and set to 7 and 2 the parameters controlling the sparsity of A and the width of the Gaussian point spread function, respectively, thus obtaining a highly blurred image. The condition number of the resulting matrix A is about 10^{10}. According to the observations at the end of Section 2, we chose $h = 3$ for SDA and SDC.

In Table 1, we report, for each method and each noise level, the values of the relative errors e_{dp} and e_{min}, along with the corresponding iterations k_{dp} and k_{min}, and the value of $|\Omega|$. The data are averaged over the 20 runs associated with the 20 realizations of noise. All the methods generally provide comparable solutions in terms of relative error. As expected, SD is much slower than the other methods; in particular, the value of e_{min} for $nl = 0.01$ can be further reduced by increasing the maximum number of iterations. SDA, SDC and DY show similar behaviors in terms of iterations. For $nl = 0.01$ and $nl = 0.025$, SDC appears slightly faster than SDA and DY in reducing the error; on the other hand, SDA generally achieves the largest value of $|\Omega|$, corresponding to the slowest error increase. As the noise level grows, SDA, SDC and DY become comparable with CG in terms of speed, but have the advantage that their error curves increase much slowly.

To further illustrate the behavior of the five methods, in Figure 3 we plot their relative error histories for two realizations of noise with $nl = 0.025, 0.075$ (for the sake of readability we consider only the first 150 iterations; iteration 0 corresponds to the starting guess). In particular, we see that the error increases more slowly for SDA than for SDC; this agrees with the fact that on the average SDC is faster than SDA in recovering the components of the solution according to the decreasing order of the singular values, and hence it is also faster in reconstructing the components strongly affected by noise.

We also show, in Figure 4, the original `blur` image to be restored, noisy and blurred versions of it for $nl = 0.025$ and $nl = 0.075$, and the best images reconstructed with SDA and CG (the best images obtained with the other gradient methods are not shown because they are practically indistinguishable from these ones). We see that the quality of the restored images is the same for the two methods. Of course, we did not expect SDA, SDC or any of the other methods considered here to provide reconstructions of better visual quality. Actually, other image restoration techniques, using a priori information on the exact solution, have to be used in order to obtain high quality images, especially when high noise levels are considered (see [7, 9, 27, 38] and the references therein). We rather want to show that the regularizing properties of SDA and SDC make them comparable or preferable to other well-known iterative methods for the solution of large-scale ill-posed problems, and therefore they can be potentially exploited in combination with gradient projection techniques (see, e.g., [33, 11, 5]).

nl	*method*	e_{dp}	k_{dp}	e_{min}	k_{min}	$\|\Omega\|$
	SDA	0.355	50	0.344	156	332
	SDC	0.355	34	0.344	110	196
0.01	DY	0.354	50	0.344	122	143
	SD	0.355	205	0.348	500	296
	CG	0.354	29	0.345	60	70
	SDA	0.373	19	0.361	61	100
	SDC	0.369	21	0.361	48	58
0.025	DY	0.372	20	0.361	54	66
	SD	0.373	53	0.361	292	448
	CG	0.372	15	0.362	29	30
	SDA	0.387	11	0.376	27	44
	SDC	0.387	11	0.376	26	21
0.05	DY	0.388	10	0.376	30	38
	SD	0.388	20	0.376	95	259
	CG	0.387	9	0.377	17	16
	SDA	0.395	10	0.387	18	20
	SDC	0.397	8	0.387	19	21
0.075	DY	0.392	9	0.387	18	21
	SD	0.398	12	0.386	49	120
	CG	0.395	7	0.388	12	10
	SDA	0.403	7	0.395	15	14
	SDC	0.402	7	0.394	15	14
0.1	DY	0.405	7	0.395	13	18
	SD	0.405	9	0.394	30	69
	CG	0.402	6	0.396	9	7

Table 1: Numerical results for the `blur` test problem (mean values over 20 realizations of noise).

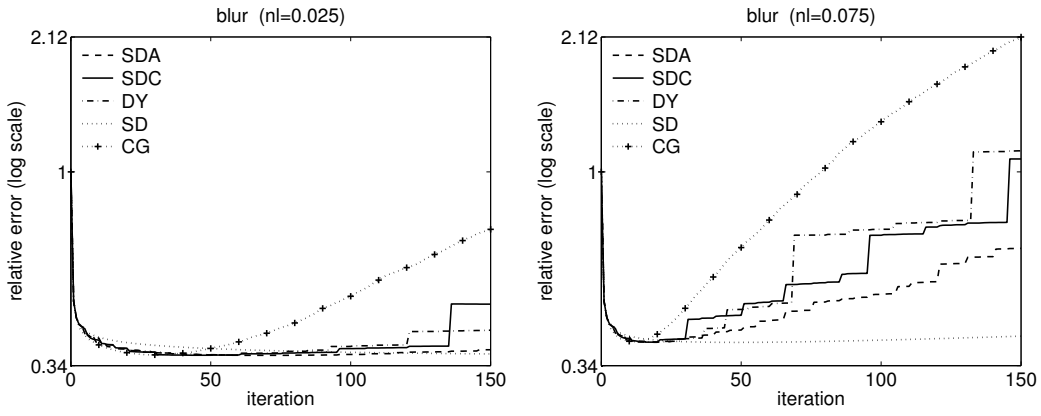

Figure 3: Relative error histories of the gradient and CG methods applied to `blur`. Left: $nl = 0.025$; right: $nl = 0.075$.

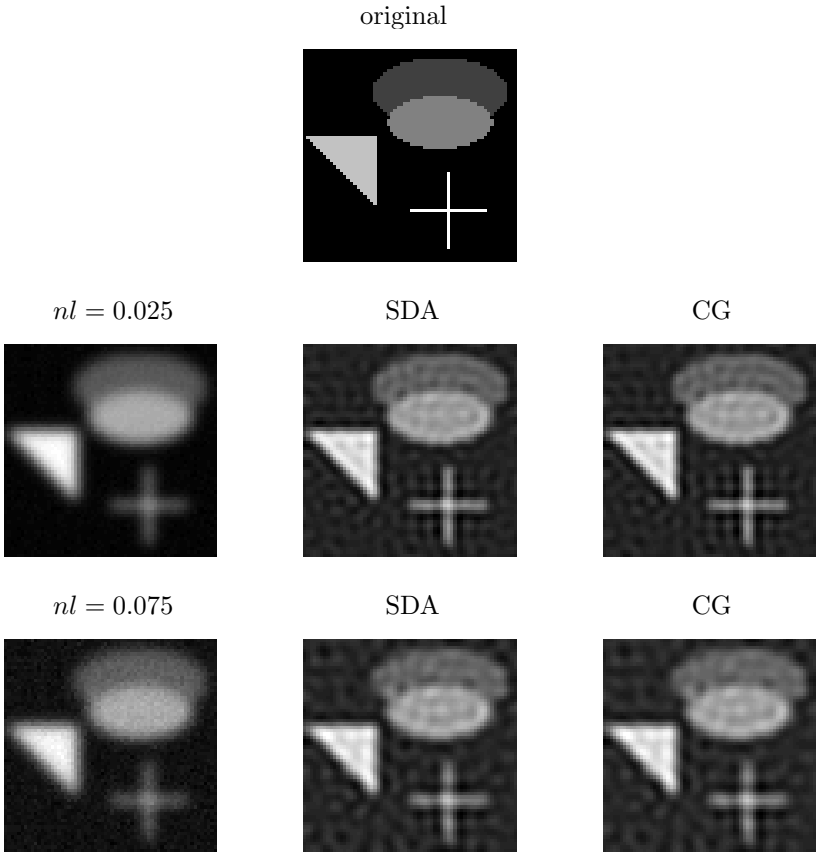

Figure 4: `blur` test problem – top: original image; middle: blurred and noisy image and best reconstructions with SDA and CG for $nl = 0.025$; bottom: blurred and noisy image and best reconstructions with SDA and CG for $nl = 0.075$.

The second set of experiments was performed on a parallel-beam tomography test problem created with the Matlab function `paralleltomo` from Hansen's AIR Tools package [30]. We considered a 50×50 image, 36 angles with values from 0 to 179 degrees, and 75 parallel rays for each angle. The resulting matrix A has dimension 2700×2500 and condition number of order 10^{15}. We used $h = 3$ for SDA and SDC.

The results reported in Table 2 (averaged over 20 realizations of noise) show that all the methods have the same general behavior as for the `blur` test case. In summary, SDA and SDC, as well as DY, are much faster than SD in reducing the error and are only slightly slower than CG; furthermore, for SDA, SDC and DY the error increase due to semiconvergence is much slower than it is for CG. For small values of nl, SDC appears faster than SDA and DY, but SDA shows a better semiconvergence. The relative error histories plotted in Figure 5, corresponding to two realizations of noise with $nl = 0.025, 0.075$, confirm the previous findings.

Figure 6 shows the exact image and the best images obtained with CG and SDA for $nl = 0.025$. As for the `blur` test case, the visual quality of the restored images is comparable; furthermore, they are comparable with the images obtained with the other gradient methods considered here. The same comments apply to the other noise levels.

nl	*method*	e_{dp}	k_{dp}	e_{min}	k_{min}	$\|\Omega\|$
	SDA	0.313	36	0.274	203	425
	SDC	0.316	30	0.275	152	349
0.01	DY	0.316	29	0.275	165	378
	SD	0.317	144	0.295	500	357
	CG	0.316	16	0.277	51	88
	SDA	0.347	17	0.316	72	131
	SDC	0.342	16	0.316	59	96
0.025	DY	0.337	21	0.316	70	86
	SD	0.347	57	0.316	499	444
	CG	0.344	9	0.318	23	28
	SDA	0.382	12	0.357	34	55
	SDC	0.385	13	0.358	27	36
0.05	DY	0.387	13	0.358	28	47
	SD	0.389	31	0.357	127	470
	CG	0.380	7	0.359	13	14
	SDA	0.408	11	0.394	20	27
	SDC	0.416	11	0.397	23	22
0.075	DY	0.420	9	0.397	21	30
	SD	0.427	22	0.393	64	255
	CG	0.411	6	0.399	10	9
	SDA	0.462	10	0.431	15	24
	SDC	0.459	8	0.430	15	18
0.1	DY	0.440	9	0.429	19	12
	SD	0.460	16	0.427	41	127
	CG	0.452	5	0.437	7	7

Table 2: Numerical results for the `paralleltomo` test problem (mean values over 20 realizations of noise).

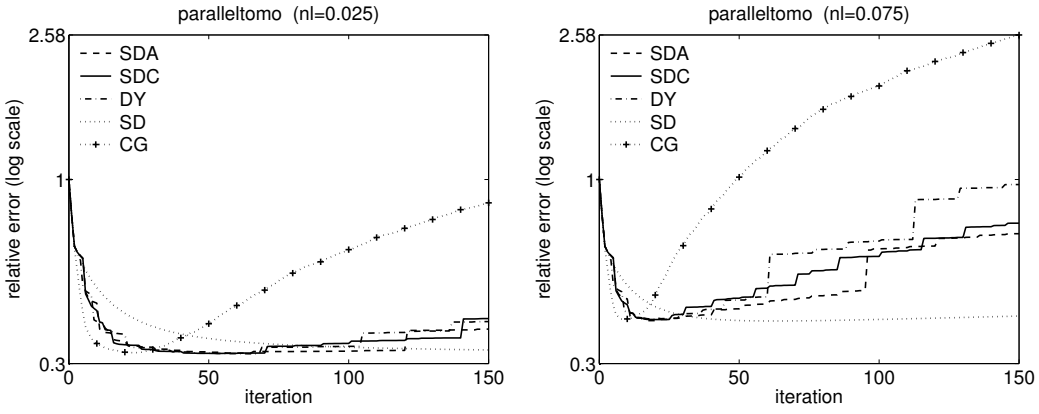

Figure 5: Relative error histories of the gradient and CG methods applied to `paralleltomo`. Left: $nl = 0.025$; right: $nl = 0.075$.

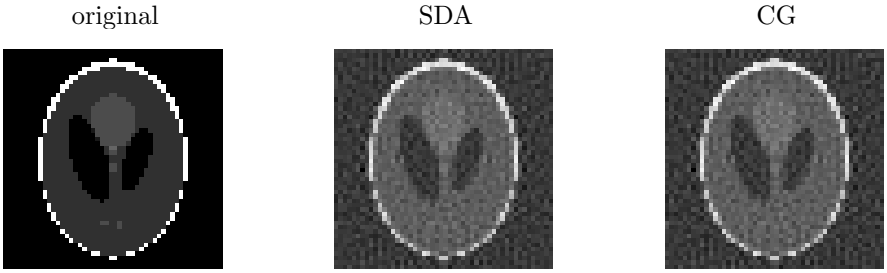

Figure 6: `paralleltomo` test problem ($nl = 0.025$) – original image and best reconstructions with SDA and CG.

nl	*method*	e_{dp}	k_{dp}	e_{min}	k_{min}	$\|\Omega\|$
	SDA	0.300	194	0.293	360	277
	SDC	0.300	170	0.293	321	268
0.01	DY	0.301	172	0.293	339	288
	SD	———	—	0.368	500	—
	CG	0.301	55	0.293	84	55
	SDA	0.327	111	0.319	195	151
	SDC	0.328	100	0.319	178	147
0.025	DY	0.328	110	0.319	185	157
	SD	———	—	0.370	500	—
	CG	0.328	37	0.320	49	30
	SDA	0.356	82	0.351	115	55
	SDC	0.358	70	0.350	102	58
0.05	DY	0.357	82	0.350	115	58
	SD	———	—	0.375	500	—
	CG	0.358	27	0.351	34	13
	SDA	0.381	60	0.373	86	45
	SDC	0.381	58	0.373	83	42
0.075	DY	0.380	65	0.374	80	36
	SD	0.386	457	0.382	500	44
	CG	0.380	23	0.373	27	10
	SDA	0.399	51	0.390	76	44
	SDC	0.396	50	0.390	67	29
0.1	DY	0.399	55	0.392	75	34
	SD	0.405	343	0.393	500	158
	CG	0.395	21	0.391	24	6

Table 3: Numerical results for the `satellite` test problem (mean values over 20 realizations of noise). "—" indicates that e_{dp} was not reached.

The last set of experiments reported here was performed on the well-known `satellite` image from Nagy's RestoreTools package [35]. The observed image was generated by convolving the original image with the RestoreTools PSF that simulates the blurring effect of a ground-based telescope, and then adding Gaussian noise. The image size is 256×256, corresponding to a square matrix A of dimension 256^2; the condition number

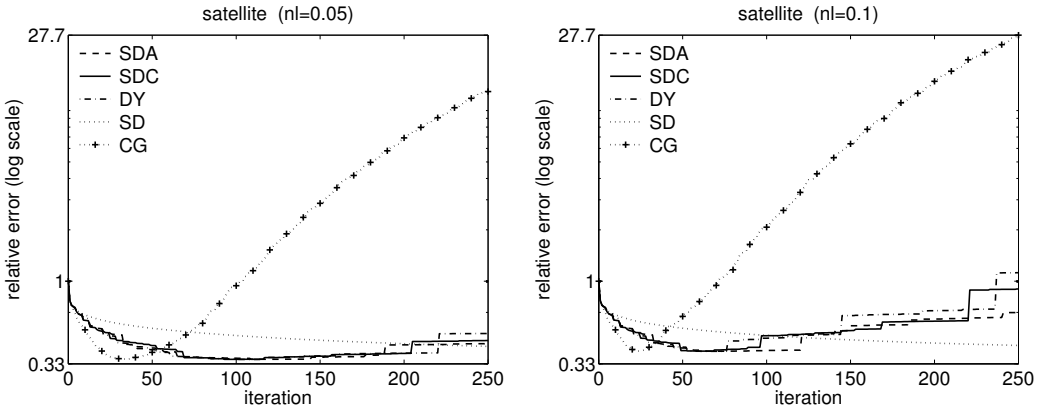

Figure 7: Relative error histories of the gradient and CG methods applied to `satellite`. Left: $nl = 0.05$; right: $nl = 0.1$.

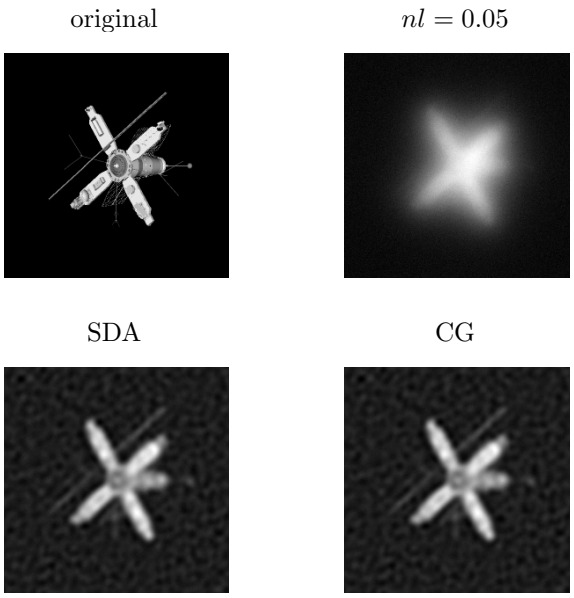

Figure 8: `satellite` test problem ($nl = 0.05$) – original image, noisy and blurred image, and best reconstructions with SDA and CG.

of A is about 10^6. For this test problem, which is less ill-conditioned than the previous ones, using $h = 2$ instead of $h = 3$ turned out to be somewhat more effective. The results shown next were obtained with $h = 2$.

The data in Table 3 show that SDA and SDC, as well as DY, are much faster than SD in reducing the error; for $nl = 0.01, 0.025, 0.05$, SD is not able to achieve the discrepancy error e_{dp} within 500 iterations. On the other hand, SDA, SDC and DY are slower than CG; on the average, they require about three times the number of iterations of CG to satisfy the discrepancy principle. From the values of $|\Omega|$ we can deduce that the corresponding errors grow much slowly than the error of CG; this is confirmed by the

relative error histories reported in Figure 7 for $nl = 0.05, 0.1$, which are representative of the semiconvergence behavior obtained with all the noise levels. We also note that SDC is often slightly faster than SDA and DY, but in practice the three methods can be considered comparable.

We conclude this section showing the original satellite image, the noisy and blurred image corresponding to $nl = 0.05$, and the best SDA and CG reconstructions. Once again, the SDA image is practically the same as the CG image. This also holds for the best images obtained with SDC and DY.

5 Conclusions

Our analysis shows that the SDA and SDC methods applied to discrete linear inverse problems have nice filtering properties. More precisely, the tendency of the two methods to push toward zero the eigencomponents of the gradient, according to the decreasing order of the singular values, allows to approximate first the most significant part of the solution. Therefore, SDA and SDC not only are much faster than the classical SD method, but also have a regularizing effect. Furthermore, our numerical experiments show that SDA and SDC are competitive with CG on severely ill-conditioned problems with high noise levels. In this case, the two methods are slightly slower than CG in reducing the error, but exhibit a better semiconvergence behavior, i.e., the associated error increases more slowly after reaching its minimum value. In comparing SDA and SDC with DY, we also analyzed the regularization properties of DY, which resulted to behave similarly to SDA and SDC. However, SDC often appears slightly faster than the other two methods, while SDA generally shows a slightly slower error increase.

Finally, we observe that effective and popular approaches to the numerical treatment of inverse problems reformulate the original linear inverse problem as a linear least squares problem with constraints that take into account a priori information on the solution, such as non-negativity, sparsity, or other statistical properties. Therefore, there has been an increasing interest in the development of projected methods able to effectively solve such constrained problems (see, e.g., [8, 19, 32, 4, 5, 31]). For this reason, we intend to investigate the behavior of SDA, SDC, and other efficient gradient methods with regularization properties, within projected gradient frameworks such as those discussed in [33, 15].

References

[1] H. Akaike. On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method. *Ann. Inst. Stat. Math. Tokyo*, 11(1):1–16, 1959.

[2] U. M. Ascher, K. van den Doel, H. Huang, and B. F. Svaiter. Gradient descent and fast artificial time integration. *ESAIM: Math. Model. Numer. Anal.*, 43(4):689–708, 2009.

[3] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA J. Numer. Anal.*, 8(1):141–148, 1988.

[4] S. Becker, J. Bobin, and E. Candes. NESTA: a fast and accurate first-order method for sparse recovery. *SIAM J. Imaging Sci.*, 4(1):1–39, 2011.

[5] F. Benvenuto, R. Zanella, L. Zanni, and M. Bertero. Nonnegative least-squares image deblurring: improved gradient projection approaches. *Inverse Problems*, 26(2):025004, 2010.

[6] S. Berisha and J. G. Nagy. Iterative methods for image restoration. In R. Chjellappa and S. Theodoridis, editors, *Academic Press Library in Signal Processing: Volume 4. Image, Video Processing and Analysis, Hardware, Audio, Acoustic and Speech Processing*, chapter 7, pages 193–247. Academic Press, first edition, 2014.

[7] M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging*. IOP Publishing, Bristol, 1998.

[8] D. Calvetti, G. Landi, L. Reichel, and F. Sgallari. Non-negativity and iterative methods for ill-posed problems. *Inverse Problems*, 20(6):1747–1758, 2004.

[9] T. Chan and J. Shen. *Image Processing And Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. SIAM, Philadelphia, PA, USA, 2005.

[10] A. Cornelio, F. Porta, M. Prato, and L. Zanni. On the filtering effect of iterative regularization algorithms for discrete inverse problems. *Inverse Problems*, 29(12):125013, 2013.

[11] Y.-H. Dai and R. Fletcher. Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numer. Math.*, 100(1):21–47, 2005.

[12] Y.-H. Dai, W.W. Hager, K. Schittkowski, and H. Zhang. The cyclic Barzilai-Borwein method for unconstrained optimization. *IMA J. Numer. Anal.. Anal.*, 26(3):604–627, July 2006.

[13] Y.-H. Dai and Y. Yuan. Alternate minimization gradient method. *IMA J. Numer. Anal.*, 23(3):377–393, 2003.

[14] Y.-H. Dai and Y. Yuan. Analysis of monotone gradient methods. *J. Ind. Manag. Optim.*, 1(2):181–192, 2005.

[15] P. L. De Angelis and G. Toraldo. On the identification property of a projected gradient method. *SIAM J. Numer. Anal.*, 30(5):1483–1497, 1993.

[16] R. De Asmundis, D. di Serafino, W.W. Hager, G. Toraldo, and H. Zhang. An efficient gradient method using the Yuan steplength. *Comput. Optim. Appl.*, DOI: 10.1007/s10589-014-9669-5, 2014.

[17] R. De Asmundis, D. di Serafino, F. Riccio, and G. Toraldo. On spectral properties of steepest descent methods. *IMA J. Numer. Anal.*, 33(4):1416–1435, 2013.

[18] H.W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*, volume 375 of *Mathematics and Its Applications*. Springer, 2000.

[19] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright. Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. *IEEE J. Sel. Top. Signal Process.*, 1(4):586–597, 2007.

[20] R. Fletcher. A limited memory steepest descent method. *Math. Program., Ser. A*, 135(1–2):413–436, 2012.

[21] G. Frassoldati, L. Zanni, and G. Zanghirati. New adaptive stepsize selections in gradient methods. *J. Ind. Manag. Optim.*, 4(2):299–312, 2008.

[22] A. Friedlander, J. M. Martínez, B. Molina, and M. Raydan. Gradient method with retards and generalizations. *SIAM J. Numer. Anal.*, 36(1):275–289, 1999.

[23] G. Golub, M. Heath, and W. Wahba. Generalized Cross-Validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

[24] M. Hanke. *Conjugate gradient type methods for ill-posed Problems.* Pitman Research Notes in Mathematics. Longman Scientific & Technical, Harlow, Essex, 1995.

[25] P. C. Hansen. Regularization Tools: A Matlab package for analysis and solution of discrete ill-posed problems. *Numer. Algorithms*, 6(1):1–35, 1994.

[26] P. C. Hansen and T. K. Koldborg. Noise propagation in regularizing iteration for image deblurring. *ETNA*, 31:204–220, 2008.

[27] P. C. Hansen, J. G. Nagy, and D. P. O'Leary. *Deblurring images. Matrices, spectra and filtering.* SIAM, Philadelphia, PA, USA, 2006.

[28] P. C. Hansen and D. P. O'Leary. The use of the L-curve in the regularization of discrete ill-posed problems. *SIAM J. Sci. Comput.*, 14(6):1487–1503, 1993.

[29] P.C. Hansen. *Rank-deficient and discrete ill-posed problems.* SIAM, Philadelphia, 1998.

[30] P.C. Hansen and M. Saxild-Hansen. AIR Tools - a MATLAB package of algebraic iterative reconstruction methods. *J. Comput. Appl. Math.*, 236(8):2167–2178, 2012.

[31] G. Landi and E. Loli Piccolomini. An improved Newton projection method for nonnegative deblurring of Poisson-corrupted images with Tikhonov regularization. *Numer. Algorithms*, 60(1), 2012.

[32] I. Loris, M. Bertero, C. De Mol, R. Zanella, and L. Zanni. Accelerating gradient projection methods for ℓ_1-constrained signal recovery by steplength selection rules. *Applied and Computational Harmonic Analysis*, 27(2):247–254, 2009.

[33] J. J. Moré and G. Toraldo. Algorithms for bound constrained quadratic programming problems. *Numer. Math.*, 55(4):377–400, 1989.

[34] V. A. Morozov. *Regularization methods for ill-posed problems.* CRC Press, Boca Raton, FL, 1993.

[35] J. G. Nagy. RestoreTools: an object oriented Matlab package for image restoration. `http://www.mathcs.emory.edu/~nagy/`.

[36] J. G. Nagy and K. M. Palmer. Steepest descent, CG, and iterative regularization of ill-posed problems. *BIT*, 43(5):1003–1017, 2003.

[37] J. Nocedal, A. Sartenaer, and C. Zhu. On the behavior of the gradient norm in the steepest descent method. *Comp. Optim. Appl.*, 22(1):5–35, 2002.

[38] C. R. Vogel. *Computational Methods for Inverse Problems.* SIAM, Philadelphia, PA, USA, 2002.

[39] Y. Yuan. A new stepsize for the steepest descent method. *J. Comp. Math.*, 24(2):149–156, 2006.

[40] Y. Yuan. Step–sizes for the gradient method. *AMS/IP Studies in Advanced Mathematics*, 42(2):785–796, 2008.