On the steplength selection in gradient methods for unconstrained optimization

Daniela di Serafino[a,*], Valeria Ruggiero[b], Gerardo Toraldo[c], Luca Zanni[d]

[a] *Department of Mathematics and Physics, University of Campania Luigi Vanvitelli,*
viale A. Lincoln 5, I-81100 Caserta, Italy
[b] *Department of Mathematics and Computer Science, University of Ferrara,*
via Saragat 1, I-44122 Ferrara, Italy
[c] *Department of Mathematics and Applications, University of Naples Federico II,*
Via Cintia 21, I-80126, Naples, Italy
[d] *Department of Physics, Informatics and Mathematics,*
University of Modena and Reggio Emilia, via Campi 213/B, I-41125, Modena, Italy

January 25, 2017

Abstract

The seminal paper by Barzilai and Borwein [IMA J. Numer. Anal. 8 (1988)] has given rise to an extensive investigation aimed at developing effective gradient methods, able to deal with large-scale optimization problems. Several steplength rules have been first designed for unconstrained quadratic problems and then extended to general nonlinear problems; these rules share the common idea of attempting to capture, in an inexpensive way, some second-order information. Our aim is to investigate the relationship between the steplengths of some gradient methods and the spectrum of the Hessian of the objective function, in order to provide insight into the computational effectiveness of these methods. We start the analysis in the framework of strongly convex quadratic problems, where the role of the eigenvalues of the Hessian matrix in the behaviour of gradient methods is better understood. Then we move to general unconstrained problems, focusing on natural extensions of some steplength rules analysed in the previous case. Our study suggests that, in the quadratic case, the methods that tend to use groups of small steplengths followed by some large steplengths, attempting to approximate the inverses of some eigenvalues of the Hessian matrix, exhibit better numerical behaviour. The methods considered in the general case seem to preserve the behaviour of their quadratic counterparts, in the sense that they appear to follow somehow the spectrum of the Hessian of the objective function during their progress toward a stationary point.

Keywords: gradient methods, steplength selection, Hessian spectral properties.
2010 MSC: 65K05, 90C20, 90C30

1. Introduction

Many real life applications lead to nonlinear optimization problems whose very large size makes first-order methods the most suitable choice. Among first-order approaches, gradient

[*] Corresponding author
Email addresses: daniela.diserafino@unina2.it (Daniela di Serafino), valeria.ruggiero@unife.it (Valeria Ruggiero), toraldo@unina.it (Gerardo Toraldo), luca.zanni@unimore.it (Luca Zanni)

methods have widely proved their effectiveness in solving challenging unconstrained and constrained problems arising in signal and image processing, compressive sensing, machine learning, optics, chemistry and other areas (see, e.g., [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] and the references therein).

These methods underwent a renaissance since the work by Barzilai and Borwein [12], which showed how a suitable choice of the steplength can significantly accelerate the classical Steepest Descent method [13, 14]. Since then, several steplength rules have been designed in order to increase the efficiency of gradient methods, while preserving their simplicity and low memory requirement. Most of these rules have been first developed for the unconstrained convex quadratic problem [15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29], which is not only of practical importance in itself, but also provides a simple setting to design effective methods for more general problems. The extension of steplength selection strategies from convex quadratic to general nonlinear optimization has involved interesting theoretical issues, leading to the exploitation of line search strategies in order to guarantee convergence to stationary points [30, 31, 32, 17, 18, 33, 34, 35, 24, 36].

The theoretical convergence results of gradient methods based on the previous steplength rules do not explain their effectiveness, and a full understanding of their practical behaviour is still missing. A feature shared by most of these methods consists in exploiting spectral properties of the Hessian of the objective function through (usually implicit) low cost approximations of expensive second-order information. This appears to be the main reason for their good behaviour (see, e.g., [33, 21, 24, 26, 37]); however, a deeper and more systematic analysis is needed.

The aim of this work is to investigate the relationship between the steplengths exploited by some well known gradient methods and the spectrum of the Hessian of the objective function, for convex quadratic and general problems of the form

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. In this case, the gradient method iteration reads

$$x_{k+1} = x_k - \alpha_k g_k, \tag{2}$$

where $g_k = \nabla f(x_k)$ and $\alpha_k > 0$ denotes the steplength. Our analysis highlights basic principles for "capturing" spectral properties of the Hessian matrix and how these principles have been implemented in recent gradient methods in order to achieve computational effectiveness.

We first consider the convex quadratic problem

$$\min_{x\in\mathbb{R}^n} f(x) := \frac{1}{2}x^T Ax - b^T x, \tag{3}$$

where $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite and $b \in \mathbb{R}^n$. It provides a simple framework for investigating the role of the eigenvalues of the Hessian matrix in the behaviour of gradient methods; furthermore, convergence results involving the spectrum of the Hessian are available in this case, which provide a sound basis for our analysis. We deal with a selection of approaches, representative of a wide class of gradient methods, as explained later in this paper. We consider the following methods: Barzilai-Borwein and Adaptive Barzilai-Borwein variants [20, 22], Limited Memory Steepest Descent [24], Steepest Descent with Alignment and Steepest Descent with Constant (Yuan) steps [26, 27]; we also consider methods such that the inverses of their steplengths follow predefined distributions obtained exploiting the Golden Arcsin rule [38] or the Chebyshev nodes [29]. In the second part of the paper, we deal with the general unconstrained problem, focusing on gradient methods whose steplengths are natural extensions of the rules developed for the convex quadratic case, combined with line search strategies forcing convergence. In particular, we investigate methods based on the Barzilai-Borwein, the $\mathrm{ABB}_{\min}$ Adaptive Barzilai-Borwein [22] and the Limited Memory Steepest Descent rules. Our analysis is supported by numerical experiments on a variety of test problems.

The paper is organized as follows. In Section 2, after some preliminary results on gradient methods applied to strongly convex quadratic problems, we discuss the relationship between the steplengths and the spectrum of the Hessian in the quadratic case, showing the results of a set of numerical experiments. This analysis is extended to the non-quadratic case in Section 3. Some conclusions are provided in Section 4.

2. Convex quadratic problems

We first consider the strongly convex quadratic problem (3), in order to highlight the strict relationship between the behaviour of gradient methods and the eigenvalues of the Hessian of the objective function. In particular, we show how some choices of the steplength exploit spectral properties of the Hessian matrix in order to achieve efficiency in the corresponding methods. We start by giving some preliminary results, which will be useful in our analysis.

2.1. Notation and preliminaries

Let $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ be the eigenvalues of the matrix A in (3), and $\{d_1, d_2, \dots, d_n\}$ a set of associated orthonormal eigenvectors. The gradient g_k can be expressed as

$$g_k = \sum_{i=1}^{n} \mu_i^k d_i, \qquad \mu_i^k \in \mathbb{R}, \tag{4}$$

where μ_i^k is called the i-th eigencomponent of g_k. The following assumptions are not restrictive (see, e.g., [27, Section 1] and [24, Section 2]) and will be considered in the sequel.

Assumption 1. *The eigenvalues $\lambda_1, \dots, \lambda_n$ are such that*

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

Assumption 2. *At the starting point x_0,*

$$\mu_1^0 \neq 0 \ \text{ and } \ \mu_n^0 \neq 0.$$

For any gradient method applied to problem (3),

$$g_k = g_{k-1} - \alpha_{k-1} A g_{k-1} = \prod_{j=0}^{k-1} (I - \alpha_j A) g_0, \tag{5}$$

and then the eigencomponents of g_k satisfy the recurrence formula

$$\mu_i^k = \mu_i^0 \prod_{j=0}^{k-1}(1-\alpha_j\lambda_i) = \mu_i^{k-1}(1-\alpha_{k-1}\lambda_i). \tag{6}$$

The previous recurrence allows to analyse the behaviour of gradient methods in terms of the spectrum of the Hessian matrix A. In particular, the following properties are a straightforward consequence of (6):

1. if $\mu_i^k = 0$ for some i, then $\mu_i^h = 0$ for $h \geq k$;
2. if $\alpha_k = 1/\lambda_i$, then $\mu_i^{k+1} = 0$;
3. $\left|\mu_i^{k+1}\right| < \left|\mu_i^k\right|$ if and only if $\alpha_k < 2/\lambda_i$;
4. if α_k is sufficiently close to $1/\lambda_j$, then $\left|\mu_i^{k+1}\right| > \left|\mu_i^k\right|$ for $i < j$ and $\lambda_i > 2\lambda_j$.

Thus, small steplengths (say close to $1/\lambda_1$) tend to decrease a large number of eigencomponents, with negligible reduction of those corresponding to small eigenvalues. The latter can be significantly reduced by using large steplengths, but this increases the eigencomponents corresponding to large eigenvalues, fostering also a nonmonotone behaviour of the sequence $\{f(x_k)\}$.

The classical Steepest Descent (SD) method for problem (3) uses the Cauchy steplength

$$\alpha_k^{\mathrm{SD}} = \operatorname*{argmin}_{\alpha>0} f(x_k - \alpha g_k) = \frac{g_k^T g_k}{g_k^T A g_k}, \tag{7}$$

which guarantees monotonicity of $\{f(x_k)\}$. It is well known that the SD method has Q-linear convergence rate which depends on $\rho = (\lambda_1 - \lambda_n)/(\lambda_1 + \lambda_n)$ [14]. Furthermore, equality (5) implies that SD has finite termination if and only if at some iteration the gradient is an eigenvector of A.

The convergence behaviour of the SD method has been deeply investigated (see, e.g., [14, 39, 26, 28]). Some key theoretical results are summarized next.

Theorem 1. *Let Assumptions 1 and 2 hold and let $\{x_k\}$ be a sequence generated by the SD method applied to problem* (3)*. Then*

$$\lim_{k\to\infty} \frac{(\mu_1^k)^2}{\sum_{j=1}^n (\mu_j^k)^2} = \begin{cases} \dfrac{1}{1+c^2} & \textit{if } k \textit{ odd} \\[2ex] \dfrac{c^2}{1+c^2} & \textit{if } k \textit{ even} \end{cases}$$

$$\lim_{k\to\infty} \frac{(\mu_n^k)^2}{\sum_{j=1}^n (\mu_j^k)^2} = \begin{cases} \dfrac{c^2}{1+c^2} & \textit{if } k \textit{ odd} \\[2ex] \dfrac{1}{1+c^2} & \textit{if } k \textit{ even} \end{cases}$$

$$\lim_{k\to\infty} \frac{(\mu_i^k)^2}{\sum_{j=1}^n (\mu_j^k)^2} = 0, \quad \textit{for } i = 2, \ldots, n-1,$$

where

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

Furthermore,

$$\lim_{k\to\infty} \frac{g_{2k}}{\|g_{2k}\|} = p, \quad \lim_{k\to\infty} \frac{g_{2k+1}}{\|g_{2k+1}\|} = p',$$

where $p, p' \in span\{d_1, d_n\}$ *and* $\|\cdot\|$ *is the Euclidean norm.*

Theorem 1 shows that the SD method tends to reduce the gradient eigencomponents corresponding to the largest and smallest eigenvalues more slowly than the other components. It eventually performs its search in the space spanned by the eigenvectors corresponding to the largest and smallest eigenvalues of A, with normalized gradient approaching the vectors p and p' in a cyclic way. This explains the well-known SD zigzagging behaviour, which generally yields slow convergence.

A possibility for avoiding the zigzagging pattern of the gradient is to foster the sequence $\{1/\alpha_k\}$ to sweep all the spectrum of the Hessian matrix. Furthermore, a suitable alternation of small and large steplengths appears to be a key issue to reduce the gradient eigencomponents in a more balanced way. In the last three decades, several more efficient gradient methods have been designed whose behaviour can be explained in light of the previous considerations, as discussed in the next section.

2.2. Steplengths and Hessian eigenvalues

Starting from the seminal work by Barzilai and Borwein [12], there has been a renewed interest for gradient methods, and many strategies for computing steplengths have been devised with the objective of overcoming the inherent drawbacks of the SD method. In our opinion, three main concepts can be identified which underlie most of these strategies:

1. injecting some second-order information into the steplengths;
2. breaking the cycling behaviour of the SD gradients by using special steplengths at selected iterations;
3. using steplengths following some predefined distribution over $[1/\lambda_1, 1/\lambda_n]$.

These ideas are not mutually exclusive and they often give the possibility of interpreting gradient methods from different points of view. Their application in different methods is discussed next for the quadratic problem (3). We focus on a selection of gradient methods, whose behaviours can be considered representative of many gradient methods; therefore, our discussion is not meant to be exhaustive.

The idea of using steplengths attempting to capture some second-order information clearly underlies the Barzilai-Borwein (BB) methods, which paved the way for the reinassance of gradient methods. In this case the steplength is defined by a secant condition, imposing either

$$\alpha_k = \operatorname*{argmin}_{\alpha} \left\|\alpha^{-1} s_{k-1} - y_{k-1}\right\| \tag{8}$$

or

$$\alpha_k = \operatorname*{argmin}_{\alpha} \|s_{k-1} - \alpha y_{k-1}\|, \tag{9}$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$. Hence, the BB methods can be regarded as quasi-Newton methods where the Hessian is approximated by $(1/\alpha_k)I$. The following steplengths are obtained from (8) and (9), respectively:

$$\alpha_k^{BB1} = \frac{\|s_{k-1}\|^2}{s_{k-1}^T y_{k-1}} = \frac{{g_{k-1}}^T g_{k-1}}{{g_{k-1}}^T A g_{k-1}}, \tag{10}$$

$$\alpha_k^{BB2} = \frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2} = \frac{{g_{k-1}}^T A g_{k-1}}{{g_{k-1}}^T A^2 g_{k-1}},$$

which satisfy

$$\frac{1}{\lambda_1} \leq \alpha_k^{\mathrm{BB2}} \leq \alpha_k^{\mathrm{BB1}} \leq \frac{1}{\lambda_n}.$$

Note that α_k^{BB1} is equal to the Cauchy steplength at iteration $k-1$, i.e., α_{k-1}^{SD}, while α_k^{BB2} is equal to the steplength of the Minimal Gradient method at iteration $k-1$, i.e.,

$$\alpha_{k-1}^{MG} = \operatorname*{argmin}_{\alpha>0} \|\nabla f(x_{k-1} - \alpha g_{k-1})\|.$$

In other words, both BB steplengths can be regarded as the result of exact line searches applied to sequences with delay 1.

The BB methods applied to strongly convex quadratic problems have R-linear convergence [40], which does not explain why they are in practice much faster than the SD method. However, it has been experimentally observed in [33] that these methods are able to generate sequences $\{1/\alpha_k\}$ sweeping the spectrum of A, thus preventing the gradient from asymptotically cycling between two fixed directions. It is worth noting that this behaviour of the BB steplengths may produce significant nonmonotonicity in the sequence $\{f(x_k)\}$.

Several gradient methods have been proposed which generalise the BB methods. They are based either on the alternation of Cauchy and BB steplengths or on their cyclic use (see, e.g., [16, 17, 34]); some of them fit into the framework of Gradient Methods with Retards [15], which, following the BB methods, use delayed Cauchy steplengths. The convergence rate of these BB-related methods is generally R-linear, but their practical convergence behaviour is superior than the SD one, like the original BB methods.

Among these methods we focus on the Adaptive Barzilai-Borwein (ABB) one as originally formulated in [20], and on its modification ABB_{min} [22], whose steplenghts are defined by the following rules:

$$\alpha_k^{\mathrm{ABB}} = \begin{cases} \alpha_k^{\mathrm{BB2}} & \text{if } \dfrac{\alpha_k^{\mathrm{BB2}}}{\alpha_k^{\mathrm{BB1}}} < \tau \\ \alpha_k^{\mathrm{BB1}} & \text{otherwise} \end{cases}$$

and

$$\alpha_k^{\mathrm{ABB_{min}}} = \begin{cases} \min\left\{\alpha_j^{\mathrm{BB2}} \,:\, j = \max\{1, k-m_a\}, \ldots, k\right\}, & \text{if } \dfrac{\alpha_k^{\mathrm{BB2}}}{\alpha_k^{\mathrm{BB1}}} < \tau \\ \alpha_k^{\mathrm{BB1}}, & \text{otherwise} \end{cases} \tag{11}$$

where m_a is a nonnegative integer and $\tau \in (0,1)$. Both methods tend to compute BB2 steplengths, which are likely to be small, spaced out with some BB1 steplengths, which are inclined to be large. Adaptive criteria are used to switch between the two steplengths, based on the value

$$\frac{\alpha_k^{\mathrm{BB2}}}{\alpha_k^{\mathrm{BB1}}} = \cos^2 \psi_{k-1},$$

where ψ_{k-1} is the angle between g_{k-1} and Ag_{k-1}. The rationale behind these criteria is to select α_k^{BB1} when g_{k-1} is a sufficiently good approximation of an eigenvector of A. In other words, the methods tend to generate a sequence of (small) BB2 steplengths in order to foster the BB1 steplength to become a suitable approximation of the inverse of some small eigenvalue. We note that $\mathrm{ABB_{min}}$ tends to adopt smaller steplengths than ABB. In conclusion, the steplength rules used by the two methods aim to follow the BB behaviour in sweeping the spectrum of A, but try to mitigate the nonmonotone behaviour of the objective function through a "wise" alternation of short and long steps.

A different approach aimed at using second-order information by capturing the spectrum of the Hessian is exploited by the Limited Memory Steepest Descent (LMSD) method proposed in [24]. The basic idea is to divide the sequence of LMSD iterations into groups of m_s iterations referred to as sweeps, where m_s is a small positive integer, and to compute the steplengths for each sweep as the inverses of some Ritz values of the Hessian matrix [41], obtained by exploiting the gradients of the previous sweep. In order to briefly describe the LMSD method, we consider an iteration $k \geq m_s$ and define the matrices $G \in \mathbb{R}^{n \times m_s}$ and $J \in \mathbb{R}^{(m_s+1) \times m_s}$ as follows:

$$G = [g_{k-m_s},\ g_{k-m_s+1}, \dots,\ g_{k-1}],$$

and

$$J = \begin{pmatrix} \frac{1}{\alpha_{k-m_s}^{\text{LMSD}}} & & \\ -\frac{1}{\alpha_{k-m_s}^{\text{LMSD}}} & \ddots & \\ & \ddots & \frac{1}{\alpha_{k-1}^{\text{LMSD}}} \\ & & -\frac{1}{\alpha_{k-1}^{\text{LMSD}}} \end{pmatrix},$$

where α_i^{LMSD} is the steplength associated with the gradient g_i. Then, the first equality in (5) can be written in matrix form as

$$AG = [G,\ g_k]J.$$

This can be used to compute the tridiagonal matrix T resulting from the application of m_s iterations of the Lanczos process to the matrix A, with starting vector $q_1 = g_{k-m_s}/\|g_{k-m_s}\|$. This process generates a matrix $Q = [q_1,\ q_2,\ \dots,\ q_{m_s}]$ whose columns are an orthonormal basis for the Krylov space

$$\operatorname{span}\left\{g_{k-m_s},\ Ag_{k-m_s},\ A^2 g_{k-m_s},\ \dots,\ A^{m_s-1} g_{k-m_s}\right\},$$

such that

$$T = Q^T A Q.$$

Since the columns of G can be obtained as suitable combinations of the columns of Q, we can write $G = QR$, where R is upper triangular and nonsingular if G is full rank, and hence

$$T = Q^T A\, G\, R^{-1} = [R,\ Q^T g_k] J R^{-1}, \tag{12}$$

(for now we assume that G is full rank; the case G rank-deficient is addressed later). The steplengths for the next m_s gradient iterations are defined as the inverses of the eigenvalues θ_i of T:

$$\alpha_{k-1+i}^{\text{LMSD}} = \frac{1}{\theta_i}, \quad i = 1, \dots, m_s. \tag{13}$$

The quantities θ_i are the so-called Ritz values, which belong to the spectrum of A and provide m_s approximations of the eigenvalues of A [41]. Note that for $m_s = 1$ we obtain the BB method with steplength α_k^{BB1}. As in the BB-like approaches, the sequence $\{f(x_k)\}$ is nonmonotone.

So far we have assumed that a group of m_s iterations have been performed before starting a new sweep; nevertheless, the LMSD method can be initialized with a single steplength α_0, as done in other gradient methods. In this case, two initial sweeps of length $m_s = 1$ are performed, followed by a sweep in which two back gradients can be exploited to define the steplengths allowing two new iterations; in the next sweep, up to four back gradients can be exploited, and so on.

Furthermore, equation (12) shows that T can be obtained without involving the matrix A explicitly; this is useful for generalizing the method to the non-quadratic case (see Section 3). The use of the matrix Q can be avoided too, by observing that $G^T G = R^T R$ and then

$$T = Q^T AQ = R^{-T} G^T AGR^{-1} = R^{-T} G^T [G,\ g_k] JR^{-1} = [R,\ r] JR^{-1}, \tag{14}$$

where the vector r is the solution of the linear system $R^T r = G^T g_k$. In this way, R can be obtained from the Cholesky factorization of $G^T G$ and the computation of Q is not required.

In our implementation (see Section 2.3) we compute T as in (14). If $G^T G$ is (numerically) indefinite, we eliminate the oldest gradient from G and repeat the Cholesky factorization; in this case, fewer than m_s steplengths are provided for the new sweep and fewer than m_s new gradients are computed. Thus, back gradients from the previous sweep are kept for defining the m_s columns of the next matrix G.
As the BB methods and their aforementioned extensions, the LMSD method has R-linear convergence [42]. However, an improvement over BB is reported in [24] for $m_s > 1$.

A different philosophy to define the steplengths is behind the SDA and SDC gradient methods, proposed in [26, 27]. They alternate a number of SD steplengths with a number of constant steplenghts, computed by using rules that exploit previous SD steplenghts, with the aim of escaping from the two-dimensional space where the SD method asymptotically reduces its search. Given two integers $h \geq 2$ and $m_c \geq 1$, the SDA and SDC steplength are computed as

$$\alpha_k = \begin{cases} \alpha_k^{\mathrm{SD}} & \text{if } \bmod (k, h + m_c) < h, \\ \hat{\alpha}_s & \text{otherwise, with } s = \max\{i \leq k \,:\, \bmod (i, h + m_c) = h\}, \end{cases} \tag{15}$$

where $\hat{\alpha}_s$ is a "special" steplength built at a certain iteration s by using $\alpha_{s-1}^{\mathrm{SD}}$ and α_s^{SD}. 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_c consecutive iterations. In the SDA method $\hat{\alpha}_s = \alpha_s^{\mathrm{A}}$, where

$$\alpha_s^{\mathrm{A}} = \left(\frac{1}{\alpha_{s-1}^{\mathrm{SD}}} + \frac{1}{\alpha_s^{\mathrm{SD}}} \right)^{-1},$$

while in the SDC method $\hat{\alpha}_s = \alpha_s^{\mathrm{Y}}$, where

$$\alpha_s^{\mathrm{Y}} = 2 \left(\sqrt{ \left(\frac{1}{\alpha_{s-1}^{\mathrm{SD}}} - \frac{1}{\alpha_s^{\mathrm{SD}}} \right)^2 + 4 \frac{\|g_s\|^2}{\left(\alpha_{s-1}^{\mathrm{SD}} \|g_{s-1}\| \right)^2} } + \frac{1}{\alpha_{s-1}^{\mathrm{SD}}} + \frac{1}{\alpha_s^{\mathrm{SD}}} \right)^{-1}. \tag{16}$$

Note that α_s^{Y} is the so-called Yuan steplengths [43], used in the Dai-Yuan method. The latter alternates some Cauchy steplengths with some Yuan steplengths in a way that resembles (15), but recomputes α_s^{Y} at each iteration instead of keeping it constant.

The choice of the steplengths in the SDA and SDC methods is motivated by some properties of α_k^{A} and α_k^{Y}. Specifically, in [26, 27] it is proved that

$$\lim_{k\to\infty} \alpha_k^{\mathrm{A}} = \frac{1}{\lambda_1 + \lambda_n}, \tag{17}$$

$$\lim_{k\to\infty} \alpha_k^{\mathrm{Y}} = \frac{1}{\lambda_1}. \tag{18}$$

where α_k^{A} and α_k^{Y} are computed by using the sequence $\{\alpha_k^{\mathrm{SD}}\}$ generated by applying the SD method to problem (3).

Equality (17) and the properties of the SD method suggest that the SDA method combines the tendency of SD to choose its search direction in $\mathrm{span}\{d_1, d_n\}$ with the tendency of the gradient method with constant steplength $1/(\lambda_1 + \lambda_n)$ to align the search direction with d_n. This significantly accelerates the convergence with respect to the SD method, as shown by the numerical results in [26]. Note that the name SDA stands for Steepest Descent with Alignment, i.e., it refers to the aforementioned alignment property.

In SDC the use of a finite sequence of Cauchy steps has a twofold goal: forcing the search in $\mathrm{span}\{d_1, d_n\}$ and computing a suitable approximation of $1/\lambda_1$ (see (18)), in order to drive toward zero the first eigencomponent of the gradient, μ_1^k. If this eigencomponent were completely removed, a sequence of Cauchy steplengths followed by constant Yuan steplengths would drive toward zero the second eigencomponent μ_2^k, and so on. Thus, the alternation of Cauchy and constant Yuan steplengths is considered as an attempt to eliminate the eigencomponents of the gradient according to the decreasing order of the eigenvalues of A. By the way, we note that the name SDC comes from Steepest Descent with Constant (Yuan) steps. We also point out that, if the Hessian matrix is ill conditioned, $1/(\lambda_1 + \lambda_n) \approx 1/\lambda_1$ and then SDA and SDC are expected to have very close behaviours.

Again, SDA and SDC have R-linear convergence, but in practice are competitive with the fastest gradient methods currently available [26]. Furthermore, the resulting sequences $\{f(x_k)\}$ show a nonmonotone behaviour.

Some of the methods considered so far fit into a more general strategy described in [28]: breaking the cycling behaviour of the SD gradients by periodically enforcing either a very small or a very large step. Some key observations are made in [28]: first, since large steps tend to increase the gradient eigencomponents associated with large eigenvalues and may increase the objective function value (see Section 2.1), very long steps should be performed after Cauchy steps, which always reduce the function value; second, if small steplengths are enforced when the eigencomponents associated with the large and "middle" eigenvalues are already small, then the gradient is dominated by the eigencomponents associated with the smallest eigenvalues and the next Cauchy steplength becomes large.[1] Based on these observations, the Cauchy-short method and its alternated variant [28] enforce short steplengths after performing Cauchy steplengths, in order to break the SD cycle. The short steplengths are Cauchy ones themselves, so that all the steplengths belong to $[1/\lambda_1, 1/\lambda_n]$ (see [28] for details).

[1] This can be explained by noting that

$$\alpha_k^{\mathrm{SD}} = \frac{g_k^T g_k}{g_k^T A g_k} = \frac{\sum_{i=1}^n (\mu_i^k)^2}{\sum_{i=1}^n (\mu_i^k)^2 \lambda_i}.$$

Likewise, the SDA and SDC methods break the SD cycle by suitably alternating Cauchy steplengths with the small steplenghts $\alpha_k^{\mathrm{A}} \approx 1/(\lambda_1 + \lambda_n)$ and $\alpha_k^{\mathrm{Y}} \approx 1/\lambda_1$, respectively. ABB and $\mathrm{ABB}_{\min}$ can be re-interpreted in light of the previous ideas too, since they enforce a large steplength α_k^{BB1} after short/medium steps of type α_k^{BB2} have been performed to reduce the eigencomponents associated with the large/medium eigenvalues.

Finally, we briefly describe gradient methods devised with the objective of approaching the optimal complexity bound for first-order methods applied to stongly-convex quadratic functions. This goal is achieved by using steplengths that are distributed in $[1/\lambda_1, 1/\lambda_n]$ according to some predefined distribution.

In [23, 25, 38] some gradient methods are proposed which select their steplengths according to the following result: if the sequence $\{1/\alpha_k\}$ is asymptotically distributed wih the arcsin probability density in $[\lambda_n, \lambda_1]$, then the asymptotic convergence rate of the corresponding gradient method approaches that of the Conjugate Gradient method [23], which is the optimal one (see, e.g., [44]). The inverses of the steplengths must be chosen as symmetric pairs, in the sense that $1/\alpha_{2k+1} = \lambda_1 + \lambda_n - 1/\alpha_{2k}$ for sufficiently large k. The previous results have been obtained by looking at the normalized gradients as probability measures over the eigenvalues of the matrix A, following the approach originally proposed in [14]. We note that λ_1 and λ_n are usually not known; therefore, practical algorithms based on this approach must provide estimates of them. Estimates based on the evaluation of moments of probability measures generated by the gradient methods are analysed in [25].

Next we report a rule for the computation of the steplength, which we refer to as Golden Arcsine (GA) rule, devised according to the previous ideas [38]:

$$\alpha_k^{GA} = \frac{1}{\beta_k}, \qquad \beta_k = \underline{\lambda}_k + (\overline{\lambda}_k - \underline{\lambda}_k) z_k, \tag{19}$$

where $\underline{\lambda}_k$ and $\overline{\lambda}_k$ are suitable approximations of the smallest and largest eigenvalues of A, respectively, and

$$z_k = (1 + cos(\pi u_k))/2, \quad u_{2j} = \min(v_j, 1 - v_j), \quad u_{2j+1} = \max(v_j, 1 - v_j), \tag{20}$$

$$v_j = \{\phi(j+1)\}, \quad \phi = \frac{\sqrt{5}+1}{2}, \tag{21}$$

with $\{a\}$ denoting the fractional part of a. The number ϕ is the well-known golden ratio. The sequence $\{\beta_k\}$ asymptotically has the arcsin distribution in $[\underline{\lambda}_k, \overline{\lambda}_k]$. More details are given in [38].

Another technique to build steplengths such that the corresponding gradient method approach the optimal complexity is based on the use of the Chebyshev nodes, i.e., the roots of the Chebyshev polynomial of the first kind. This approach has been developed in [23] and in [29], by taking different points of view. In [29] it is proved that if the steplengths are defined as

$$\alpha_k^{CH} = 1/\gamma_k, \quad \gamma_k = \frac{\overline{\lambda} - \underline{\lambda}}{2}\, t_k + \frac{\overline{\lambda} + \underline{\lambda}}{2}, \quad k = 0, \dots, N-1,$$

where $[\underline{\lambda}, \overline{\lambda}] \supset [\lambda_n, \lambda_1]$, t_k are the roots of the Chebyshev polynomial of the first kind of degree N,

$$t_k = \cos\frac{(2k+1)\pi}{2N},$$

and

$$N \approx \left\lceil \frac{1}{2}\sqrt{\frac{\overline{\lambda}}{\underline{\lambda}}}\,\log\frac{2}{\varepsilon} \right\rceil$$

($\lceil a \rceil$ denotes the smallest integer $\bar{a}$ such that $\bar{a} \geq a$), then the gradient method reduces the error in the computed solution by a factor ε in N iterations. The closer the values of $\overline{\lambda}$ and $\underline{\lambda}$ to λ_1 and λ_n, respectively, the better the complexity bound is.

An algorithm using α_k^{CH} must also provide good estimates of the extremal eigenvalues of the matrix A. Some techniques to build these estimates are discussed in [23, 25, 29]. It is worth noting that the author of [29] points out that the gradient method described there is not proposed as a practical algorithm, but only to prove that a complexity bound is achievable. However, the steplengths α_k^{CH} can be exploited to accelerate other gradient methods, as suggested in [29].

We conclude this section by observing that the previous strategies based on predefined distributions of the stepengths take into account only the extremal eigenvalues of A; they also tend to generate more steplengths near the endpoints of the interval $[1/\lambda_1, 1/\lambda_n]$. This behaviour and its outcome are discussed in the next section.

2.3. Numerical results for quadratic problems

In order to illustrate the effects of the different steplength rules described in the previous section, we analyse the numerical results obtained by solving some problems of the form (3) with gradient methods using those rules. For the sake of space, we do not consider all the methods presented in Section 2.2, but only a selection of them which, in our opinion, is representative of the approaches analysed there.

Specifically, we discuss the results obtained by running Matlab implementations[2] of the following methods:

- BB, with BB1 steplength (see (10));
- ABB$_{\min}$, with $\tau = 0.8$ and $m_a = 5$ (see (11));
- LMSD, with $m_s = 6$ (see (13));
- SDC, with $h = 3$ and $m_c = 4$ (see (15)–(16));
- GA with estimates of the extremal eigenvalues of A (see (19)–(21); we use the implementation available from `http://www.i3s.unice.fr/∼pronzato/Matlab/goldenArcsineQ.m`).

The parameters of these methods were chosen on the basis of the literature and our past numerical experience. In the LMSD method, the Ritz values used within a sweep were sorted in decreasing order, as proposed in [24], with the aim of applying large steplenghs after some iterations in which smaller steplengths had reduced the eigencomponents of the gradient corresponding to large eigenvalues (the ones that are considerably increased by the large steplengths).

[2] We used Matlab R2016a on an Intel core i7-3517U.

Table 1: Number of iterations of the selected gradient methods. The mark '—' indicates that the stopping criterion (22) has not been satisfied within 1000 iterations.

problem	BB1	ABB$_{\min}$	LMSD	SDC	GA
QP1	173	147	165	149	178
QP2	—	754	—	954	932
QP3	236	199	181	192	246

The following criterion was used to stop the iterations:

$$\|g_k\| < \varepsilon, \tag{22}$$

where $\varepsilon = 10^{-6}$; a maximum number of 1000 iterations was considered too. We modified the original GA implementation in order to stop the method as soon as (22) had been satisfied. For all the methods, the same random vector from a uniform distribution on the unit sphere was used as starting point.

Following [25, 38, 29], we considered three test problems of dimension $n = 10^3$, with Hessian matrices having different distributions of the eigenvalues. Without loss of generality, we set

$$A = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n),$$

so that d_i is the i-th vector of the canonical basis of $\mathbb{R}^n$. We chose as optimal solution a random vector x^* from a uniform distribution on the unit sphere and set $b = Ax^*$. The eigenvalues of A were defined as follows:

QP1: n eigenvalues $\lambda_i = (\underline{\lambda} b - \overline{\lambda} a)/(b-a) + (\underline{\lambda} - \overline{\lambda})/(b-a)\xi_i$, where $\underline{\lambda} = 1$, $\overline{\lambda} = 10^3$, $a = (1-c)^2$, $b = (1+c)^2$, $c = 1/2$ and the values ξ_i are distributed according to the Marčenko-Pastur density $p_c(x) = \sqrt{(b-x)(x-a)}/(2\pi x c^2)$ (roughly speaking, this distribution describes the asymptotic behaviour of the eigenvalues of a class of covariance matrices [45]);

QP2: n eigenvalues in $[\underline{\lambda}, \overline{\lambda}]$, with $\underline{\lambda} = \lambda_n = 1$, $\overline{\lambda} = \lambda_1 = 10^4$ and λ_i/λ_{i-1} constant.

QP3: n eigenvalues having a two-block distribution: $\lambda_i = \underline{\lambda} + (\overline{\lambda} - \underline{\lambda}) s_{n-i+1}$, $i = 1, ..., n$, where $\underline{\lambda} = 1$, $\overline{\lambda} = 10^3$ and the values s_i are generated from a uniform random distribution in $(0, 0.2)$ for $i = 1, ..., n/2$, and in $(0.8, 1)$ for $i = n/2 + 1, \ldots, n$.

In Figures 1–6 we show, for each problem and each method, the distribution with the iterations of the inverse of the steplength, $1/\alpha_k$, the history of the gradient norm, $\|g_k\|$, and of the function error, $f(x_k) - f(x^*)$. The horizontal lines in the pictures illustrating the distribution of $1/\alpha_k$ represent 20 eigenvalues of A with linearly spaced indices (these indices have been computed by using `round(linspace(1,n,20))`); for problem QP2, a logarithmic scale has been used on the y axis, in order to better display the eigenvalues. For all the problems we also report, in Table 1, the number of iterations performed by each method.

From Figures 1–3 we see that the ABB$_{\min}$ and SDC methods behave as described in Section 2.2: they tend to compute groups of small steplengths, interleaved with some larger steplengths, thus attempting to reduce first the eigencomponents of the gradient associated with large eigenvalues and then the remaining eigencomponents. Conversely, BB1 does not

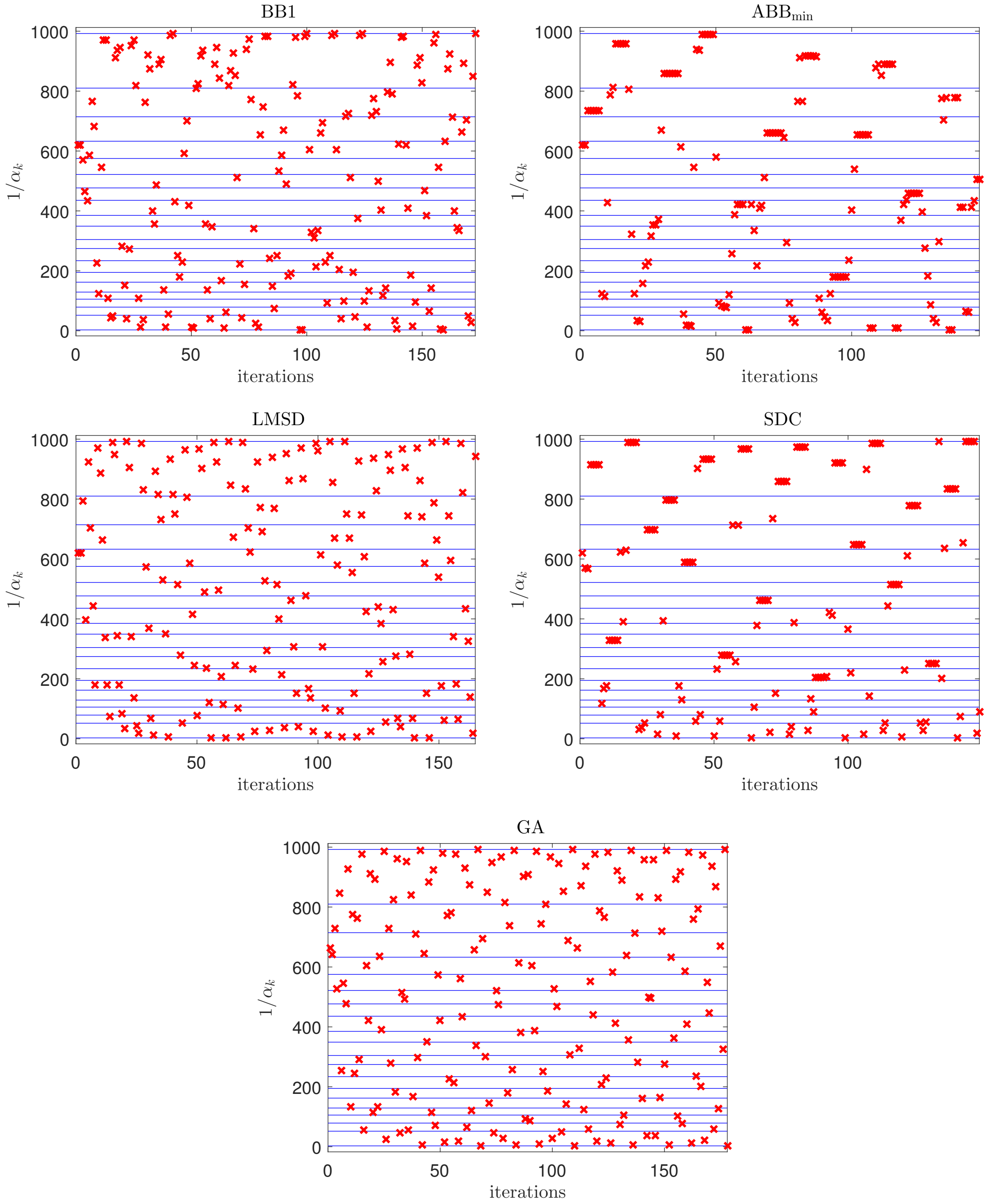

Figure 1: Test problem QP1. Distribution of $1/\alpha_k$ with the iterations. The horizontal lines correspond to 20 eigenvalues of A with linearly spaced indices.

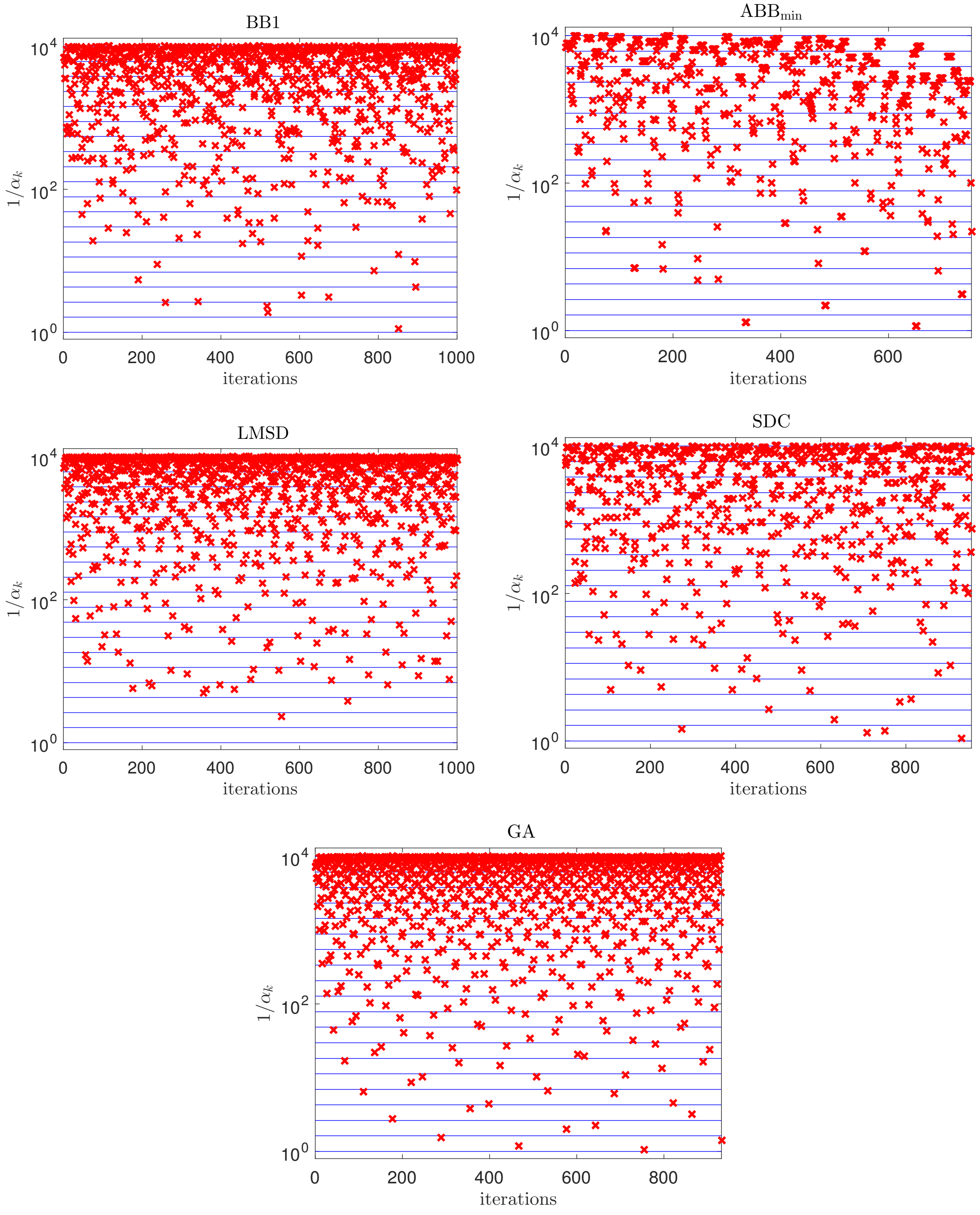

Figure 2: Test problem QP2. Distribution of $1/\alpha_k$ with the iterations. The horizontal lines correspond to 20 eigenvalues of A with linearly spaced indices.

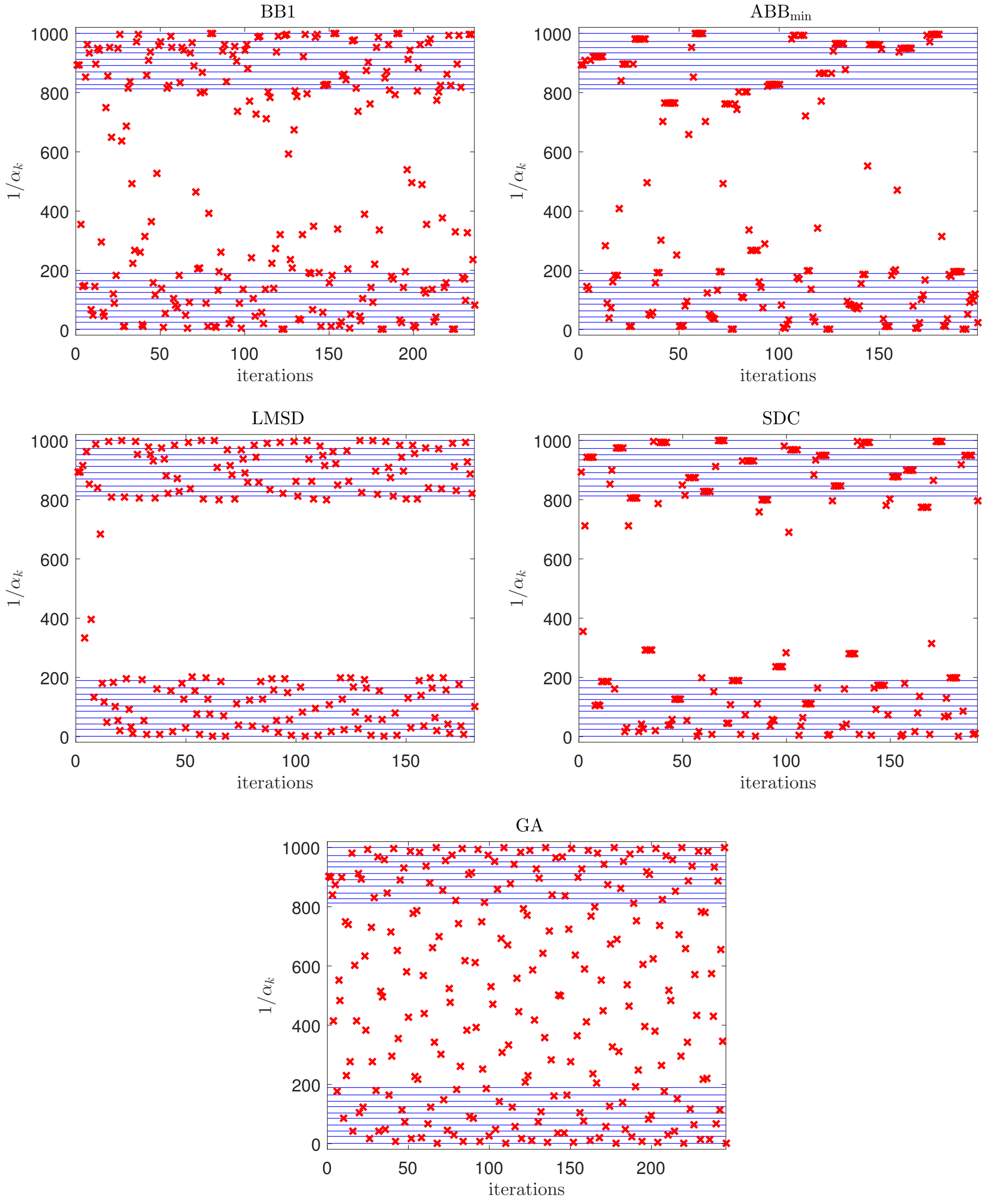

Figure 3: Test problem QP3. Distribution of $1/\alpha_k$ with the iterations. The horizontal lines correspond to 20 eigenvalues of A with linearly spaced indices.

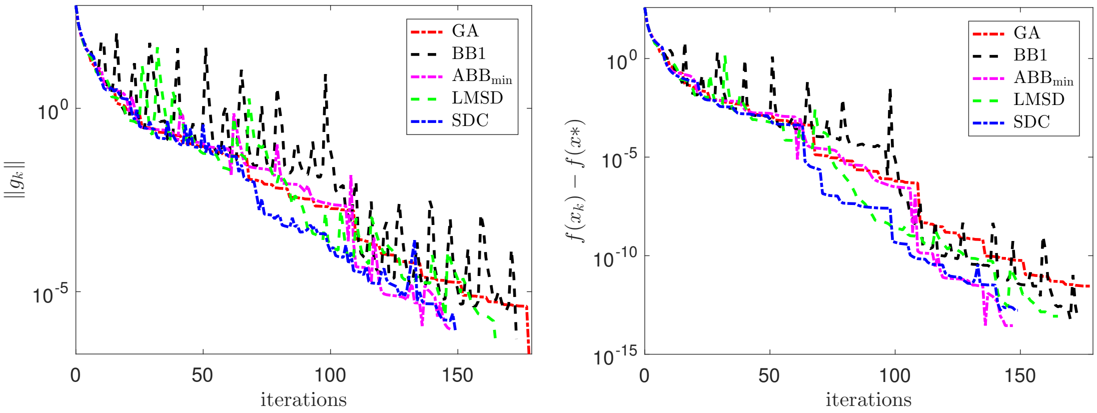

Figure 4: Test problem QP1. History of gradient norm (left) and function error (right).

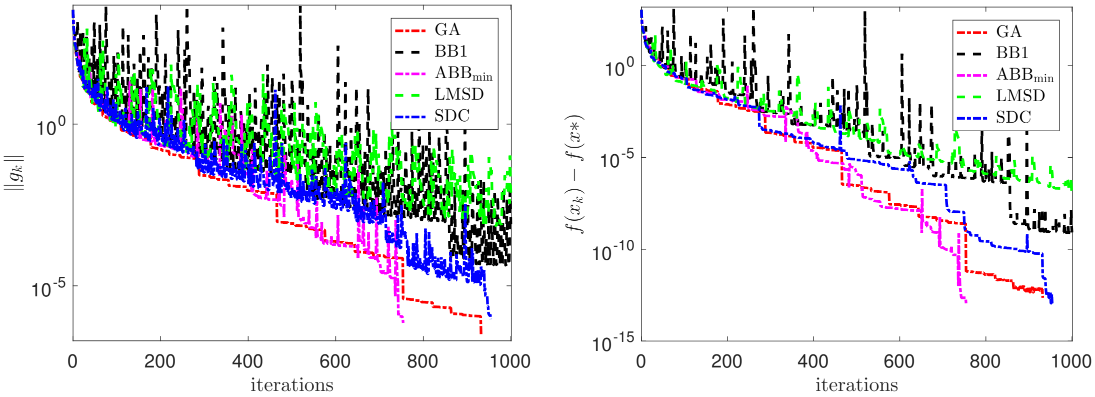

Figure 5: Test problem QP2. History of gradient norm (left) and function error (right).

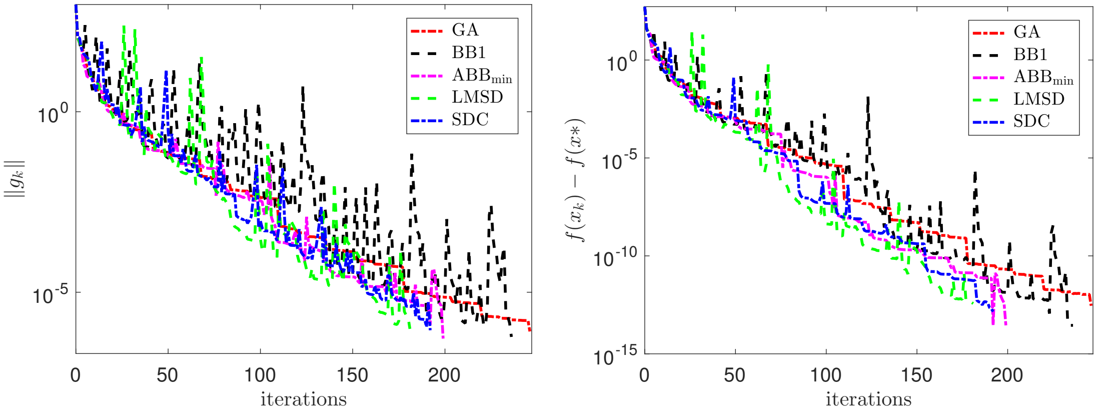

Figure 6: Test problem QP3. History of gradient norm (left) and function error (right).

appear to foster any order in the decrease of the eigencomponents of the gradient, but seems to travel in the spectrum of A in a more chaotic way. Concerning the behaviour of LMSD, the repeated use of the Lanczos procedure provides at most m_s Ritz values at each sweep, which attempt to approximate the extreme eigenvalues and a subset of the interior eigenvalues of A. The pictures also show that the steplengths of GA, according to their definition, follow a predefined path, which does not take into account the actual distribution of the eigenvalues of A in $[\lambda_n, \lambda_1]$. Furthermore, we observe that ABB$_{\min}$, LMSD and SDC are able to "catch" the actual distribution of the eigenvalues, as clearly illustrated by Figure 3.

The convergence histories, as well as the numbers of iterations, show that ABB$_{\min}$ and SDC better adapt to different distributions of the eigenvalues of A. ABB$_{\min}$ is comparable with SDC on problems QP1 and QP3 and requires fewer iterations on QP2. The performance of the remaining methods varies with the distribution of the eigenvalues of A: the number of iterations executed by BB1 and GA on QP1 is slightly larger than the number of iterations of LMSD and all the three methods appear slower than ABB$_{\min}$ and SDC; BB1 and LMSD are not able to achieve the required accuracy on QP2, while GA is comparable with SDC on this problem; finally, BB1 and GA perform more iterations than the remaining methods on problem QP3, because they do not catch the two-block distribution of the eigenvalues, while LMSD appears to be the fastest of all methods. We also observe that BB1 and LMSD produce more oscillating function values than the other methods and GA shows a monotone behaviour of both the gradient norm and function error.

3. Extension to general unconstrained minimization problems

Among the gradient methods analysed in the previous section, BB1, LMSD and ABB$_{\min}$ can be extended in a natural way to the general minimization problem (1), using line search strategies to ensure convergence to a stationary point [30, 46, 24]. In this section, after describing some generalizations of the aforementioned methods, we study their practical behaviour on selected test problems, with the aim of understanding if and how the spectral properties identified in the strongly convex quadratic case are preserved in the general one.

Henceforth the basic gradient iteration (2) is rewritten as follows:

$$x_{k+1} = x_k - \nu_k g_k, \tag{23}$$

where ν_k is the line search parameter obtained by reducing, if necessary, the tentative value α_k suggested by an appropriate steplength rule.

3.1. Gradient methods for general minimization problems

The generalizations of the BB1, ABB$_{\min}$ and LMSD methods considered in our analysis are described in Algorithms 1–3. The tentative steplengths in the BB1 and ABB$_{\min}$ methods are defined with the same updating rules introduced in the Section 2.2, except in the case $(x_{k+1} - x_k)^T (g_{k+1} - g_k) \leq 0$, where the steplength $\alpha_{k+1} = \alpha_{max}$ is used. For LMSD, the strategy for defining Ritz-like values follows the rules described in Section 2.2, but needs further explanation [24]. For general unconstrained problems, the matrix T in (14) is upper Hessenberg, but generally not tridiagonal; thus, we compute a symmetric tridiagonal matrix $\overline{T}$ by replacing the strictly upper triangle of T by the transpose of its strictly lower triangle (in Matlab notation $\overline{T} = tril(T) + tril(T, -1)')$. The Ritz-like values θ_i, $i = 1, \ldots, m_s$, defining the tentative steplengths for the next sweep via equation (13), are the eigenvalues

Algorithm 1: BB1 for general unconstrained minimization problems

```
Initialization:  $x_0 \in \mathbb{R}^n$,  $0 < \alpha_{min} \leq \alpha_{max}$,  $\alpha_0 \in [\alpha_{min}, \alpha_{max}]$,  $\varepsilon > 0$,
                 $\delta, \sigma \in (0,1)$,  $M \in \mathbb{N}$;
for  $k = 0, 1, \ldots$
    $\nu_k = \alpha_k$;   $f_{ref} = \max\{f(x_{k-j}),\ 0 \leq j \leq \min(k, M)\}$;
    while  $f(x_k - \nu_k g_k) > f_{ref} - \sigma \nu_k g_k^T g_k$                    (line search)
        $\nu_k = \delta \nu_k$;
    end
    $x_{k+1} = x_k - \nu_k g_k$;
    if  $\|g_{k+1}\| \leq \varepsilon \|g_0\|$  stop;
    $z = -g_k^T (g_{k+1} - g_k)$;
    if  $z > 0$                                                     (tentative steplength)
        $\alpha_{k+1} = \max\left\{\alpha_{min}, \min\left\{\frac{\nu_k g_k^T g_k}{z}, \alpha_{max}\right\}\right\}$;
    else
        $\alpha_{k+1} = \alpha_{max}$;
    end
end
```

of $\overline{T}$. The case of non-positive eigenvalues is handled by simply discarding these values, hence providing fewer than m_s steplengths for the next sweep; if no positive eigenvalues are available, any tentative steplength can be adopted for a sweep of length 1 (e.g., we use the initial steplength). The presence of non-positive eigenvalues highlights critical situations, which can originate from either a non-positive curvature or an inadequate approximation of the eigenvalues of the current Hessian. In this case, in addition to discarding the non-positive eigenvalues, we find convenient to discard also the oldest back gradients. Furthermore, regardless of the steplength rule, all the methods keep the sequence of tentative steplengths $\{\alpha_k\}$ bounded below and above by the positive constants α_{min} and α_{max}.

Concerning the line search strategy, our choice is driven not only by the theoretical need to introduce some form of monotonicity in the sequence $\{f(x_k)\}$, but also by the purpose of keeping unchanged as much as possible the steplength provided by the selected rule. To this end, we exploit the Grippo-Lampariello-Lucidi (GLL) nonmonotone line search [47]. When the tentative steplength is provided by the BB1 or $\mathrm{ABB}_{\mathrm{min}}$ rule, we use this line search strategy with a predefined value for the memory parameter M. In the LMSD case, following the proposal in [24], we modify the line search strategy by setting f_{ref} equal to the value of the objective function at the beginning of the sweep to which x_{k+1} belongs; when a tentative steplength does not produce a sufficient reduction with respect to f_{ref}, the steplength is adjusted by backtracking and the current sweep is interrupted. As a consequence, the memory parameter may vary during the sweep, with a maximum value bounded by m_s. As suggested in [24], the sweep is terminated also in the iterations where the gradient norm increases, since this situation is likely to generate an unproductive new steplength because of the increasing order in which the tentative steplengths are applied. In all the situations where a sweep is prematurely ended after $\bar{l}$ steps, only the most recent $\bar{l}$ gradients are kept and a smaller matrix $\overline{T}$ is computed to generate the next sweep.

Algorithm 2: ABB$_{\min}$ for general unconstrained minimization problems

```
Initialization:  $x_0 \in \mathbb{R}^n$,  $0 < \alpha_{min} \le \alpha_{max}$,  $\alpha_0 \in [\alpha_{min}, \alpha_{max}]$,  $\varepsilon > 0$,
                 $\delta, \sigma, \tau \in (0,1)$,  $M, m_a \in \mathbb{N}$;
for  $k = 0, 1, \ldots$
    $\nu_k = \alpha_k$;   $f_{ref} = \max\{f(x_{k-j}),\ 0 \le j \le \min(k, M)\}$;
    while  $f(x_k - \nu_k g_k) > f_{ref} - \sigma \nu_k g_k^T g_k$                    (line search)
        $\nu_k = \delta \nu_k$;
    end
    $x_{k+1} = x_k - \nu_k g_k$;
    if  $\|g_{k+1}\| \le \varepsilon \|g_0\|$  stop;
    $y = g_{k+1} - g_k$;
    $z = -g_k^T y$;
    if  $z > 0$                                                         (tentative steplength)
        $\alpha_{k+1}^{\mathrm{BB1}} = \max\left\{\alpha_{min}, \min\left\{\frac{\nu_k g_k^T g_k}{z}, \alpha_{max}\right\}\right\}$;
        $\alpha_{k+1}^{\mathrm{BB2}} = \max\left\{\alpha_{min}, \min\left\{\frac{\nu_k z}{y^T y}, \alpha_{max}\right\}\right\}$;
        if $\frac{\alpha_{k+1}^{\mathrm{BB2}}}{\alpha_{k+1}^{\mathrm{BB1}}} < \tau$
            $\alpha_{k+1} = \min\left\{\alpha_j^{\mathrm{BB2}} \ : \ j = \max\{1, k+1-m_a\}, \ldots, k+1\right\}$
        else
            $\alpha_{k+1} = \alpha_{k+1}^{\mathrm{BB1}}$
        end
    else
        $\alpha_{k+1} = \alpha_{max}$;
    end
end
```

Thanks to the line search strategy and the boundedness of $\{\alpha_k\}$, the gradient methods considered in this section satisfy a basic convergence result [47, p. 709], which we state in the following theorem for completeness.

Theorem 2. *Assume that $\Omega = \{x \in \mathbb{R}^n : f(x) \le f(x_0)\}$ is a bounded set and f is continuously differentiable in some neighborhood of Ω. Let $\{x_k\}$ be the sequence defined by*

$$x_{k+1} = x_k - \nu_k g_k,$$

with $\nu_k = \alpha_k \delta^{h_k}$, where $\alpha_k \in [\alpha_{min}, \alpha_{max}]$, $0 < \alpha_{min} \le \alpha_{max}$, $\delta \in (0,1)$ and h_k is the first nonnegative integer such that

$$f(x_k - \alpha_k \delta^{h_k} g_k) \le \max_{0 \le j \le m(k)} f(x_{k-j}) - \sigma \alpha_k \delta^{h_k} \|g_k\|^2, \tag{24}$$

with $m(0) = 0$, $m(k) \le \min(m(k-1)+1, M)$, $k \ge 1$, $M \in \mathbb{N}$, $\sigma \in (0,1)$. Then, either $g_j = 0$ for some j, or the following properties hold:

Algorithm 3: LMSD for general unconstrained minimization problems

```
Initialization:  $x_0 \in \mathbb{R}^n$,  $0 < \alpha_{min} \leq \alpha_{max}$,  $\alpha_0 \in [\alpha_{min}, \alpha_{max}]$,  $\varepsilon > 0$,
                 $\theta_1 = 1/\alpha_0$,  $\delta, \sigma \in (0,1)$,  $l = 1$,  $m_s \in \mathbb{N}_+$;
for  $k = 0, 1, \dots$
    $f_{ref} = f(x_k)$;
    while  $l > 0$
        $\alpha_k = \theta_l^{-1}$;                                        (tentative steplength)
        $\nu_k = \max\{\alpha_{min}, \min\{\alpha_k, \alpha_{max}\}\}$;
        $x_{k+1} = x_k - \nu_k g_k$;
        if $f(x_{k+1}) \leq f_{ref} - \sigma \nu_k g_k^T g_k$
            $l = l - 1$;
            if  $\|g_{k+1}\| \geq \|g_k\|$
                $l = 0$;
            end
        else
            repeat                                                   (line search)
                $\nu_k = \delta \nu_k$;
            until  $f(x_k - \nu_k g_k) \leq f_{ref} - \sigma \nu_k g_k^T g_k$
            $x_{k+1} = x_k - \nu_k g_k$;
            $l = 0$;
        end
        if  $\|g_{k+1}\| \leq \varepsilon \|g_0\|$  stop;
    end
    Compute up to $m_s$ new Ritz-like values:
    $0 < \theta_1 \leq \theta_2 \leq \cdots \leq \theta_l$,  $l \leq m_s$;
end
```

(i) $\lim_k \|g_k\| = 0$*;*

(ii) no limit point of $\{x_k\}$ *is a local maximum of* f*;*

(iii) if the number of stationary points of f *in* Ω *is finite, then the sequence* $\{x_k\}$ *converges.*

In [46] the R-linear rate of convergence is discussed for nonmonotone line search methods when f is bounded below, strongly convex and with Lipschitz–continuous gradient. Under these assumptions, R-linear convergence to a minimum value is established for the sequence $\{f(x_k)\}$, where x_k is generated by any iterative method of the form $x_{k+1} = x_k + \nu_k d_k$, with d_k such that $g_k^T d_k \leq -c_1 \|g_k\|^2$ and $\|d_k\| \leq c_2 \|g_k\|$, $c_1, c_2 > 0$, equipped with a nonmonotone line search to update ν_k. Obviously, the negative gradient direction $d_k = -g_k$ satisfies the previous assumptions with $c_1 = c_2 = 1$. Furthermore, in [46], with the same assumptions on f and d_k, the conditions are established under which the tentative steplength is always accepted when suitable parameters are used in the nonmonotone line search. These results allow to obtain local R-linear convergence of the BB1 method for general objective functions when the iterate is close to the solution and a convenient choice of parameters for the nonmonotone line search is made [40].

Other nonmonotone line search strategies have been proposed besides the classical GLL one (see, e.g., [32, 48]). However, the performance of these strategies seems to be related to specific steplength choices; therefore, they do not appear convenient for the analysis which is the focus of this section. Finally, we recall that when the globalization of the gradient algorithm is obtained by a simple monotone line search, the convergence of the sequence of iterates $\{x_k\}$ to a minimizer of f is proved under the assumption that f is bounded below, convex and continuously differentiable [49, 50]. When ∇f is also Lipschitz-continuous, the rate of convergence of $\{f(x_k)\}$ to a minimum is $\mathcal{O}(\frac{1}{k})$ [51].

3.2. Numerical results for general minimization problems

In order to analyse the practical behaviour of the methods described in Section 3.1, we applied them to the well known test problems described next.

NQP1: Trigonometric test problem [52]. The objective function is

$$f(x) = \|b - (Av(x) + Bu(x))\|^2,$$

where $v(x) = (\sin(x_1), ..., \sin(x_n))^T$, $u(x) = (\cos(x_1), ..., \cos(x_n))^T$, and A and B are square matrices of order n with entries generated as random integers in $(-100, 100)$. A vector $x^* \in \mathbb{R}^n$ is computed with entries randomly generated from a uniform distribution in $(-\pi, \pi)$ and the vector b is defined so that $f(x^*) = 0$, i.e. x^* is a minimum point. The starting vector is set as $x_0 = x^* + 0.1\, r$, where $r \in \mathbb{R}^n$ has random entries from a uniform distribution in $[-\pi, \pi]$.

NQP2: Convex2 test problem [30]. The objective function is

$$f(x) = \sum_{i=1}^{n} \frac{i}{10}(e^{x_i} - x_i);$$

this is a strictly convex problem, since the Hessian is a diagonal matrix with diagonal entries equal to $\frac{i}{10}e^{x_i}$, $i = 1, ..., n$. The solution x^* is the zero vector and the minimum value is $f(x^*) = \frac{n(n+1)}{20}$; the starting vector is $x_0 = (1, 1, ..., 1)^T$.

NQP3: Chained Rosenbrock test problem [53]. The objective function is

$$f(x) = \sum_{i=2}^{n}(4\varphi_i(x_{i-1} - x_i^2)^2 - (1 - x_i)^2),$$

where the values φ_i are defined in [53, Table 1] for $n = 50$. In our experiments we also consider $n = 100, 200$ and set $\varphi_{i+50} = \varphi_i$, $i = 1, ..., 50$, for $n = 100$, and $\varphi_{i+50j} = \varphi_i$, $i = 1, ..., 50$, $j = 1, 2, 3$, for $n = 200$. A solution of the problem is $x^* = (1, 1, ..., 1)^T$; the starting vector x_0 is the zero vector.

NQP4: Laplace2 test problem [33]. The objective function is

$$f(x) = \frac{1}{2}x^T Ax - b^T x + \frac{1}{4}h^2 \sum_i x_i^4,$$

where A is a square matrix of order $n = N^3$, arising from the discretization of a 3D Laplacian on the unit box by a standard seven-point finite difference formula. The discretization spacing on each coordinate direction is $h = \frac{1}{N+1}$ and the vector b is chosen so that the entries of the solution x^* of the minimization problem are

$$x_i \equiv x(kh, rh, sh) = h^3 krs(kh-1)(rh-1)(sh-1)e^{-\frac{1}{2}d^2((kh-d_1)^2+(rh-d_2)^2+(sh-d_3)^2)},$$

where the index i is associated with the mesh point (kh, rh, sh), $k, r, s = 1, ..., N$. The parameters d, d_1, d_2 and d_3 are chosen in two different ways:

(a) $d = 20, d_1 = d_2 = d_3 = 0.5$

(b) $d = 50, d_1 = 0.4, d_2 = 0.7, d_3 = 0.5$.

In both cases, $N = 100$ and a starting vector with entries randomly generated from a uniform distribution in $(0, 1)$ is considered.

The experiments were carried out by using the same setting for the parameters common to the different methods: $\delta = 0.5$, $\sigma = 10^{-4}$, $\alpha_{min} = 10^{-10}$, $\alpha_{max} = 10^5$. The remaining parameters were chosen as follows: $M = 9$ in BB1 and ABB_{min}, $\tau = 0.5$ and $m_\alpha = 5$ in ABB_{min}, $m_s = 3$ and $m_s = 5$ in LMSD. An initial steplength equal to 1 was used by all the methods; the value of ε in the relative stopping criterion (see Algorithms 1-3) was set as 10^{-7} for NQP1, NQP2 and NQP3, and 10^{-6} for NQP4 and a maximum number of 5000 iterations was considered too. Note that, for all the test problems, the sequence $\{x_k\}$ generated by each method approached x^*.

A first set of experiments was aimed at evaluating how the sequences $\{1/\nu_k\}$ generated by the different methods are distributed with respect to the eigenvalues of the current Hessian. To this end, we considered small-size instances of NQP1, NQP2 and NQP3, i.e., $n = 50$ for NQP1 and NQP3, and $n = 100$ for NQP2. The corresponding values of $\{1/\nu_k\}$ at each iteration are shown in Figures 7–9, using a logarithmic scale on the y axis for better readability. In the pictures, the mark 'o' denotes a value of $1/\nu_k$ obtained by backtracking, while '×' indicates a value of $1/\nu_k$ satisfying the nonmonotone sufficient decrease condition without backtracking. At each iteration, we also depict a subset of the eigenvalues of the Hessian matrix, by using blue dots. More precisely, at each iteration we sort the eigenvalues of the Hessian and plot those corresponding to 20 linearly spaced indices, provided that they take positive values (otherwise, we plot a smaller number of eigenvalues). We also represent by green squares a subset of the eigenvalues of the Hessian matrix at the solution, selected with the same procedure.

The figures show a behaviour similar to that observed in the quadratic case. The sequence $\{1/\nu_k\}$ generated by BB1 takes values that travel in the spectra of the Hessian matrices in a chaotic way. ABB_{min} favours, through the BB2 rule, the computation of steplengths whose inverse values approximate the largest eigenvalues of the Hessian matrices; when s_{k-1} and y_{k-1} tend to be aligned, the method attempts to catch small eigenvalues by using the BB1 rule. The values of $1/\nu_k$ generated by LMSD during a sweep attempt to travel in the spectra of the Hessian matrices corresponding to that sweep; in particular, the extreme Ritz values obtained in a sweep can be considered as an attempt to approximate the extreme eigenvalues of the Hessians in that sweep. Nevertheless, as shown in Figure 8, when x_k is far from x^*, the LMSD method with $m_s = 5$ generates some very small steplengths whose inverses fall out of the spectra of the Hessian matrices; the choice $m_s = 3$ mitigates this drawback, thanks to the smaller number of previous gradients taken into account. However, as x_k approaches x^*,

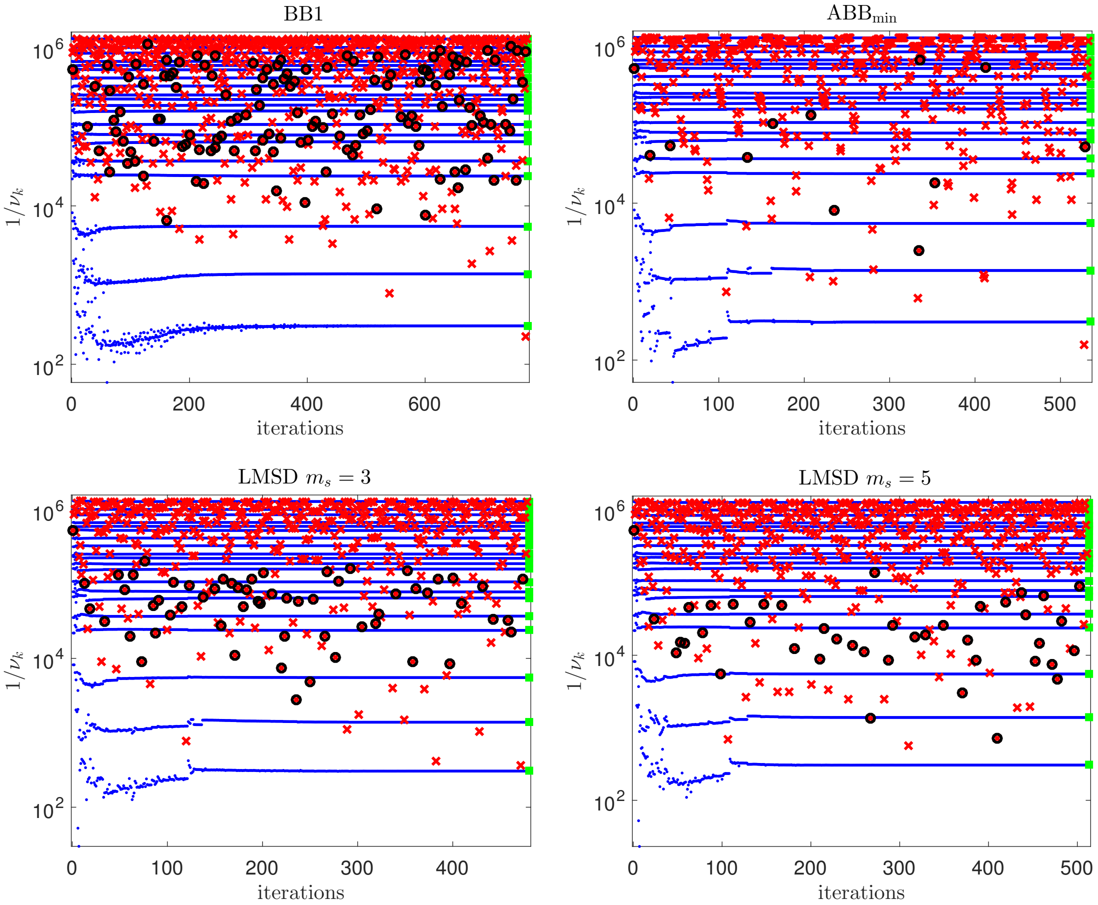

Figure 7: Test problem NQP1, $n = 50$. Distribution of $1/\nu_k$ with the iterations. (At most) 20 positive eigenvalues of the Hessian, with linearly spaced indices, are also represented at each iteration.

LMSD shows a behaviour closer to that observed in the convex quadratic case. We also see that the steplength reduction occurs in a few iterations, especially for ABB$_{\min}$; in general, BB1 applies backtracking more often than the other methods, and LMSD with $m_s = 3$ more often than LMSD with $m_s = 5$.

In Figures 10–12 we show the histories of the gradient norm and the error function for the previous small-size problems. All the methods have the oscillating behaviour observed in the quadratic case; furthermore, ABB$_{\min}$ and LMSD appear more effective, according to their capability of better catching significant information about the spectrum of the Hessian.

Further experiments were performed to confirm the behaviour of the methods as the size of the problems increases. We run BB1, ABB$_{\min}$ and LMSD, with $m_s = 3$ and $m_s = 5$, on larger instances of NQP1 ($n = 100, 200$), NQP2 ($n = 10000, 100000$) and NQP3 ($n = 100, 200$), as well as on the NQP4(a) and NQP4(b) test problems. In Tables 2-3 we report the number, it, of iterations performed by each method, the number, H, of iterations where the steplength was reduced, and the errors in the computed solution and the associated function value, $err_x = \|x_{it} - x^*\|$ and $err_f = f(x_{it}) - f(x^*)$. For LMSD, in the column

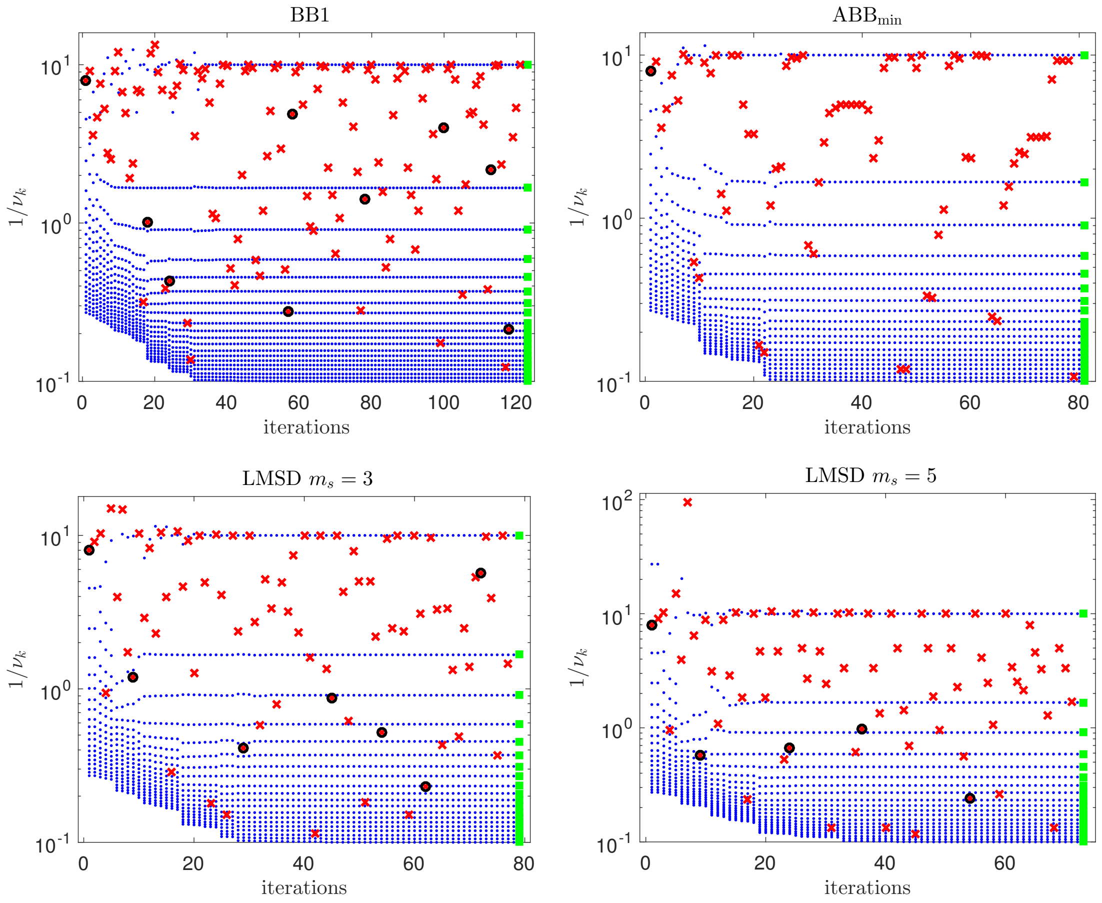

Figure 8: Test problem NQP2, $n = 100$. Distribution of $1/\nu_k$ with the iterations. (At most) 20 positive eigenvalues of the Hessian, with linearly spaced indices, are also represented at each iteration.

labelled by *it* we also report, in brackets, the number of sweeps.

The results confirm that the number of steplength reductions is generally very small with respect to the total number of iterations; we remark that H takes the smallest value for $\mathrm{ABB}_{\min}$ and is generally smaller for LMSD than for BB1. Except for NQP1, $\mathrm{ABB}_{\min}$ requires fewer iterations than LMSD. Furthermore, the number of iterations of $\mathrm{ABB}_{\min}$ ranges between 27% and 69% of the number of iterations of BB1; on NQP1, the latter method is not able to achieve the required accuracy within 5000 iterations. The number of iterations of LMSD is generally a greater percentage of the number of iterations of BB1; nevertheless, for large problems this may save computing time.

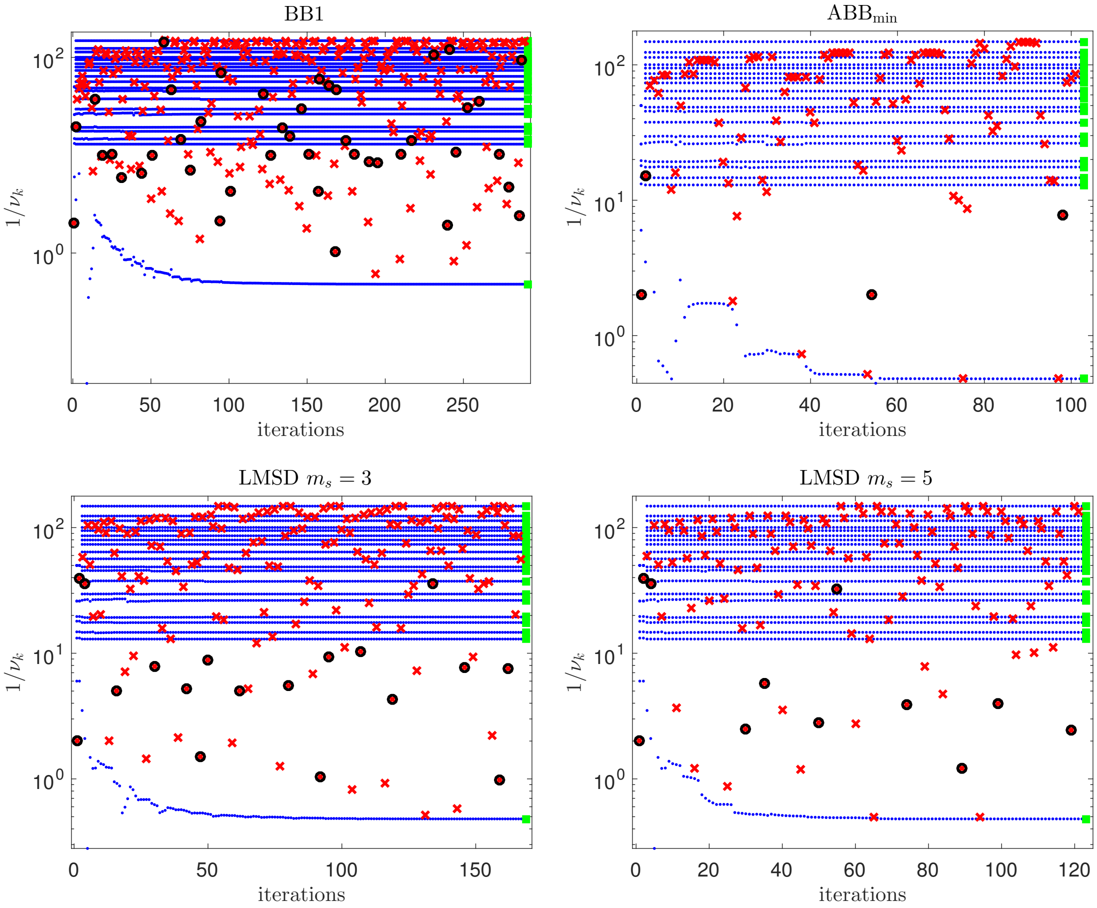

Figure 9: Test problem NQP3, $n = 50$. Distribution of $1/\nu_k$ with the iterations. (At most) 20 positive eigenvalues of the Hessian, with linearly spaced indices, are also represented at each iteration.

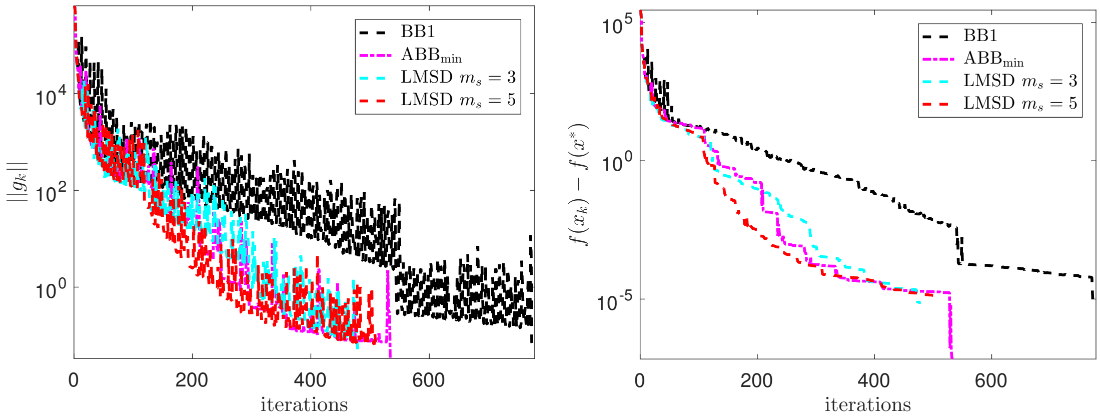

Figure 10: Test problem NQP1, $n = 50$. History of gradient norm (left) and function error (right).

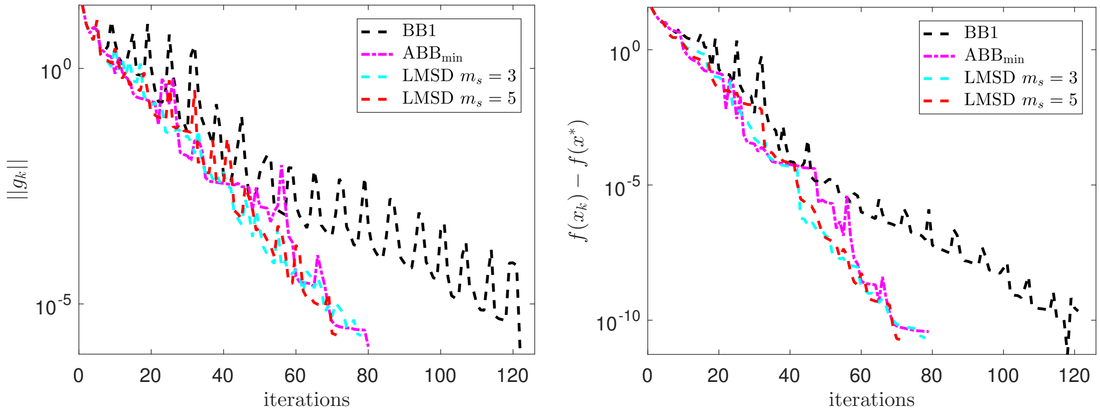

Figure 11: Test problem NQP2, $n = 100$. History of gradient norm (left) and function error (right).

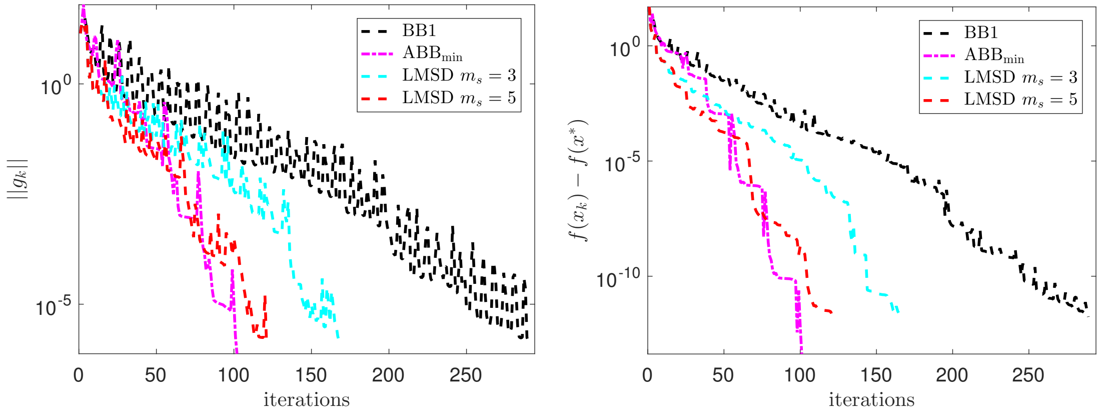

Figure 12: Test problem NQP3, $n = 50$. History of gradient norm (left) and function error (right).

Table 2: Numerical results for problems NQP1, NQP2, NQP3. The mark '—' indicates that the stopping criterion has not been satisfied within 5000 iterations.

method	it	H	err_x	err_f
NQP1 – $n = 100$, $\|g_0\|$ =1.62e+6, ε =1.00e–7				
BB1	—	975	6.40e–2	5.90e–2
ABB$_{\min}$	2953	24	6.30e–4	2.09e–4
LMSD ($m_s = 3$)	3932 (1340)	496	2.56e–3	9.97e–5
LMSD ($m_s = 5$)	2542 (531)	183	8.38e–4	1.77e–5
NQP1 – $n = 200$, $\|g_0\|$ =4.51e+6, ε =1.00e–7				
BB1	—	906	4.42e–2	1.86e–2
ABB$_{\min}$	2316	19	3.46e–3	3.06e–4
LMSD ($m_s = 3$)	3211 (1097)	391	2.19e–2	4.38e–3
LMSD ($m_s = 5$)	2076 (429)	148	8.01e–3	6.42e–4
NQP2 – $n = 10000$, $\|g_0\|$ =2.20e+1, ε =1.00e–7				
BB1	1533	269	1.38e–3	9.82e–10
ABB$_{\min}$	410	13	7.94e–4	7.87e–9
LMSD ($m_s = 3$)	706 (268)	98	1.54e–3	1.22e–9
LMSD ($m_s = 5$)	612 (179)	49	1.77e–3	1.68e–9
NQP2 – $n = 100000$, $\|g_0\|$ =2.20e+1, ε =1.00e–7				
BB1	2615	463	1.85e–2	1.80e–8
ABB$_{\min}$	729	19	1.21e–2	1.94e–8
LMSD ($m_s = 3$)	2226 (830)	334	5.00e–3	1.30e–9
LMSD ($m_s = 5$)	1864 (506)	124	2.02e–2	2.10e–8
NQP3 – $n = 100$, $\|g_0\|$ =1.99e+1, ε =1.00e–7				
BB1	147	21	6.83e–7	5.69e–13
ABB$_{\min}$	102	3	4.00e–6	3.89e–12
LMSD ($m_s = 3$)	175 (61)	24	6.57e–7	1.04e–13
LMSD ($m_s = 5$)	138 (32)	10	1.96e–6	2.19e–13
NQP3 – $n = 200$, $\|g_0\|$ =1.99e+1, ε =1.00e–7				
BB1	290	43	3.98e–6	1.15e–11
ABB$_{\min}$	95	4	2.10e–6	2.53e–12
LMSD ($m_s = 3$)	147 (51)	16	5.68e–6	7.74e–12
LMSD ($m_s = 5$)	135 (31)	12	5.02e–6	6.06e–12

4. Conclusions

The analysis of the relationship between the steplengths of some gradient methods and the spectrum of the Hessian of the objective function seems to provide insight into the computational effectiveness of those methods. For convex quadratic problems, it is especially interesting to follow the distribution of the inverse of the steplength with the iterations. This distribution shows that the way the different rules alternate small and large steplengths strongly affects the effectiveness of the methods. In particular, the methods that tend to use groups of small steplengths followed by some large steplengths, attempting to approximate the inverses of some eigenvalues of the Hessian matrix, exhibit better numerical behaviour. Furthermore, some methods better adapt to different distributions of

Table 3: Numerical results for problems NQP4(a) and NQP4(b).

method	it	H	err_x	err_f
NQP4(a) – $n = 10^6$, $\lVert g_0 \rVert = 1.87e + 3$, ε =1.00e–6				
BB1	1122	217	6.32e–1	5.83e–4
ABB$_{\min}$	306	9	1.71e–1	5.87e–4
LMSD ($m_s = 3$)	430 (147)	46	6.16e–1	5.51e–4
LMSD ($m_s = 5$)	427 (90)	34	2.08e–1	9.98e–5
NQP4(b) – $n = 10^6$, $\lVert g_0 \rVert = 1.87e + 3$, ε =1.00e–6				
BB1	624	114	4.63e–1	3.11e–4
ABB$_{\min}$	291	9	4.76e–1	3.43e–4
LMSD ($m_s = 3$)	568 (194)	76	1.37e–1	2.73e–5
LMSD ($m_s = 5$)	441 (93)	38	4.00e–1	2.33e–4

the eigenvalues, thus resulting more efficient. For general unconstrained problems, gradient methods using steplength strategies that are natural extensions of effective rules devised for the convex quadratic case seem to preserve the behaviour of their quadratic counterparts. More precisely, they appear to follow somehow the spectrum of the Hessian of the objective function during their progress toward a stationary point, achieving a significant improvement over the standard Barzilai-Borwein approach.

Acknowledgements

This work was partially supported by INdAM-GNCS (Progetti 2016).

References

[1] E. G. Birgin, I. Chambouleyron, J. M. Martínez, Estimation of the optical constants and the thickness of thin films using unconstrained optimization, J. Comput. Phys. 151 (1999) 862–880.

[2] T. Serafini, G. Zanghirati, L. Zanni, Gradient projection methods for large quadratic programs and applications in training support vector machines, Optim. Methods Softw. 20 (2005) 353–378.

[3] Y. H. Dai, R. Fletcher, New algorithms for singly linearly constrained quadratic programming problems subject to lower and upper bounds, Math. Programming 106 (2006) 403–421.

[4] M. A. Figueiredo, R. D. Nowak, S. J. Wright, Projection for sparse reconstruction: Application to compressed sensing and other inverse problems, IEEE Journal of Selected Topics in Signal Processing 1 (2007) 586–597.

[5] M. Zhu, S. J. Wright, T. F. Chan, Duality-based algorithms for total-variation-regularized image restoration, Comput. Optim. Appl. 47 (3) (2008) 377–400.

[6] G. Yu, L. Qi, Y. H. Dai, On nonmonotone Chambolle gradient projection algorithms for total variation image restoration, J. Math. Imaging Vis. 35 (2009) 143–154.

[7] I. Loris, M. Bertero, C. De Mol, R. Zanella, L. Zanni, Accelerating gradient projection methods for l1-constrained signal recovery by steplength selection rules, Applied and Computational Harmonic Analysis 27 (2009) 247–154.

[8] R. Zanella, G. Zanghirati, R. Cavicchioli, L. Zanni, P. Boccacci, M. Bertero, G. Vicidomini, Towards real-time image deconvolution: application to confocal and sted microscopy, Sci. Rep. 3 (2013) 2523.

[9] E. G. Birgin, J. M. Martínez, M. Raydan, Spectral projected gradient methods: Review and perspectives, J. Stat. Soft. 60 (3) (2014) 1–21.

[10] L. Antonelli, V. De Simone, D. di Serafino, On the application of the spectral projected gradient method in image segmentation, J. Math. Imaging Vis. 54 (1) (2015) 106–116.

[11] R. De Asmundis, D. di Serafino, G. Landi, On the regularizing behavior of the SDA and SDC gradient methods in the solution of linear ill-posed problems, J. Comput. Appl. Math. 302 (2016) 81–93.

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

[13] A. Cauchy, Méthodes générales pour la résolution des systèmes d'équations simultanées, CR. Acad. Sci. Par. 25 (1847) 536–538.

[14] 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 (1959) 1–16.

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

[16] M. Raydan, B. F. Svaiter, Relaxed steepest descent and Cauchy-Barzilai-Borwein method, Comput. Optim. Appl. 21 (2002) 155–167.

[17] Y. H. Dai, Alternate step gradient method, Optimization 53 (2003) 395–415.

[18] Y. H. Dai, Y. Yuan, Alternate minimization gradient method, IMA J. Numer. Anal. 23 (2003) 377–393.

[19] Y. H. Dai, R. Fletcher, On the asymptotic behaviour of some new gradient methods, Math. Programming 103 (2005) 541–559.

[20] B. Zhou, L. Gao, Y. H. Dai, Gradient methods with adaptive step-sizes, Comput. Optim. Appl. 35 (1) (2006) 69–86.

[21] Y. H. Dai, Y. Yuan, Analyses of monotone gradient methods, J. Ind. Manag. Optim. 1 (2005) 181–192.

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

[23] L. Pronzato, A. Zhigljavsky, Gradient algorithms for quadratic optimization with fast convergence rates, Comput. Optim. Appl. 50 (2011) 597–617.

[24] R. Fletcher, A limited memory steepest descent method, Math. Program., Ser. A 135 (2012) 413–436.

[25] L. Pronzato, A. Zhigljavsky, E. Bukina, Estimation of spectral bounds in gradient algorithms, Acta Appl. Math. 127 (2013) 117–136.

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

[27] R. De Asmundis, D. di Serafino, W. Hager, G. Toraldo, H. Zhang, An efficient gradient method using the Yuan steplength, Comput. Optim. Appl. 59 (3) (2014) 541–563.

[28] C. Gonzaga, R. M. Schneider, On the steepest descent algorithm for quadratic functions, Comput. Optim. Appl. 63 (2) (2016) 523–542.

[29] C. C. Gonzaga, On the worst case performance of the steepest descent algorithm for quadratic functions, Math. Program., Ser. A 160 (2016) 307–320.

[30] M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, SIAM J. Optim. 7 (1) (1997) 26–33.

[31] E. G. Birgin, J. M. Martínez, M. Raydan, Nonmonotone spectral projected gradient methods on convex sets, SIAM J. Optim. 10 (2000) 1196–1211.

[32] Y. H. Dai, H. Zhang, Adaptive two-point stepsize gradient algorithm, Numerical Algorithms 27 (2001) 377–385.

[33] R. Fletcher, On the Barzilai-Borwein method, in: L. Qi, K. Teo, X. Yang, P. M. Pardalos, D. Hearn (Eds.), Optimization and Control with Applications, Vol. 96 of Applied Optimization, Springer, US, 2005, pp. 235–256.

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

[35] S. Bonettini, R. Zanella, L. Zanni, A scaled gradient projection method for constrained image deblurring, Inverse Probl. 25 (1) (2009) 015002.

[36] F. Porta, M. Prato, L. Zanni, A new steplength selection for scaled gradient methods with application to image deblurring, J. Sci. Comp. 65 (2015) 895–919.

[37] D. di Serafino, V. Ruggiero, G. Toraldo, L. Zanni, A note on spectral properties of some gradient methods, in: Numerical Computations: Theory and Algorithms (NUMTA-2016), Vol. 1776 of AIP Conference Proceedings, 2016.

[38] L. Pronzato, A. Zhigljavsky, E. Bukina, An asymptotically optimal gradient algorithm for quadratic optimization with low computational cost, Optimization Letters 7 (6) (2013) 1047–1059.

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

[40] Y. H. Dai, L. Z. Liao, R-linear convergence of the Barzilai and Borwein gradient method, IMA J Numer Anal 22 (1) (2002) 1–10.

[41] G. H. Golub, C. F. Van Loan, Matrix Computations, 3rd edn., Applied Optimization, John Hopkins University Press, Baltimore and London, 1996.

[42] F. E. Curtis, W. Guo, R-linear convergence of limited memory steepest descent, Tech. Rep. arXiv:1610.03831 (2016).

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

[44] A. S. Nemirovski, D. B. Yudin, Problem Complexity and Method Efficiency in Optimization, Interscience Series in Discrete Mathematics, Wiley, 1983.

[45] V. A. Marčenko, L. A. Pastur, Distribution of eigenvalues for some sets of random matrices, Mathematics of the USSSR - Sbornik 1 (4) (1967) 457–483.

[46] Y. H. Dai, On the nonmonotone line search, J. Optim. Theory Appl. 112 (2002) 315–330.

[47] L. Grippo, F. Lampariello, S. Lucidi, A nonmonotone line search technique for Newton's method, SIAM J. Numer. Anal. 23 (1986) 707–716.

[48] H. Zhang, W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, SIAM J. Optim. 14 (4) (2004) 1045–1056.

[49] Y. E. Nesterov, Effective Methods in Nonlinear Programming, Moscow, 1989.

[50] R. Burachik, L. M. Graña Drummond, A. N. Iusem, B. F. Svaiter, Full convergence of the steepest descent method with inexact line searches, Optimization 32 (1995) 137–146.

[51] Y. Nesterov, Introductory lectures on convex optimization: a basic course, Applied optimization, Kluwer Academic Publ., Boston, Dordrecht, London, 2004.

[52] R. Fletcher, M. J. D. Powell, A rapidly convergent descent method for minimization, Comput. J. 6 (1963) 163–168.

[53] Ph. L. Toint, Some numerical results using a sparse matrix updating formula in unconstrained optimization, Math. Comput. 32 (1978) 839–852.