Noname manuscript No.
(will be inserted by the editor)

On the acceleration of the Barzilai-Borwein method

Yakui Huang · Yu-Hong Dai · Xin-Wei Liu · Hongchao Zhang

Received: date / Accepted: date

Abstract The Barzilai-Borwein (BB) gradient method is efficient for solving large-scale unconstrained problems to the modest accuracy and has a great advantage of being easily extended to solve a wide class of constrained optimization problems. In this paper, we propose a new stepsize to accelerate the BB method by requiring finite termination for minimizing two-dimensional strongly convex quadratic function. Combing with this new stepsize, we develop gradient methods which adaptively take the nonmonotone BB stepsizes and certain monotone stepsizes for minimizing general strongly convex quadratic function. Furthermore, by incorporating nonmonotone line searches and gradient projection techniques, we extend these new gradient methods to solve general smooth unconstrained and bound constrained optimization. Extensive numerical experiments show that our strategies of properly inserting monotone gradient steps into the nonmonotone BB method could significantly improve its performance and the new resulted methods can outperform the most successful gradient decent methods developed in the recent literature.

Mathematics Subject Classification (2000) 90C20 · 90C25 · 90C30

Yakui Huang
Institute of Mathematics, Hebei University of Technology, Tianjin 300401, China
E-mail: huangyakui2006@gmail.com

Yu-Hong Dai
LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, 100190, Beijing, China
E-mail: dyh@lsec.cc.ac.cn

Xin-Wei Liu
Institute of Mathematics, Hebei University of Technology, Tianjin 300401, China
E-mail: mathlxw@hebut.edu.cn

Hongchao Zhang
Department of Mathematics, Louisiana State University, Baton Rouge, LA 70803-4918, USA
E-mail: hozhang@math.lsu.edu

1 Introduction

Gradient descent methods have been widely used for solving smooth unconstrained nonlinear optimization

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

by generating a sequence of iterates

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable, $g_k = \nabla f(x_k)$ and $\alpha_k > 0$ is the stepsize along the negative gradient. Different gradient descent methods would have different rules for determining the stepsize α_k. The classic steepest descent (SD) method proposed by Cauchy [6] determines its stepsize by the so-called exact line search

$$\alpha_k^{SD} = \arg\min_{\alpha\in\mathbb{R}} \; f(x_k - \alpha g_k). \tag{3}$$

Although the SD method locally has the most function value reduction along the negative gradient direction, it often performs poorly in practice. Theoretically, when f is a strongly convex quadratic function, i.e.,

$$f(x) = \frac{1}{2}x^T Ax - b^T x, \tag{4}$$

where $b \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n\times n}$ is symmetric and positive definite, SD method converges Q-linearly [1] and will asymptotically perform zigzag between two orthogonal directions [19,26].

In 1988, Barzilai and Borwein [3] proposed the following two novel stepsizes that significantly improve the performance of gradient descent methods:

$$\alpha_k^{BB1} = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}} \qquad \text{and} \qquad \alpha_k^{BB2} = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}, \tag{5}$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$. Clearly, when $s_{k-1}^T y_{k-1} > 0$, one has $\alpha_k^{BB1} \geq \alpha_k^{BB2}$. Hence, α_k^{BB1} is often called the *long* BB stepsize while α_k^{BB2} is called the *short* BB stepsize. When the objective function is quadratic (4), the BB stepsize α_k^{BB1} will be exactly the steepest descent stepsize, but with one step retard, while α_k^{BB2} will be just the stepsize of minimal gradient (MG) method [11], that is

$$\alpha_k^{BB1} = \frac{g_{k-1}^T g_{k-1}}{g_{k-1}^T A g_{k-1}} = \alpha_{k-1}^{SD} \quad \text{and} \quad \alpha_k^{BB2} = \frac{g_{k-1}^T A g_{k-1}}{g_{k-1}^T A^2 g_{k-1}} = \alpha_{k-1}^{MG}.$$

It is proved that the Barzilai-Borwein (BB) method converges R-superlinearly for minimizing two-dimensional strongly convex quadratic function [3] and R-linearly for the general n-dimensional case [10]. Although the BB method does not decrease the objective function value monotonically, extensive numerical experiments show that it performs much better than the SD method [17,30, 35]. And it is commonly accepted that when a not high accuracy is required, BB-type methods could be even competitive with nonlinear conjugate gradient (CG) methods for solving smooth unconstrained optimization [17,30]. Furthermore, by combing with gradient projection techniques, the BB-type methods have a great advantage of easy extension to solve a wide class of constrained optimization, for example the bound or simplex constrained optimization [8]. Hence, BB-type methods enjoy many important applications, such as image restoration [32], signal processing [29], eigenvalue problems [28], nonnegative matrix factorization [27], sparse reconstruction [33], machine learning [31], etc.

Recently, Yuan [34,35] propose a gradient descent method which combines a new stepsize

$$\alpha_k^Y = \frac{2}{\frac{1}{\alpha_{k-1}^{SD}} + \frac{1}{\alpha_k^{SD}} + \sqrt{\left(\frac{1}{\alpha_{k-1}^{SD}} - \frac{1}{\alpha_k^{SD}}\right)^2 + \frac{4\|g_k\|^2}{(\alpha_{k-1}^{SD}\|g_{k-1}\|)^2}}}, \tag{6}$$

in the SD method so that the new method enjoys finite termination for minimizing a two-dimensional strongly convex quadratic function. Based on this new stepsize α_k^Y, Dai and Yuan [12] further develop the DY method, which alternately employs α_k^{SD} and α_k^Y stepsizes as follows

$$\alpha_k^{DY} = \begin{cases} \alpha_k^{SD}, \text{ if mod}(k,4) < 2; \\ \alpha_k^Y, \quad \text{otherwise.} \end{cases} \tag{7}$$

It is easy to see that $\alpha_k^Y \leq \alpha_k^{SD}$. Hence, DY method (7) is a monotone method. Moreover, it is shown that DY method (7) performs better than the nonmonotone BB method [12].

The property of nonmonotonically reducing objective function values is an intrinsic feature that causes the efficiency of BB method. However, it is also pointed out by Fletcher [18] that retaining monotonicity is important for a gradient method, especially for minimizing general objective functions. On the other hand, although the monotone DY method performs well, using α_k^{SD} and α_k^Y in a nonmonotone fashion may yield better performance, see [14] for example. Moreover, it is usually difficult to compute the exact monotone stepsize α_k^{SD} in general optimization. Hence, in this paper, motivated by the great success of the BB method and the previous considerations, we want to further improve and accelerate the *nonmonotone* BB method by incorporating some *monotone* steps. For a more general and uniform analysis, we first consider to accelerate the class of gradient descent methods (2) for quadratic optimization (4) using the following stepsize

$$\alpha_k(\Psi(A)) = \frac{g_{k-1}^T \Psi(A) g_{k-1}}{g_{k-1}^T \Psi(A) A g_{k-1}}, \tag{8}$$

where $\Psi(\cdot)$ is a real analytic function on $[\lambda_1, \lambda_n]$ that can be expressed by a Laurent series

$$\Psi(z) = \sum_{k=-\infty}^{\infty} c_k z^k, \quad c_k \in \mathbb{R},$$

such that $0 < \sum_{k=-\infty}^{\infty} c_k z^k < +\infty$ for all $z \in [\lambda_1, \lambda_n]$. Here, λ_1 and λ_n are the smallest and largest eigenvalues of A, respectively. Clearly, the method (8) is generally nonmonotone and the two BB stepsizes α_k^{BB1} and α_k^{BB2} can be obtained by setting $\Psi(A) = I$ and $\Psi(A) = A$ in (8), respectively.

More particularly, we will derive a new stepsize, say $\tilde{\alpha}_k(\Psi(A))$, which together with the stepsize $\alpha_k(\Psi(A))$ in (8) can minimize the two-dimensional convex quadratic function in no more than 5 iterations. To the best of our knowledge, this is the first nonmonotone gradient method with finite termination property. We will see that $\tilde{\alpha}_k(I) \leq \alpha_k^{SD}$ and $\tilde{\alpha}_k(A) \leq \alpha_k^{MG}$. Hence, this finite termination property is essentially obtained by inserting monotone stepsizes into the generally nonmonotone stepsizes (8). In fact, we show that this finite termination property can be maintained even the algorithm uses different function Ψ's during its iteration. Based on this observation, to achieve good numerical performance, we propose an adaptive nonmonotone gradient method (ANGM), which adaptively takes some nonmonotone steps involving the long and short BB stepsizes (5), and some monotone steps using $\tilde{\alpha}_k(A)$. Moreover, to efficiently minimize more general nonlinear objective function, we propose two variants of ANGM, called ANGR1 and ANGR2, using certain retard stepsize. By combing nonmonotone line search and gradient projection techniques, these two variants of gradient methods are further extended to solve bound constrained optimization. Our numerical experiments show that the new proposed methods significantly accelerate the BB method and perform much better on minimizing quadratic function (4) than the most successful gradient decent methods developed in the recent literature, such as DY method [12], ABBmin2 method [20] and SDC method [14]. In addition, we also compare ANGR1 and ANGR2 with the spectral projected gradient (SPG) method [4,5] and the BB method using Dai-Zhang nonmonotone line search (BB1-DZ) [13] for solving the general unconstrained problems from [2] and the bound constrained problems from the CUTEst collection [23]. The numerical results highly suggest the potential benefits of our new proposed methods for solving more general unconstrained and bound constrained optimization.

The paper is organized as follows. In Section 2, we derive the new stepsize $\tilde{\alpha}_k(\Psi(A))$ by requiring finite termination on minimizing two-dimensional strongly convex quadratic function. In Section 3, we first derive the ANGM, ANGR1 and ANGR2 methods for minimizing general strongly convex quadratic function and then generalize ANGR1 and ANGR2 to solve bound constrained optimization. Our extensive numerical experiments on minimizing strongly convex quadratic function and solving general unconstrained and bound constrained optimization are presented in Section 4. We finally draw some conclusion remarks in Section 5.

2 Derivation of new stepsize

In this section, we derive a new monotone stepsize based on the nonmonotone gradient method (8) to minimize quadratic function (4). This new stepsize is motivated by requiring finite termination for minimizing two-dimensional strongly convex quadratic function. Such an idea was originally proposed by Yuan [34] to accelerate SD method. However, new techniques need to be developed for accelerating the class of nonmonotone gradient descent methods (8) since the key orthogonal property of successive two gradients generated by SD method no longer holds for methods (8).

Observe that the method (8) is invariant under translations and rotations when minimizing quadratics. Hence, for theoretical analysis to minimize (4), without loss of generality, we may simply assume that the matrix A is diagonal, i.e.,

$$A = \operatorname{diag}\{\lambda_1, \dots, \lambda_n\}, \tag{9}$$

where $0 < \lambda_1 \leq \ldots \leq \lambda_n$.

2.1 Motivation

First, let us investigate the behavior of gradient method (8) with $\Psi(A) = I$ (i.e., the BB1 method). Particularly, we apply it to the non-random quadratic minimization problem proposed in [14], which has the form (4) with a diagonal matrix A given by

$$A_{jj} = 10^{\frac{ncond}{n-1}(n-j)}, \quad j = 1, \dots, n, \tag{10}$$

and b being a null vector. Here, $ncond = \log_{10} \kappa$ and $\kappa > 0$ is the condition number of A. We set $n = 10$, $\kappa = 10^3$ and use $(10, 10, \dots, 10)^T$ as the initial point. The iteration was stopped once the gradient norm is reduced by a factor of 10^{-6}. Denote the i-th component of g_k by $g_k^{(i)}$ and the indices of the components of g_k with two largest magnitudes by i_1 and i_2, respectively. Then, the percentage of the magnitudes of the first two largest components to that of the whole gradient can be computed by

$$\Upsilon(g_k) = \frac{|g_k^{(i_1)}| + |g_k^{(i_2)}|}{\sum_{i=1}^n |g_k^{(i)}|}.$$

This $\Upsilon(g_k)$ is plotted in Fig. 1 (left), where we can see that $\Upsilon(g_k) \geq 0.8$ holds for more than half of the iterations (145 out of 224 total iterations). Hence, roughly speaking, the searches of the BB1 method are often dominated in some two-dimensional subspaces. The history of index i_1 against the iteration number is also plotted in Fig. 1 (right), where we can see that $|g_k^{(i_1)}|$ corresponds more frequently to the largest eigenvalues λ_{10} or λ_9. Since

$$g_{k+1}^{(j)} = (1 - \alpha_k \lambda_j) g_k^{(j)}, \ j = 1, \dots, n. \tag{11}$$

and $1/\lambda_n \leq \alpha_k \leq 1/\lambda_1$, the history of i_1 in Fig. 1 (right) in fact indicates that, most stepsizes generated by the BB1 method are often much larger than $1/\lambda_{10}$ or $1/\lambda_9$. As a result, the BB1 method may need many iterations to reduce the corresponding components of the gradients $g_k^{(9)}$ or $g_k^{(10)}$.

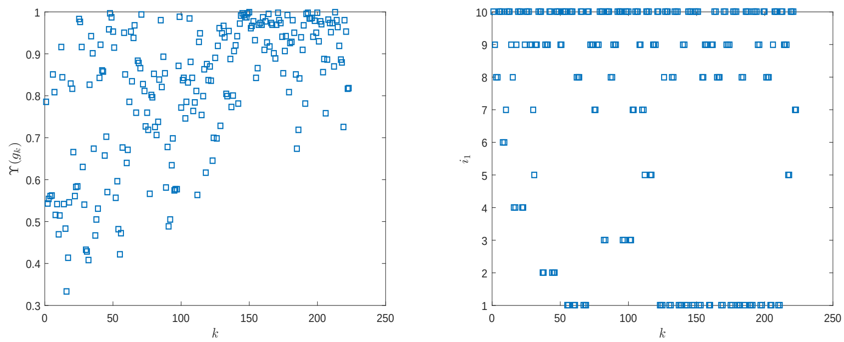

Fig. 1: Problem (10) with $n = 10$: history of $\Upsilon(g_k)$ (left) and the index i_1 (right) generated by the BB1 method

In [26], we showed that a family of gradient methods including SD and MG will asymptotically reduce their searches in a two-dimensional subspace and could be accelerated by exploiting certain orthogonal properties in this two-dimensional subspace. In a similar spirit, we could also accelerate the convergence of the class of gradient methods (8) in a lower dimensional subspace if certain orthogonal properties hold.

Suppose that, for a given $k > 0$, there exists a q_k satisfying

$$(I - \alpha_{k-1}A)q_k = g_{k-1}. \tag{12}$$

Since this q_k is also invariant under translations and rotations, for later analysis we may still assume A in (12) is diagonal as in (9). The following lemma shows a generalized orthogonal property for q_k and g_{k+1}, which is a key property for deriving our new stepsize in the next subsection.

Lemma 1 (Orthogonal property) *Suppose that the sequence $\{g_k\}$ is obtained by applying gradient method* (2) *with stepsizes* (8) *to minimize a quadratic function* (4) *and q_k satisfies* (12). *Then, we have*

$$q_k^T \Psi(A) g_{k+1} = 0. \tag{13}$$

Proof By (2), (8) and (12) we get

$$\begin{aligned} q_k^T \Psi(A) g_{k+1} &= q_k^T \Psi(A)(I - \alpha_k A)(I - \alpha_{k-1}A)g_{k-1} \\ &= g_{k-1}^T \Psi(A)(I - \alpha_k A) g_{k-1} \\ &= g_{k-1}^T \Psi(A) g_{k-1} - \alpha_k g_{k-1}^T \Psi(A) A g_{k-1} = 0. \end{aligned}$$

This completes the proof. □

2.2 A new stepsize

In this subsection, we derive a new stepsize based on the iterations of gradient method (8). We show that combining the new stepsize with gradient method (8), we can achieve finite termination for minimizing two-dimensional strongly convex quadratic functions.

By Lemma 1, we have that $g_k^T \Psi(A) q_{k-1} = 0$ for $k > 0$. Now, suppose both $\Psi^r(A) q_{k-1}$ and $\Psi^{1-r}(A) g_k$ are nonzero vectors, where $r \in \mathbb{R}$. Let us consider to minimize the function f in a two-dimensional subspace spanned by $\frac{\Psi^r(A) q_{k-1}}{\|\Psi^r(A) q_{k-1}\|}$ and $\frac{\Psi^{1-r}(A) g_k}{\|\Psi^{1-r}(A) g_k\|}$, and let

$$\begin{aligned} \varphi(t,l) &:= f\left(x_k + t \frac{\Psi^r(A) q_{k-1}}{\|\Psi^r(A) q_{k-1}\|} + l \frac{\Psi^{1-r}(A) g_k}{\|\Psi^{1-r}(A) g_k\|} \right) \\ &= f(x_k) + \vartheta_k^T \begin{pmatrix} t \\ l \end{pmatrix} + \frac{1}{2} \begin{pmatrix} t \\ l \end{pmatrix}^T H_k \begin{pmatrix} t \\ l \end{pmatrix}, \end{aligned} \tag{14}$$

where

$$\vartheta_k = B_k g_k = \begin{pmatrix} \frac{g_k^T \Psi^r(A) q_{k-1}}{\|\Psi^r(A) q_{k-1}\|} \\ \frac{g_k^T \Psi^{1-r}(A) g_k}{\|\Psi^{1-r}(A) g_k\|} \end{pmatrix} \text{ with } B_k = \left(\frac{\Psi^r(A) q_{k-1}}{\|\Psi^r(A) q_{k-1}\|}, \frac{\Psi^{1-r}(A) g_k}{\|\Psi^{1-r}(A) g_k\|} \right)^T \tag{15}$$

and

$$H_k = B_k A B_k^T = \begin{pmatrix} \frac{q_{k-1}^T \Psi^{2r}(A) A q_{k-1}}{\|\Psi^r(A) q_{k-1}\|^2} & \frac{q_{k-1}^T \Psi(A) A g_k}{\|\Psi^r(A) q_{k-1}\| \|\Psi^{1-r}(A) g_k\|} \\ \frac{q_{k-1}^T \Psi(A) A g_k}{\|\Psi^r(A) q_{k-1}\| \|\Psi^{1-r}(A) g_k\|} & \frac{g_k^T \Psi^{2(1-r)}(A) A g_k}{\|\Psi^{1-r}(A) g_k\|^2} \end{pmatrix}. \tag{16}$$

Denote the components of H_k by $H_k^{(ij)}$, $i, j = 1, 2$ and notice that $B_k B_k^T = I$ by $g_k^T \Psi(A) q_{k-1} = 0$. Then, we have the following finite termination theorem.

Theorem 1 (Finite termination) *Suppose that a gradient method* (2) *is applied to minimize a two-dimensional quadratic function* (4) *with α_k given by* (8) *for all $k \neq k_0$ and uses the stepsize*

$$\tilde{\alpha}_{k_0} = \frac{2}{(H_{k_0}^{(11)} + H_{k_0}^{(22)}) + \sqrt{(H_{k_0}^{(11)} - H_{k_0}^{(22)})^2 + 4(H_{k_0}^{(12)})^2}} \tag{17}$$

at the k_0-th iteration where $k_0 \geq 2$. Then, the method will find the minimizer in at most $k_0 + 3$ iterations.

Proof Let us suppose x_k is not a minimizer for all $k = 1, \ldots, k_0 + 2$. We then show $k_0 + 3$ must be the minimizer, i.e., $g_{k_0+3} = 0$. For notation convenience, in the following proof of this theorem, let's simply use k to denote k_0. First, we show that using stepsize (17) at the k-th iteration implies

$$g_{k+1} \quad \text{is parallel to} \quad -B_k^T H_k^{-1} \vartheta_k + \tilde{\alpha}_k g_k, \tag{18}$$

where ϑ_k, B_k and H_k is given by (15) and (16). In fact, $\tilde{\alpha}_k$ given by (17) satisfies the following quadratic equation

$$\tilde{\alpha}_k^2 \Delta - \tilde{\alpha}_k (H_k^{(11)} + H_k^{(22)}) + 1 = 0, \tag{19}$$

where $\Delta = \det(H_k) = \det(A) > 0$. Let

$$\Theta = (H_k^{(12)}\vartheta_k^{(1)} + H_k^{(22)}\vartheta_k^{(2)})\vartheta_k^{(1)} - (H_k^{(11)}\vartheta_k^{(1)} + H_k^{(12)}\vartheta_k^{(2)})\vartheta_k^{(2)},$$

where $\vartheta_k^{(i)}$ are components of ϑ_k, $i = 1, 2$. Then, multiplying Θ to (19), we have

$$\tilde{\alpha}_k^2 \Delta\Theta - \tilde{\alpha}_k (H_k^{(11)} + H_k^{(22)})\Theta + \Theta = 0, \tag{20}$$

which is exactly

$$\begin{aligned} &(H_k^{(22)}\vartheta_k^{(1)} - H_k^{(12)}\vartheta_k^{(2)} - \tilde{\alpha}_k\Delta\vartheta_k^{(1)})[\vartheta_k^{(2)} - \tilde{\alpha}_k(H_k^{(12)}\vartheta_k^{(1)} + H_k^{(22)}\vartheta_k^{(2)})] \\ =&(H_k^{(11)}\vartheta_k^{(2)} - H_k^{(12)}\vartheta_k^{(1)} - \tilde{\alpha}_k\Delta\vartheta_k^{(2)})[\vartheta_k^{(1)} - \tilde{\alpha}_k(H_k^{(11)}\vartheta_k^{(1)} + H_k^{(12)}\vartheta_k^{(2)})]. \end{aligned} \tag{21}$$

The above identity (21) implies the vector

$$\begin{pmatrix} \vartheta_k^{(1)} - \tilde{\alpha}_k(H_k^{(11)}\vartheta_k^{(1)} + H_k^{(12)}\vartheta_k^{(2)}) \\ \vartheta_k^{(2)} - \tilde{\alpha}_k(H_k^{(12)}\vartheta_k^{(1)} + H_k^{(22)}\vartheta_k^{(2)}) \end{pmatrix}$$

is parallel to

$$\begin{pmatrix} H_k^{(22)}\vartheta_k^{(1)} - H_k^{(12)}\vartheta_k^{(2)} - \tilde{\alpha}_k\Delta\vartheta_k^{(1)} \\ H_k^{(11)}\vartheta_k^{(2)} - H_k^{(12)}\vartheta_k^{(1)} - \tilde{\alpha}_k\Delta\vartheta_k^{(2)} \end{pmatrix},$$

which written in a matrix format just means

$$\vartheta_k + H_k(-\tilde{\alpha}_k\vartheta_k) \quad \text{is parallel to} \quad H_k^{-1}\vartheta_k - \tilde{\alpha}_k\vartheta_k. \tag{22}$$

Since $n = 2$, we have $B_k B_k^T = B_k^T B_k = I$. Then, it follows from $g_k = B_k^T \vartheta_k$, $\vartheta_k = B_k g_k$, $H_k = B_k A B_k^T$ and $g_{k+1} = g_k - \tilde{\alpha}_k A g_k$ that $g_{k+1} = B_k^T \vartheta_k + B_k^T H_k(-\tilde{\alpha}_k\vartheta_k)$. So, we have from (22) that (18) holds. Therefore, (17) implies (18) holds.

Now, it follows from (15) and $H_k^{-1} = B_k A^{-1} B_k^T$ that

$$-B_k^T H_k^{-1}\vartheta_k + \tilde{\alpha}_k g_k = -A^{-1}g_k + \tilde{\alpha}_k g_k = -A^{-1}(g_k - \tilde{\alpha}_k A g_k) = -A^{-1}g_{k+1}.$$

Hence, (18) implies g_{k+1} is parallel to $A^{-1}g_{k+1}$. So, if $\tilde{\alpha}_k$ given by (17) is used at the k-th iteration, then g_{k+1} is parallel to $A^{-1}g_{k+1}$. Since x_{k+1} is not the minimizer, we have $g_{k+1} \neq 0$. So, g_{k+1} is an eigenvector of A, i.e. $Ag_{k+1} = \lambda g_{k+1}$ for some $\lambda > 0$. Since x_{k+2} is not the minimizer, we have $g_{k+2} \neq 0$ and the algorithm will not stop at the $k+2$-th iteration. So, by (8), we have $\alpha_{k+2} = \frac{g_{k+1}^T \Psi(A) g_{k+1}}{g_{k+1}^T \Psi(A) A g_{k+1}} = 1/\lambda$. Hence, we have $g_{k+3} = (1 - \alpha_{k+2}\lambda) g_{k+2} = 0$, which implies x_{k+3} must be the minimizer. We complete the proof. □

Notice that by setting $k_0 = 2$ in the above Theorem 1, the new gradient method in Theorem 1 will find the exact minimizer in at most 5 iterations when minimizing a two-dimensional strongly convex quadratic function. In fact, since $\Delta = \lambda_1\lambda_2$ and $H_k^{(11)} + H_k^{(22)} = \lambda_1 + \lambda_2$, the equation (19) has two positive roots $1/\lambda_1$ and $1/\lambda_2$. This observation allows us to use the stepsize $\tilde{\alpha}_{k_0}$ with some retards as stated in the following corollary, which would lead us a more convenient way for choosing stepsizes when the objective function is not quadratic.

Corollary 1 *Suppose that a gradient method is applied to a two-dimensional quadratic function* (4) *with* $\alpha_{k_0+m} = \tilde{\alpha}_{k_0}$ *for* $k_0 \geq 2$ *and some positive integer* m*, and* α_k *given by* (8) *for all* $k \neq k_0 + m$*. Then, the method stops in at most* $k_0 + m + 3$ *iterations.*

By setting $\Psi(A) = I$, $\Psi(A) = A$ and $r = 1/2$ in (16), and setting $k_0 = k$ in (17), we can derive the following two stepsizes:

$$\tilde{\alpha}_k^{BB1} = \frac{2}{\frac{q_{k-1}^T A q_{k-1}}{\|q_{k-1}\|^2} + \frac{1}{\alpha_k^{SD}} + \sqrt{\left(\frac{q_{k-1}^T A q_{k-1}}{\|q_{k-1}\|^2} - \frac{1}{\alpha_k^{SD}}\right)^2 + \frac{4(q_{k-1}^T A g_k)^2}{\|q_{k-1}\|^2\|g_k\|^2}}} \tag{23}$$

and

$$\tilde{\alpha}_k^{BB2} = \frac{2}{\frac{1}{\hat{\alpha}_{k-1}} + \frac{1}{\alpha_k^{MG}} + \sqrt{\left(\frac{1}{\hat{\alpha}_{k-1}} - \frac{1}{\alpha_k^{MG}}\right)^2 + \Gamma_k}}, \tag{24}$$

respectively, where

$$\hat{\alpha}_k = \frac{q_k^T A q_k}{q_k^T A^2 q_k} \quad \text{and} \quad \Gamma_k = \frac{4(q_{k-1}^T A^2 g_k)^2}{q_{k-1}^T A q_{k-1} \cdot g_k^T A g_k}. \tag{25}$$

By (23) and (24), we have

$$\tilde{\alpha}_k^{BB1} \leq \min\left\{\alpha_k^{SD}, \frac{\|q_{k-1}\|^2}{q_{k-1}^T A q_{k-1}}\right\} \quad \text{and} \quad \tilde{\alpha}_k^{BB2} \leq \min\{\alpha_k^{MG}, \hat{\alpha}_{k-1}\}. \tag{26}$$

Hence, both $\tilde{\alpha}_k^{BB1}$ and $\tilde{\alpha}_k^{BB2}$ are short monotone steps for reducing the value and gradient norm of the objective function, respectively. And it follows from Theorem 1 that by inserting the monotone steps $\tilde{\alpha}_k^{BB1}$ and $\tilde{\alpha}_k^{BB2}$ into the BB1 and BB2 methods, respectively, the gradient method will have finite termination for minimizing two-dimensional strongly convex quadratic functions.

To numerically verify this finite termination property, we apply the method (8) with $\Psi(A) = I$ (i.e., the BB1 method) and $\tilde{\alpha}_2^{BB1}$ given by (23) to minimize a two-dimensional quadratic function (4) with

$$A = \mathrm{diag}\{1, \lambda\} \quad \text{and} \quad b = 0. \tag{27}$$

We run the algorithm for five iterations using ten random starting points. The averaged values of $\|g_5\|$ and $f(x_5)$ are presented in Table 1. Moreover, we also run BB1 method for a comparison purpose. We can observe that for different values of λ, the values of $\|g_5\|$ and $f(x_5)$ obtained by BB1 method with $\tilde{\alpha}_2^{BB1}$ given by (23) are numerically very close to zero. However, even for the case $\lambda = 10$, $\|g_5\|$ and $f(x_5)$ obtained by pure BB1 method are far away from zero. These numerical results coincide with our analysis and show that the nonmonotone method (8) can be significantly accelerated by incorporating proper monotone steps.

Table 1: Averaged results on problem (27) with different condition numbers

λ	BB1		BB1 with $\tilde{\alpha}_2^{BB1}$ given by (23)	
	$\|g_5\|$	$f(x_5)$	$\|g_5\|$	$f(x_5)$
10	6.6873e+00	4.8701e+00	1.1457e-16	5.8735e-31
100	8.6772e+01	1.9969e+02	2.1916e-16	2.8047e-30
1000	2.0925e+02	9.7075e+01	3.4053e-19	2.3730e-29
10000	1.7943e+02	9.6935e+00	5.1688e-19	6.7870e-28

3 New methods

In this section, based on the above analysis, we propose an adaptive nonmonotone gradient method (ANGM) and its two variants, ANGR1 and ANGR2, for solving both unconstrained and box constrained optimization. These new gradient methods adaptively take some nonmonotone steps involving the long and short BB stepsizes (5), and some monotone steps using the new stepsize developed in the previous section.

3.1 Quadratic case

As mentioned in Section 2.1, the stepsizes α_k^{BB1} generated by the BB1 method may be far away from the reciprocals of the largest eigenvalues of the Hessian matrix A of the quadratic function (4). In other words, the stepsize α_k^{BB1} may be too large to effectively decrease the components of gradient g_k corresponding to the first several largest eigenvalues, which, by (11), can be greatly reduced when small stepsizes are employed. In addition, it has been observed by many works in the recent literature that gradient methods using long and short stepsizes adaptively generally perform much better than those using only one type of stepsizes, for example see [11,14,15,22,25,26]. So, we would like to develop gradient methods that combines the two nonmonotone BB stepsizes with the short monotone stepsize given by (17).

We first extend the orthogonal property developed in Lemma 1 and the finite termination result given in Theorem 1.

Lemma 2 (Generalized orthogonal property) *Suppose that a gradient method* (2) *with stepsizes in the form of* (8) *is applied to minimize a quadratic function* (4)*. In particular, at the* $k-1$*-th and* k*-th iteration, two stepsizes* $\alpha_{k-1}(\Psi(A))$ *and* $\alpha_k(\Psi_1(A))$ *are used, respectively, where* Ψ *and* Ψ_1 *may be two different analytic functions used in* (8)*. If* $q_k \in \mathbb{R}^n$ *satisfies*

$$(I - \alpha_{k-1}(\Psi(A))A)q_k = g_{k-1}, \tag{28}$$

then we have

$$q_k^T \Psi_1(A) g_{k+1} = 0. \tag{29}$$

Proof Notice that by (2), we have

$$g_k = g_{k-1} - \alpha_{k-1}(\Psi(A))Ag_{k-1} \quad \text{and} \quad g_{k+1} = g_k - \alpha_k(\Psi_1(A))Ag_k.$$

Then, the proof is essential the same as those in the proof of Lemma 1. □

Based on Lemma 2 and using the same arguments as those in the proof of Theorem 1, we can obtain the following finite termination result even different function Ψ's are used in (8) to obtain the stepsizes.

Theorem 2 (Generalized finite termination) *Suppose that a gradient method* (2) *is applied to minimize a two-dimensional quadratic function* (4) *with* α_k *given by* (8) *for all* $k \neq k_0$ *and* $k \neq k_0 - 1$*, and uses the stepsizes* $\alpha_{k-1}(\Psi_1(A))$ *and* $\alpha_k(\Psi_1(A))$ *at the* $k-1$*-th and* k*-th iteration, respectively, where* $k_0 \geq 2$*. Then, the method will find the minimizer in at most* $k_0 + 3$ *iterations.*

Theorem 2 allows us to incorporate the nonmonotone BB stepsizes α_k^{BB1} and α_k^{BB2}, and the short monotone stepsize $\tilde{\alpha}_k^{BB2}$ in one gradient method. Alternate or adaptive scheme has been employed for choosing long and short stepsizes in BB-type methods [8,36]. And recent studies show that adaptive strategies are more preferred than the alternate scheme [9,36]. Hence, we would like develop adaptive strategies to choose proper stepsizes for our new gradient methods. In particular, our adaptive nonmonotone gradient method (ANGM) takes the long BB stepsize α_k^{BB1} when $\alpha_k^{BB2}/\alpha_k^{BB1} \geq \tau_1$ for some $\tau_1 \in (0,1)$. Otherwise, a short stepsize α_k^{BB2} or $\tilde{\alpha}_k^{BB2}$ will be taken depending on the ratio $\|g_{k-1}\|/\|g_k\|$. Notice that α_k^{BB2} minimizes the gradient in the sense that

$$\alpha_k^{BB2} = \alpha_{k-1}^{MG} = \arg\min_{\alpha \in \mathbb{R}} \|g_{k-1} - \alpha A g_{k-1}\|.$$

So, when $\|g_{k-1}\|/\|g_k\| > \tau_2$ for some $\tau_2 > 1$, i.e. the gradient norm decreases, the previous stepsize α_{k-1} is often a reasonable approximation of α_k^{BB2}. By our numerical experiments, when BB method is applied the searches are often dominated in some two-dimensional subspaces. And the new gradient method in Theorem 2 would have finite convergence for minimizing two-dimensional quadratic function when the new stepsize $\tilde{\alpha}_k^{BB2}$ is applied after some BB2 steps. Hence, our ANGM would employ the new monotone stepsize $\tilde{\alpha}_k^{BB2}$ when $\|g_{k-1}\| \geq \tau_2 \|g_k\|$; otherwise, certain BB2 steps should be taken. In practice, we find that when $\|g_{k-1}\| \leq \tau_2 \|g_k\|$, ANGM often has good performance by taking the stepsize $\min\{\alpha_k^{BB2}, \alpha_{k-1}^{BB2}\}$. To summarize, our ANGM applies the following adaptive strategies for choosing stepsizes:

$$\alpha_k = \begin{cases} \min\{\alpha_k^{BB2}, \alpha_{k-1}^{BB2}\}, & \text{if } \alpha_k^{BB2} < \tau_1 \alpha_k^{BB1} \text{ and } \|g_{k-1}\| < \tau_2 \|g_k\|; \\ \tilde{\alpha}_k^{BB2}, & \text{if } \alpha_k^{BB2} < \tau_1 \alpha_k^{BB1} \text{ and } \|g_{k-1}\| \geq \tau_2 \|g_k\|; \\ \alpha_k^{BB1}, & \text{otherwise,} \end{cases} \tag{30}$$

Notice that the calculation of $\tilde{\alpha}_k^{BB2}$ needs to compute α_k^{MG} which is not easy to obtain when the objective function is not quadratic. In stead, the calculation of $\tilde{\alpha}_{k-1}^{BB2}$ will just require α_k^{BB2}, which is readily available even for general objective function. Moreover, it is found in recent research that gradient methods using retard stepsizes can often lead better performances [21]. Hence, in the first variant of ANGM, we simply replace $\tilde{\alpha}_k^{BB2}$ in (30) by $\tilde{\alpha}_{k-1}^{BB2}$, i.e. the stepsizes are chosen as

$$\alpha_k = \begin{cases} \min\{\alpha_k^{BB2}, \alpha_{k-1}^{BB2}\}, & \text{if } \alpha_k^{BB2} < \tau_1 \alpha_k^{BB1} \text{ and } \|g_{k-1}\| < \tau_2 \|g_k\|; \\ \tilde{\alpha}_{k-1}^{BB2}, & \text{if } \alpha_k^{BB2} < \tau_1 \alpha_k^{BB1} \text{ and } \|g_{k-1}\| \geq \tau_2 \|g_k\|; \\ \alpha_k^{BB1}, & \text{otherwise.} \end{cases} \tag{31}$$

We call the gradient method using stepsize (31) ANGR1. On the other hand, since the calculation of $\tilde{\alpha}_{k-1}^{BB2}$ also needs $\hat{\alpha}_{k-2}$ and Γ_{k-1} and by (26),

$$\tilde{\alpha}_{k-1}^{BB2} \leq \min\{\alpha_k^{BB2}, \hat{\alpha}_{k-2}\}, \tag{32}$$

to simplify ANGR1, we may further replace $\tilde{\alpha}_{k-1}^{BB2}$ in (31) by its upper bound in (32). As a result, we have the second variant of ANGM, which chooses stepsizes as

$$\alpha_k = \begin{cases} \min\{\alpha_k^{BB2}, \alpha_{k-1}^{BB2}\}, & \text{if } \alpha_k^{BB2} < \tau_1 \alpha_k^{BB1} \text{ and } \|g_{k-1}\| < \tau_2 \|g_k\|; \\ \min\{\alpha_k^{BB2}, \hat{\alpha}_{k-2}\}, & \text{if } \alpha_k^{BB2} < \tau_1 \alpha_k^{BB1} \text{ and } \|g_{k-1}\| \geq \tau_2 \|g_k\|; \\ \alpha_k^{BB1}, & \text{otherwise.} \end{cases} \tag{33}$$

We call the gradient method using stepsize (33) ANGR2.

In terms of global convergence for minimizing quadratic function (4), by (26), we can easily show the R-linear global convergence of ANGM since it satisfies the property in [7]. Similarly, R-linear convergence of ANGR1 and ANGR2 can be also established. See the proof of Theorem 3 in [9] for example.

Remark 1 Compared with other gradient methods, ANGM, ANGR1 and ANGR2 do not need additional matrix-vector products. In fact, it follows from (12) that $Aq_k = \frac{1}{\alpha_{k-1}}(q_k - g_{k-1})$, which gives

$$\hat{\alpha}_k = \frac{q_k^T A q_k}{q_k^T A^2 q_k} = \frac{\alpha_{k-1} q_k^T (q_k - g_{k-1})}{(q_k - g_{k-1})^T (q_k - g_{k-1})}. \tag{34}$$

Hence, no additional matrix-vector products are needed for calculation of $\hat{\alpha}_{k-1}$ in $\tilde{\alpha}_k^{BB2}$, $\hat{\alpha}_{k-2}$ in $\tilde{\alpha}_{k-1}^{BB2}$ and the stepsize used in ANGR2. Since the calculation of Ag_k is necessary for the calculation of g_{k+1}, Γ_k in $\tilde{\alpha}_k^{BB2}$ requires no additional matrix-vector products either. As for $\tilde{\alpha}_{k-1}^{BB2}$, it follows from $g_{k-1}^T A^2 q_{k-2} = \frac{1}{\alpha_{k-3}}(q_{k-2} - g_{k-3})^T A g_{k-1}$ and $Ag_{k-1} = \frac{1}{\alpha_{k-1}}(g_{k-1} - g_k)$ that

$$\begin{aligned}\Gamma_{k-1} &= \frac{4((q_{k-2} - g_{k-3})^T A g_{k-1})^2}{\alpha_{k-3}((q_{k-2} - g_{k-3})^T q_{k-2}) \cdot g_{k-1}^T A g_{k-1}} \\ &= \frac{4((q_{k-2} - g_{k-3})^T (g_{k-1} - g_k))^2}{\alpha_{k-3}\alpha_{k-1}((q_{k-2} - g_{k-3})^T q_{k-2}) \cdot g_{k-1}^T (g_{k-1} - g_k)}. \end{aligned} \tag{35}$$

Thus, no additional matrix-vector products are required for calculation of Γ_{k-1} in $\tilde{\alpha}_{k-1}^{BB2}$.

Remark 2 Notice that all the new methods, ANGM, ANGR1 and ANGR2, require the vector q_k for calculation of their stepsizes. However, computing q_k exactly from (12) maybe as difficult as minimizing the quadratic function. Notice that the q_k satisfying (12) also satisfies the secant equation

$$q_k^T g_k = \|g_{k-1}\|^2. \tag{36}$$

Hence, we may find an approximation of q_k by requiring the above secant condition holds. One efficient way to find such a q_k satisfying the secant equation (36) is to simply treat the Hessian matrix A as the diagonal matrix (9) and derive q_k from (12), that is when $g_k^{(i)} \neq 0$,

$$q_k^{(i)} = \frac{g_{k-1}^{(i)}}{1 - \alpha_{k-1}\lambda_i} = \frac{(g_{k-1}^{(i)})^2}{g_k^{(i)}}, \quad i = 1, \ldots, n, \tag{37}$$

And we can just let $q_k^{(i)} = 0$, if $g_k^{(i)} = 0$. To summarize, the approximated q_k can be computed by

$$q_k^{(i)} = \begin{cases} \frac{(g_{k-1}^{(i)})^2}{g_k^{(i)}}, & \text{if } g_k^{(i)} \neq 0; \\ 0, & \text{if } g_k^{(i)} = 0. \end{cases} \tag{38}$$

As we will see in Section 4, this simple way of calculating q_k leads very efficient algorithm.

For a simple illustration of numerical behavior of ANGR1, we again applied ANGR1 with $\tau_1 = 0.85$ and $\tau_2 = 1.3$ to solve problem (10) with $n = 10$. Fig. 2 shows the largest component $|g_k^{(i_1)}|$ of the gradient generated by BB1 and ANGR1 methods against the iteration number, where circle means the

ANGR1 method takes the new stepsize $\tilde{\alpha}_k^{BB2}$ at that iteration. It can be seen that $|g_k^{(i_1)}|$ generated by BB1 method often increases significantly with a much larger value at the iteration where the new stepsize $\tilde{\alpha}_k^{BB2}$ is applied. On the other hand, $|g_k^{(i_1)}|$ generated by the ANGR1 method is often reduced and kept small after the new stepsize $\tilde{\alpha}_k^{BB2}$ is applied. A detail correspondence of i_1 and λ_j is presented in Table 2, where n_j is the total number of $i_1's$ for which $i_1 = j$, $j = 1, \ldots, 10$. We can see from the last three columns in Table 2 that ANGR1 is much efficient than BB1 for decreasing those components of g_k corresponding to large eigenvalues. Hence, the undesired behavior of BB1 discussed in the motivation Section 2.1 is greatly eliminated by ANGR1.

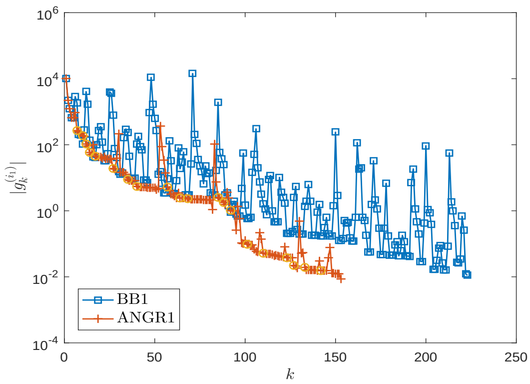

Fig. 2: Problem (10) with $n = 10$: history of $|g_k^{(i_1)}|$ generated by the BB1 and ANGR1 methods

Table 2: The correspondence of i_1 and λ_j by the BB1 and ANGR1 methods

n_j	1	2	3	4	5	6	7	8	9	10
BB1	40	5	6	4	6	2	10	21	42	87
ANGR1	51	4	22	18	16	4	1	8	12	17

3.2 Bound constrained case

In this subsection, we would like to extend ANGR1 and ANGR2 methods for solving the bound constrained optimization

$$\min_{x \in \Omega} \quad f(x), \tag{39}$$

where f is Lipschitz continuously differentiable on the feasible set $\Omega = \{x \in \mathbb{R}^n |\ l \le x \le u\}$. Here, $l \le x \le u$ means componentwise $l_i \le x_i \le u_i$ for all $i = 1, \ldots, n$. Clearly, when $l_i = -\infty$ and $u_i = +\infty$ for all i, problem (39) reduces to the smooth unconstrained optimization.

Our methods will incorporate the gradient projection strategy and update the iterates as

$$x_{k+1} = x_k + \lambda_k d_k,$$

with λ_k being a step length determined by some line searches and d_k being the search direction given by

$$d_k = P_\Omega(x_k - \alpha_k g_k) - x_k, \tag{40}$$

where $P_\Omega(\cdot)$ is the Euclidean projection onto Ω and α_k is our proposed stepsize.

It is well-known that the components of iterates generated by gradient descent methods corresponding to optimal solutions at the boundary will be finally unchanged when the problem is nondegenerate. Hence, in [25], the authors suggest to use the following modified BB stepsizes for bound constrained optimization

$$\bar{\alpha}_k^{BB1} = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T \bar{y}_{k-1}} \quad \text{and} \quad \bar{\alpha}_k^{BB2} = \frac{s_{k-1}^T \bar{y}_{k-1}}{\bar{y}_{k-1}^T \bar{y}_{k-1}}, \tag{41}$$

where $\bar{y}_{k-1}$ is given by

$$\bar{y}_{k-1}^{(i)} = \begin{cases} 0, & \text{if } s_{k-1}^{(i)} = 0; \\ g_k^{(i)} - g_{k-1}^{(i)}, & \text{otherwise.} \end{cases} \tag{42}$$

Notice that $\alpha_k^{BB1} = \bar{\alpha}_k^{BB1}$. We will also do this modifications for solving bound constrained optimization and replace the two BB stepsizes in our new methods by $\bar{\alpha}_k^{BB1}$ and $\bar{\alpha}_k^{BB2}$.

As mentioned before, we expect to get short steps using our new stepsizes. Since (32) may not hold for general functions, we would impose $\bar{\alpha}_k^{BB2}$ as a safeguard. As a result, our ANGR1 and ANGR2 methods for solving bound constrained optimization employ the following stepsizes:

$$\bar{\alpha}_k = \begin{cases} \min\{\bar{\alpha}_{k-1}^{BB2}, \bar{\alpha}_k^{BB2}\}, & \text{if } \bar{\alpha}_k^{BB2} < \tau_1 \bar{\alpha}_k^{BB1} \text{ and } \|\bar{g}_{k-1}\| < \tau_2 \|\bar{g}_k\|; \\ \min\{\bar{\alpha}_k^{BB2}, \tilde{\alpha}_{k-1}^{BB2}\}, & \text{if } \bar{\alpha}_k^{BB2} < \tau_1 \bar{\alpha}_k^{BB1} \text{ and } \|\bar{g}_{k-1}\| \ge \tau_2 \|\bar{g}_k\|; \\ \bar{\alpha}_k^{BB1}, & \text{otherwise,} \end{cases} \tag{43}$$

and

$$\bar{\alpha}_k = \begin{cases} \min\{\bar{\alpha}_{k-1}^{BB2}, \bar{\alpha}_k^{BB2}\}, & \text{if } \bar{\alpha}_k^{BB2} < \tau_1 \bar{\alpha}_k^{BB1} \text{ and } \|\bar{g}_{k-1}\| < \tau_2 \|\bar{g}_k\|; \\ \min\{\bar{\alpha}_k^{BB2}, \hat{\alpha}_{k-2}\}, & \text{if } \bar{\alpha}_k^{BB2} < \tau_1 \bar{\alpha}_k^{BB1} \text{ and } \|\bar{g}_{k-1}\| \ge \tau_2 \|\bar{g}_k\|; \\ \bar{\alpha}_k^{BB1}, & \text{otherwise,} \end{cases} \tag{44}$$

respectively, where $\tau_1 \in (0, 1)$, $\tau_2 \ge 1$, and $\bar{g}_k = P_\Omega(x_k - g_k) - x_k$.

The overall algorithm of ANGR1 and ANGR2 for solving bound constrained optimization (39) are given in Algorithm 1, where the adaptive nonmonotone line search by Dai and Zhang [13] is employed to ensure global convergence and achieve better numerical performance. In particular, the step length $\lambda_k = 1$ is accepted if

$$f(x_k + d_k) \le f_r + \sigma g_k^T d_k, \tag{45}$$

where f_r is the so-called reference function value and is adaptively updated by the rules given in [13] and $\sigma \in (0, 1)$ is a line search parameter. Once (45) is not accepted, it will perform an Armijo-type back tracking line search to find the step length λ_k such that

$$f(x_k + \lambda_k d_k) \le \min\{f_{\max}, f_r\} + \sigma \lambda_k g_k^T d_k, \tag{46}$$

where $f_{\max}$ is the maximal function value in recent M iterations, i.e.,

$$f_{\max} = \max_{0 \le i \le \min\{k, M-1\}} f(x_{k-i}).$$

This nonmonotone line search is observed specially suitable for BB-type methods [13]. Moreover, under standard assumptions, Algorithm 1 ensures convergence in the sense that $\liminf_{k\to\infty} \|\bar{g}_k\| = 0$, see [24].

Algorithm 1: Adaptive nonmonotone gradient method with retard steps

Input: $x_0 \in \mathbb{R}^n$, $\epsilon, \sigma \in (0, 1)$, $\tau > 0$, $M \in \mathbb{N}$, $0 < \alpha_{\min} \le \alpha_{\max}$, $\alpha_0 \in [\alpha_{\min}, \alpha_{\max}]$, $k := 0$.

while $\|\bar{g}_k\| > \epsilon$ **do**
  Compute the search direction d_k by (40);
  Determine λ_k by Dai-Zhang nonmonotone line search;
  $x_{k+1} = x_k + \lambda_k d_k$;
  Compute $s_k = x_{k+1} - x_k$ and $\bar{y}_k$ by (42);
  if $s_k^T \bar{y}_k > 0$ **then**
    Compute stepsize $\bar{\alpha}_{k+1}$ by (43) or (44);
    $\alpha_{k+1} = \max\{\alpha_{\min}, \min\{\bar{\alpha}_{k+1}, \alpha_{\max}\}\}$;
  else
    $\alpha_{k+1} = 1/\|\bar{g}_{k+1}\|_\infty$;
  end
  $k := k + 1$;
end

4 Numerical results

In this section, we present numerical comparisons of ANGM, ANGR1, ANGR2 with some recent very successful gradient descent methods on solving quadratic, general unconstrained and bound constrained problems. All the comparison methods were implemented in Matlab (v.9.0-R2016a) and run on a laptop with an Intel Core i7, 2.9 GHz processor and 8 GB of RAM running Windows 10 system.

4.1 Quadratic problems

In this subsection, we compare ANGM, ANGR1 and ANGR2 with the BB1 [3], DY [12], ABBmin2 [20], and SDC [14] methods on solving quadratic optimization problems.

We first solve some randomly generated quadratic problems from [34]. Particularly, we solve

$$\min_{x\in\mathbb{R}^n} f(x) = (x - x^*)^T V (x - x^*), \tag{47}$$

where x^* is randomly generated with components between -10 and 10, $V = \mathrm{diag}\{v_1, \ldots, v_n\}$ is a diagonal matrix with $v_1 = 1$ and $v_n = \kappa$, and v_j, $j = 2, \ldots, n-1$, are generated by the *rand* function between 1 and κ.

Table 3: Distributions of v_j

Problem	Spectrum
1	$\{v_2, \ldots, v_{n-1}\} \subset (1, \kappa)$
2	$\{v_2, \ldots, v_{n/5}\} \subset (1, 100)$ $\{v_{n/5+1}, \ldots, v_{n-1}\} \subset (\frac{\kappa}{2}, \kappa)$
3	$\{v_2, \ldots, v_{n/2}\} \subset (1, 100)$ $\{v_{n/2+1}, \ldots, v_{n-1}\} \subset (\frac{\kappa}{2}, \kappa)$
4	$\{v_2, \ldots, v_{4n/5}\} \subset (1, 100)$ $\{v_{4n/5+1}, \ldots, v_{n-1}\} \subset (\frac{\kappa}{2}, \kappa)$
5	$\{v_2, \ldots, v_{n/5}\} \subset (1, 100)$ $\{v_{n/5+1}, \ldots, v_{4n/5}\} \subset (100, \frac{\kappa}{2})$ $\{v_{4n/5+1}, \ldots, v_{n-1}\} \subset (\frac{\kappa}{2}, \kappa)$

We have tested five sets of problems (47) with $n = 1,000$ using different spectral distributions of the Hessian listed in Table 3. The algorithm is stopped once the number of iteration exceeds 20,000 or the gradient norm is reduced by a factor of ϵ, which is set to $10^{-6}, 10^{-9}$ and 10^{-12}, respectively. Three different condition numbers $\kappa = 10^4, 10^5$ and 10^6 are tested. For each value of κ or ϵ, 10 instances of the problem are randomly generated and the averaged results obtained by the starting point $x_0 = (0, \ldots, 0)^T$ are presented. For the ABBmin2 method, τ is set to 0.9 as suggested in [20]. The parameter pair (h, s) of the SDC method is set to $(8, 6)$ which is more efficient than other choices for this test. The q_k is calculated by (38) for our methods.

We compared the algorithms by using the performance profiles of Dolan and Moré [16] on iteration metric. In these performance profiles, the vertical

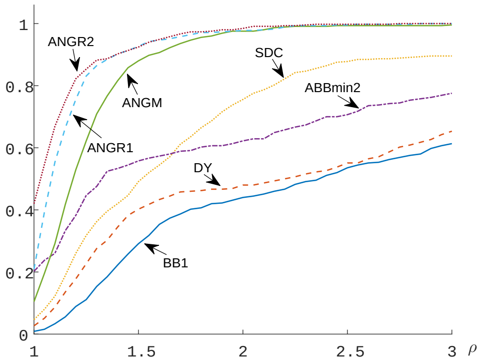

Fig. 3: Performance profiles of compared methods on solving random quadratic problems (47) with spectral distributions in Table 3, iteration metric

Table 4: The numbers of averaged iterations of the ANGM, BB1, DY, ABBmin2 and SDC methods on solving quadratic problems (47) with spectral distributions in Table 3

set	ϵ	τ_1 for ANGM									BB1	DY	ABBmin2	SDC
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9				
1	1e-06	200.6	198.6	213.3	193.6	204.1	196.5	211.1	219.0	230.3	221.0	197.1	**177.1**	183.2
	1e-09	665.0	678.1	677.5	937.3	718.7	768.5	1259.4	1198.1	1195.3	2590.0	2672.7	**428.8**	2029.6
	1e-12	939.7	941.6	901.5	1159.2	951.2	1043.7	1464.3	1339.0	1411.9	6032.9	6353.9	**560.3**	4087.2
2	1e-06	140.3	**140.2**	143.1	161.2	172.8	171.1	217.3	265.8	310.3	311.2	261.5	302.5	160.6
	1e-09	**546.1**	590.5	641.1	779.5	850.5	891.9	1055.7	1161.3	1299.0	1665.4	1340.0	1321.1	735.1
	1e-12	**895.1**	1025.2	1098.1	1328.6	1446.6	1598.2	1739.6	1908.7	2107.0	2820.8	2434.6	2267.5	1346.8
3	1e-06	**163.2**	170.1	177.3	193.3	216.7	231.1	283.3	318.4	363.9	388.4	329.4	356.3	235.5
	1e-09	**566.5**	640.0	680.7	811.0	970.8	986.0	1188.3	1218.7	1364.4	1783.1	1511.8	1470.8	818.1
	1e-12	**928.4**	1030.0	1163.7	1412.0	1575.2	1733.2	1973.6	2075.2	2191.1	2977.7	2780.2	2288.4	1310.6
4	1e-06	**212.3**	213.7	237.1	259.0	254.7	291.8	365.3	431.4	475.1	500.5	431.3	519.0	262.8
	1e-09	**616.1**	655.7	759.7	885.5	956.4	1107.9	1232.4	1405.3	1533.3	1859.4	1659.9	1489.5	805.5
	1e-12	**996.0**	1078.4	1250.9	1452.1	1629.5	1786.8	2041.6	2179.0	2427.6	3051.5	2785.4	2383.9	1469.3
5	1e-06	**623.1**	654.8	663.4	671.0	683.4	697.2	761.3	813.7	931.0	832.5	650.8	816.0	668.3
	1e-09	**2603.0**	2654.7	2847.6	2901.4	2936.9	3161.7	3228.6	3306.4	3807.2	4497.2	3185.5	2929.7	3274.5
	1e-12	**4622.7**	4675.2	4905.9	4634.2	4818.7	4944.7	5224.0	5480.7	5972.1	7446.7	7024.1	4808.7	5816.6
total	1e-06	**1339.5**	1377.5	1434.2	1478.1	1531.7	1587.7	1838.3	2048.4	2310.6	2253.7	1870.1	2170.9	1510.4
	1e-09	**4996.7**	5219.0	5606.6	6314.7	6433.3	6916.0	7964.4	8289.7	9199.2	12395.0	10370.0	7640.0	7662.7
	1e-12	**8382.0**	8750.3	9320.1	9986.1	10421.3	11106.5	12443.0	12982.6	14109.7	22329.5	21378.3	12308.9	14030.4

axis shows the percentage of the problems the method solves within the factor ρ of the metric used by the most effective method in this comparison. Fig. 3 shows the performance profiles of ANGM, ANGR1 and ANGR2 obtained by setting $\tau_1 = 0.1$, $\tau_2 = 1$ and other four compared methods. Clearly, ANGR2 outperforms all other methods. In general, we can see ANGM, ANGR1 and ANGR2 are much better than the BB1, DY and SDC methods.

To further analyze the performance of our methods, we present results of them with different values of τ_1 from 0.1 to 0.9 in Tables 4, 5 and 6. From Table 4, we can see that, for the first problem set, ANGM is much faster than the BB1, DY and SDC methods, and competitive with the ABBmin2 method.

Table 5: The numbers of averaged iterations of the ANGR1, BB1, DY, ABB-min2 and SDC methods on problems in Table 3

set	ϵ	τ_1 for ANGR1									BB1	DY	ABBmin2	SDC
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9				
1	1e-06	198.7	185.8	188.1	182.5	189.2	188.2	184.5	200.1	214.1	221.0	197.1	**177.1**	183.2
	1e-09	655.5	641.6	625.4	667.9	621.7	680.9	762.7	811.5	1012.2	2590.0	2672.7	**428.8**	2029.6
	1e-12	897.3	854.7	844.5	838.4	775.2	813.1	917.4	984.7	1172.7	6032.9	6353.9	**560.3**	4087.2
2	1e-06	137.9	119.0	119.2	118.7	**115.3**	124.5	152.4	157.5	178.9	311.2	261.5	302.5	160.6
	1e-09	**466.5**	470.9	487.2	512.0	540.2	589.9	672.6	691.0	722.3	1665.4	1340.0	1321.1	735.1
	1e-12	788.1	**783.3**	802.1	835.3	908.1	1004.5	1074.2	1110.3	1215.2	2820.8	2434.6	2267.5	1346.8
3	1e-06	164.5	149.6	**146.3**	153.0	157.8	166.8	183.3	208.9	229.1	388.4	329.4	356.3	235.5
	1e-09	507.5	**473.0**	516.6	541.2	563.3	609.0	682.5	747.2	850.5	1783.1	1511.8	1470.8	818.1
	1e-12	818.3	**801.5**	865.4	894.4	965.4	1046.0	1108.8	1200.2	1419.8	2977.7	2780.2	2288.4	1310.6
4	1e-06	189.7	**172.2**	168.8	179.6	195.2	212.7	229.4	262.9	282.1	500.5	431.3	519.0	262.8
	1e-09	539.0	**519.2**	558.7	570.2	617.1	689.6	786.3	853.1	901.8	1859.4	1659.9	1489.5	805.5
	1e-12	**849.0**	851.1	894.2	934.8	1011.8	1065.5	1197.3	1291.8	1408.0	3051.5	2785.4	2383.9	1469.3
5	1e-06	625.7	587.6	600.1	**583.1**	614.2	632.2	675.8	726.9	769.0	832.5	650.8	816.0	668.3
	1e-09	2539.9	**2518.1**	2612.7	2530.4	2636.0	2581.2	2773.6	2875.2	3026.2	4497.2	3185.5	2929.7	3274.5
	1e-12	**4207.3**	4238.2	4292.4	4256.2	4240.7	4285.1	4397.2	4534.8	4729.0	7446.7	7024.1	4808.7	5816.6
total	1e-06	1316.5	**1214.1**	1222.6	1217.0	1271.8	1324.5	1425.3	1556.4	1673.1	2253.7	1870.1	2170.9	1510.4
	1e-09	4708.5	**4622.7**	4800.7	4821.7	4978.4	5150.6	5677.7	5977.9	6512.9	12395.0	10370.0	7640.0	7662.7
	1e-12	7560.1	**7528.8**	7698.7	7759.1	7901.3	8214.1	8694.9	9121.9	9944.7	22329.5	21378.3	12308.9	14030.4

As for the other four problem sets, ANGM method with a small value of τ_1 outperforms the other compared methods though its performance seems to become worse as τ_1 increases. The results shown in Tables 5 and 6 are slightly different from those in Table 4. In particular, ANGR1 and ANGR2 outperform other four compared methods for most of the problem sets and values of τ_1. For each given τ_1 and tolerance level, ANGR1 and ANGR2 always perform better than other methods in terms of total number of iterations.

Table 6: The numbers of averaged iterations of the ANGR2, BB1, DY, ABB-min2 and SDC methods on problems in Table 3

set	ϵ	τ_1 for ANGR2									BB1	DY	ABBmin2	SDC
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9				
1	1e-06	196.9	191.8	184.8	186.8	183.3	189.0	**175.6**	182.2	189.5	221.0	197.1	177.1	183.2
	1e-09	591.0	607.8	599.4	613.3	669.5	699.6	704.0	763.6	1005.1	2590.0	2672.7	**428.8**	2029.6
	1e-12	870.4	811.3	763.1	790.7	800.0	881.4	844.8	971.0	1146.5	6032.9	6353.9	**560.3**	4087.2
2	1e-06	129.7	117.6	115.0	**111.9**	114.6	116.3	140.6	146.9	164.4	311.2	261.5	302.5	160.6
	1e-09	**455.4**	474.8	470.8	491.4	530.6	566.0	595.9	601.8	691.5	1665.4	1340.0	1321.1	735.1
	1e-12	786.0	**750.5**	811.7	819.3	938.1	908.4	1000.3	1008.5	1089.0	2820.8	2434.6	2267.5	1346.8
3	1e-06	156.2	141.2	**136.3**	149.7	146.7	167.9	179.9	188.0	206.8	388.4	329.4	356.3	235.5
	1e-09	495.8	**465.6**	490.4	529.0	530.8	606.4	667.6	718.5	726.6	1783.1	1511.8	1470.8	818.1
	1e-12	809.0	**775.1**	796.4	876.6	918.8	997.0	1080.8	1130.1	1157.8	2977.7	2780.2	2288.4	1310.6
4	1e-06	183.2	168.0	**164.9**	166.9	180.9	195.7	219.8	234.2	225.3	500.5	431.3	519.0	262.8
	1e-09	**516.2**	521.9	523.2	541.6	603.9	637.3	700.0	709.8	739.2	1859.4	1659.9	1489.5	805.5
	1e-12	842.6	**830.1**	845.8	899.5	959.4	1039.2	1099.3	1133.9	1192.2	3051.5	2785.4	2383.9	1469.3
5	1e-06	611.7	**580.5**	605.4	594.8	586.2	605.1	659.3	644.1	653.1	832.5	650.8	816.0	668.3
	1e-09	2472.7	**2394.4**	2510.8	2504.4	2447.2	2502.1	2551.9	2648.6	2763.9	4497.2	3185.5	2929.7	3274.5
	1e-12	4122.4	4108.7	**4103.2**	4107.2	4161.8	4227.9	4363.6	4324.0	4483.7	7446.7	7024.1	4808.7	5816.6
total	1e-06	1277.8	**1199.1**	1206.3	1210.1	1211.6	1274.0	1375.2	1395.4	1439.1	2253.7	1870.1	2170.9	1510.4
	1e-09	4531.1	**4464.5**	4594.7	4679.7	4781.9	5011.5	5219.4	5442.3	5926.3	12395.0	10370.0	7640.0	7662.7
	1e-12	7430.3	**7275.7**	7320.2	7493.4	7778.1	8053.8	8388.8	8567.6	9069.2	22329.5	21378.3	12308.9	14030.4

Secondly, we compared the methods on solving the non-rand quadratic problem (10) with $n = 10,000$. For ANGM, ANGR1 and ANGR2, τ_1 and τ_2 were set to 0.4 and 1, respectively. The parameter pair (h, s) used for the S-

DC method was set to $(30, 2)$. Other settings are the same as above. Table 7 presents averaged number of iterations over 10 different starting points with entries randomly generated in $[-10, 10]$. It can be seen that ANGM, ANGR1 and ANGR2 are significantly better than the BB1 and DY methods. In addition, ANGR1 and ANGR2 often outperform the ABBmin2 and SDC methods while ANGM is very competitive with them.

Table 7: The numbers of iterations of the compared methods on problem (10) with $n = 10,000$

κ	ϵ	BB1	DY	ABBmin2	SDC	ANGM	ANGR1	ANGR2
	1e-06	626.8	527.7	513.0	597.1	539.1	**500.6**	512.1
10^4	1e-09	1267.0	972.1	894.5	1000.6	976.7	893.7	**890.2**
	1e-12	1741.9	1396.8	1277.8	1409.2	1399.1	1298.0	**1257.4**
	1e-06	1597.8	1326.7	1266.3	1254.3	1209.7	**1046.0**	1127.9
10^5	1e-09	3687.5	3168.3	2559.8	2647.4	2605.2	2424.3	**2399.8**
	1e-12	5564.8	4892.4	3895.0	4156.4	4139.0	3858.5	**3663.3**
	1e-06	4060.9	2159.4	3130.2	1986.2	2112.9	1992.0	**1936.0**
10^6	1e-09	10720.4	10134.3	7560.8	7178.5	7381.1	**6495.1**	6550.1
	1e-12	17805.5	18015.6	12193.6	11646.7	11922.9	10364.9	**10280.2**
	1e-06	6285.5	4013.8	4909.5	3837.6	3861.7	**3538.6**	3576.0
total	1e-09	15674.9	14274.7	11015.1	10826.5	10963.0	**9813.1**	9840.1
	1e-12	25112.2	24304.8	17366.4	17212.3	17461.0	15521.4	**15200.9**

Finally, we compared the methods on solving two large-scale real problems Laplace1(a) and Laplace1(b) described in [17]. The two problems require the solution of a system of linear equations derived from a 3D Laplacian on a box, discretized using a standard 7-point finite difference stencil. Each problem has $n = N^3$ variables with N being the number of interior nodes taken in each coordinate direction. The solution is fixed by a Gaussian function centered at (α, β, γ) and multiplied by $x(x-1)y(y-1)z(z-1)$. The parameter σ is used to control the rate of decay of the Gaussian. See [17] for more details on these problems. Here, we set the parameters as follows:

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

The null vector was used as the starting point. We again stop the iteration when $\|g_k\| \leq \epsilon\|g_0\|$ with different values of ϵ.

For ANGM, ANGR1 and ANGR2, τ_1 and τ_2 were set to 0.7 and 1.2, respectively. The parameter pair (h, s) used for the SDC method was chosen for the best performance in our test, i.e., $(2, 6)$ and $(8, 6)$ for Laplace1(a) and Laplace1(b), respectively. Other settings are the same as above. The number of iterations required by the compared methods for solving the two problems are listed in Table 8. It can be seen that our methods are significantly better than the BB1, DY and SDC methods and is often faster than ABBmin2 especially when a tight tolerance is used.

Table 8: The numbers of iterations of BB1, DY, ABBmin2, SDC, ANGM, ANGR1 and ANGR2 on solving the 3D Laplacian problem

n	ϵ	BB1	DY	ABBmin2	SDC	ANGM	ANGR1	ANGR2
Laplace1(a)								
60^3	1e-06	259	249	**192**	213	245	195	233
	1e-09	441	373	329	393	313	322	**308**
	1e-12	680	546	401	529	367	373	**364**
80^3	1e-06	359	383	289	297	291	332	**288**
	1e-09	591	570	430	553	408	446	**396**
	1e-12	882	789	608	705	620	**516**	591
100^3	1e-06	950	427	351	513	450	**303**	416
	1e-09	1088	651	485	609	584	519	**503**
	1e-12	1241	918	687	825	694	604	**597**
total	1e-06	1568	1059	832	1023	986	**830**	937
	1e-09	2120	1594	1244	1555	1305	1287	**1207**
	1e-12	2803	2253	1696	2059	1681	**1493**	1552
Laplace1(b)								
60^3	1e-06	246	236	217	**213**	242	217	214
	1e-09	473	399	365	437	**333**	338	409
	1e-12	651	532	502	555	**451**	478	573
80^3	1e-06	288	454	294	309	296	**290**	324
	1e-09	607	567	**433**	485	517	499	495
	1e-12	739	794	634	766	686	**590**	645
100^3	1e-06	544	371	369	379	381	406	**358**
	1e-09	646	700	585	653	638	**558**	648
	1e-12	937	1038	880	965	854	**785**	810
total	1e-06	1078	1061	**880**	901	919	913	896
	1e-09	1726	1666	**1383**	1575	1488	1395	1552
	1e-12	2327	2364	2016	2286	1991	**1853**	2028

4.2 Unconstrained problems

Now we present comparisons of ANGR1 and ANGR2 with SPG method[1] in [4,5], and the BB1 method using Dai-Zhang nonmonotone line search [13] (BB1-DZ) on solving general unconstrained problems.

For our methods, the parameter values are set as the following:

$$\alpha_{\min} = 10^{-30},\ \alpha_{\max} = 10^{30},\ M = 8,\ \sigma = 10^{-4},\ \alpha_0 = 1/\|g_0\|_\infty.$$

[1] codes available at `https://www.ime.usp.br/~egbirgin/tango/codes.php`

In addition, the parameter τ_1 and τ_2 for ANGR1 and ANGR2 are set to 0.8 and 1.2, respectively. Default parameters were used for SPG. Each method was stopped if the number of iteration exceeds 200,000 or $\|g_k\|_\infty \leq 10^{-6}$.

Our test problems were taken from [2]. We have tested 59 problems listed in Table 9 with $n = 1,000$ and the performance profiles are shown in Fig. 4, which shows that ANGR1 and ANGR2 outperform SPG and BB1-DZ in terms of the iteration number, and ANGR2 is faster than ANGR1. Moreover, BB1-DZ is slightly better than SPG. Detail numerical results are also presented in Table 10. Since the only difference between BB1-DZ with ANGR1 and ANGR2 lies in the choice of stepsizes, these numerical results show our adaptive choices of stepsizes in ANGR1 and ANGR2 are very effective and can indeed greatly accelerate the convergence of BB-type methods.

Table 9: Test problems from [2]

Problem	name	Problem	name
1	Extended Freudenstein & Roth function	31	NONDIA
2	Extended Trigonometric	32	DQDRTIC
3	Extended White & Holst	33	Partial Perturbed Quadratic
4	Extended Beale	34	Broyden Tridiagonal
5	Extended Penalty	35	Almost Perturbed Quadratic
6	Perturbed Quadratic	36	Perturbed Tridiagonal Quadratic
7	Raydan 1	37	Staircase 1
8	Raydan 2	38	Staircase 2
9	Diagonal 1	39	LIARWHD
10	Diagonal 2	40	POWER
11	Diagonal 3	41	ENGVAL1
12	Hager	42	EDENSCH
13	Generalized Tridiagonal 1	43	BDEXP
14	Extended Tridiagonal 1	44	GENHUMPS
15	Extended TET	45	NONSCOMP
16	Generalized Tridiagonal 2	46	VARDIM
17	Diagonal 4	47	QUARTC
18	Diagonal 5	48	Extended DENSCHNB
19	Extended Himmelblau	49	Extended DENSCHNF
20	Extended PSC1	50	LIARWHD
21	Generalized PSC1	51	BIGGSB1
22	Extended Powell	52	Generalized Quartic
23	Extended Cliff	53	Diagonal 7
24	Perturbed quadratic diagonal	54	Diagonal 8
25	Quadratic QF1	55	Full Hessian FH3
26	Extended quadratic exponential EP1	56	SINCOS
27	Extended Tridiagonal 2	57	Diagonal 9
28	BDQRTIC	58	HIMMELBG
29	TRIDIA	59	HIMMELH
30	ARWHEAD		

4.3 Bound constrained problems

We further compare ANGR1 and ANGR2, with SPG [4,5] and the BB1-DZ method [13] combined with gradient projection techniques on solving bound constrained problems from the CUTEst collection [23] with dimension more than 50. We deleted 3 problems from this list since none of these comparison algorithms can solve them. Hence, in total there are 47 problems left in our test. The iteration was stopped if the number of iteration exceeds 200,000 or

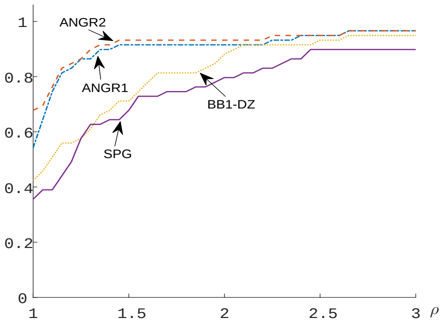

Fig. 4: Performance profiles of compared methods on 59 unconstrained problems in Table 9, iteration metric

$\|P_{\Omega}(x_k - g_k) - x_k\|_\infty \leq 10^{-6}$. The parameters τ_1 and τ_2 for ANGR1 and ANGR2 are set to 0.4 and 1.5, respectively. Other settings are the same as before. Fig. 5 shows the performance profiles of all the compared methods on iteration metric. Similar as the unconstrained case, from Fig. 5, we again see that both ANGR1 and ANGR2 perform significantly better than SPG and BB1-DZ. Hence, our new gradient methods also have potential great benefits for solving constrained optimization.

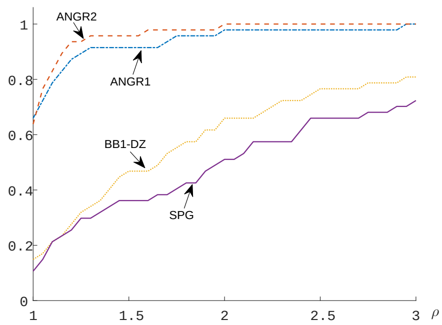

Fig. 5: Performance profiles of compared methods on solving 47 bound constrained problems from CUTEst, iteration metric

Table 10: Results of compared methods on unconstrained problems in Table 9

Problem	SPG			BB1-DZ			ANGR1			ANGR2		
	iter	f_k	$\|g_k\|_\infty$	iter	f_k	$\|g_k\|_\infty$	iter	f_k	$\|g_k\|_\infty$	iter	f_k	$\|g_k\|_\infty$
1	87	2.45e+04	8.95e-07	41	2.45e+04	4.06e-08	26	2.45e+04	1.46e-07	25	2.45e+04	1.46e-07
2	45	5.72e-13	3.83e-07	87	5.26e-07	3.97e-07	100	4.89e-07	9.32e-07	100	4.73e-07	2.20e-07
3	110	5.80e-14	3.05e-08	120	6.30e-20	4.95e-11	91	9.95e-12	2.93e-07	88	1.05e-11	2.90e-07
4	46	3.92e-10	6.57e-07	65	8.28e-18	3.74e-10	31	9.52e-15	4.15e-08	31	1.94e-12	6.12e-08
5	41	8.83e+02	9.02e-07	107	8.83e+02	1.24e-08	107	8.83e+02	1.24e-08	107	8.83e+02	1.24e-08
6	597	2.39e-13	9.83e-07	457	2.16e-13	9.24e-07	300	7.98e-16	5.66e-08	287	1.69e-13	8.03e-07
7	465	5.01e+04	9.64e-07	339	5.01e+04	9.04e-07	330	5.01e+04	5.56e-08	301	5.01e+04	8.39e-07
8	1	1.00e+03	0.00e+00	1	1.00e+03	0.00e+00	1	1.00e+03	0.00e+00	1	1.00e+03	0.00e+00
9	415	-2.71e+06	8.50e-07	361	-2.71e+06	5.43e-07	294	-2.71e+06	9.05e-07	276	-2.71e+06	7.04e-07
10	168	3.13e+01	6.05e-07	173	3.13e+01	5.68e-07	135	3.13e+01	5.93e-07	131	3.13e+01	8.61e-07
11	668	-4.96e+05	8.82e-07	404	-4.96e+05	8.63e-07	336	-4.96e+05	8.85e-07	286	-4.96e+05	9.93e-07
12	63	-4.47e+04	2.91e-07	58	-4.47e+04	5.70e-07	52	-4.47e+04	5.14e-07	52	-4.47e+04	5.59e-07
13	29	9.97e+02	6.53e-07	27	9.97e+02	9.67e-07	26	9.97e+02	4.21e-07	25	9.97e+02	9.79e-07
14	30	2.13e-07	3.75e-07	18	2.68e-07	4.46e-07	20	5.20e-07	7.33e-07	20	5.20e-07	7.33e-07
15	10	1.28e+03	4.34e-10	5	1.28e+03	1.33e-08	5	1.28e+03	1.33e-08	5	1.28e+03	1.33e-08
16	69	2.38e+00	8.58e-07	58	9.58e-01	7.14e-07	56	9.58e-01	8.40e-07	57	9.58e-01	6.13e-07
17	3	0.00e+00	0.00e+00	3	0.00e+00	0.00e+00	3	0.00e+00	0.00e+00	3	0.00e+00	0.00e+00
18	4	6.93e+02	4.81e-08	1	6.93e+02	0.00e+00	1	6.93e+02	0.00e+00	1	6.93e+02	0.00e+00
19	15	4.21e-19	1.92e-10	15	4.21e-19	1.92e-10	15	4.21e-19	1.92e-10	15	4.21e-19	1.92e-10
20	23	9.99e+02	8.27e-07	22	9.99e+02	8.65e-07	22	9.99e+02	8.65e-07	22	9.99e+02	8.65e-07
21	15	3.87e+02	5.87e-07	13	3.87e+02	3.78e-07	13	3.87e+02	3.78e-07	13	3.87e+02	3.78e-07
22	246	3.19e-07	6.80e-07	266	3.67e-07	5.97e-07	164	7.08e-07	9.62e-07	205	5.55e-07	9.82e-07
23	154	9.99e+01	7.79e-07	126	9.99e+01	1.57e-07	541	9.99e+01	2.61e-07	539	9.99e+01	6.10e-08
24	373	2.41e-11	8.95e-07	246	1.41e-11	6.83e-07	249	2.05e-12	2.98e-07	314	4.05e-12	4.14e-07
25	612	-5.00e-04	9.45e-07	575	-5.00e-04	9.95e-07	319	-5.00e-04	8.92e-08	282	-5.00e-04	7.45e-07
26	4	7.93e+03	4.19e-08	4	7.93e+03	3.71e-09	4	7.93e+03	3.71e-09	4	7.93e+03	3.71e-09
27	34	3.89e+02	7.50e-07	39	3.89e+02	9.74e-07	36	3.89e+02	8.49e-07	36	3.89e+02	5.15e-07
28	77	3.98e+03	7.64e-07	98	3.98e+03	8.91e-07	69	3.98e+03	1.49e-07	64	3.98e+03	9.57e-07
29	3910	2.47e-13	6.97e-07	2403	9.93e-15	6.71e-07	816	9.62e-16	4.67e-07	611	6.71e-14	5.86e-07
30	4	0.00e+00	1.50e-09	4	0.00e+00	1.50e-09	4	0.00e+00	1.50e-09	4	0.00e+00	1.50e-09
31	15	1.35e-14	9.86e-09	15	1.35e-14	9.86e-09	16	1.10e-12	8.20e-08	16	1.10e-12	8.20e-08
32	41	4.20e-16	2.86e-07	26	1.80e-13	8.47e-07	17	2.99e-16	2.14e-07	17	2.99e-16	2.14e-07
33	312	2.51e-14	5.96e-07	229	1.42e-13	9.63e-07	257	6.00e-15	2.16e-07	294	7.03e-14	8.24e-07
34	47	5.31e-15	3.20e-07	51	7.94e-14	8.28e-07	41	7.67e-14	7.35e-07	47	2.62e-14	7.66e-07
35	621	2.36e-13	9.76e-07	363	2.47e-13	9.97e-07	358	1.53e-13	7.77e-07	272	4.53e-15	6.22e-07
36	595	2.49e-13	9.97e-07	606	2.48e-13	9.95e-07	368	3.46e-13	1.00e-06	306	1.82e-13	8.61e-07
37	15936	1.81e-12	7.80e-07	10239	1.54e-12	7.68e-07	7234	1.39e-13	1.89e-07	6543	1.07e-12	9.13e-07
38	28528	1.97e-14	2.43e-07	25640	2.74e-12	9.97e-07	60283	2.86e-12	9.93e-07	27783	2.98e-12	9.74e-07
39	61	2.64e-14	2.06e-08	59	2.62e-13	4.93e-07	47	9.15e-14	6.15e-08	47	1.30e-12	2.36e-07
40	5435	2.84e-13	9.98e-07	5472	2.94e-13	9.98e-07	327	1.89e-13	6.99e-07	366	5.63e-13	9.98e-07
41	33	1.11e+03	8.58e-07	31	1.11e+03	8.09e-07	29	1.11e+03	1.70e-07	29	1.11e+03	1.70e-07
42	32	6.00e+03	7.33e-07	32	6.00e+03	7.33e-07	31	6.00e+03	4.46e-07	31	6.00e+03	9.35e-07
43	15	6.52e-05	9.93e-07	15	6.52e-05	9.93e-07	15	6.52e-05	9.93e-07	15	6.52e-05	9.93e-07
44	747	1.39e-17	1.50e-09	3	4.49e-23	3.00e-12	3	4.49e-23	3.00e-12	3	4.49e-23	3.00e-12
45	48	9.98e-12	8.75e-07	89	1.97e-14	2.57e-07	59	2.30e-13	5.85e-07	63	2.18e-13	7.89e-07
46	1	4.62e-30	4.44e-16	30	2.17e-27	5.77e-15	30	2.17e-27	5.77e-15	30	2.17e-27	5.77e-15
47	1	0.00e+00	0.00e+00	1	0.00e+00	0.00e+00	1	0.00e+00	0.00e+00	1	0.00e+00	0.00e+00
48	8	1.37e-11	4.68e-07	8	1.37e-11	4.68e-07	8	1.37e-11	4.68e-07	8	1.37e-11	4.68e-07
49	12	1.78e-19	4.76e-10	17	6.46e-21	5.91e-11	17	6.46e-21	5.91e-11	17	6.46e-21	5.91e-11
50	61	2.64e-14	2.06e-08	59	2.62e-13	4.93e-07	47	9.15e-14	6.15e-08	47	1.30e-12	2.36e-07
51	30459	1.24e-05	9.95e-07	14877	3.73e-07	6.46e-07	7558	6.51e-07	8.62e-07	8969	6.08e-12	6.45e-07
52	9	4.98e-15	1.99e-07	9	4.98e-15	1.99e-07	9	4.98e-15	1.99e-07	9	4.98e-15	1.99e-07
53	7	-8.17e+02	1.97e-09	7	-8.17e+02	1.97e-09	7	-8.17e+02	1.97e-09	7	-8.17e+02	1.97e-09
54	6	-4.80e+02	5.85e-08	6	-4.80e+02	4.36e-10	6	-4.80e+02	4.36e-10	6	-4.80e+02	4.36e-10
55	3	-2.50e-01	4.56e-10	3	-2.50e-01	4.56e-10	3	-2.50e-01	4.56e-10	3	-2.50e-01	4.56e-10
56	15	3.87e+02	5.87e-07	13	3.87e+02	3.78e-07	13	3.87e+02	3.78e-07	13	3.87e+02	3.78e-07
57	553	-2.70e+06	1.91e-07	762	-2.70e+06	6.06e-07	305	-2.70e+06	9.96e-07	346	-2.70e+06	5.60e-07
58	16	4.79e-04	8.77e-07	17	3.78e-04	6.91e-07	17	3.78e-04	6.91e-07	17	3.78e-04	6.91e-07
59	12	-5.00e+02	1.09e-07	12	-5.00e+02	7.41e-11	12	-5.00e+02	7.41e-11	12	-5.00e+02	7.41e-11

5 Conclusions

We have developed techniques to accelerate the Barzilai-Borwein (BB) method motivated from finite termination for minimizing two-dimensional strongly convex quadratic functions. More particularly, by exploiting certain orthogonal properties of the gradients, we derive a new monotone stepsize that can be combined with BB stepsizes to significantly improve their performance for minimizing general strongly convex quadratic functions. By adaptively using this new stepsize and the two BB stepsizes, we develop a new gradient method called ANGM and its two variants ANGR1 and ANGR2, which are further extended for solving unconstrained and bound constrained optimization. Our extensive numerical experiments show that all the new developed methods are significantly better than the BB method and are faster than some very successful gradient methods developed in the recent literature for solving quadratic, general smooth unconstrained and bound constrained optimization.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant Nos. 11631013, 11701137, 11671116), the National 973 Program of China (Grant No. 2015CB856002), China Scholarship Council (No. 201806705007), and USA National Science Foundation (1522654, 1819161).

References

1. Akaike, H.: On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method. Ann. Inst. Stat. Math. **11**(1), 1–16 (1959)
2. Andrei, N.: An unconstrained optimization test functions collection. Adv. Model. Optim. **10**(1), 147–161 (2008)
3. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA J. Numer. Anal. **8**(1), 141–148 (1988)
4. Birgin, E.G., Martínez, J.M., Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. SIAM J. Optim. **10**(4), 1196–1211 (2000)
5. Birgin, E.G., Martínez, J.M., Raydan, M.: Spectral projected gradient methods: review and perspectives. J. Stat. Softw. **60**(3), 539–559 (2014)
6. Cauchy, A.: Méthode générale pour la résolution des systemes déquations simultanées. Comp. Rend. Sci. Paris **25**, 536–538 (1847)
7. Dai, Y.H.: Alternate step gradient method. Optimization **52**(4-5), 395–415 (2003)
8. Dai, Y.H., Fletcher, R.: Projected Barzilai–Borwein methods for large-scale box-constrained quadratic programming. Numer. Math. **100**(1), 21–47 (2005)
9. Dai, Y.H., Huang, Y., Liu, X.W.: A family of spectral gradient methods for optimization. Comp. Optim. Appl. **74**(1), 43–65 (2019)
10. Dai, Y.H., Liao, L.Z.: R-linear convergence of the Barzilai and Borwein gradient method. IMA J. Numer. Anal. **22**(1), 1–10 (2002)
11. Dai, Y.H., Yuan, Y.X.: Alternate minimization gradient method. IMA J. Numer. Anal. **23**(3), 377–393 (2003)
12. Dai, Y.H., Yuan, Y.X.: Analysis of monotone gradient methods. J. Ind. Manag. Optim. **1**(2), 181 (2005)
13. Dai, Y.H., Zhang, H.: Adaptive two-point stepsize gradient algorithm. Numer. Algorithms **27**(4), 377–385 (2001)
14. De Asmundis, R., Di Serafino, D., Hager, W.W., Toraldo, G., Zhang, H.: An efficient gradient method using the yuan steplength. Comp. Optim. Appl. **59**(3), 541–563 (2014)

15. Di Serafino, D., Ruggiero, V., Toraldo, G., Zanni, L.: On the steplength selection in gradient methods for unconstrained optimization. Appl. Math. Comput. **318**, 176–195 (2018)
16. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002)
17. Fletcher, R.: On the Barzilai–Borwein method. In: Qi, L., Teo, K., Yang, X. (eds.) Optimization and Control with Applications. pp. 235–256. Springer, Boston, (2005)
18. Fletcher, R.: A limited memory steepest descent method. Math. Program. **135**(1-2), 413–436 (2012)
19. Forsythe, G.E.: On the asymptotic directions of the s-dimensional optimum gradient method. Numer. Math. **11**(1), 57–76 (1968)
20. Frassoldati, G., Zanni, L., Zanghirati, G.: New adaptive stepsize selections in gradient methods. J. Ind. Manag. Optim. **4**(2), 299 (2008)
21. Friedlander, A., Martínez, J.M., Molina, B., Raydan, M.: Gradient method with retards and generalizations. SIAM J. Numer. Anal. **36**(1), 275–289 (1998)
22. Gonzaga, C.C., Schneider, R.M.: On the steepest descent algorithm for quadratic functions. Comp. Optim. Appl. **63**(2), 523–542 (2016)
23. Gould, N.I., Orban, D., Toint, P.L.: CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. Comp. Optim. Appl. **60**(3), 545–557 (2015)
24. Hager, W.W., Zhang, H.: A new active set algorithm for box constrained optimization. SIAM J. Optim. **17**(2), 526–557 (2006)
25. Huang, Y., Dai, Y.H., Liu, X.W., Zhang, H.: Gradient methods exploiting spectral properties. arXiv preprint arXiv:1905.03870 (2019)
26. Huang, Y., Dai, Y.H., Liu, X.W., Zhang, H.: On the asymptotic convergence and acceleration of gradient methods. arXiv preprint arXiv:1908.07111 (2019)
27. Huang, Y., Liu, H., Zhou, S.: Quadratic regularization projected Barzilai–Borwein method for nonnegative matrix factorization. Data Min. Knowl. Disc. **29**(6), 1665–1684 (2015)
28. Jiang, B., Dai, Y.H.: Feasible Barzilai–Borwein-like methods for extreme symmetric eigenvalue problems. Optim. Method Softw. **28**(4), 756–784 (2013)
29. Liu, Y.F., Dai, Y.H., Luo, Z.Q.: Coordinated beamforming for miso interference channel: Complexity analysis and efficient algorithms. IEEE Trans. Signal Process. **59**(3), 1142–1157 (2011)
30. Raydan, M.: The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. SIAM J. Optim. **7**(1), 26–33 (1997)
31. Tan, C., Ma, S., Dai, Y.H., Qian, Y.: Barzilai–Borwein step size for stochastic gradient descent. In: Advances in Neural Information Processing Systems, pp. 685–693 (2016)
32. Wang, Y., Ma, S.: Projected Barzilai–Borwein method for large-scale nonnegative image restoration. Inverse Probl. Sci. En. **15**(6), 559–583 (2007)
33. Wright, S.J., Nowak, R.D., Figueiredo, M.A.: Sparse reconstruction by separable approximation. IEEE Trans. Signal Process. **57**(7), 2479–2493 (2009)
34. Yuan, Y.X.: A new stepsize for the steepest descent method. J. Comput. Math. **24**(2), 149–156 (2006)
35. Yuan, Y.X.: Step-sizes for the gradient method. AMS IP Studies in Advanced Mathematics. **42**(2), 785–796 (2008)
36. Zhou, B., Gao, L., Dai, Y.H.: Gradient methods with adaptive step-sizes. Comp. Optim. Appl. **35**(1), 69–86 (2006)