New Adaptive Stepsize Selections in Gradient Methods

G. Frassoldati* L. Zanni* G. Zanghirati**

*Department of Pure and Applied Mathematics
University of Modena and Reggio Emilia
via Campi 213/b
I-41100 Modena, Italy
giacomofrassoldati@gmail.com zanni.luca@unimore.it

**Department of Mathematics, University of Ferrara
Scientific-Technological Campus, Building B
via Saragat, 1
I-44100 Ferrara, Italy
g.zanghirati@unife.it

Technical Report n. 77, University of Modena and Reggio Emilia, Italy
January 2007

Abstract

This paper deals with gradient methods for minimizing n-dimensional strictly convex quadratic functions. Two new adaptive stepsize selection rules are presented and some key properties are proved. Practical insights on the effectiveness of the proposed techniques are given by a numerical comparison with the Barzilai-Borwein (BB) method, the cyclic/adaptive BB methods and two recent monotone gradient methods.

Keywords: unconstrained optimization, strictly convex quadratics, gradient methods, adaptive stepsize selections.

1 Introduction

We consider some recent gradient methods to minimize the quadratic function

$$\min f(x) = \frac{1}{2}x^T Ax - b^T x \tag{1}$$

where A is a real symmetric positive definite (SPD) $n \times n$ matrix and $b, x \in \mathbb{R}^n$. Given a starting point x_0 and using the notation $g_k = g(x_k) = \nabla f(x_k)$, the gradient methods for (1) are defined by the iteration

$$x_{k+1} = x_k - \alpha_k g_k, \qquad k = 0, 1, \ldots, \tag{2}$$

where the stepsize $\alpha_k > 0$ is determined through an appropriate selection rule. Classical examples of stepsize selections are the line searches used by the Steepest Descent (SD) [4] and the Minimal Gradient (MG) [11, 19] methods, which minimize $f(x_k - \alpha g_k)$ and $\|g(x_k - \alpha g_k)\|$, respectively:

$$\alpha_k^{\mathrm{SD}} = \operatorname*{argmin}_{\alpha \in \mathbb{R}} f(x_k - \alpha g_k) = \frac{g_k^T g_k}{g_k^T A g_k},$$

$$\alpha_k^{\mathrm{MG}} = \operatorname*{argmin}_{\alpha \in \mathbb{R}} \|g(x_k - \alpha g_k)\| = \frac{g_k^T A g_k}{g_k^T A^2 g_k}.$$

Many other rules for the stepsize selection have been proposed to accelerate the slow convergence exhibited in most cases by SD and MG (refer to [1] for an explanation of the zigzagging phenomenon associated to the SD method). The literature shows that very promising performance can be obtained by using selection rules derived by the ingenious stepsizes proposed by Barzilai and Borwein [2]:

$$\alpha_k^{\mathrm{BB1}} = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}} = \frac{g_{k-1}^T g_{k-1}}{g_{k-1}^T A g_{k-1}} = \alpha_{k-1}^{\mathrm{SD}}\,, \tag{3}$$

$$\alpha_k^{\mathrm{BB2}} = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}} = \frac{g_{k-1}^T A g_{k-1}}{g_{k-1}^T A^2 g_{k-1}} = \alpha_{k-1}^{\mathrm{MG}}\,, \tag{4}$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$. Starting from (3) and (4), special stepsize selections have been developed, that allow the corresponding gradient methods to largely improve the SD method. In some cases, they can even get competitive with the conjugate gradient method, which is the method of choice for problem (1). Furthermore, successful extensions of these BB-like gradient methods to non-quadratic functions [10, 15] and to constrained optimization problems [3, 7, 8, 9, 17] have been proposed. Hence, the study of new effective stepsizes becomes an interesting research topic for a wide range of mathematical programming problems.

Here we discuss some of the most recent stepsize selections. The first class of selection rules we consider exploits the cyclic use of the same stepsize in some consecutive iterations. This idea was first proposed in [14] for the so called Gradient Method with Retards (GMR): given a positive integer m and a set of real numbers $q_j \geq 1,\ j = 1, \ldots, m$, define

$$\alpha_k^{\mathrm{GMR}} = \frac{g_{\nu(k)}^T A^{\mu(k)-1} g_{\nu(k)}}{g_{\nu(k)}^T A^{\mu(k)} g_{\nu(k)}}, \tag{5}$$

where

$$\nu(k) \in \big\{k, k-1, \ldots, \max\{0, k-m\}\big\}, \qquad \mu(k) \in \{q_1, q_2, \ldots, q_m\}\,.$$

Special implementations of (5) that exploit the cyclic use of the SD step [5, 6, 16] or the BB1 step [5, 10] have been investigated, showing a meaningful convergence acceleration on ill-conditioned problems. These cyclic methods are further improved by introducing an adaptive choice of the cycle length m (also known as "memory"), as it is the case for the Adaptive Cyclic Barzilai-Borwein (ACBB) method [10]:

$$\begin{cases} (\alpha_k = \alpha_k^{\mathrm{BB1}}, \quad j = 1) & \text{if } k = 1 \text{ or } j = 10 \text{ or } \beta_k \geq 0.95\,, \\ (\alpha_k = \alpha_{k-1}, \quad j = j+1) & \text{otherwise,} \end{cases}$$

where

$$\beta_k = \frac{g_k^T A g_k}{\|g_k\| \|A g_k\|} = \sqrt{\frac{\alpha_k^{\mathrm{MG}}}{\alpha_k^{\mathrm{SD}}}} = \cos(g_k, A g_k)\,.$$

Another effective strategy included in some recent techniques consists in alternating different stepsize rules: in this case too, an adaptively controlled switching criterion improves the convergence performances. Promising approaches based on the rules alternation are the Adaptive Barzilai-Borwein (ABB) method [19], whose stepsize selection is

$$\begin{cases} \alpha_k = \alpha_k^{\mathrm{BB2}} & \text{if } \alpha_k^{\mathrm{BB2}} / \alpha_k^{\mathrm{BB1}} < \tau\,, \\ \alpha_k = \alpha_k^{\mathrm{BB1}} & \text{otherwise,} \end{cases} \tag{6}$$

(note that $\alpha_k^{\mathrm{BB2}}/\alpha_k^{\mathrm{BB1}} = \cos^2(g_{k-1}, Ag_{k-1})$), and the Adaptive Steepest Descent (ASD) method [19], which updates the stepsize according to

$$\begin{cases} \alpha_k = \alpha_k^{\mathrm{MG}} & \text{if } \alpha_k^{\mathrm{MG}}/\alpha_k^{\mathrm{SD}} > \tau\,, \\ \alpha_k = \alpha_k^{\mathrm{SD}} - 0.5\alpha_k^{\mathrm{MG}} & \text{otherwise,} \end{cases} \tag{7}$$

where τ is a prefixed threshold. Numerical experiments suggest to set $\tau \in [0.1, 0.2]$ in the ABB method and τ slightly larger than 0.5 in the ASD method. Throughout this paper we let $\tau = 0.15$ and $\tau = 0.55$, respectively. The computational study reported in [19] shows that ABB and ASD methods generally outperform BB1 method (we recall that ASD is a monotone scheme) and also that they behave similarly, even if the ABB scheme seems to be preferable on ill-conditioned problems and when high accuracy is required.

Finally, competitive results with respect to the BB1 method are also obtained with stepsize selections derived by a new rule proposed by Yuan [18]:

$$\alpha_k^{\mathrm{Y}} = \frac{2}{\sqrt{(1/\alpha_{k-1}^{\mathrm{SD}} - 1/\alpha_k^{\mathrm{SD}})^2 + 4\|g_k\|^2/\|s_{k-1}\|^2} + (1/\alpha_{k-1}^{\mathrm{SD}} + 1/\alpha_k^{\mathrm{SD}})}\,. \tag{8}$$

The derivation of this stepsize is based on an analysis of (1) in the two-dimensional case: here, if a Yuan's step is taken after exactly one SD step, then only one more SD step is needed to get to the minimizer. In [12], a variant of (8) has been suggested:

$$\alpha_k^{\mathrm{YV}} = \frac{2}{\sqrt{(1/\alpha_{k-1}^{\mathrm{SD}} - 1/\alpha_k^{\mathrm{SD}})^2 + 4\|g_k\|^2/(\alpha_{k-1}^{\mathrm{SD}}\|g_{k-1}\|)^2} + (1/\alpha_{k-1}^{\mathrm{SD}} + 1/\alpha_k^{\mathrm{SD}})}\,,$$

which coincides with (8) if x_k is obtained by taking an SD step. Starting from the new formula, Dai and Yuan [12] suggested a gradient method whose stepsize is given by

$$\alpha_k^{\mathrm{DY}} = \begin{cases} \alpha_k^{\mathrm{SD}} & \text{if} \quad \mathrm{mod}\,(k,4) < 2\,, \\ \alpha_k^{\mathrm{YV}} & \text{otherwise.} \end{cases}$$

The numerical experiments in [12] show that this last monotone method performs better than BB1 on problem (1): thus, it is interesting to evaluate its behaviour together with the above BB1 improvements.

From a theoretical point of view, convergence results may be given for the considered gradient methods. For instance, since the BB1, ACBB and ABB methods belong to the GMR class, their R-linear convergence can be obtained by proceeding as in [5]; for the ASD method, the Q-linear convergence of $\{x_k\}$ is established in [19] and, in a very similar way, the same result may be derived for the DY method. However, these results don't explain the great improvement of BB1 over SD and the further improvements of the most recent gradient methods.

In this work, to better understand the behaviour of the considered methods, we focus on the stepsizes sequences they generate. The analysis of these sequences emphasizes key differences in the stepsize distributions and it leads us to introduce two improved selection rules.

The paper is organized as follows. In Section 2 we consider the behaviour of BB1, ACBB, ABB, ASD and DY schemes on a small test problem, to illustrate and discuss the different stepsize distributions. In Section 3 we propose two new stepsize selections and we prove some useful properties to explain their behaviour. Numerical evidence of the improvements due to the new selection rules are given in Section 4 on medium-to-large test problems. Finally, in Section 5 we discuss some conclusions and future developments.

2 Comparing recent gradient methods

To analyse the convergence of any gradient method for a quadratic function, we can assume without loss of generality that A is diagonal with distinct eigenvalues [13]:

$$A = diag(\lambda_1, \lambda_2, \ldots, \lambda_n), \qquad 0 < \lambda_1 < \lambda_2 < \ldots < \lambda_n \,. \tag{9}$$

It follows from (2) and the definition of g_k that

$$g_{k+1}^{(i)} = (1 - \alpha_k \lambda_i) g_k^{(i)}, \qquad i = 1, 2, \ldots, n. \tag{10}$$

Thus, we can also assume that $g_1^{(i)} \neq 0$ for all $i = 1, 2, \ldots, n$, since if there is a component of the gradient such that $g_1^{(i)} = 0$, then $g_k^{(i)} = 0$ for all k, hence this component could be disregarded.

To investigate the differences between the gradient methods described in the previous section, we have to inspect the stepsize distributions. To this end, let's consider a simple test problem obtained by modifying the one given in [12]:

$$A = diag(\lambda_1, \lambda_2, \ldots, \lambda_{10}) \tag{11}$$

where

$$\lambda_i = 111i - 110, \qquad i = 1, \ldots, 10. \tag{12}$$

We test BB1, ACBB, ABB, ASD and DY on this problem by setting the starting point x_0 such that $g_0^{(i)} = \sqrt{1+i}$, the stopping condition as $\|g_k\| \leq 10^{-8}$ and $\alpha_0 = \alpha_0^{\mathrm{SD}}$ when a starting stepsize is needed[1].

The results are summarized in Tables 1 and 2, where the data in the last two columns will be discussed later. In the second row of Table 1 we put the number of iterations required by each algorithm. Then, for each method, we classify the sequence $\{\alpha_k\}$ in 10 subsets depending on which eigenvalue α_k^{-1} is nearest to.

Furthermore, we focus our attention on the role of the longest steps (that is to say $\alpha_k \geq \frac{2}{\lambda_1+\lambda_2}$) and we study the effect they have on each component of the gradient. Table 2 reports for each method the numbers $\log_{10} |\rho^{(i)}|$ at the end of the run, where

$$\rho^{(i)} = (1 - \alpha_{j_1}\lambda_i)(1 - \alpha_{j_2}\lambda_i) \ldots (1 - \alpha_{j_h}\lambda_i), \quad i = 1, 2, \ldots, 10, \quad h \leq k$$

and $j_1, j_2, \ldots, j_h$ are those indices such that $\alpha_{j_\ell} \geq \frac{2}{\lambda_1+\lambda_2}$, $\ell = 1, \ldots, h$. The value $|\rho^{(i)}|$, which is clearly independent on the order of the indices j_ℓ, quantifies how the i-th component of the gradient is reduced or amplified due to the h longest stepsizes.

We may observe that the stepsizes larger than $\frac{2}{\lambda_1+\lambda_2}$ are fewer in ACBB and ABB than in the other methods; nevertheless, they induce a larger reduction of the first gradient component. Moreover, these improvements are obtained with a less remarkable increase in the other components with respect to BB1, ASD and DY. This suggests that ACBB and ABB distribute the longest stepsizes near to λ_1^{-1} in a more fruitful way than the other schemes. Since this behaviour can be observed in many other test problems where ACBB and ABB outperform BB1, ASD and DY, it is worthwhile to investigate if more efficient schemes could be derived by improving the ability to approximate λ_1^{-1}. Roughly speaking, we are looking for stepsize rules that exploit the benefit arising from a fast reduction of the first gradient component. It is well known (see for example the discussion in [13]) that in BB-like stepsize selections, after a meaningful reduction of the first gradient components (that is after the largest stepsizes), an iteration occurs where g_{k-1} is likely to be dominated by large components: then a small α_k will be generated, which in turn will force both a large decrease in the large gradient components and a remarkable reduction of the objective function.

	BB1	ACBB	ABB	ASD	DY	ABB_{min1}	ABB_{min2}
iterations	363	108	132	360	199	61	44
$\frac{2}{\lambda_1+\lambda_2} \le \alpha_k$	54	10	16	46	29	3	2
$\frac{2}{\lambda_2+\lambda_3} \le \alpha_k < \frac{2}{\lambda_1+\lambda_2}$	43	8	11	39	22	7	4
$\frac{2}{\lambda_3+\lambda_4} \le \alpha_k < \frac{2}{\lambda_2+\lambda_3}$	33	11	19	45	16	6	6
$\frac{2}{\lambda_4+\lambda_5} \le \alpha_k < \frac{2}{\lambda_3+\lambda_4}$	24	13	10	26	14	5	3
$\frac{2}{\lambda_5+\lambda_6} \le \alpha_k < \frac{2}{\lambda_4+\lambda_5}$	25	4	8	26	16	4	4
$\frac{2}{\lambda_6+\lambda_7} \le \alpha_k < \frac{2}{\lambda_5+\lambda_6}$	25	5	6	23	13	2	2
$\frac{2}{\lambda_7+\lambda_8} \le \alpha_k < \frac{2}{\lambda_6+\lambda_7}$	28	11	12	29	18	4	5
$\frac{2}{\lambda_8+\lambda_9} \le \alpha_k < \frac{2}{\lambda_7+\lambda_8}$	33	9	16	25	19	11	3
$\frac{2}{\lambda_9+\lambda_{10}} \le \alpha_k < \frac{2}{\lambda_8+\lambda_9}$	39	24	11	54	25	2	8
$\alpha_k < \frac{2}{\lambda_9+\lambda_{10}}$	59	13	23	47	27	17	7

Table 1: Total number of iterations and stepsize distribution with respect to the eigenvalues.

	BB1 $h=54$	ACBB $h=10$	ABB $h=16$	ASD $h=46$	DY $h=29$	ABB_{min1} $h=3$	ABB_{min2} $h=2$
$\log_{10}\lvert\rho^{(1)}\rvert$	−8.2	−10.9	−10.5	−7.7	−8.0	−11.0	−8.6
$\log_{10}\lvert\rho^{(2)}\rvert$	53.5	15.5	16.1	52.2	24.6	6.1	4.1
$\log_{10}\lvert\rho^{(3)}\rvert$	71.8	18.6	21.7	67.6	35.1	7.0	4.7
$\log_{10}\lvert\rho^{(4)}\rvert$	81.9	20.4	24.8	76.2	40.7	7.6	5.0
$\log_{10}\lvert\rho^{(5)}\rvert$	89.0	21.7	26.9	82.2	44.6	7.9	5.3
$\log_{10}\lvert\rho^{(6)}\rvert$	94.4	22.7	28.5	86.8	47.5	8.2	5.5
$\log_{10}\lvert\rho^{(7)}\rvert$	98.8	23.5	29.8	90.5	49.9	8.5	5.6
$\log_{10}\lvert\rho^{(8)}\rvert$	102.5	24.1	30.9	93.6	51.9	8.7	5.8
$\log_{10}\lvert\rho^{(9)}\rvert$	105.6	24.7	31.9	96.3	53.6	8.8	5.9
$\log_{10}\lvert\rho^{(10)}\rvert$	108.5	25.2	32.7	98.7	55.1	9.0	6.0

Table 2: Effects of the longest steps on the components of the gradient.

The importance of reducing $|g_k^{(1)}|$ may be easily illustrated also on the test problem (11)–(12): we solve this problem by the BB1 method with the starting point and the stopping rule previously described, but with two different values for α_0, that are

$$\alpha_0 = \alpha_0^{\text{SD}} \qquad \text{or} \qquad \alpha_0 = (\lambda_1 + 10^{-9})^{-1}.$$

The values of $f(x_k)$ obtained in these two experiments are plotted in Figure 1. For $\alpha_0 = (\lambda_1 + 10^{-9})^{-1}$ we have from (10)

$$|g_1^{(1)}| \approx 10^{-9}|g_0^{(1)}| \tag{13}$$

so the first component of the gradient becomes negligible until all the other components will be significantly reduced. In this phase, in a sense the problem turns into a simpler one and a more effective behaviour of the method may be expected. In fact, after an increasing function value in the first iteration, a very fast convergence is observed (the method requires only 45 iterations) and no other stepsizes near λ_1^{-1} are selected. In the next section, we will introduce

[1] All the experiments presented in the paper are performed with Matlab 6.0 on a 2.0GHz AMD Sempron 3000+ with 512MB of RAM.

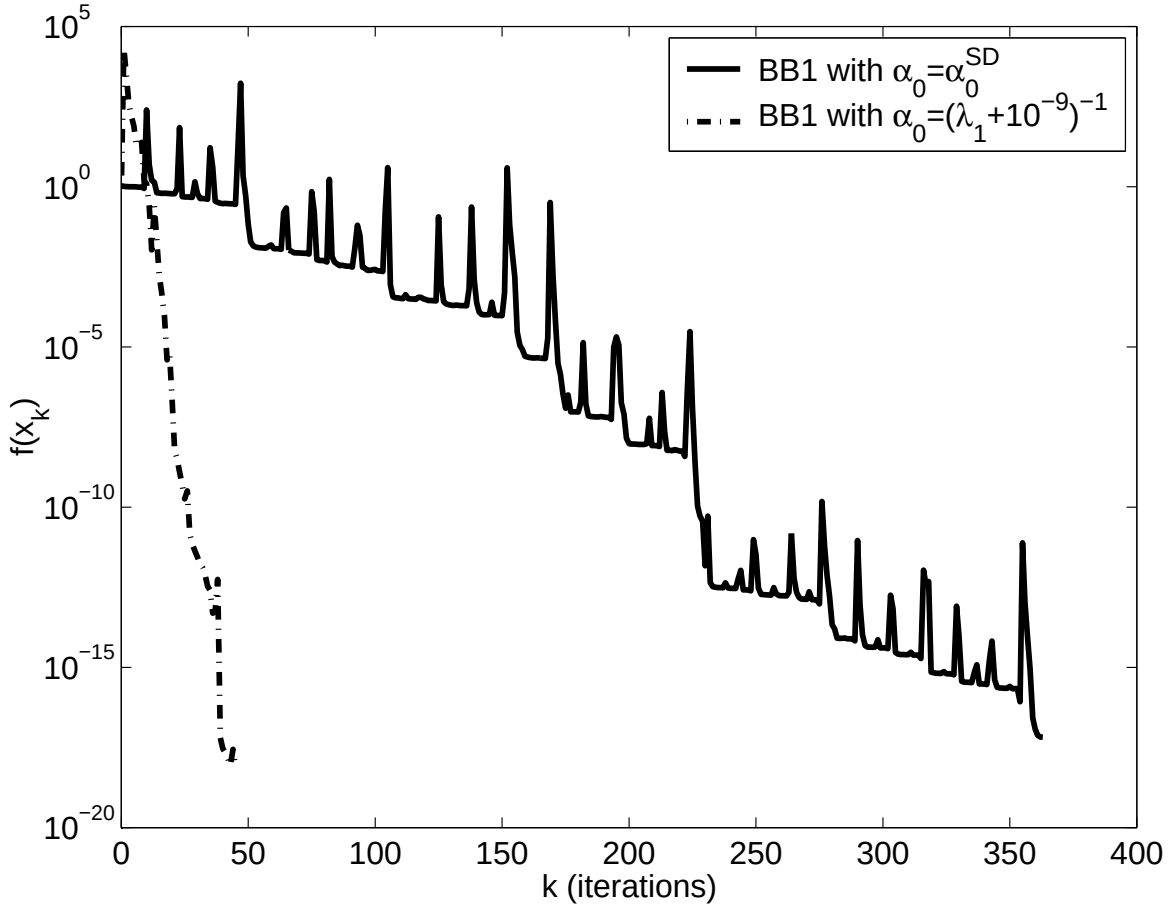

Figure 1: Behaviour of the BB1 method started with different initial stepsizes.

two stepsize selections that are appropriately designed to better capture the inverse of the smallest eigenvalues than the above rules.

3 Derivation of the new methods

The recent literature shows that ACBB and ABB can be considered very effective approaches that often outperform other BB-like gradient methods. In our experience the two schemes behave rather similarly, but in general ABB performs better when large and ill-conditioned quadratic problems are faced. Thus, we develop new stepsize selections starting from the ABB rule. Following the considerations in the previous section, we look for ABB-like algorithms able to exploit BB1 steps close to λ_1^{-1}. We point this goal by forcing stepsizes that reduce the components $|g_k^{(i)}|$ for large i, in such a way that a following BB1 step will likely depend on a gradient dominated by small components. Our first implementation of this idea, denoted by ABB_{min1}, consists in substituting the BB2 step in (6) with the following shorter step:

$$\begin{cases} \alpha_k = \min\{\alpha_j^{\text{BB2}} \mid j = \max\{1, k-m\}, \ldots, k\} & \text{if } \alpha_k^{\text{BB2}}/\alpha_k^{\text{BB1}} < \tau\,, \\ \alpha_k = \alpha_k^{\text{BB1}} & \text{otherwise.} \end{cases} \tag{14}$$

This method can be regarded as a particular member of the GMR class, so it is R-linearly convergent [5]. Furthermore, it could allow the same step to be reused in some consecutive iterations, as it is in the ACBB method.

To state our second variant of the ABB scheme we consider $g_{k+1} = g_{k+1}(\alpha_k)$, so that $\alpha_{k+1}^{\mathrm{SD}}$ is a function of α_k as well. We then introduce the stepsize

$$\alpha_k^{\mathrm{new}} = \operatorname*{argmax}_{\alpha_k \in \mathbb{R}} (\alpha_{k+1}^{\mathrm{SD}}) \tag{15}$$

and discuss its main properties. The following result is necessary.

Lemma 1. *Let A be a SPD matrix and let $g_k \neq 0$ be such that $\cos^2(g_k, Ag_k) < 1$. Let*

$$c_j = g_k^T A^j g_k > 0 \qquad j = 0, 1, 2, 3, \tag{16}$$

and

$$R = c_1 c_3 - c_2^2\,, \qquad S = c_0 c_3 - c_1 c_2\,, \qquad T = c_0 c_2 - c_1^2\,. \tag{17}$$

Then

$$R, S, T > 0\,, \tag{18}$$
$$S = (c_0 R + c_2 T)/c_1\,, \tag{19}$$
$$c_2 S = c_1 R + c_3 T\,, \tag{20}$$
$$S^2 - 4RT > 0\,.$$

Proof. By the definition (16) and by applying the Cauchy-Schwartz inequality to $c_1 = g_k^T(Ag_k)$ it follows that $c_0/c_1 > c_1/c_2$, hence $T > 0$. Now, observe that $c_j = y^T A^{(j-1)} y$, $j = 1, 2, 3$, where $y = A^{1/2} g_k$: then $c_1/c_2 > c_2/c_3$ follows in the same way, so $R > 0$. By using these inequalities in sequence, we also have $S = c_0 c_3 - c_1 c_2 > 0$, which complete (18). Then (19) and (20) easily follows by substitution. Finally,

$$\begin{aligned} S^2 - 4RT &= \frac{c_0^2 R^2 + c_2^2 T^2 + 2RT c_0 c_2 - 4RT c_1^2}{c_1^2} \\ &= \frac{c_0^2 R^2 + c_2^2 T^2 + 2RT(T - c_1^2)}{c_1^2} \\ &> \frac{c_0^2 R^2 + c_2^2 T^2 + 2RT(T - c_0 c_2)}{c_1^2} \\ &= \frac{(c_0 R - c_2 T)^2 + 2RT^2}{c_1^2} > 0\,. \end{aligned}$$

□

We may now give the explicit form of α_k^{new}. We consider

$$\begin{aligned} \alpha_{k+1}^{\mathrm{SD}} = F(\alpha_k) &= \frac{g_{k+1}^T g_{k+1}}{g_{k+1}^T A g_{k+1}} = \frac{g_k^T (I - \alpha_k A)(I - \alpha_k A) g_k}{g_k^T (I - \alpha_k A) A (I - \alpha_k A) g_k} \\ &= \frac{g_k^T g_k - 2\alpha_k g_k^T A g_k + \alpha_k^2 g_k^T A^2 g_k}{g_k^T A g_k - 2\alpha_k g_k^T A^2 g_k + \alpha_k^2 g_k^T A^3 g_k} \\ &= \frac{c_0 - 2\alpha_k c_1 + \alpha_k^2 c_2}{c_1 - 2\alpha_k c_2 + \alpha_k^2 c_3} \end{aligned}$$

and look at $F'(\alpha_k)$. It is easy to see that the roots of $F'(\alpha_k) = 0$ must satisfy

$$(c_1 - 2\alpha_k c_2 + \alpha_k^2 c_3)(-c_1 + \alpha_k c_2) - (c_0 - 2\alpha_k c_1 + \alpha_k^2 c_2)(-c_2 + \alpha_k c_3) = 0\,, \tag{21}$$

that is

$$R\alpha_k^2 - S\alpha_k + T = 0\,,$$

where R, S and T are defined in (17). From $R > 0$ and $S^2 - 4RT > 0$ we have

$$\alpha_k^{\text{new}} = \alpha_{k,1} = \frac{S - \sqrt{S^2 - 4RT}}{2R} \tag{22}$$

and

$$\alpha_{k,2} = \operatorname*{argmin}_{\alpha_k \in \mathbb{R}} \big(\alpha_{k+1}^{\text{SD}}\big) = \frac{S + \sqrt{S^2 - 4RT}}{2R}\,.$$

We report in the following theorem some interesting properties of α_k^{new}.

Theorem 1. *Let A be a SPD matrix and let $g_k \neq 0$ be such that $\cos^2(g_k, Ag_k) < 1$. The stepsize α_k^{new} satisfies the following properties:*

$$\alpha_k^{new} = \min_{\alpha_k \in \mathbb{R}} F(\alpha_k) = F(\alpha_{k,2})\,, \tag{23}$$

$$\frac{1}{\lambda_n} \leq \alpha_k^{new} \leq \frac{1}{\lambda_2}\,, \tag{24}$$

$$\textit{if} \quad n = 2 \quad \textit{then} \quad \alpha_k^{new} = \frac{1}{\lambda_2}\,, \tag{25}$$

$$\alpha_k^{new} < \frac{c_2}{c_3} < \alpha_k^{MG}\,. \tag{26}$$

Proof. From (21) we can write $F(\alpha_{k,2})$ as follows:

$$F(\alpha_{k,2}) = \frac{c_0 - 2\alpha_{k,2}c_1 + \alpha_{k,2}^2 c_2}{c_1 - 2\alpha_{k,2}c_2 + \alpha_{k,2}^2 c_3} = \frac{c_2\alpha_{k,2} - c_1}{c_3\alpha_{k,2} - c_2}\,.$$

By observing that $\alpha_k^{\text{new}}\alpha_{k,2} = T/R$, we obtain

$$F(\alpha_{k,2}) = \alpha_k^{\text{new}} \frac{c_2\alpha_{k,2} - c_1}{c_3\alpha_{k,2}\alpha_k^{\text{new}} - c_2\alpha_k^{\text{new}}} = \alpha_k^{\text{new}} \frac{c_2\alpha_{k,2} - c_1}{c_3\frac{T}{R} - c_2\alpha_k^{\text{new}}}\,.$$

To prove (23), we show that

$$c_2\alpha_{k,2} - c_1 = c_3\frac{T}{R} - c_2\alpha_k^{\text{new}}\,.$$

In fact, by substituting α_k^{new} and $\alpha_{k,2}$ and by using (20) we have

$$\begin{aligned} c_2\alpha_{k,2} - c_1 &= c_2\frac{S + \sqrt{S^2 - 4RT}}{2R} - c_1 \\ &= \frac{c_2 S + c_2\sqrt{S^2 - 4RT} - 2Rc_1}{2R} \\ &= \frac{c_3 T - c_1 R + c_2\sqrt{S^2 - 4RT}}{2R} \end{aligned}$$

and

$$\begin{aligned} c_3\frac{T}{R} - c_2\alpha_k^{\text{new}} &= c_3\frac{T}{R} - c_2\frac{S - \sqrt{S^2 - 4RT}}{2R} \\ &= \frac{2c_3 T - c_2 S + c_2\sqrt{S^2 - 4RT}}{2R} \\ &= \frac{c_3 T - c_1 R + c_2\sqrt{S^2 - 4RT}}{2R}\,. \end{aligned}$$

The left inequality in (24) follows from the Rayleigh's quotient property

$$\alpha_k^{\text{new}} = F(\alpha_{k,2}) \geq \frac{1}{\lambda_n}\,.$$

The right part in (24) follows from

$$\alpha_k^{\text{new}} = \min_{\alpha_k \in \mathbb{R}} F(\alpha_k) \leq F\left(\frac{1}{\lambda_1}\right) = \frac{g_k^T(I-\lambda_1^{-1}A)(I-\lambda_1^{-1}A)g_k}{g_k^T(I-\lambda_1^{-1}A)A(I-\lambda_1^{-1}A)g_k} \leq \frac{1}{\lambda_2}\,,$$

where the last inequality holds true because the vector $(I-\lambda_1^{-1}A)g_k$ is orthogonal to the eigenvector corresponding to λ_1. When $n=2$ the last result obviously yields (25).

Now, to show (26) we observe that

$$\frac{c_2}{c_3} - \alpha_k^{\text{new}} = \frac{c_2}{c_3} - \frac{S-\sqrt{S^2-4RT}}{2R} = \frac{2Rc_2 - Sc_3 + c_3\sqrt{S^2-4RT}}{2c_3R}\,.$$

If $(2Rc_2 - Sc_3) \geq 0$, then $(c_2/c_3 - \alpha_k^{\text{new}}) > 0$; otherwise, we have

$$c_3\sqrt{S^2-4RT} - (2Rc_2 - Sc_3) > 0$$

and

$$\begin{aligned}
\frac{c_2}{c_3} - \alpha_k^{\text{new}} &= \left(\frac{2Rc_2 - Sc_3 + c_3\sqrt{S^2-4RT}}{2c_3R}\right)\left(\frac{c_3\sqrt{S^2-4RT} - (2Rc_2 - Sc_3)}{c_3\sqrt{S^2-4RT} - (2Rc_2 - Sc_3)}\right) \\
&= \frac{c_3^2(S^2-4RT) - (4R^2c_2^2 + S^2c_3^2 - 4RSc_2c_3)}{2c_3R\big(c_3\sqrt{S^2-4RT} - (2Rc_2 - Sc_3)\big)} \\
&= \frac{-4c_3^2RT - 4R^2c_2^2 + 4RSc_2c_3}{2c_3R\big(c_3\sqrt{S^2-4RT} - (2Rc_2 - Sc_3)\big)}\,.
\end{aligned}$$

It follows from (18) and (20) that

$$\begin{aligned}
-4c_3^2RT - 4R^2c_2^2 + 4RSc_2c_3 &= -4c_3^2RT - 4R^2c_2^2 + 4R(c_1R + c_3T)c_3 \\
&= -4R^2c_2^2 + 4R^2c_1c_3 \\
&= 4R^2(c_1c_3 - c_2^2) = 4R^3 > 0\,,
\end{aligned}$$

hence $(c_2/c_3 - \alpha_k^{\text{new}}) > 0$ holds true, which gives the first part of (26). Finally, the right part of (26) follows from $\alpha_k^{\text{MG}} = c_1/c_2$ and the positivity of R. □

Remark 1. The properties (23) and (26) well emphasize the ability of the new selection rule to produce short stepsizes, so that it should be useful within adaptive alternation schemes similar to (14). The inequalities (24) explain why a sequence of stepsizes computed by (22) can allow meaningful reductions of the components $g_k^{(i)}$ with $i = 2, \ldots, n$, without forcing a too much remarkable reduction in $g_k^{(1)}$; thus, after this sequence of stepsizes, we most likely end up with a gradient vector where the first component dominates. Finally, in the special case $n = 2$, from (25) we have that the gradient method where $\alpha_0 = \alpha_0^{\text{new}}$ and $\alpha_1 = \alpha_1^{\text{SD}}$ will find the solution after two iterations.

The stepsize (22) considered with one iteration of retard satisfies

$$\alpha_{k-1}^{\text{new}} < \alpha_{k-1}^{\text{MG}} = \alpha_k^{\text{BB2}}$$

and then it allows a shorter step than BB2. Thus, also $\alpha_{k-1}^{\rm new}$ can be exploited within an ABB-like scheme to achieve a better reduction of the components $|g_k^{(i)}|$ for large i. The corresponding selection rule, denoted by ABB_{min2}, is the following:

$$\begin{cases} \alpha_k = \alpha_{k-1}^{\rm new} & \text{if } \alpha_k^{\rm BB2}/\alpha_k^{\rm BB1} < \tau\,, \\ \alpha_k = \alpha_k^{\rm BB1} & \text{otherwise.} \end{cases} \tag{27}$$

The computational cost per iteration is essentially the same as the other methods, because no additional matrix-vector multiplications are needed. In fact, if we keep into memory $w = Ag_{k-1}$ and compute, at each iteration, the vector $z = Ag_k$, then $\alpha_{k-1}^{\rm new}$ can be obtained by

$$c_0 = g_{k-1}^T g_{k-1}\,, \quad c_1 = g_{k-1}^T w\,, \quad c_2 = w^T w\,, \quad c_3 = \frac{g_k^T z - c_1 + 2\alpha_{k-1} c_2}{\alpha_{k-1}^2}\,,$$

via (17) and (22). Of course, different ways to obtain c_3 without additional matrix-vector products are also available.

Concerning the convergence properties of ABB_{min2}, taking into account that we have $\alpha_k \le \alpha_k^{\rm BB1}$ for all k, the R-linear convergence can be proved by proceeding as in [5].

The behaviour of ABB_{min1} and ABB_{min2} on the test problem (11)–(12) is described in Tables 1 and 2. The starting point and α_0 are the same as in the other methods. The parameters setting is the following: $m = 9$ and $\tau = 0.8$ in ABB_{min1}, $\tau = 0.9$ in ABB_{min2}. In our experience this setting gives satisfactory results in many situations and it will be exploited also in the numerical experiments of the next section. From Table 1 we observe that both the new methods generate much less stepsizes larger than $\frac{2}{\lambda_1+\lambda_2}$. These few stepsizes seem able to reduce the first gradient component to such an extent, that the other methods can only get to after many large stepsizes (see Table 2). The effect of this behaviour on the convergence rate can be observed by looking at the iteration counts reported in Table 1.

The next section gives more insights into the effectiveness of the new methods, by showing an additional numerical experience.

4 Numerical experiments

In this section we present the results of a numerical investigation on different kinds of test problems. A group of randomly generated test problems is analyzed first, then another group of tests is considered, which comes from a PDE-like prototype problem.

Table 3 reports the results on the test problem (9) with four different Euclidean condition numbers $\kappa_2 = \kappa_2(A)$, ranging from 10^2 to 10^5, and with the three different sizes $n = 10^2$, 10^3, 10^4. We let $\lambda_1 = 1$ and $\lambda_n = \kappa_2$. Two subsets of experiments are carried out, depending on the spectral distribution:

$\mathcal{S}_1$) λ_i, $i = 2, \ldots, n-1$, is randomly sampled from the uniform probability distribution in $(1, \kappa_2)$;

$\mathcal{S}_2$) $\lambda_i = 10^{p_i}$, $i = 2, \ldots, n-1$, where p_i is randomly sampled from the uniform probability distribution in $(0, \log_{10}(\kappa_2))$.

The entries of the starting points x_0 are randomly sampled in the interval $(-5, 5)$, the stopping condition is $\|g_k\| \le 10^{-8}$ and, for all the methods, the parameter setting is as described in the previous sections. For a given value of n and κ_2, 10 problems are randomly generated and the number of iterations averaged over the 10 runs of each algorithm is listed in Table 3 (these numbers are meaningful, given the similar computational cost per iteration

n	κ_2	BB1	ACBB	ABB	ASD	DY	ABB_{min1}	ABB_{min2}
Spectral distribution $\mathcal{S}_1$								
10^2	10^2	142.4	135.4	123.0	152.3	133.5	118.9	**112.2**
	10^3	530.8	379.4	288.0	451.5	376.3	247.4	**215.3**
	10^4	1518.3	873.4	481.7	1197.0	1151.5	397.7	**303.3**
	10^5	5182.6	1860.3	1087.9	3765.8	4379.6	525.9	**342.6**
10^3	10^2	147.7	149.1	138.0	162.5	147.6	141.9	**133.6**
	10^3	514.1	444.4	422.1	475.3	442.7	403.8	**390.2**
	10^4	1583.3	1293.4	955.5	1476.2	1422.1	818.3	**721.2**
	10^5	5179.7	3391.7	1467.0	4765.0	5094.7	1215.7	**956.0**
10^4	10^2	154.9	154.9	144.5	166.1	149.9	147.1	**140.9**
	10^3	529.1	476.4	451.6	490.3	464.8	441.0	**440.9**
	10^4	1918.6	1567.2	1212.0	1600.3	1484.2	1216.1	**1154.3**
	10^5	6142.3	4897.4	2532.9	4681.3	5866.1	2358.8	**2050.9**
Spectral distribution $\mathcal{S}_2$								
10^2	10^2	146.7	149.8	136.1	158.1	135.3	137.5	**129.5**
	10^3	508.1	470.0	441.2	484.5	453.8	423.9	**417.7**
	10^4	1735.3	1520.8	1389.7	1545.9	1493.6	**1350.4**	1376.5
	10^5	5734.1	5274.3	4458.0	5514.2	5816.6	**4175.3**	4402.7
10^3	10^2	156.1	152.6	147.8	173.3	152.0	145.1	**139.6**
	10^3	538.9	504.1	462.5	517.0	503.9	453.6	**448.3**
	10^4	1862.8	1752.6	1528.6	1797.5	1630.9	1467.6	**1454.7**
	10^5	7400.7	5349.4	4903.3	5834.4	6182.8	**4596.9**	4882.8
10^4	10^2	162.7	162.8	152.5	172.0	151.9	152.7	**146.4**
	10^3	545.8	541.6	475.5	535.7	505.6	476.8	**462.7**
	10^4	2004.0	1775.9	1568.8	1971.7	1763.5	**1500.3**	1514.3
	10^5	7577.0	5892.4	5056.2	5645.4	6726.6	**4784.8**	4980.0

Table 3: Iteration counts of randomly generated test problems.

of the considered methods). For each value of κ_2, the winner method is marked in bold: the new methods win in all cases. In particular, if the eigenvalues are uniformly distributed (distribution $\mathcal{S}_1$) the algorithm ABB_{min2} is clearly the better choice and can greatly improve the efficiency of the ABB method (BB1, ACBB, ASD and DY seem less effective than ABB). For instance, in the case $n = 10^2$ and $\kappa_2 = 10^5$ the new scheme requires on average 342.6 iterations only, that is less than one third of the averaged ABB iterations. In most of the other cases ABB_{min1} is the second choice.

Looking at the second test subset, a different pattern appears: the new methods still outperform the others, but the iteration counts are less dissimilar. Furthermore, the ABB_{min1} method is the winner scheme for large condition numbers. A possible explanation is that the eigenvalue density near λ_1 reduces the benefits of our strategy.

In the second group of experiments we evaluate the algorithms in solving a large scale real problem proposed in [13, problem Laplace1 (L1)]: it requires the solution of an elliptic system of linear equations, arising from a 3D Laplacian on the unitary cube, discretized using a standard 7-point finite difference stencil. Here N interior nodes are taken in each coordinate direction, so that the problem has $n = N^3$ variables. The solution is a Gaussian function centered at the point $(\alpha, \beta, \gamma)^T$ multiplied by a quadratics, which vanishes on the boundary. A parameter σ controls the decay rate of the Gaussian. We refer the reader to [13] for additional details on this problem. In our experiments we set the parameters in two

n	θ	BB1	ACBB	ABB	ASD	DY	ABB_{min1}	ABB_{min2}
Problem L1(a)								
60^3	10^{-3}	137	123	100	134	96	118	**95**
	10^{-6}	374	334	282	238	299	229	**191**
	10^{-9}	526	478	408	431	421	357	**347**
80^3	10^{-3}	161	173	171	170	**125**	167	172
	10^{-6}	471	328	384	322	323	**248**	271
	10^{-9}	610	681	557	558	560	425	**346**
100^3	10^{-3}	325	260	223	**147**	163	208	226
	10^{-6}	610	414	476	416	389	358	**338**
	10^{-9}	886	579	582	690	675	495	**423**
Problem L1(b)								
60^3	10^{-3}	50	49	51	62	**44**	62	53
	10^{-6}	337	257	262	264	235	227	**205**
	10^{-9}	649	394	370	452	419	370	**361**
80^3	10^{-3}	65	78	**58**	67	63	87	82
	10^{-6}	**274**	371	342	325	306	303	309
	10^{-9}	527	673	482	553	568	490	**444**
100^3	10^{-3}	101	122	82	**75**	77	86	85
	10^{-6}	499	381	393	402	**375**	384	380
	10^{-9}	875	789	**567**	764	790	631	591

Table 4: Iteration counts for the 3D Laplacian problem.

θ	BB1	ACBB	ABB	ASD	DY	ABB_{min1}	ABB_{min2}
10^{-3}	839	805	685	655	**568**	728	713
10^{-6}	2565	2085	2139	1967	1927	1749	**1694**
10^{-9}	4073	3594	2966	3448	3433	2768	**2512**

Table 5: Total number of iterations.

different ways:

$$\begin{aligned} &\text{(a)} \quad \sigma = 20, \quad \alpha = \beta = \gamma = 0.5\,; \\ &\text{(b)} \quad \sigma = 50, \quad \alpha = 0.4, \quad \beta = 0.7, \quad \gamma = 0.5\,. \end{aligned}$$

The null vector is the starting point and we stop the iterations when $\|g_k\| \leq \theta \|g_0\|$, with different values of θ. Table 4 reports the iteration counts.

We summarize the algorithms performances in Table 5, where, for each accuracy level, the total number of iterations accumulated by each method in all problems is reported.

The numbers show clearly that the new stepsize selections are preferable when high accuracy is required.

From both the groups of test problems one can observe how the proposed stepsize selections make the new methods perform often better than other recent successful gradient schemes. Furthermore, even if the ABB_{min2} method seems preferable, the performances of the ABB_{min1} method are very similar, but the latter uses a simpler adaptive selection involving BB1 and BB2 stepsizes only. This means the new ABB_{min1} method is well suited to be extended to general nonlinear optimization problems, as it is the standard ABB method.

5 Conclusions and developments

In this work we analyzed convergence properties of recent classes of gradient methods that have shown to be effective in minimizing strictly convex quadratic functions. To better understand the improvements exhibited by the adaptive-stepsize gradient methods over the standard Barzilai-Borwein approaches, the sequences of stepsizes generated by these schemes are studied with respect to the Hessian's eigenvalues. Based on this analysis, new adaptive stepsize selection rules are proposed, which are appropriately designed to better capture the inverse of the minimum eigenvalue. For one of the new stepsizes some theoretical properties and meaningful bounds are proved. Numerical results carried out on randomly generated test problems as well as on a classical large-scale problem show that the schemes based on the new proposals often outperform other modern gradient methods. Future works will concern with the possible application of the new rules to non-quadratic optimization and to constrained optimization. In particular, one of the proposed selection rule seems promising also for these settings, given that it simply exploits the two Barzilai-Borwein stepsizes, that are successfully used in many gradient methods for general optimization problems.

References

[1] H. Akaike, *On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method*, Ann. Inst. Statist. Math Tokyo, 11 (1959), pp. 1–17.

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

[3] E. G. Birgin, J. M. Martínez, M. Raydan, *Nonmonotone spectral projected gradient methods on convex sets*, SIAM Journal on Optimization, 10:4 (2000), pp. 1196–1211.

[4] A. Cauchy, *Méthode générale pour la résolution des systèmes d'equations simultanées*, Comp. Rend. Sci. Paris, 25 (1847), pp. 46–89.

[5] Y. H. Dai, *Alternate stepsize gradient method*, Optimization, 52 (2003), pp. 395–415.

[6] Y. H. Dai, R. Fletcher, *On the asymptotic behaviour of some new gradient methods*, Mathematical Programming (Series A), 103:3 (2005), pp. 541–559.

[7] Y. H. Dai, R. Fletcher, *Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming*, Numerische Mathematik, 100 (2005), pp. 51–47.

[8] Y. H. Dai, R. Fletcher, *New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds*, Mathematical Programming (Series A), 106:3 (2006), pp. 403–421.

[9] W. W. Hager, H. Zhang, *A new active set algorithm for box constrained optimization*, SIAM J. Optim., 17 (2006), pp. 526-557.

[10] Y. H. Dai, W. W. Hager, K. Schittkowski, H. Zhang, *The Cyclic Barzilai-Borwein Method for Unconstrained Optimization*, IMA J. Numer. Anal., 26 (2006), pp. 604–627.

[11] Y. H. Dai, Y. X. Yuan, *Alternate Minimization Gradient Method*, IMA J. Numer. Anal., 23 (2003), pp. 377–393.

[12] Y. H. Dai, Y. X. Yuan, *Analyses of Monotone Gradient Methods*, Journal of Industry and Management Optimization, 1:2 (2005), pp. 181–192.

[13] R. Fletcher, *On the Barzilai-Borwein Method*, in "Optimization and Control with Applications", Appl. Optim., vol. 96, Springer, New York (2005), pp. 235–256.

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

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

[16] M. Raydan, B. F. Svaiter, *Relaxed Steepest Descent and Cauchy-Barzilai-Borwein Method*, Computational Optimization and Applications, 21 (2002), pp. 155–167.

[17] T. Serafini, G. Zanghirati, L. Zanni, *Gradient projection methods for quadratic programs and applications in training support vector machines*, Optimization Methods and Software, 20 (2005), pp. 347–372.

[18] Y. X. Yuan, *A new stepsize for the steepest descent method*, Journal of Computational Mathematics, 24 (2006), pp. 149–156.

[19] B. Zhou, L. Gao, Y. H. Dai, *Gradient methods with adaptive step-sizes*, Computional Optimization and Applications, 35 (2006), pp. 69–86.