IMA Journal of Numerical Analysis (2013) **33**, 1416–1435
doi:10.1093/imanum/drs056
Advance Access publication on May 6, 2013

On spectral properties of steepest descent methods

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

Daniela di Serafino
Dipartimento di Matematica e Fisica, Seconda Università di Napoli, Viale A. Lincoln 5, 81100 Caserta, Italy
and
Istituto di Calcolo e Reti ad Alte Prestazioni, CNR, Via P. Castellino 111, 80131 Napoli, Italy
daniela.diserafino@unina2.it

Filippo Riccio
Institut für Mathematik, Universität Würzburg, Campus Hubland Nord, Emil-Fischer-Straße 31, 97074 Würzburg, Germany
filippo.riccio@mathematik.uni-wuerzburg.de

and

Gerardo Toraldo*
Dipartimento di Matematica e Applicazioni "R. Caccioppoli", Università di Napoli Federico II, Complesso Universitario Monte Sant'Angelo, Via Cinthia, 80126 Napoli, Italy
*Corresponding author: toraldo@unina.it

[Received on 8 February 2012; revised on 29 November 2012]

In recent years, it has become increasingly clear that the critical issue in gradient methods is the choice of the step length, whereas using gradient as the search direction may lead to very effective algorithms, whose surprising behaviour has only been partially explained, mostly in terms of the spectrum of the Hessian matrix. On the other hand, the convergence of the classical Cauchy steepest descent (SD) method has been analysed extensively and related to the spectral properties of the Hessian matrix, but the connection with the spectrum of the Hessian has not been exploited much to modify the method in order to improve its behaviour. In this work, we show how, for convex quadratic problems, moving from some theoretical properties of the SD method, second-order information provided by the step length can be exploited to dramatically improve the usually poor practical behaviour of this method. This allows us to achieve computational results comparable with those of the Barzilai and Borwein algorithm, with the further advantage of monotonic behaviour.

Keywords: steepest descent methods; quadratic optimization; Hessian spectral properties.

1. Introduction

Gradient methods for the unconstrained minimization problem

$$\min_{x\in\Re^n} f(x) \tag{1.1}$$

generate a sequence $\{x_k\}$ by the following rule:

$$x_{k+1} = x_k - \alpha_k g_k, \tag{1.2}$$

where $g_k = \nabla f(x_k)$ and the step length $\alpha_k > 0$ depends on the method under consideration. In particular, in the classical (optimal) SD method proposed by Cauchy (1847) for the solution of nonlinear systems of equations, α_k is chosen as

$$\alpha_k^{\mathrm{SD}} = \operatorname*{argmin}_{\alpha} f(x_k - \alpha g_k). \tag{1.3}$$

Since the theoretical properties of gradient methods derive from the minimization of a convex quadratic function, we focus our attention on the model problem

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{2}x^{\mathrm{T}}Ax - b^{\mathrm{T}}x, \tag{1.4}$$

where $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite and $b \in \mathbb{R}^n$. This is a simple setting suitable for analysing the relevance of the eigenvalues of the Hessian of the objective function to the behaviour of the algorithms we consider; furthermore, it allows us to highlight the ability of the SD method to automatically reveal some second-order information about the problem, which can be conveniently exploited to dramatically improve the usually poor behaviour of the method. For problem (1.4), the Cauchy step length α_k^{SD} can be computed exactly as the reciprocal of the Rayleigh quotient of A at g_k, i.e.,

$$\alpha_k^{\mathrm{SD}} = \frac{g_k^{\mathrm{T}} g_k}{g_k^{\mathrm{T}} A g_k}. \tag{1.5}$$

The SD method, despite the minimal storage requirements and very low computational cost per iteration, has long been considered very bad and ineffective because of its slow convergence rate and its oscillatory behaviour. However, in the last twenty years, the interest in gradient methods has been renewed after the innovative approach of Barzilai and Borwein (BB) (Barzilai & Borwein, 1988), which stimulated novel choices for α_k in (1.2), and proved to be largely superior to the Cauchy step length (1.5). In the BB approach, α_k is computed through a secant condition by imposing either

$$\min_{\alpha} \|s_{k-1} - \alpha y_{k-1}\| \tag{1.6}$$

or

$$\min_{\alpha} \left\| \frac{1}{\alpha} s_{k-1} - y_{k-1} \right\|, \tag{1.7}$$

where $\|\cdot\|$ is the L_2 vector norm, $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$, thus obtaining the following step lengths, respectively:

$$\alpha_k^{\mathrm{BB1}} = \frac{\|s_{k-1}\|^2}{s_{k-1}^{\mathrm{T}} y_{k-1}}, \tag{1.8}$$

$$\alpha_k^{\mathrm{BB2}} = \frac{s_{k-1}^{\mathrm{T}} y_{k-1}}{\|y_{k-1}\|^2}, \tag{1.9}$$

for which the inequality $\alpha_k^{\mathrm{BB1}} \geqslant \alpha_k^{\mathrm{BB2}}$ holds (see, for instance, Raydan & Svaiter, 2002, Lemma 2.1).

The step length α_k^{BB1} is equal to $\alpha_{k-1}^{\mathrm{SD}}$, i.e., the Cauchy step length at the previous iteration, while α_k^{BB2} is equal to $\alpha_{k-1}^{\mathrm{SD}g}$, where

$$\alpha_k^{\mathrm{SD}g} = \frac{g_k^{\mathrm{T}} A g_k}{g_k^{\mathrm{T}} A^2 g_k} = \operatorname*{argmin}_{\alpha} \|\nabla f(x_k - \alpha g_k)\| = \operatorname*{argmin}_{\alpha} \|(I - \alpha A) g_k\|. \tag{1.10}$$

We note that (1.10) can be interpreted as the Cauchy step for the convex quadratic problem

$$\min_x \|Ax - b\| = \min_x \tfrac{1}{2} x^{\mathrm{T}} A^2 x - (Ab)^{\mathrm{T}} x, \tag{1.11}$$

which is obviously equivalent to (1.4). Therefore, both the BB step lengths (1.8) and (1.9) can be seen as Cauchy step lengths with one delay. The use of larger delays was investigated in Friedlander *et al.* (1999), extending the convergence results which hold for the BB method (Raydan, 1993; Dai & Liao, 2002). A deeper analysis of the asymptotic behaviour of the BB and related methods is proposed in Dai & Fletcher (2005). Fletcher (2005) makes some intuitive considerations about the relationship between the nonmonotonicity of such methods and their surprising computational performance; he also discusses the circumstances under which the BB (and related) methods might be competitive with the conjugate gradient (CG) method and he argues that the former represent an effective alternative to the latter when moving from (1.4) to constrained or nonquadratic problems (see also Birgin *et al.*, 2000; Dai & Fletcher, 2006; Hager & Zhang, 2006; Andretta *et al.*, 2010). As observed in Friedlander *et al.* (1999), gradient methods are very competitive with the CG method when low accuracy in the solution is required, for instance, in the context of inexact Newton methods. Furthermore, in the last years, gradient methods have been successfully used in practice, for instance, in the application to certain ill-posed inverse problems, where the SD method shows a smoothing, regularizing effect and where a strict optimization solution is not necessary such as in image deblurring and denoising problems (Bertero *et al.*, 2008; Huang, 2008).

All of these observations, illustrated in Friedlander *et al.* (1999), Raydan & Svaiter (2002), Fletcher (2005) and Bonettini *et al.* (2008), justify the interest in designing effective gradient methods and the need for better understanding their behaviour. In recent years, it has become increasingly clear that the critical issue in gradient methods is the choice of the step length, whereas using the gradient as search direction may lead to very effective algorithms. The surprising behaviour of these algorithms has been only partially explained (Raydan, 1997; Fletcher, 2005; Dai & Yuan, 2005), pointing out that the effectiveness of the approach is related to the way the eigencomponents of the gradient with respect to A decrease.

For the SD method, convergence has been analysed extensively and related to the spectral properties of the Hessian matrix A, for instance, in the pioneering works of Akaike (1959) and Forsythe (1968).

However, the connection with the spectrum of A has not been exploited much to modify the SD method in order to improve its behaviour. The recurrence

$$g_{k+1} = g_k - \alpha_k A g_k = \alpha_k \left(\frac{1}{\alpha_k} g_k - A g_k \right), \tag{1.12}$$

which holds for any gradient method, suggests that in order to get faster convergence, a greedy approach like (1.3) might be unsatisfactory, whereas fostering the search direction to align with an eigendirection of A could speed up the convergence of the algorithm (Frassoldati *et al.*, 2008).

We will show how, moving from some theoretical properties of the SD method, second-order information provided by the step length (1.5) can be exploited in order to improve dramatically the usually poor practical behaviour of the Cauchy method, achieving computational results comparable with those of the BB method, while preserving monotonicity.

This paper is organized as follows. In Section 2, some classical convergence results for the SD method are briefly reviewed, which are the theoretical basis of the analysis carried out in the rest of the paper. In Section 3, we highlight that the sequence of Cauchy step lengths has the nice feature of providing an approximation to the sum of the extreme eigenvalues of the Hessian. Based on that we propose a modification of the SD method, called steepest descent with alignment (SDA), aimed at aligning the search direction with the eigendirection corresponding to the smallest eigenvalue and then to eventually force the algorithm into the one-dimensional subspace spanned by that eigendirection. In Section 4, we show that a gradient method where the step length is twice the Cauchy step length (1.5) eventually ends up in a one-dimensional subspace spanned by the eigenvector associated with the largest eigenvalue. This result gives further motivation for the relaxed Cauchy steepest descent (RSD) method by Raydan & Svaiter (2002) and actually suggests that it is worth fostering an over-relaxation. Finally, in Section 5, we provide some numerical evidence of the performance of the algorithmic approaches presented in Sections 3 and 4, compared with the standard BB algorithm.

In the rest of this paper, we denote by $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ the eigenvalues of the matrix A and by $\{d_1, d_2, \ldots, d_n\}$ a set of associated orthonormal eigenvectors. We make the following assumptions.

Assumption 1.1 The eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ are such that

$$\lambda_1 > \lambda_2 > \lambda_3 \cdots > \lambda_n > 0.$$

Assumption 1.2 For all the methods considered in this work, any starting point x_0 is such that

$$g_0^{\mathrm{T}} d_1 \neq 0, \quad g_0^{\mathrm{T}} d_n \neq 0.$$

Finally, we denote by x^* the solution of problem (1.4) and by $\kappa(A)$ the spectral condition number of A.

2. The gradient method

The most general gradient method for problem (1.4) iterates according to the following algorithmic framework.

Algorithm 1 (Gradient method)

choose $x_0 \in \Re^n$
$g_0 \leftarrow Ax_0 - b$; $k \leftarrow 0$
while (not stop_condition)
  choose a suitable step length $\alpha_k > 0$
  $x_{k+1} \leftarrow x_k - \alpha_k g_k$; $\quad g_{k+1} \leftarrow g_k - \alpha_k A g_k$
  $k \leftarrow k + 1$
endwhile

For the optimal choice (1.5) of the step length, it is well known that the algorithm has a q-linear rate of convergence that depends on the spectral radius of the Hessian matrix; more precisely, the following result holds.

PROPOSITION 2.1 (Akaike, 1959) The sequence $\{x_k\}$ generated by the SD algorithm converges q-linearly to x^* with a rate of convergence

$$\rho = \frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n}. \tag{2.1}$$

The convergence of Algorithm 1 holds for a choice of the step length much more general than (1.5). If we consider $2\alpha_k^{\mathrm{SD}}$ as the step length, then

$$f(x_k - 2\alpha_k^{\mathrm{SD}} g_k) = f(x_k)$$

and the following decrease condition holds:

$$f(x_k - \alpha g_k) < f(x_k) \quad \text{for all } \alpha \in (0, 2\alpha_k^{\mathrm{SD}}). \tag{2.2}$$

The next proposition states the condition under which Algorithm 1, with a step length inspired by (2.2), converges to the solution of (1.4).

PROPOSITION 2.2 (Raydan & Svaiter, 2002) The sequence $\{x_k\}$ generated by Algorithm 1 with $\alpha_k = \rho_k \alpha_k^{\mathrm{SD}}$, $\rho_k \in [0, 2]$, converges to x^* provided $\{\rho_k\}$ has an accumulation point in $(0, 2)$.

We now present some known formulas that hold for the gradients of the sequence generated by Algorithm 1, for any choice of α_k. First, we observe that

$$g_{k+1} = g_k - \alpha_k A g_k = \prod_{j=0}^{k} (I - \alpha_j A) g_0. \tag{2.3}$$

Furthermore, if

$$g_0 = \sum_{i=1}^{n} \mu_i d_i,$$

then, by (2.3), we have

$$g_{k+1} = \sum_{i=1}^{n} \mu_i^{k+1} d_i, \tag{2.4}$$

where

$$\mu_i^{k+1} = \mu_i \prod_{j=0}^{k} (1 - \alpha_j \lambda_i). \tag{2.5}$$

Formulas (2.3–2.5) have very high relevance in the analysis of the gradient methods, since they allow us to study the convergence in terms of the spectrum of the matrix A. If at the kth iteration, $\mu_i^k = 0$ for some i, it follows from (2.4–2.5) that for $h > k$ it will be $\mu_i^h = 0$ and therefore the component of the gradient along d_i will be zero at all subsequent iterations. We note that the condition $\mu_i^k = 0$ holds if and only if $\mu_i = 0$ or $\alpha_j = 1/\lambda_i$ for some $j \leqslant k$. Furthermore, from (2.3), it follows that the SD method has finite termination if and only if at some iteration the gradient is an eigenvector of A.

The next proposition gives the asymptotic rate of convergence of $\{\mu_1^k\}$ for a quite general choice of the step length in a gradient method.

PROPOSITION 2.3 (Friedlander *et al.*, 1999) In Algorithm 1, if the step length α_k is chosen as the reciprocal of the Rayleigh quotient of A at any nonzero vector, then the sequence $\{\mu_1^k\}$ converges q-linearly to zero with convergence factor $1 - \lambda_n/\lambda_1$.

Friedlander *et al.* (1999) also present a large collection of possible choices of α_k (including some well-known methods) for which $\{\mu_i^k\}$ vanishes for all i.

The next result extends and summarizes previous results of Akaike (1959) concerning the behaviour of the sequences $\{\mu_i^k\}$ in the SD method.

PROPOSITION 2.4 (Nocedal *et al.*, 2002) Let us consider the sequence $\{x_k\}$ generated by the SD method and suppose that Assumptions 1.1–1.2 hold. Then

$$\lim_k \frac{(\mu_n^k)^2}{\sum_{j=1}^n (\mu_j^k)^2} = \begin{cases} \dfrac{c^2}{1+c^2} & \text{for } k \text{ odd}, \\ \dfrac{1}{1+c^2} & \text{for } k \text{ even}, \end{cases} \tag{2.6}$$

$$\lim_k \frac{(\mu_1^k)^2}{\sum_{j=1}^n (\mu_j^k)^2} = \begin{cases} \dfrac{1}{1+c^2} & \text{for } k \text{ odd}, \\ \dfrac{c^2}{1+c^2} & \text{for } k \text{ even}, \end{cases} \tag{2.7}$$

$$\lim_k \frac{(\mu_i^k)^2}{\sum_{j=1}^n (\mu_j^k)^2} = 0 \quad \text{for } 1 < i < n, \tag{2.8}$$

where c is a constant satisfying

$$c = \lim_k \frac{\mu_1^{2k}}{\mu_n^{2k}} = -\lim_k \frac{\mu_n^{2k+1}}{\mu_1^{2k+1}}.$$

Proposition 2.4 shows that the Cauchy method eventually performs its search in the two-dimensional subspace generated by d_1 and d_n, zigzagging between two directions, without being able to eliminate from the basis of the current search direction any of the two components d_1 and d_n and hence to align

the gradient with an eigendirection of the Hessian matrix. Conversely, the nice behaviour of the BB methods is often explained by saying that the nonmonotonicity of such methods produces an erratic path of $1/\alpha_k$ in the interior of the spectrum of A which fosters the sequences $\{\mu_i^k\}$ to go to zero together (Fletcher, 2005; Dai & Yuan, 2005).

3. A new steepest descent method

In this section, we suggest a simple way of modifying the SD method to force the gradients into a one-dimensional subspace as the iterations progress, to avoid the classical zigzag pattern which is the main reason for the slow convergence of the SD method.

We first show that the sequence of step lengths $\{\alpha_k^{\mathrm{SD}}\}$ in the SD method gives asymptotically some meaningful information about the spectrum of the Hessian matrix.

PROPOSITION 3.1 Let us consider the sequence $\{x_k\}$ generated by the SD method applied to problem (1.4) and suppose that Assumptions 1.1–1.2 hold. Then, the sequences $\{\alpha_{2k}^{\mathrm{SD}}\}$ and $\{\alpha_{2k+1}^{\mathrm{SD}}\}$ are converging and

$$\lim_k \left(\frac{1}{\alpha_{2k}^{\mathrm{SD}}} + \frac{1}{\alpha_{2k+1}^{\mathrm{SD}}} \right) = \lambda_1 + \lambda_n. \tag{3.1}$$

Proof. By Nocedal *et al.* (2002, Lemma 3.3), we have

$$\lim_k \alpha_{2k}^{\mathrm{SD}} = \frac{1+c^2}{\lambda_n(1+c^2\gamma)},$$
$$\lim_k \alpha_{2k+1}^{\mathrm{SD}} = \frac{1+c^2}{\lambda_n(\gamma+c^2)},$$

where c is the same constant as in Proposition 2.4 and $\gamma = \kappa(A)$; then (3.1) trivially follows. □

PROPOSITION 3.2 Under Assumptions 1.1–1.2, the sequence $\{x_k\}$ generated by Algorithm 1, with constant step length

$$\hat{\alpha} = \frac{1}{\lambda_1 + \lambda_n}, \tag{3.2}$$

converges to x^*. Moreover,

$$\lim_k \frac{\mu_h^k}{\mu_n^k} = \frac{\mu_h}{\mu_n} \lim_k \left(\frac{\lambda_n}{\lambda_1} + \frac{\lambda_1 - \lambda_h}{\lambda_1} \right)^k = 0, \quad h = 1, 2, \ldots, n-1, \tag{3.3}$$

where μ_i^k $(i = 1, 2, \ldots, n)$ is defined in (2.5).

Proof. Since $\alpha_k^{\mathrm{SD}} \geqslant 1/\lambda_1$ for any k, then $\alpha_k^{\mathrm{SD}} \geqslant \hat{\alpha}$; therefore, Proposition 2.2 applies and $\lim_k x_k = x^*$. From (2.5), we have that

$$\mu_h^k = \mu_h \left(\frac{\lambda_1 + \lambda_n - \lambda_h}{\lambda_n + \lambda_1} \right)^k, \quad \mu_n^k = \mu_n \left(\frac{\lambda_1}{\lambda_n + \lambda_1} \right)^k$$

and (3.3) clearly holds. □

Relation (3.3) indicates that if the hypotheses of Proposition 3.2 hold, then the sequence $\{\mu_h^k\}$, for $h < n$, goes to zero faster than $\{\mu_n^k\}$. Thus, a gradient method with step length (3.2) tends to align the search direction with the eigendirection of A corresponding to the minimum eigenvalue λ_n.

We note that the constant step length (3.2) is half of the theoretically 'optimal' constant step length (see Elman & Golub, 1994)

$$\alpha^{\mathrm{OPT1}} = \frac{2}{\lambda_1 + \lambda_n}, \tag{3.4}$$

which minimizes $\|I - \alpha A\|$. Dai & Yang (2006) proposed a gradient method with

$$\alpha_k^{\mathrm{OPT2}} = \frac{\|g_k\|}{\|Ag_k\|} \tag{3.5}$$

and showed that this step length converges to (3.4) and allows the extreme eigenvalues of A to be approximated. However, despite its nice theoretical features, the step length (3.5) leads only to a slight reduction in the number of iterations with respect to the SD method.

Propositions 2.4 and 3.1 suggest an approach different from that in Dai & Yang (2006), aimed at speeding up the convergence of the SD method by forcing the algorithm search into the one-dimensional subspace spanned by the eigendirection d_n. Of course, computing the exact value of (3.2) is unrealistic, but Proposition 3.1 suggests that, for k sufficiently large,

$$\tilde{\alpha}_k = \left(\frac{1}{\alpha_k^{\mathrm{SD}}} + \frac{1}{\alpha_{k-1}^{\mathrm{SD}}} \right)^{-1} \tag{3.6}$$

can be used as an approximate value for (3.2). Since Proposition 2.4 shows that in the SD method

$$g_k = \mu_1^k d_1 + \mu_n^k d_n + \zeta_k, \tag{3.7}$$

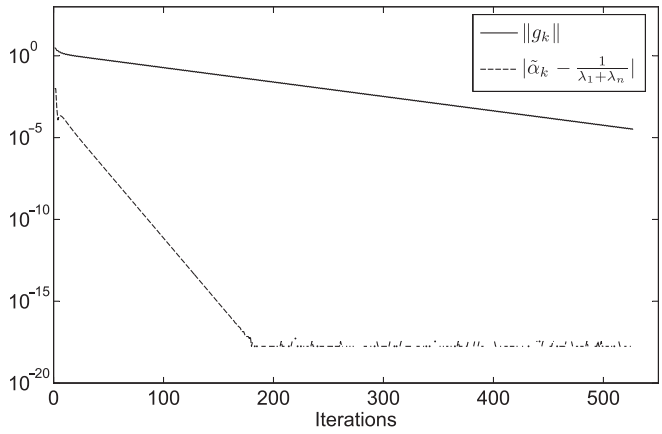
with ζ_k going to zero faster than $\mu_1^k d_1 + \mu_n^k d_n$, our approach is based on the idea of using sequences of Cauchy steps (1.5) which force the search into a two-dimensional space and, at the same time, supplying a suitable approximation of (3.2) to be used in aligning the search direction with d_n.

We consider a modified version of the SD method, called SDA, where step lengths of the form (3.6) are chosen at some selected iterations (see Algorithm 2). More precisely, when the sequence $\{\tilde{\alpha}_k\}$ settles down (see the *switch condition* in Algorithm 2), the SDA method performs h consecutive iterations using as step length the last computed $\tilde{\alpha}_k$, provided it produces a decrease in the objective function (otherwise, SDA adopts the double Cauchy step).

In Fig. 1, we show the values of the sequence $\{|\tilde{\alpha}_k - \hat{\alpha}|\}$ computed by using the Cauchy step lengths resulting from the application of the SD method to problem (1.4), where $n = 10$, A is a randomly generated matrix with $\kappa(A) = 100$, $b = (1, \ldots, 1)^{\mathrm{T}}$ and $x_0 = (0, \ldots, 0)^{\mathrm{T}}$; the stop condition $\|g_k\| < 10^{-5}\|g_0\|$ is used. We note that the sequence goes to zero very fast, although the SD method performs very poorly and needs more than 500 iterations to find a solution with the required accuracy.

In Fig. 2, we report the behaviour of the gradient norm in the SDA method for the above problem with $\varepsilon = 10^{-4}$, for $h = 1$ and $h = 5$. We observe the SDA method largely outperforms the SD method; furthermore, the $\overline{\alpha}$ steps (big dots in the graph) have a rather negligible effect in terms of reduction in the gradient, but a very strong effect in reducing the overall number of iterations. This is because, as expected, such steps have an important role in aligning the search direction with the eigendirection d_n, as shown in Fig. 3. We also note that a value of $h > 1$ tends to further speed up this alignment.

We conclude this section by observing that the step length (3.6) is related to the step length



Q4 FIG. 1. Behaviour of the sequences $\{|\widetilde{\alpha}_k - 1/(\lambda_1 + \lambda_n)|\}$ and $\{\|g_k\|\}$ for the SD method.

Algorithm 2 (SDA)

choose $x_0 \in \Re^n$, $\varepsilon > 0$, h integer
$g_0 \leftarrow Ax_0 - b$
$\alpha_0^{\mathrm{SD}} \leftarrow \frac{g_0^{\mathrm{T}} g_0}{g_0^{\mathrm{T}} A g_0}$; $x_1 \leftarrow x_0 - \alpha_0^{\mathrm{SD}} g_0$; $g_1 \leftarrow Ax_1 - b$
$\alpha_1^{\mathrm{SD}} \leftarrow \frac{g_1^{\mathrm{T}} g_1}{g_1^{\mathrm{T}} A g_1}$; $x_2 \leftarrow x_1 - \alpha_1^{\mathrm{SD}} g_1$; $g_2 \leftarrow Ax_2 - b$
$\widetilde{\alpha}_1 \leftarrow \frac{\alpha_1^{\mathrm{SD}} \alpha_0^{\mathrm{SD}}}{\alpha_1^{\mathrm{SD}} + \alpha_0^{\mathrm{SD}}}$
$k \leftarrow 1$; $s \leftarrow 1$
while (not stop_condition)
  repeat
    $p \leftarrow s$; $k \leftarrow k + 1$
    $\alpha_k^{\mathrm{SD}} \leftarrow \frac{g_k^{\mathrm{T}} g_k}{g_k^{\mathrm{T}} A g_k}$; $x_{k+1} \leftarrow x_k - \alpha_k^{\mathrm{SD}} g_k$; $g_{k+1} \leftarrow g_k - \alpha_k^{\mathrm{SD}} A g_k$
    $\widetilde{\alpha}_k \leftarrow \frac{\alpha_k^{\mathrm{SD}} \alpha_p^{\mathrm{SD}}}{\alpha_k^{\mathrm{SD}} + \alpha_p^{\mathrm{SD}}}$
    $s \leftarrow k$
  until ($|\widetilde{\alpha}_k - \widetilde{\alpha}_p| < \varepsilon$) *switch condition*
  $\widetilde{\alpha} \leftarrow \widetilde{\alpha}_k$
  for $i = 1, h$
    $k \leftarrow k + 1$
    $\alpha_k^{\mathrm{SD}} \leftarrow \frac{g_k^{\mathrm{T}} g_k}{g_k^{\mathrm{T}} A g_k}$
    $\overline{\alpha} \leftarrow \min\{\widetilde{\alpha}, 2\alpha_k^{\mathrm{SD}}\}$
    $x_k \leftarrow x_k - \overline{\alpha} g_k$; $g_k \leftarrow g_k - \bar{\alpha} A g_k$
  endfor
endwhile

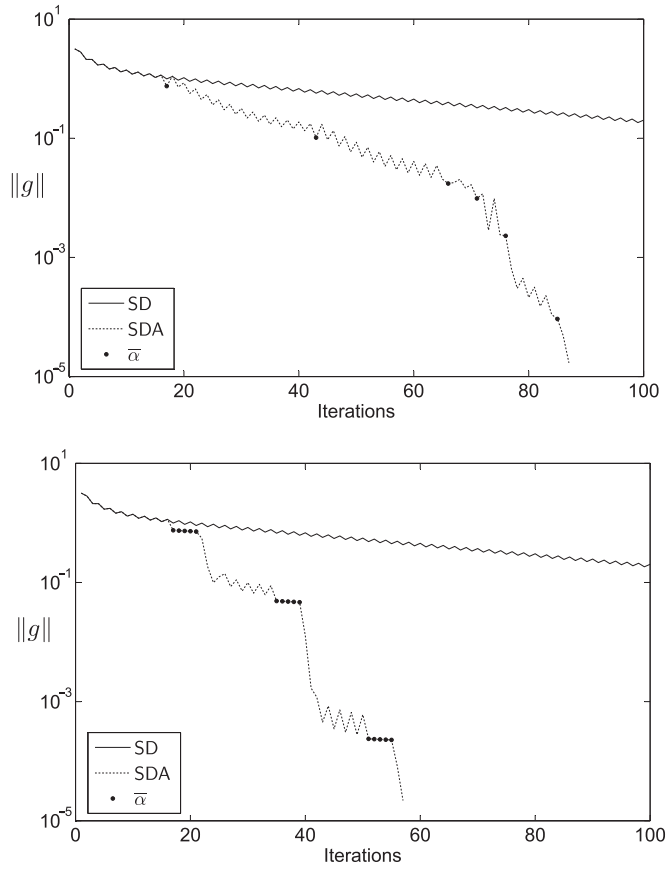


FIG. 2. Convergence of the SDA method, for $h = 1$ (top) and $h = 5$ (bottom).

$$\alpha_k^{\mathrm{DY}} = 2\left(\sqrt{\left(\frac{1}{\alpha_{k-1}^{\mathrm{SD}}} - \frac{1}{\alpha_k^{\mathrm{SD}}}\right)^2 + 4\frac{\|g_k\|^2}{(\alpha_{k-1}^{\mathrm{SD}}\|g_{k-1}\|)^2}} + \frac{1}{\alpha_{k-1}^{\mathrm{SD}}} + \frac{1}{\alpha_k^{\mathrm{SD}}}\right)^{-1}, \tag{3.8}$$

determined by imposing finite termination for two-dimensional quadratic problems (Dai & Yuan, 2005; Yuan, 2006) and that

$$\tilde{\alpha}_k < \alpha_k^{\mathrm{DY}} < \min\{\alpha_{k-1}^{\mathrm{SD}}, \alpha_k^{\mathrm{SD}}\}. \tag{3.9}$$

In their computational analysis, Dai & Yuan (2005) show that the gradient method with

$$\alpha_k = \begin{cases} \alpha_k^{\mathrm{SD}} & \text{if } \mathrm{mod}(k,4) = 1, 2, \\ \alpha_k^{\mathrm{DY}} & \text{otherwise}, \end{cases} \tag{3.10}$$

outperforms other monotone gradient methods and the BB method.

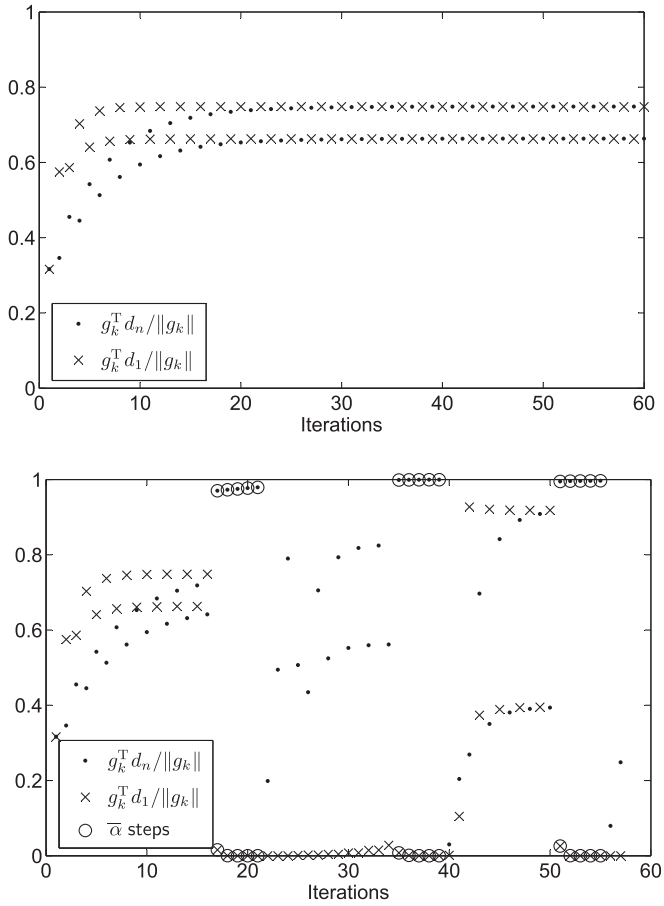


FIG. 3. Behaviour of the (normalized) components of the gradient along the eigendirections d_1 and d_n for the SD method (top) and the SDA method with $h = 5$ (bottom).

4. Relaxed steepest descent method

In this section, we discuss a choice of the step length that fosters the SD algorithm to make its search in the one-dimensional space spanned by d_1. The relaxed steepest descent (RSD) method was suggested by Raydan & Svaiter (2002) who proposed a relaxation of the step length (1.5) in order to accelerate the convergence. They adopt in Algorithm 1 a step length α_k chosen at random in $[0, 2\alpha_k^{\mathrm{SD}}]$, in order to escape from the zigzagging behaviour of the SD method (see Algorithm 3).

Under the hypotheses of Proposition 2.2, the RSD method converges monotonically to x^*. Numerical experiments in Raydan & Svaiter (2002) show that this method largely outperforms the SD method, but the BB method and its Cauchy Barzilai Borwein (CBB) variant, which are nonmonotone, are still the fastest and most effective ones. From their numerical experiments, the authors observe the tendency of the BB and CBB methods to force gradient directions to approximate eigenvectors of the Hessian matrix A; this explains, to some extent, the good behaviour of these methods.

Algorithm 3 (RSD)

choose $x_0 \in \Re^n$
$g_0 \leftarrow Ax_0 - b; \quad k = 0$
while (not stop_condition)
  randomly choose $\alpha_k \in [0, 2\alpha_k^{\mathrm{SD}}]$
  $x_{k+1} \leftarrow x_k - \alpha_k g_k; \quad g_{k+1} \leftarrow g_k - \alpha_k A g_k$
  $k \leftarrow k + 1$
endwhile

The next proposition shows that an over-relaxation of the Cauchy step fosters a similar tendency and suggests a slightly different form of relaxation that produces better effects than a simple random choice of the step length in $[0, 2\alpha_k^{\mathrm{SD}}]$.

PROPOSITION 4.1 Let us consider the sequences $\{x_k\}$ and $\{g_k\}$ generated by the gradient method with

$$\alpha_k = 2\alpha_k^{\mathrm{SD}}; \tag{4.1}$$

then

$$\lim_k \frac{g_{k+1}}{\prod_{j=0}^{k}(1 - \alpha_j\lambda_1)} = \mu_1 d_1, \tag{4.2}$$

$$\lim_k \alpha_k = \frac{2}{\lambda_1}, \tag{4.3}$$

$$\lim_k \nabla f(x_k - \alpha_k^{\mathrm{SD}} g_k) = 0. \tag{4.4}$$

Proof. We have

$$g_{k+1} = \mu_1 \left(\prod_{j=0}^{k}(1 - \alpha_j\lambda_1) \right) d_1 + \sum_{i=2}^{n} \mu_i \left(\prod_{j=0}^{k}(1 - \alpha_j\lambda_i) \right) d_i$$

and hence

$$\frac{g_{k+1}}{\prod_{j=0}^{k}(1 - \alpha_j\lambda_1)} = \mu_1 d_1 + \sum_{i=2}^{n} \mu_i \prod_{j=0}^{k} \frac{(1 - \alpha_j\lambda_i)}{(1 - \alpha_j\lambda_1)} d_i. \tag{4.5}$$

Furthermore,

$$\frac{\lambda_n}{2} \leqslant \frac{1}{\alpha_j} \leqslant \frac{\lambda_1}{2} \tag{4.6}$$

and then

$$1 - \alpha_j\lambda_n \geqslant -1, \quad 1 - \alpha_j\lambda_1 \leqslant -1. \tag{4.7}$$

If we set $\theta = \lambda_1 - \lambda_2$, then $\lambda_1 \geqslant \lambda_i + \theta$ for $i > 1$ and it follows that

$$1 - \alpha_j\lambda_i \geqslant 1 - \alpha_j(\lambda_1 - \theta)$$

and hence, by (4.7),

$$\frac{1-\alpha_j\lambda_i}{1-\alpha_j\lambda_1} \leqslant 1 + \frac{\theta\alpha_j}{1-\alpha_j\lambda_1}. \tag{4.8}$$

By using (4.6), we get

$$\frac{\theta\alpha_j}{1-\alpha_j\lambda_1} = \frac{\theta}{1/\alpha_j - \lambda_1} \leqslant \frac{\theta}{\lambda_n/2 - \lambda_1}$$

and thus

$$\frac{1-\alpha_j\lambda_i}{1-\alpha_j\lambda_1} \leqslant 1-\rho, \tag{4.9}$$

with

$$\rho = \frac{2\theta}{2\lambda_1 - \lambda_n}. \tag{4.10}$$

Since

$$\frac{1-\alpha_j\lambda_i}{1-\alpha_j\lambda_1} = -1 + \frac{2-\alpha_j(\lambda_1+\lambda_i)}{1-\alpha_j\lambda_1} \tag{4.11}$$

and, by (4.6),

$$\begin{aligned}
\frac{2-\alpha_j(\lambda_1+\lambda_i)}{1-\alpha_j\lambda_1} &\geqslant \frac{2-\frac{2}{\lambda_n}(\lambda_1+\lambda_i)}{1-\alpha_j\lambda_1} \\
&= \frac{2\lambda_n - 2\lambda_1 - 2\lambda_i}{\lambda_n(1-\alpha_j\lambda_1)} \\
&= \frac{2\lambda_1 - 2\lambda_n + 2\lambda_i}{\alpha_j\lambda_1\lambda_n - \lambda_n} \\
&\geqslant \frac{2\theta + 2\lambda_i}{2\lambda_1 - \lambda_n} \\
&\geqslant \rho,
\end{aligned}$$

we get

$$-1+\rho \leqslant \frac{1-\alpha_j\lambda_i}{1-\alpha_j\lambda_1} \leqslant 1-\rho.$$

Therefore, by (4.5), we have (4.2).

Because of (4.2)

$$\lim_k \alpha_k = 2\frac{\mu_1^2 d_1^{\mathrm{T}} d_1}{\mu_1^2 d_1^{\mathrm{T}} A d_1}$$

and, since $Ad_1 = \lambda_1 d_1$, we have

$$\lim_k \alpha_k = 2\frac{\mu_1^2 d_1^{\mathrm{T}} d_1}{\mu_1^2 \lambda_1 d_1^{\mathrm{T}} d_1} = \frac{2}{\lambda_1}.$$

Thus (4.3) holds.

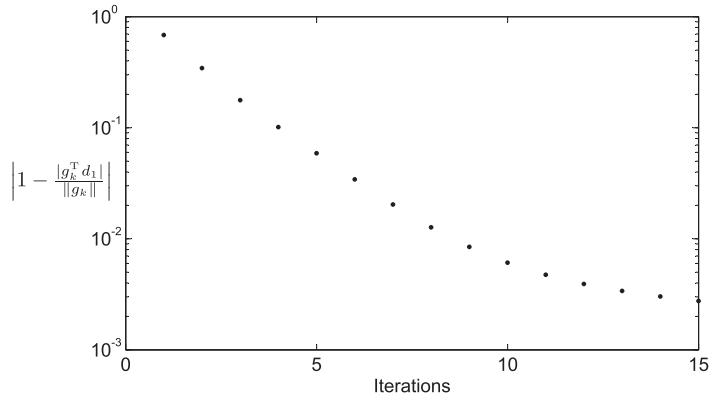


FIG. 4. Behaviour of the (normalized) component of the gradient along the eigendirection d_1 in 15 consecutive double Cauchy steps.

Finally, in order to prove (4.4), we first note that the sequence $\{\|g_k\|\}$ is bounded above and so is $\{\prod_{j=0}^{k}(1-\alpha_j\lambda_1)\}$ because of (4.2). Then

$$\begin{aligned}\lim_k \nabla f\left(x_k-\frac{\alpha_k}{2}g_k\right) &= \lim_k \left(g_k-\frac{\alpha_k}{2}Ag_k\right)\\ &= \lim_k \prod_{j=0}^{k-1}(1-\alpha_j\lambda_1)\left(\frac{g_k}{\prod_{j=0}^{k-1}(1-\alpha_j\lambda_1)}-\frac{\alpha_k}{2}A\frac{g_k}{\prod_{j=0}^{k-1}(1-\alpha_j\lambda_1)}\right)\\ &= \lim_k \prod_{j=0}^{k-1}(1-\alpha_j\lambda_1)\left(\mu_1 d_1-\frac{1}{\lambda_1}\mu_1 A d_1\right) = \lim_k \prod_{j=0}^{k-1}(1-\alpha_j\lambda_1)(\mu_1 d_1-\mu_1 d_1)=0;\end{aligned}$$

hence (4.4) holds and the proof is complete. □

Proposition 4.1 suggests that the double Cauchy step, although meaningless in terms of function reduction, might have a significant impact in terms of alignment of the gradient with the eigenvector d_1 and this might be of some support in a general gradient framework. To verify such alignment, we applied 15 consecutive double Cauchy steps to the problem described in Section 3. As predicted by Proposition 4.1, the component of the gradient along the eigendirection corresponding to the maximum eigenvalue of A soon becomes dominant, as shown in Fig. 4.

For this problem, we also considered a modified version of the SD method (SDM), in which 5 consecutive double Cauchy steps are performed every 10 Cauchy steps; the results in Fig. 5 show that this simple modification of the SD method produces a rather meaningful speed-up of the convergence.

Concerning the RSD method, Proposition 4.1 seems to suggest an over-relaxation rather than an under-relaxation of the Cauchy step and therefore we consider a modified version of the RSD method, called relaxed steepest descent with alignment (RSDA), where

$$\alpha_k \in [0.8\alpha_k^{\mathrm{SD}}, 2\alpha_k^{\mathrm{SD}}]. \tag{4.12}$$

Figure 6 shows the convergence of the RSD and the RSDA methods applied to the same problem considered above. Of course, because of the randomness in (4.12), a careful and deeper analysis is

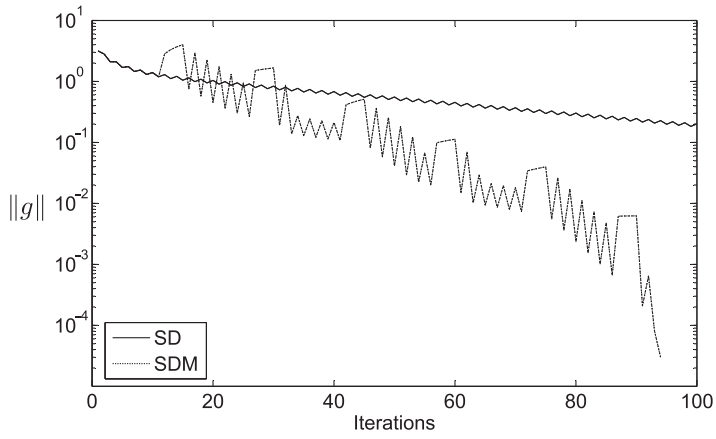


FIG. 5. Convergence history of the SD and SDM methods.

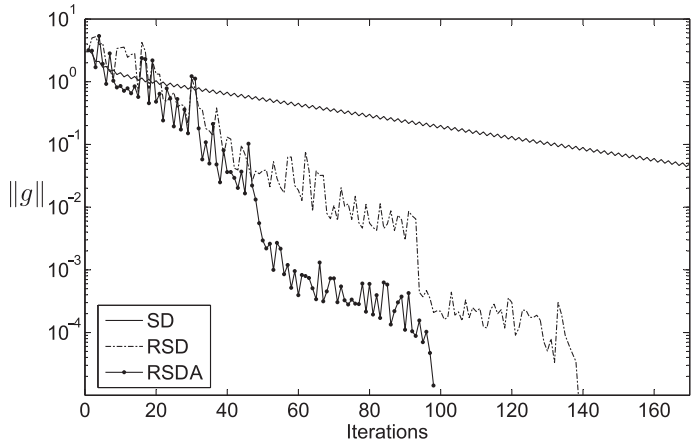


FIG. 6. Convergence history of the SD, RSD, and RSDA methods.

needed in order to evaluate the effectiveness of the method, especially to check the validity of our claim about the advantage in using the RSDA rather than the RSD method. Extensive numerical tests will be considered in Section 5 to get a clear picture of the numerical behaviour of the algorithmic approaches we have proposed in the last two sections.

5. Numerical experiments

In this section, we report some numerical results that compare the SDA and RSDA methods with the BB algorithm using the step length (1.8) and the Dai–Yuan (DY) algorithm using (3.10). A more extensive comparison with other gradient methods would be interesting, but outside of the scope of this paper, which is mainly to show how powerful and probably underestimated, although well known, is the SD method in revealing the spectral properties of problem (1.4). These properties can be easily plugged

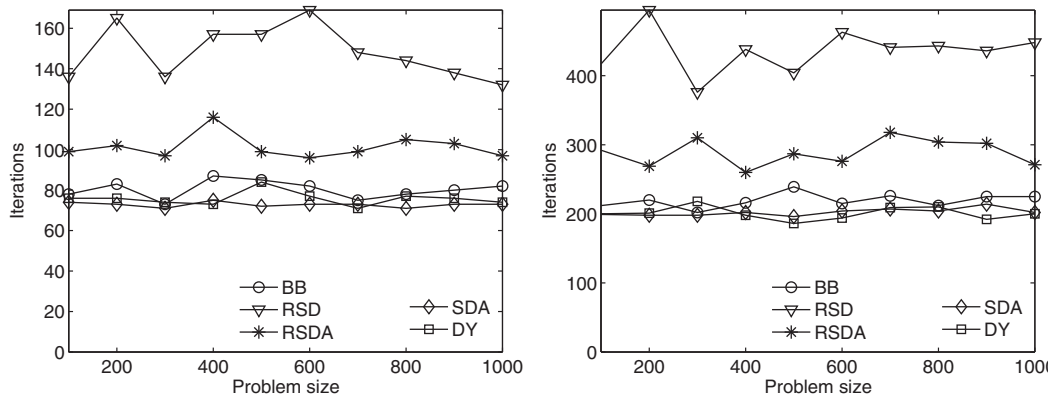


FIG. 7. Iterations for the randomly generated test problems with $\kappa(A) = 10^2$ (left) and $\kappa(A) = 10^3$ (right).

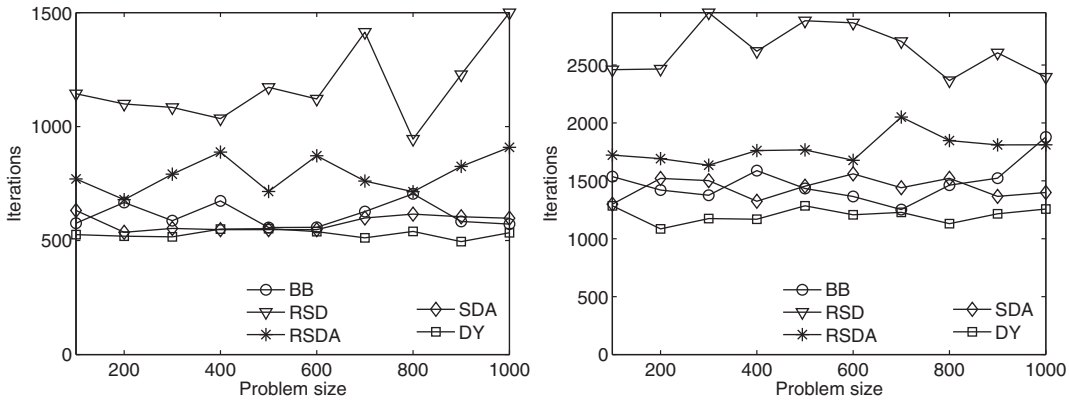


FIG. 8. Iterations for the randomly generated test problems with $\kappa(A) = 10^4$ (left) and $\kappa(A) = 10^5$ (right).

into this method with rather surprising results. On the other hand, the BB method is considered a quite efficient nonmonotone strategy, well representative of the so-called gradient methods with retard, even competitive with CG methods when low accuracy is required (Friedlander *et al.*, 1999; Fletcher, 2005). The choice of the DY approach is motivated by the analysis of monotone gradient methods in Dai & Yuan (2005), which suggests the superiority of the step length (3.10). Therefore, the BB and DY methods are valid benchmarks for testing the effectiveness of the SDA and RSDA methods. We also show the results obtained with the RSD method, to verify the conjecture in Section 4 about the advisability of using (4.12), as suggested by Proposition 4.1.

We considered two sets of test problems of type (1.4). The problems of the first set were randomly generated, by using MATLAB functions, with dimensions $100, 200, \ldots, 1000$. The Hessian matrices A were obtained by running `sprandsym` with `density` $= 0.8$, `kind` $= 1$ and condition number $\kappa(A) = 10^2, 10^3, 10^4, 10^5$. For each instance of A, x^* was generated by `rand` with entries in $[-10, 10]$ and $b = Ax^*$ was used in the linear term. Furthermore, for each problem, five starting points were generated

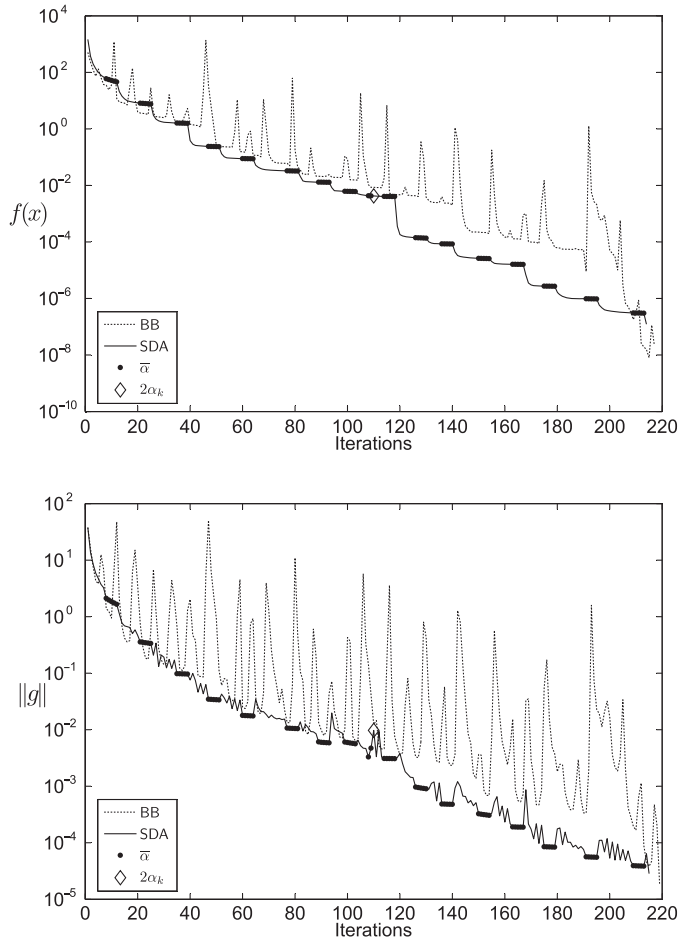


FIG. 9. Convergence history for the BB and SDA algorithms.

by `rand` with entries in $[-10, 10]$. As the stopping criterion, we used

$$\|g_k\| \leqslant 10^{-6}\|g_0\|.$$

All algorithms were implemented in MATLAB. In the SDA method, h and ε were set to 5 and 10^{-2}, respectively. We note that, because of their randomness, the RSD and RSDA algorithms were run 10 times on each problem with each starting point, varying the seed in the `rand` function used in the choice of the step length.

In Figs 7–8, we report the number of iterations of the five algorithms, fixing the condition number and varying the matrix dimension. For the SDA, BB and DY methods, the number of iterations for each problem is the mean of the results obtained with the 5 different starting points. For the RSD and RSDA methods, the number of iterations is averaged over the 50 runs associated with each problem.

TABLE 1 *Iterations for the Laplace problems, with stop condition* $\|g_k\| < 10^{-2}\|g_0\|$

Problem	CG	BB	DY	SDA	RSDA	RSD
Laplace1(a)	16	14	12	17	14	18
Laplace1(b)	16	14	12	17	14	18

TABLE 2 *Iterations for the Laplace problems, with stop condition* $\|g_k\| < 10^{-4}\|g_0\|$

Problem	CG	BB	DY	SDA	RSDA	RSD
Laplace1(a)	135	225	185	186	269	406
Laplace1(b)	135	205	196	184	282	397

TABLE 3 *Iterations for the Laplace problems, with stop condition* $\|g_k\| < 10^{-6}\|g_0\|$

Problem	CG	BB	DY	SDA	RSDA	RSD
Laplace1(a)	181	484	389	392	596	900
Laplace1(b)	181	495	397	416	593	913

We first note that the poorest results were obtained by the two random Cauchy algorithms RSD and RSDA. This is not surprising at all (see Friedlander *et al.*, 1999; Raydan & Svaiter, 2002); however, it is worth noting the clear superiority of the RSDA over the RSD method. The SDA, BB and DY methods give the overall best results, with the DY method performing slightly better for larger values of $\kappa(A)$. As expected, the performance of all the algorithms deteriorates as the ill conditioning increases, while the problem size appears to be a much less critical issue.

In Fig. 9, we compare the complete convergence history (gradient norm and function value) of the SDA and BB methods for a specific instance of the test problems ($n = 300$, $\kappa(A) = 10^3$). The difference in the behaviour of the two algorithms clearly emerges. The SDA iterates with step length $\overline{\alpha}$ are highlighted in the picture, making clear their role in accelerating the decrease of the objective function. A notable feature of the SDA method is that it adopted the double Cauchy step only once in order to preserve the algorithm monotonicity, and actually, in the overall set of 400 random test problems, it took this step only 20 times.

Similar results were obtained with the second set of test problems, consisting of the Laplace1(a) and Laplace1(b) problems described in Fletcher (2005), which arise from a uniform seven-point finite difference discretization of the three-dimensional Poisson equation on a box, with homogeneous Dirichlet boundary conditions. These problems have 10^6 variables and a highly sparse Hessian matrix with condition number $10^{3.61}$. For each problem, five starting points were generated by `rand` with entries in $[0, 1]$; the iteration was terminated when $\|g_k\| < \eta\|g_0\|$, with $\eta = 10^{-2}, 10^{-4}, 10^{-6}$, to check the effects of different accuracy requirements. The algorithms were also compared with the CG method implemented in the MATLAB `pcg` function. In Tables 1–3, for each problem, we report the average number of iterations for the six algorithms (as in the random test problems, for each starting point, the RSD and RSDA methods were run 10 times varying the seed in the `rand` function used in the choice of the step length).

The results in Table 3 show that the CG method outperforms the other methods when high accuracy is required. In this case, the RSDA and RSD methods achieve the poorest results, with the RSDA

method showing a significant improvement over the RSD one; the BB, DY and SDA methods take a smaller number of iterations than the previous methods, but are still much slower than the CG. Very interesting are the results in Tables 1 and 2, which suggest that, for low accuracy requirements, gradient methods, especially the DY and SDA methods, provide reasonable alternatives to the CG algorithm, for instance, in the computational contexts outlined in Fletcher (2005) and Huang (2008). The results in Table 3 show that the performance of the gradient algorithms with respect to the CG algorithm seriously deteriorates as the stopping condition becomes stronger. Concerning the SDA method, we note that its behaviour depends on the SD ability to force the search into a two-dimensional space (see (3.7)) and on the approximation of $\hat{\alpha}$ through $\tilde{\alpha}_k$ (Proposition 3.1) which fosters the alignment of the gradient with d_n. A rather inaccurate alignment (which, in our experience, is usually achieved very soon by the SDA method) can be sufficient to get a low-accuracy solution in few iterations. Conversely, getting high accuracy in the solution requires a strong alignment of the gradient with d_n, and therefore many SD iterates, both to get a very small value of ζ_k in (3.7) and to compute a reliable approximation of $\hat{\alpha}$.

In conclusion, for the SDA, BB and DY methods, we do not feel it is fair to state the clear superiority of one method over the others, although the DY method appears to be more efficient on the most ill-conditioned random problems. We just believe that our numerical experiences support the alignment-based approaches motivated by the theoretical results in Sections 3 and 4, which highlight some potentialities of the SD algorithm, related to the spectral properties of A revealed by the method. We also note that numerical experiments showed that the SDA algorithm can be made more efficient on the problems with the largest ill conditioning by reducing the value of ε. Conversely, numerical tests with different values of h showed that the performance of the SDA method depends very little on h, unless very small values of it, say 1 or 2, are taken (varying h between 3 and 10 was almost uninfluential on the performance of the algorithm).

Motivated by the encouraging numerical results, we hope the analysis in this paper can be further refined in order to design effective gradient methods for nonquadratic functions, for which the monotonicity property of the SDA method might represent a remarkable advantage over BB-like algorithms. Finally, we believe that using step lengths able to force the algorithm search into low-dimensional subspaces should retain its benefits also in the more general framework of constrained optimization; therefore, a possible further development of this research might be to incorporate the ideas outlined here in a projected gradient framework (De Angelis & Toraldo, 1993), to deal with bound-constrained problems.

Acknowledgements

The authors would like to thank the anonymous referees for their careful reading and their constructive and valuable comments.

Funding

This research was partially supported by the Italian Ministry of University and Research under the PRIN 2008 Project *Optimization methods and software for inverse problems* (grant no. 2008T5KA4L).

References

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

Andretta, M., Birgin, E. G. & Martínez, J. M. (2010) Partial spectral projected gradient method with active-set strategy for linearly constrained optimization. *Numer. Algorithms*, **53**, 23–52.

Barzilai, J. & Borwein, J. M. (1988) Two-point step size gradient methods. *IMA J. Numer. Anal.*, **8**, 141–148.
Bertero, M., Lanteri, H. & Zanni, L. (2008) Iterative image reconstruction: a point of view. *Mathematical Methods in Biomedical Imaging and Intensity-Modulated Radiation Therapy (IMRT)*. CRM, vol. 7. Edizioni della Normale, Pisa, 37–63.
Birgin, E. G., Martínez, J. M. & Raydan, M. (2000) Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optimiz.*, **10**, 1196–1211.
Bonettini, S., Zanella, R. & Zanni, L. (2008) A scaled gradient projection method for constrained image deblurring. *Inverse Probl.*, **25**, 015002.
Cauchy, A. (1847) Méthodes générales pour la résolution des systèmes d'équations simultanées. *CR. Acad. Sci. Par.*, **25**, 536–538.
Dai, Y.-H. & Fletcher, R. (2005) On the asymptotic behaviour of some new gradient methods. *Math. Program. (Series A)*, **13**, 541–559.
Dai, Y.-H. & Fletcher, R. (2006) New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds. *Math. Program. (Series A)*, **106**, 403–421.
Dai, Y.-H. & Liao, L.-Z. (2002) R-linear convergence of the Barzilai and Borwein gradient method. *IMA J Numer. Anal.*, **22**, 1–10.
Dai, Y. H. & Yang, X. Q. (2006) A new gradient method with an optimal stepsize property. *Comput. Optim. Appl.*, **33**, 73–88.
Dai, Y. H. & Yuan, Y. (2005) Analyses of monotone gradient methods. *J. Ind. Manag. Optim.*, **1**, 181–192.
De Angelis, P. L. & Toraldo, G. (1993) On the identification property of a projected gradient method. *SIAM J. Numer. Anal.*, **30**, 1483–1497.
Elman, H. C. & Golub, H. G. (1994) Inexact and preconditioned Uzawa algorithms for saddle point problems. *SIAM J. Numer. Anal.*, **31**, 1645–1661.
Fletcher, R. (2005) On the Barzilai–Borwein method. *Optimization and Control with Applications* (L. Qi, K. Teo, X. Yang, P. M. Pardalos & D. Hearn eds), Applied Optimization, vol. 96. USA: Springer, pp. 235–256.
Forsythe, G. E. (1968) On the asymptotic directions of the s-dimensional optimum gradient method. *Numer. Math.*, **11**, 57–76.
Frassoldati, G., Zanni, L. & Zanghirati, G. (2008) New adaptive stepsize selections in gradient methods. *J. Ind. Manag. Optim.*, **4**, 299–312.
Friedlander, A., Martínez, J. M., Molina, B. & Raydan, M. (1999) Gradient method with retards and generalizations. *SIAM J. Numer. Anal.*, **36**, 275–289.
Hager, W. W. & Zhang, H. (2006) A new active set algorithm for box constrained optimization. *SIAM J. Optim.*, **17**, 526–557.
Huang, H. (2008) *Efficient Reconstruction of 2D Images and 3D Surfaces*. Vancouver: University of BC.
Nocedal, J., Sartenaer, A. & Zhu, C. (2002) On the behavior of the gradient norm in the steepest descent method. *Comp. Optim. Appl.*, **22**, 5–35.
Raydan, M. (1993) On the Barzilai and Borwein choice of steplength for the gradient method. *IMA J. Numer. Anal.*, **13**, 321–326.
Raydan, M. (1997) The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM J. Optimiz.*, **7**, 26–33.
Raydan, M. & Svaiter, B. F. (2002) Relaxed steepest descent and Cauchy–Barzilai–Borwein method. *Comput. Optim. Appl.*, **21**, 155–167.
Yuan, Y. (2006) A new stepsize for the steepest descent method. *J. Comp. Math.*, **24**, 149–156.