Journal of Computational Mathematics
Vol.37, No.6, 2019, 914–934.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1911-m2019-0171

STABILIZED BARZILAI-BORWEIN METHOD*

Oleg Burdakov
Department of Mathematics, Linköping University, Linköping, Sweden
Email: oleg.burdakov@liu.se
Yuhong Dai
LSEC, ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China
Email: dyh@lsec.cc.ac.cn
Na Huang[1)]
Department of Applied Mathematics, College of Science, China Agricultural University, Beijing, China
Email: hna@cau.edu.cn

Abstract

The Barzilai-Borwein (BB) method is a popular and efficient tool for solving large-scale unconstrained optimization problems. Its search direction is the same as for the steepest descent (Cauchy) method, but its stepsize rule is different. Owing to this, it converges much faster than the Cauchy method. A feature of the BB method is that it may generate too long steps, which throw the iterates too far away from the solution. Moreover, it may not converge, even when the objective function is strongly convex. In this paper, a stabilization technique is introduced. It consists in bounding the distance between each pair of successive iterates, which often allows for decreasing the number of BB iterations. When the BB method does not converge, our simple modification of this method makes it convergent. For strongly convex functions with Lipschits gradients, we prove its global convergence, despite the fact that no line search is involved, and only gradient values are used. Since the number of stabilization steps is proved to be finite, the stabilized version inherits the fast local convergence of the BB method. The presented results of extensive numerical experiments show that our stabilization technique often allows the BB method to solve problems in a fewer iterations, or even to solve problems where the latter fails.

Mathematics subject classification: 65K05, 90C06, 90C30.
Key words: Unconstrained optimization, Spectral algorithms, Stabilization, Convergence analysis.

1. Introduction

In this paper, we consider spectral gradient methods for solving the unconstrained optimization problem

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

where $f : R^n \to R^1$ is a sufficiently smooth function. Its minimizer is denoted by x^*. Gradient-type iterative methods used for solving problem (1.1) have the form

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

* Received July 16, 2019 / Revised version received September 30, 2019 / Accepted November 11, 2019 / Published online November 21, 2019 /
1) Corresponding author

where $g_k = \nabla f(x_k)$ and $\alpha_k > 0$ is a stepsize. Methods of this type differ in the stepsize rules which they follow.

We focus here on the two choices of α_k proposed in 1988 by Barzilai and Borwein [1], usually refereed to as the BB method. The rationale behind these choices is related to viewing the gradient-type methods as quasi-Newton methods, where α_k in (1.2) is replaced by the matrix $D_k = \alpha_k I$. This matrix is served as an approximation of the inverse Hessian matrix. Following the quasi-Newton approach, the stepsize is calculated by forcing either D_k^{-1} (BB1 method) or D_k (BB2 method) to satisfy the secant equation in the least squares sense. The corresponding two problems are formulated as

$$\min_{D=\alpha I} \|D^{-1}s_{k-1} - y_{k-1}\| \quad \text{and} \quad \min_{D=\alpha I} \|s_{k-1} - Dy_{k-1}\|, \tag{1.3}$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$. The solutions to these problems are

$$\alpha_k^{BB1} = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}} \quad \text{and} \quad \alpha_k^{BB2} = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}, \tag{1.4}$$

respectively. Here and in what follows, $\|\cdot\|$ denotes the Euclidean vector norm and the induced matrix norm. Other norms used in this paper will be denoted in a different way.

Barzilai and Borwein [1] proved that their method converges R-superlinearly for two-dimensional strictly convex quadratics. Dai and Fletcher [7] analyzed the asymptotic behavior of BB-like methods, and they obtained R-superlinear convergence of the BB method for the three-dimensional case. Global convergence of the BB method for the n-dimensional case was established by Raydan [20] and further refined by Dai and Liao [10] for obtaining the R-linear rate. For nonquadratic functions, local convergence proof of the BB method with R-linear rate was, first, sketched in some detail by Liu and Dai [19], and then it was later rigorously proved by Dai et al. [9]. Extensive numerical experiments show that the two BB stepsize rules significantly improve the performance of gradient methods (see, e.g., [14, 21]), both in quadratic and nonquadratic cases.

A variety of modifications and extensions have been developed, such as gradient methods with retards [15], alternate BB method [8], cyclic BB method [9], limited memory gradient method [4] etc. Several approaches were proposed for dealing with nonconvex objective functions, in which case the BB stepsize (1.4) may become negative. In our numerical experiments, we use the one proposed in [6]. The BB method has been extended to solving symmetric and nonsymmetric linear equations [6, 11]. Furthermore, by incorporating the nonmontone line search by Grippo et al. [17], Raydan [21] and Grippo et al. [18] developed the global BB method for general unconstrained optimization problems. Later, Birgin et al. [2] proposed the so-called spectral projected gradient method which extends Raydan's method to smooth convex constrained problems. For more works on BB-like methods, see [3, 14, 23] and references therein.

As it was observed by many authors, the BB method may generate too long steps, which throw the iterates too far away from the solution. In practice, it may not converge even for strongly convex functions (see, e.g., [14]). The purpose of this paper is to introduce a simple stabilization technique and to justify its efficiency both theoretically and practically. Our stabilization does not assume any objective function evaluations. It consists in uniformly bounding $\|s_k\|$, the distance between each pair of successive iterates. It should be emphasized that, if the BB method safely converges for a given function, then there is no necessity in stabilizing it. In such cases, the stabilization may increase the number of iterations. In other cases, as it will be demonstrated by results of our numerical experiments, the stabilization may allow for decreasing the number of iterations or even to make the BB method convergent.

Although we focus here on stabilizing the conventional BB method, our approach can directly be combined with the existing modifications of the BB method, where a nonmonotone line search is used.

The paper is organized as follows. In the next section, we present an example of a strictly convex function and show that the BB method does not converge in this case. This contributes to a motivation for stabilizing this method. In the same section, its stabilized version is introduced. In Section 3, a global convergence of our stabilized BB algorithm as well as its R-linear rate of convergence are proved under suitable assumptions. Results of numerical experiments are reported and discussed in Section 4. Finally, some conclusions are included in the last section of the paper.

2. Stabilized Algorithm

Before formulating our stabilized algorithm, we wish to begin with a motivation based on presenting an example of a strongly convex function for which we theoretically prove that neither of the BB methods converge. To the best of our knowledge, no theoretical evidence of BB methods being divergent is available in the literature.

In the review paper by Fletcher [14], it is claimed that the BB method diverges in practice for certain initial points in the test problem referred to as Strictly Convex 2 by Raydan [21], in which

$$f(x) = \sum_{i=1}^{n} i(e^{x_i} - x_i)/10. \tag{2.1}$$

This strongly convex function will be used in Section 4 for illustrating the efficiency of the stabilized algorithm. Our numerical experiments show that, in this specific case, the failure of the BB method is related to the underflow and overflow effects in the computer arithmetic. We are not acquainted with any theoretical justification of the divergence of the BB method for this or any other functions.

We will present now an instance of a function for which the BB method does not converge in the exact arithmetic. For this purpose, the notation

$$a = \sqrt{5} - 1, \quad b = \sqrt{5} + 3, \quad c_1 = \frac{3\sqrt{5} + 8}{4}, \quad c_2 = -\frac{5\sqrt{5} + 11}{32}, \quad f(a) = \frac{c_1 a^2}{2} + \frac{c_2 a^4}{4}$$

will be used. Consider the univariate function

$$f(x) = \begin{cases} \frac{1}{4}(x+a)^2 - (\sqrt{5}+1)(x+a) + f(a), & x < -a, \\ \frac{c_1}{2}x^2 + \frac{c_2}{4}x^4, & -a \le x \le a, \\ \frac{1}{4}(x-a)^2 + (\sqrt{5}+1)(x-a) + f(a), & x > a. \end{cases} \tag{2.2}$$

Its first derivative

$$g(x) = \begin{cases} \frac{1}{2}(x+a) - \sqrt{5} - 1, & x < -a, \\ c_1 x + c_2 x^3, & -a \le x \le a, \\ \frac{1}{2}(x-a) + \sqrt{5} + 1, & x > a \end{cases}$$

is continuously differentiable, and $g(x)$ is an odd monotonically increasing function (see Figure 2.1). It can be easily verified that the function $f(x)$ is twice continuously differentiable with

$$1/2 \leq f''(x) \leq c_1, \quad \forall x \in R^1.$$

This means that this function is strongly convex, and its first derivative is Lipschitz-continuous.

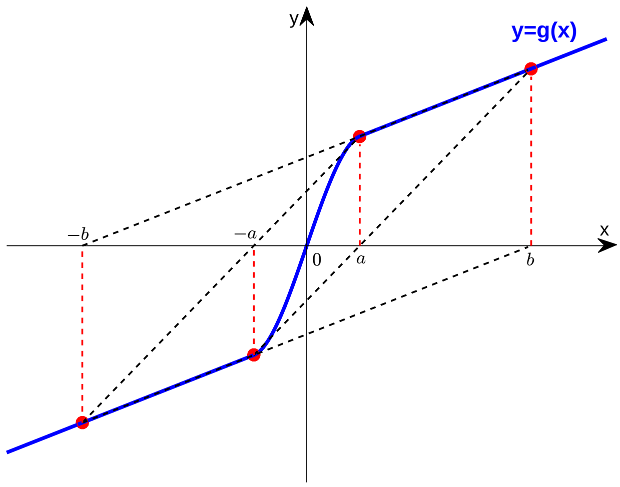

Fig. 2.1. Cyclic iterates generated by the BB method for function (2.2).

For any univariate objective function, there is no difference between BB1 and BB2 versions, and they are equivalent to the secant method applied to the first derivative. For function (2.2), if to initiate the BB method with $x_0 = -b$ and $x_1 = -a$, then the subsequent iterates are

$$\begin{aligned}
x_2 &= x_1 - \frac{x_1 - x_0}{g(x_1) - g(x_0)} g(x_1) = b, \\
x_3 &= x_2 - \frac{x_2 - x_1}{g(x_2) - g(x_1)} g(x_2) = a, \\
x_4 &= x_3 - \frac{x_3 - x_2}{g(x_3) - g(x_2)} g(x_3) = -b = x_0, \\
x_5 &= x_4 - \frac{x_4 - x_3}{g(x_4) - g(x_3)} g(x_4) = -a = x_1.
\end{aligned}$$

This clearly shows that the BB method cycles between four points (see Figure 2.1). The presented counter-example can be easily extended to n-dimensional case. As an example, one can consider a separable objective function equal to the sum of any number of functions of the form (2.2), where no variable appears in more than one of these functions.

After motivating the necessity of stabilizing the BB method, we can now proceed to presenting the basic idea of our stabilized BB algorithm, where $\Delta > 0$ is a parameter. It consists in choosing the stepsize in (1.2) in the way that $\|x_{k+1} - x_k\| = \Delta$, whenever $\|\alpha_k^{BB} g_k\| > \Delta$, i.e. $\alpha_k^{BB} > \Delta / \|g_k\|$. In other cases, we choose $\alpha_k = \alpha_k^{BB}$, which results in $\|x_{k+1} - x_k\| \leq \Delta$. Thus, denoting

$$\alpha_k^{stab} = \frac{\Delta}{\|g_k\|},$$

we propose to choose

$$\alpha_k = \min\{\alpha_k^{BB},\, \alpha_k^{stab}\}. \tag{2.3}$$

Here $\alpha_k^{BB} = \alpha_k^{BB1}$ or $\alpha_k^{BB} = \alpha_k^{BB2}$, depending on the specific BB method in (1.4). A formal description of our stabilized BB algorithm follows.

Algorithm 2.1. *BBstab.*

Given: initial points $x_0, x_1 \in \mathbb{R}^n$ such that $x_0 \neq x_1$, and scalar $\Delta > 0$.
Evaluate g_0 and g_1.
for $k = 1, 2, \ldots$ **do**
  if $g_k = 0$ **then** stop.
  Set $s_{k-1} \leftarrow x_k - x_{k-1}$ and $y_{k-1} \leftarrow g_k - g_{k-1}$.
  Compute α_k by formula (2.3).
  Set $x_{k+1} \leftarrow x_k - \alpha_k g_k$ and evaluate g_{k+1}.
end (for)

This algorithm will be refereed to as BB1stab or BB2stab depending on the corresponding choice of α_k^{BB} in (1.4). Note that, for $\Delta = +\infty$, it reduces to the underlying standard BB algorithm.

3. Convergence Analysis

In this section, global convergence of the BBstab algorithm will be proved. Whenever iterates $\{x_k\}$ are mentioned, they are assumed to be generated by BBstab, where it is required that $x_0 \neq x_1$.

Throughout this section, the objective function is assumed to comply with the following requirement.

A1. The function $f : R^n \to R^1$ is twice continuously differentiable, and there exist positive constants $\Lambda_1 \leq \Lambda_2$ such that

$$\Lambda_1 \|v\|^2 \leq v^T \nabla^2 f(x) v \leq \Lambda_2 \|v\|^2, \quad \forall x, v \in R^n. \tag{3.1}$$

This assumption implies that

$$\Lambda_1 \|x - x^*\| \leq \|g(x)\| \leq \Lambda_2 \|x - x^*\|, \quad \forall x \in R^n. \tag{3.2}$$

Extra assumptions are introduced below in proper places.

We shall use the following notation:

$$\begin{aligned}
\Omega_1 &= \{x \in R^n :\ \|g(x)\| \leq \Lambda_1 \Delta\},\\
\Omega_2 &= \{x \in R^n :\ \Lambda_1 \Delta\ < \|g(x)\| \leq \Lambda_2 \Delta\},\\
\Omega_3 &= \{x \in R^n :\ \Lambda_2 \Delta\ < \|g(x)\|\},\\
\Omega_{3'} &= \{x \in R^n :\ \Lambda_2 \Delta\ < \|g(x)\| \leq \varkappa \Lambda_2 \Delta\},
\end{aligned}$$

which will be motivated later. Here

$$\varkappa = \frac{\Lambda_2}{\Lambda_1}.$$

Obviously, $\Omega_{3'} \subset \Omega_3$, and $\Omega_{1,2,3} = R^n$, where $\Omega_{1,2,3} = \Omega_1 \cup \Omega_2 \cup \Omega_3$. We shall use similar notation for other unions of sets Ω_i.

Inequalities (3.1) ensure that

$$\frac{1}{\Lambda_2} \le \alpha_k^{BB} \le \frac{1}{\Lambda_1}, \quad \forall k \ge 1, \tag{3.3}$$

which in turn means that

$$\alpha_k \le \min\left\{\frac{\Delta}{\|g_k\|}, \frac{1}{\Lambda_1}\right\}, \qquad \forall k \ge 1, \tag{3.4}$$

$$\frac{1}{\varkappa\Lambda_2} \le \alpha_k \le \frac{1}{\Lambda_1}, \qquad \forall x_k \in \Omega_{1,2,3'}. \tag{3.5}$$

These bounds justify the implications

$$\begin{aligned} x_k \in \Omega_1 \quad &\Rightarrow \quad \alpha_k = \alpha_k^{BB}, \\ x_k \in \Omega_2 \quad &\Rightarrow \quad \alpha_k = \min\{\alpha_k^{BB}, \alpha_k^{stab}\}, \\ x_k \in \Omega_3 \quad &\Rightarrow \quad \alpha_k = \alpha_k^{stab}. \end{aligned} \tag{3.6}$$

We can now prove the following result.

Lemma 3.1. *Let $x_0, x_1 \in R^n$ be arbitrary starting points. Then for any $\Delta > 0$, the iterates $\{x_k\}$ have the property that*

$$\|g_{k+1}\| \le \begin{cases} q_k\|g_k\|, & \text{if } x_k \in \Omega_3, \\ \varkappa\|g_k\|, & \text{otherwise,} \end{cases} \qquad \forall k \ge 1, \tag{3.7}$$

where

$$q_k = 1 - \frac{\Lambda_1\Delta}{\|g_k\|}.$$

Proof. Using Assumption A1, we get

$$g_{k+1} = g_k - \alpha_k H_k g_k,$$

where the matrix $H_k = \int_0^1 \nabla^2 f(x_k + ts_k)dt$ is symmetric, and it fulfills the relations

$$\Lambda_1 I \preceq H_k \preceq \Lambda_2 I.$$

Clearly,

$$\|g_{k+1}\| \le \|I - \alpha_k H_k\|\|g_k\|. \tag{3.8}$$

Consider, first, the case when $x_k \in \Omega_3$. Using the inequality $\Lambda_2\Delta < \|g(x)\|$ and relations (3.6), we can derive for (3.8) the following upper bound

$$\|I - \alpha_k^{stab}H_k\| = \max_{\|v\|=1} \left|1 - \alpha_k^{stab}v^T H_k v\right| = 1 - \alpha_k^{stab} \min_{\|v\|=1} v^T H_k v \le 1 - \frac{\Lambda_1\Delta}{\|g_k\|}.$$

This proves the upper inequality in (3.7).

Suppose now that $x_k \in \Omega_{1,2}$, i.e., $\|g_k\| \le \Lambda_2\Delta$. Then, using (3.4), we get the bounds $\Lambda_2^{-1} \le \alpha_k \le \Lambda_1^{-1}$, which together with the inequalities $\Lambda_1 \le \|H_k\| \le \Lambda_2$ yield

$$\|I - \alpha_k H_k\| \le \max\{1 - \varkappa^{-1}, \varkappa - 1\} = \varkappa - 1 < \varkappa.$$

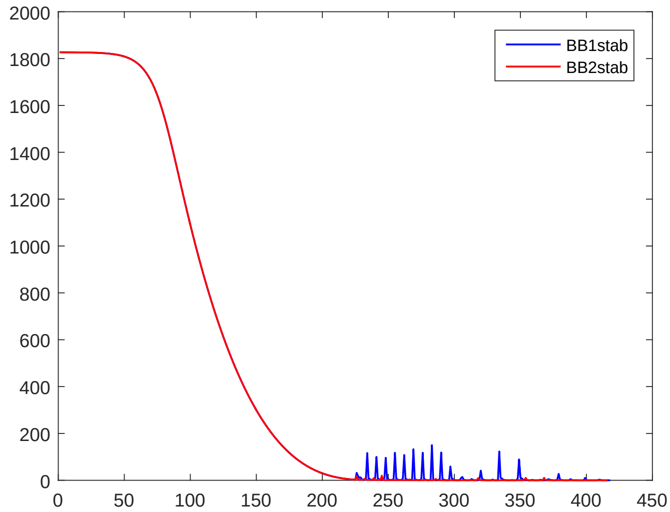

Fig. 3.1. Graphs of $\|g_k\|$ for BB1stab and BB2stab with $\Delta = 2$ for Raydan function (2.1).

By combining this estimate with (3.8), we finally prove the lower inequality in (3.7). $\square$

Lemma 3.1 implies that the stabilization steps have the following properties

$$q_k \in (0,1), \qquad \forall x_k \in \Omega_3, \tag{3.9}$$

$$q_{k+1} < q_k, \qquad \forall x_k, x_{k+1} \in \Omega_3. \tag{3.10}$$

Next, we prove that, after a finite number of iterations, all iterates belong to the bounded set $\Omega_{1,2,3'}$.

Lemma 3.2. *For any $x_0, x_1 \in R^n$ and $\Delta > 0$, there exists an integer $K \geq 1$ such that the inequality*

$$\|g_k\| \leq \varkappa \Lambda_2 \Delta \tag{3.11}$$

holds, that is $x_k \in \Omega_{1,2,3'}$, for all $k \geq K$. Moreover, K is the iteration number corresponding to the first iterate x_K that belongs to $\Omega_{1,2,3'}$.

Proof. Notice that (3.11) is satisfied if and only if $x_k \in \Omega_{1,2,3'}$. We first show that if $x_k \in \Omega_{1,2,3'}$, then so does the next iterate. Indeed, in view of (3.7) and (3.9), if $x_k \in \Omega_{3'}$, then $x_{k+1} \in \Omega_{1,2,3'}$. On the other hand, if $x_k \in \Omega_{1,2}$, i.e. $\|g_k\| \leq \Lambda_2\Delta$, then, by Lemma 3.1, we have $\|g_{k+1}\| \leq \varkappa\Lambda_2\Delta$.

Suppose now that $x_1 \in \Omega_3 \setminus \Omega_{3'}$. Then it immediately follows from relations (3.9) and (3.10), that there exists $K > 1$ such that $x_K \in \Omega_{1,2,3'}$. As it was shown above, this means that $x_k \in \Omega_{1,2,3'}$ for all $k \geq K$. $\square$

It follows from (3.9) that, when iterates belong to the set Ω_3, the value $\|g_k\|$ monotonically decreases as indicated by (3.7). Furthermore, the actual decrease may speed-up in accordance with (3.10). When the iterates reach $\Omega_{1,2}$, the decrease is naturally expected to slow down, and this is followed by a non-monotonic behavior of $\|g_k\|$, which is a typical feature of the BB steps. One can observe all these stages in the behavior of BBstab in Figure 3.1. It presents changes of $\|g_k\|$ with k in the process of minimizing Raydan function (2.1). Details of these runs are discussed in Section 4. Note that both BB1 and BB2 fail to solve this problem starting from the same points. The figure illustrates the role of stabilization in providing convergence of BBstab. One can clearly recognize the first stage of the process when the stabilization steps ensure a monotonic decrease of $\|g_k\|$. For the BB1stab and BB2stab, the iteration when the standard BB step was used for the first time is 228 and 226, respectively. For them, the last stabilization step was used in iteration 379 and 353, respectively. Observe that the spikes of $\|g_k\|$ produced by BB1 is much larger than those for BB2.

Lemma 3.2 allows us to deduce an interesting property of the BB method, namely, that if it generates bounded steps, it cannot generate unbounded iterates because one can choose a sufficiently large Δ, which is not binding. The same lemma indicates that a proper choice of Δ allows for BBstab to reach any neighborhood of x^*. We use the notation

$$B_\delta(x^*) = \{x \in R^n : \ \|x - x^*\| \leq \delta\}.$$

in the following formulation of this useful feature of BBstab.

Lemma 3.3. *Let $x_0, x_1 \in R^n$ be any starting points. Then for any $\delta > 0$ and positive $\Delta \leq \frac{\delta}{\varkappa^2}$, there exists $K(\Delta) \geq 1$ such that the iterates $\{x_k\}$ satisfy the condition*

$$x_k \in B_\delta(x^*), \quad \forall k \geq K(\Delta).$$

Proof. Combining (3.2) and Lemma 3.2, we get the relations

$$\|x_k - x^*\| \leq \frac{\|g_k\|}{\Lambda_1} \leq \varkappa^2 \Delta \leq \delta,$$

which are satisfied for all sufficiently large k. This completes the proof. □

We shall make use of Lemma 3.2 for proving global convergence result for BBstab. We show also that its local rate of convergence is R-linear, which means that there exist positive γ and $c \in (0,1)$ such that

$$\|x_{k+1} - x^*\| \leq \gamma c^k \|x_1 - x^*\|. \tag{3.12}$$

These convergence results are based on our convergence analysis presented in the next sub-section for convex quadratic functions.

3.1. Convergence in Quadratic Case

In this sub-section, we focus on minimizing convex quadratic functions of the form

$$f(x) = \frac{1}{2} x^T A x - b^T x, \tag{3.13}$$

where the matrix $A \in R^{n\times n}$ is positive definite, and $b \in R^n$. For these functions, we derive the convergence with R-linear rate. To this end, we will make use of the following property which is the same as Property A in [5].

Definition 3.1. *We say that the choice of the stepsize in* (1.2) *has property **P** if there exist an integer m and positive constants M_1 and M_2 such that, for all $k \geq 1$,*
(i) $\Lambda_1 \leq \alpha_k^{-1} \leq M_1$;
(ii) for any integer $\ell \in [1, n-1]$ and real number $\epsilon > 0$, if $R(k-j, \ell) \leq \epsilon$ and $(g_{k-j}^{(\ell+1)})^2 \geq M_2 \epsilon$ hold for $j \in [0, \min\{k, m\} - 1]$, then $\alpha_k^{-1} \geq \frac{2}{3} \lambda_{\ell+1}$.

Theorem 3.1. *Let $x_0, x_1 \in R^n$ be arbitrary starting points. Then for any $\Delta > 0$, the sequence $\{x_k\}$ converges to x^* with* R*-linear rate. Moreover, there exists a positive integer $\bar{j}$, such that, for any $\Delta > 0$, $x_0 \in R^n$ and $x_1 \in \Omega_{1,2,3'}$, the inequality*

$$\|g_{k+\bar{j}}\| \le \frac{1}{2}\|g_k\|$$

holds for all $k \ge 1$.

Proof. It is well known that the BB method is invariant under orthogonal transformation of the variables and, as it can be easily seen, so does its stabilized version. Hence, we can assume without loss of generality that the matrix A is of the form

$$A = \mathrm{diag}(\lambda_1, \ldots, \lambda_n), \tag{3.14}$$

where $0 < \Lambda_1 = \lambda_1 < \lambda_2 < \ldots < \lambda_n = \Lambda_2$. Here, like it is often done for the gradient methods (see, e.g., [20]), it is assumed without loss of generality that the matrix A has distinct eigenvalues. Then denoting the i-th component of g_k by $g_k^{(i)}$, we have

$$g_{k+1}^{(i)} = (1 - \alpha_k \lambda_i) g_k^{(i)}, \quad i = 1, \ldots, n. \tag{3.15}$$

We will also make use of the following notation:

$$R(k,\, \ell) = \sum_{i=1}^{\ell} (g_k^{(i)})^2.$$

Firstly, we prove that the step size α_k has property $\mathbf{P}$. Lemma 3.2 ensures that $x_k \in \Omega_{1,2,3'}$ for all $k \ge 1$. Then the bounds (3.5) show that α_k has property $\mathbf{P}$(i) with $M_1 = \varkappa\Lambda_2$.

Next, we will show, for any integer $\ell \in [1,\, n-1]$ and real number $\epsilon > 0$, that the inequality $\alpha_k^{-1} \ge \frac{2}{3}\lambda_{\ell+1}$ is satisfied, whenever $R(k-1,\, \ell) \le \epsilon$ and $(g_{k-1}^{(\ell+1)})^2 \ge 2\epsilon$. This will be done separately for BB1- and BB2-based iterates.

For the BB1 case, we have

$$\begin{aligned} \alpha_k^{-1} &\ge \frac{g_{k-1}^T A g_{k-1}}{\|g_{k-1}\|^2} = \frac{\sum_{i=1}^{n} \lambda_i (g_{k-1}^{(i)})^2}{R(k-1,\, n)} \ge \frac{\lambda_{\ell+1} \sum_{i=\ell+1}^{n} (g_{k-1}^{(i)})^2}{R(k-1,\, \ell) + \sum_{i=\ell+1}^{n} (g_{k-1}^{(i)})^2} \\ &\ge \frac{\lambda_{\ell+1} \sum_{i=\ell+1}^{n} (g_{k-1}^{(i)})^2}{\epsilon + \sum_{i=\ell+1}^{n} (g_{k-1}^{(i)})^2} \ge \frac{2\lambda_{\ell+1}\epsilon}{\epsilon + 2\epsilon} = \frac{2}{3}\lambda_{\ell+1}. \end{aligned}$$

For BB2, we obtain

$$\begin{aligned} (\alpha_k)^{-1} &\ge \frac{g_{k-1}^T A^2 g_{k-1}}{g_{k-1}^T A g_{k-1}} \ge \frac{\lambda_{\ell+1} \sum_{i=\ell+1}^{n} \lambda_i (g_{k-1}^{(i)})^2}{\lambda_{\ell+1} R(k-1,\, \ell) + \sum_{i=\ell+1}^{n} \lambda_i (g_{k-1}^{(i)})^2} \\ &\ge \frac{\lambda_{\ell+1}^2 (g_{k-1}^{(\ell+1)})^2}{\lambda_{\ell+1} R(k-1,\, \ell) + \lambda_{\ell+1} (g_{k-1}^{(\ell+1)})^2} \ge \frac{2\lambda_{\ell+1}\epsilon}{\epsilon + 2\epsilon} = \frac{2}{3}\lambda_{\ell+1}. \end{aligned}$$

Thus, **P**(ii) holds for $m = 2$ and $M_2 = 2$. This implies that BBstab stepsize α_k satisfies **P**. Then we can skip the rest of the proof because it is similar to the proof of Theorem 4.1 in [5]. $\square$

It should be emphasized that, in this lemma, the value of $\bar{j}$ depends only on Λ_1 and Λ_2.

3.2. Convergence in General Case

For nonquadratic functions, we shall first prove local R-linear convergence of BBstab. This result will then be used for showing that it converges from any starting point.

Throughout this sub-section, we need to additionally assume that the Hessian matrix $\nabla^2 f(x)$ is Lipschitz-continuous at x^*. In what follows, we use the notation $H = \nabla^2 f(x^*)$.

A2. There exist a radius $\rho > 0$ and a Lipschitz constant $L \geq 0$ such that

$$\|\nabla^2 f(x) - H\| \leq L\|x - x^*\|, \quad \forall x \in B_\rho(x^*).$$

This assumption implies that

$$\|g(x) - H(x - x^*)\| \leq \frac{L}{2}\|x - x^*\|^2, \quad \forall x \in B_\rho(x^*). \tag{3.16}$$

The second-order Taylor approximation to f around x^* is given by the quadratic function

$$\hat{f}(x) = f(x^*) + \frac{1}{2}(x - x^*)^T H(x - x^*). \tag{3.17}$$

Define new iterates $\hat{x}_{k,j}$ associated with $\hat{f}$ as follows:

$$\begin{cases} \hat{x}_{k,0} = x_k, \\ \hat{x}_{k,j+1} = \hat{x}_{k,j} - \hat{\alpha}_{k,j}\hat{g}_{k,j}, \quad j \geq 0, \end{cases} \tag{3.18}$$

where

$$\hat{\alpha}_{k,j} = \begin{cases} \alpha_k, & \text{if } j = 0, \\ \min\{\hat{\alpha}^{BB}_{k,j},\, \hat{\alpha}^{stab}_{k,j}\}, & \text{otherwise.} \end{cases}$$

Here $\hat{\alpha}^{BB}_{k,j} = \hat{\alpha}^{BB1}_{k,j}$ or $\hat{\alpha}^{BB}_{k,j} = \hat{\alpha}^{BB2}_{k,j}$ and $\hat{\alpha}^{stab}_{k,j} = \frac{\Delta}{\|\hat{g}_{k,j}\|}$ with

$$\hat{\alpha}^{BB1}_{k,j} = \frac{\hat{s}^T_{k+j-1}\hat{s}_{k+j-1}}{\hat{s}^T_{k+j-1}\hat{y}_{k+j-1}}, \qquad \hat{\alpha}^{BB2}_{k,j} = \frac{\hat{s}^T_{k+j-1}\hat{y}_{k+j-1}}{\hat{y}^T_{k+j-1}\hat{y}_{k+j-1}},$$

$\hat{s}_{k+j-1} = \hat{x}_{k,j} - \hat{x}_{k,j-1}$, $\hat{g}_{k,j} = H(\hat{x}_{k,j} - x^*)$ and $\hat{y}_{k+j-1} = \hat{g}_{k,j} - \hat{g}_{k,j-1}$. In what follows, whenever we mention $\hat{x}_{k,j}$ and $\hat{\alpha}_{k,j}$, they are assumed to be generated as defined above.

The next result follows immediately from Theorem 3.1.

Lemma 3.4. *Let $\Delta > 0$ be any scalar, such that $\Omega_{1,2,3'} \subseteq B_\rho(x^*)$. Then there exists a positive integer $\bar{j}$, dependent only on Λ_1 and Λ_2, such that, for any $x_{k-1} \in R^n$ and $x_k \in \Omega_{1,2,3'}$, the inequality holds*

$$\|\hat{g}_{k,\bar{j}}\| \leq \frac{1}{2}\|\hat{g}_{k,0}\|.$$

It can be easily seen that if $x_k \in \Omega_{1,2,3'}$, then all corresponding $\hat{x}_{k,j} \in \Omega_{1,2,3'}$. In this case, BBstab stepsize α_k satisfies the bounds (3.5), and similarly for $\hat{\alpha}_{k,j}$, we have the bounds

$$\frac{1}{\varkappa\Lambda_2} \leq \hat{\alpha}_{k,j} \leq \frac{1}{\Lambda_1}, \quad \forall j \geq 0. \tag{3.19}$$

The following result will be used for proving local R-linear convergence.

Lemma 3.5. *Let integer $\bar{j} \geq 1$ be arbitrary. Then there exist positive scalars $\bar{\Delta}$ and γ with the following property: for any $\Delta \in (0,\ \bar{\Delta}]$, $x_{k-1} \in R^n$, $x_k \in \Omega_{1,2,3'} \subset B_\rho(x^*)$ and $m \in [0,\ \bar{j}]$, for which*

$$\|\hat{g}_{k,j}\| \geq \frac{1}{2}\|\hat{g}_{k,0}\|, \quad \forall j \in [0, \max\{0, m-1\}], \tag{3.20}$$

we have the inequality

$$\|x_{k+j} - \hat{x}_{k,j}\| \leq \gamma\|x_k - x^*\|^2 \tag{3.21}$$

satisfied for all $j \in [0, m]$.

Proof. Throughout the proof, let c denote a generic positive constant, which may depend on some of fixed constants, such as $\bar{\Delta}$, $\bar{j}$, Λ_1, Λ_2 or L, but not on the choice of Δ or $x_k \in \Omega_{1,2,3'} \subset B_\rho(x^*)$. For brevity, we will use the same notation in all inequalities, even though every specific value of c depends on the one, where it is used. What is important is that the number of these inequalities is finite.

We first notice that, by Lemma 3.2, the relation $x_{k+j} \in \Omega_{1,2,3'} \subseteq B_\rho(x^*)$ holds for all $j \geq 0$. The process of proving (3.21) will be combined with showing that the inequalities

$$\|g(x_{k+j}) - \hat{g}(\hat{x}_{k,j})\| \leq c\|x_k - x^*\|^2, \tag{3.22}$$

$$\|s_{k+j}\| \leq c\|x_k - x^*\|, \tag{3.23}$$

$$|\alpha_{k+j} - \hat{\alpha}_{k,j}| \leq c\|x_k - x^*\|, \tag{3.24}$$

are satisfied for all $j \in [0,\ m]$.

The proof of (3.21)-(3.24) is by induction on m. For $m = 0$, noticing that $\hat{x}_{k,0} = x_k$, $\hat{\alpha}_{k,0} = \alpha_k$ and $s_k = -\alpha_k g_k$, by (3.2), (3.5) and (3.16), we can immediately get (3.21)-(3.24) satisfied for $j = 0$.

Suppose that there exist $M \in [1,\ \bar{j})$ and $\bar{\Delta} > 0$ with the property that if (3.20) holds for any $m \in [0,\ M-1]$, then (3.21)-(3.24) are satisfied for all $j \in [0,\ m]$. Next, we shall show that for a smaller choice of $\bar{\Delta} > 0$, we can replace M by $M+1$. Hence, we suppose that (3.20) holds for all $j \in [0,\ M]$. Since (3.20) holds for all $j \in [0,\ M-1]$, it follows from the induction hypothesis and (3.23) that

$$\|x_{k+M+1} - x^*\| \leq \|x_k - x^*\| + \sum_{i=0}^{M} \|s_{k+i}\| \leq c\|x_k - x^*\|. \tag{3.25}$$

By analogy with the proof of Lemma 2.2 in [9], we derive from (3.2), (3.5), (3.16), (3.19), (3.25) and the induction hypothesis that (3.21)-(3.23) hold for $j = M+1$. Then we just need to show that

$$|\alpha_{k+M+1} - \hat{\alpha}_{k,M+1}| \leq c\|x_k - x^*\|. \tag{3.26}$$

It follows from (3.2) that

$$\|x_k - x^*\| \leq \frac{\|g_k\|}{\Lambda_1} \leq \varkappa^2\Delta \leq \varkappa^2\bar{\Delta}.$$

Then by choosing any $\bar{\Delta} < 1/(2\gamma\varkappa^3)$, using relations (3.1), (3.5), (3.19)-(3.21), (3.23) and the same reasoning as in the proof of Lemma 2.2 in [9], we obtain

$$|\alpha_{k+M+1}^{BB} - \hat{\alpha}_{k,M+1}^{BB}| \leq c\|x_k - x^*\|. \tag{3.27}$$

In the following, the proof of (3.26) will be done by separately considering four different cases.

Case I: $\alpha_{k+M+1}^{BB} \leq \alpha_{k+M+1}^{stab}$ and $\hat{\alpha}_{k,M+1}^{BB} \leq \hat{\alpha}_{k,M+1}^{stab}$. Then (3.27) directly leads to

$$|\alpha_{k+M+1} - \hat{\alpha}_{k,M+1}| = |\alpha_{k+M+1}^{BB} - \hat{\alpha}_{k,M+1}^{BB}| \leq c\|x_k - x^*\|.$$

Case II: $\alpha_{k+M+1}^{BB} \leq \alpha_{k+M+1}^{stab}$ and $\hat{\alpha}_{k,M+1}^{BB} > \hat{\alpha}_{k,M+1}^{stab}$. If $\hat{\alpha}_{k,M+1}^{stab} \geq \alpha_{k+M+1}^{BB}$, then (3.27) implies

$$|\alpha_{k+M+1} - \hat{\alpha}_{k,M+1}| = \hat{\alpha}_{k,M+1}^{stab} - \alpha_{k+M+1}^{BB} < \hat{\alpha}_{k,M+1}^{BB} - \alpha_{k+M+1}^{BB} \leq c\|x_k - x^*\|.$$

Suppose now that $\hat{\alpha}_{k,M+1}^{stab} < \alpha_{k+M+1}^{BB}$. Then we have

$$|\alpha_{k+M+1} - \hat{\alpha}_{k,M+1}| = \alpha_{k+M+1}^{BB} - \hat{\alpha}_{k,M+1}^{stab} \leq \alpha_{k+M+1}^{stab} - \hat{\alpha}_{k,M+1}^{stab}. \tag{3.28}$$

It follows from (3.5) and (3.19) that

$$\|\hat{g}_{k,M+1}\| = \frac{\Delta}{\hat{\alpha}_{k,M+1}^{stab}} > \frac{\Delta}{\hat{\alpha}_{k,M+1}^{BB}} \geq \Delta\Lambda_1. \tag{3.29}$$

By (3.2) and (3.16), we get

$$\|g_{k+M+1} - \hat{g}_{k,M+1}\| \leq \frac{L}{2}\|x_{k+M+1} - x^*\|^2 \leq \frac{L}{2\Lambda_1^2}\|g_{k+M+1}\|^2 \leq \frac{1}{2}\varkappa^4\Delta^2 L.$$

This along with (3.29) leads to

$$\|g_{k+M+1}\| \geq \|\hat{g}_{k,M+1}\| - \|g_{k+M+1} - \hat{g}_{k,M+1}\| \geq \Delta\Big(\Lambda_1 - \frac{1}{2}\varkappa^4\Delta L\Big) \geq \Delta C(\bar{\Delta}),$$

where $C(\bar{\Delta}) = \Lambda_1 - \varkappa^4\bar{\Delta}L/2 > 0$ whenever $\bar{\Delta} < 2\Lambda_1/(\varkappa^4 L)$. Then we obtain

$$\begin{aligned}
|\alpha_{k+M+1}^{stab} - \hat{\alpha}_{k,M+1}^{stab}| &= \left|\frac{\Delta}{\|g_{k+M+1}\|} - \frac{\Delta}{\|\hat{g}_{k,M+1}\|}\right| = \Delta\frac{|\|\hat{g}_{k,M+1}\| - \|g_{k+M+1}\||}{\|g_{k+M+1}\|\|\hat{g}_{k,M+1}\|} \\
&\leq \Delta\frac{\|\hat{g}_{k,M+1} - g_{k+M+1}\|}{\|g_{k+M+1}\|\|\hat{g}_{k,M+1}\|} \leq \frac{c\|x_k - x^*\|^2}{\Delta\Lambda_1 C(\bar{\Delta})} \leq \frac{c\|g_k\|\|x_k - x^*\|}{\Delta\Lambda_1^2 C(\bar{\Delta})} \\
&\leq \frac{c\varkappa\Lambda_2\Delta\|x_k - x^*\|}{\Delta\Lambda_1^2 C(\bar{\Delta})} = \frac{c\varkappa^2}{\Lambda_1 C(\bar{\Delta})}\|x_k - x^*\|.
\end{aligned}$$

This together with (3.28) shows that (3.26) holds.

Case III: $\alpha_{k+M+1}^{BB} > \alpha_{k+M+1}^{stab}$ and $\hat{\alpha}_{k,M+1}^{BB} \leq \hat{\alpha}_{k,M+1}^{stab}$. If $\alpha_{k+M+1}^{stab} \geq \hat{\alpha}_{k,M+1}^{BB}$, then by (3.27), we have

$$|\alpha_{k+M+1} - \hat{\alpha}_{k,M+1}| = \alpha_{k+M+1}^{stab} - \hat{\alpha}_{k,M+1}^{BB} \leq \alpha_{k+M+1}^{BB} - \hat{\alpha}_{k,M+1}^{BB} \leq c\|x_k - x^*\|.$$

Suppose now that $\alpha_{k+M+1}^{stab} < \hat{\alpha}_{k,M+1}^{BB}$. Then we get

$$|\alpha_{k+M+1} - \hat{\alpha}_{k,M+1}| = \hat{\alpha}_{k,M+1}^{BB} - \alpha_{k+M+1}^{stab} \leq \hat{\alpha}_{k,M+1}^{stab} - \alpha_{k+M+1}^{stab}.$$

To use the same reasoning as in Case II, we need to have lower bounds for $\|g_{k+M+1}\|$ and $\|\hat{g}_{k,M+1}\|$. To this end, applying (3.5) and (3.19), we obtain

$$\|g_{k+M+1}\| = \frac{\Delta}{\alpha^{stab}_{k+M+1}} > \frac{\Delta}{\alpha^{BB}_{k+M+1}} \geq \Delta\Lambda_1. \tag{3.30}$$

Furthermore, (3.2), (3.16) and (3.30) yield

$$\|\hat{g}_{k,M+1}\| \geq \|g_{k+M+1}\| - \|g_{k+M+1} - \hat{g}_{k,M+1}\| \geq \Delta C(\bar{\Delta}).$$

This lower bound is positive whenever $\bar{\Delta} < 2\Lambda_1/(\varkappa^4 L)$. The two lower bounds allows us to conclude, by analogy with Case II, that (3.26) holds.

Case IV: $\alpha^{BB}_{k+M+1} > \alpha^{stab}_{k+M+1}$ and $\hat{\alpha}^{BB}_{k,M+1} > \hat{\alpha}^{stab}_{k,M+1}$. It follows from (3.2), (3.22), (3.29) and (3.30) that

$$\begin{aligned}
&|\alpha_{k+M+1} - \hat{\alpha}_{k,M+1}| \\
\leq & \Delta \frac{\|\hat{g}_{k,M+1} - g_{k+M+1}\|}{\|g_{k+M+1}\|\|\hat{g}_{k,M+1}\|} \leq \Delta \frac{c\|x_k - x^*\|^2}{\Delta^2\Lambda_1^2} \leq \frac{c\|g_k\|\|x_k - x^*\|}{\Delta\Lambda_1^3} \\
\leq & \frac{c\varkappa\Lambda_2\Delta\|x_k - x^*\|}{\Delta\Lambda_1^3} \leq \frac{c\varkappa\Lambda_2}{\Lambda_1^3}\|x_k - x^*\|.
\end{aligned}$$

Collecting the results in the considered four cases, one can see that (3.26) is satisfied for any

$$\Delta < \min\left\{\frac{1}{2\gamma\varkappa^3}, \frac{2\Lambda_1}{\varkappa^4 L}\right\}.$$

This completes the induction and finally proves that inequalities (3.21)-(3.24) hold for all $j \in [0, m]$. $\square$

Next we will establish the local convergence property of BBstab for nonquadratic functions.

Theorem 3.2. *There exists positive $\bar{\Delta}$ such that, for any positive $\Delta \leq \bar{\Delta}$ and any starting points $x_0,\ x_1 \in \Omega_{1,2,3'}$, the sequence $\{x_k\}$ converges to x^* with* R-*linear rate.*

Lemma 3.5 allows us to skip the proof of this theorem because the reasoning is similar to the proof of Theorem 2.3 in [9].

We complete the analysis by presenting the following global convergence result.

Theorem 3.3. *There exists positive $\bar{\Delta}$ such that, for any positive $\Delta \leq \bar{\Delta}$ and any starting points $x_0,\ x_1 \in R^n$, the sequence $\{x_k\}$ converges to x^* with* R-*linear rate.*

Proof. Let $\bar{\Delta} > 0$ be given by Theorem 3.2, which ensures local convergence to x^*. According to Lemma 3.2, after a finite number of BBstab iterations, all iterates will belong to $\Omega_{1,2,3'}$. This finally proves global convergence with R-linear rate. $\square$

4. Numerical Results

Our algorithms were implemented in MATLAB. The algorithms are terminated when either the number of iterations exceeds 10^5, or

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

In the next two subsections, results of numerical experiments are presented separately for quadratic and nonquadratic test functions.

A successful value of Δ is obviously problem dependent. In our implementation, we try to estimate its order of magnitude by setting $\Delta = +\infty$ for the first few iterations and making use of $\|s_k\|$ produced at these iterations by the standard BB algorithm. At the subsequent iterations, the constant value

$$\Delta = c \cdot \min\{\|s_1\|, \|s_2\|, \|s_3\|\}, \tag{4.1}$$

is applied, where $c > 0$ is a parameter. It turns out that this adaptive choice of Δ is less problem dependent.

Table 4.1: Numerical results for linear systems from the SuiteSparse Matrix Collection, Part I.

PROBLEM		BB1	BB1stab		PROBLEM		BB1	BB1stab	
name	n	it	it	c	name	n	it	it	c
1138_bus	1 138	35 202	**21 384**	0.3	ex33	1 733	1 303	**958**	0.2
2cubes_sphere	101 492	5 576	**4 662**	0.3	Flan_1565	1 564 794	13 781	16 537	0.25
af_0_k101	503 625	4 433	**2 634**	0.2	fv3	9 801	449	**449**	0.2
af_1_k101	503 625	2 473	2 766	0.25	G2_circuit	150 102	1 139	**1 139**	0.25
af_2_k101	503 625	4 034	**2 499**	0.25	G3_circuit	1 585 478	2 177	**2 177**	0.2
af_3_k101	503 625	3 627	**2 378**	0.2	Geo_1438	1 437 960	32 134	**29 095**	0.3
af_4_k101	503 625	3 047	5 368	0.3	gyro	17 361	10 611	11 925	0.3
af_5_k101	503 625	2 397	2 753	0.2	gyro_m	17 361	3 325	**2 225**	0.25
af_shell3	504 855	1 956	4 565	0.3	hood	220 542	4 073	4 308	0.25
af_shell7	504 855	2 495	5 515	0.3	Hook_1498	1 498 023	7 839	**7 358**	0.25
apache1	80 800	18 017	**9 143**	0.2	inline_1	503 712	20 490	**16 833**	0.3
apache2	715 176	17 807	**17 807**	0.2	jnlbrng1	40 000	124	**108**	0.2
audikw_1	943 695	92 730	**65 818**	0.2	Kuu	7 102	1 733	**949**	0.3
bcsstk08	1 074	4 627	5 113	0.3	ldoor	952 203	9 133	9 281	0.3
bcsstk09	1 083	747	**713**	0.3	LF10000	19 998	48 867	**38 250**	0.2
bcsstk10	1 086	3 416	**2 383**	0.25	LFAT5000	19 994	22 358	**22 358**	0.25
bcsstk11	1 473	2 204	**1 699**	0.2	m_t1	97 578	1 826	**1 826**	0.2
bcsstk13	2 003	6 848	8 171	0.3	mhd3200b	3 200	2 065	**2 065**	0.2
bcsstk14	1 806	3 577	**2 682**	0.25	mhd4800b	4 800	2 466	**2 466**	0.2
bcsstk15	3 948	7 006	**4 872**	0.25	msc01050	1 050	15 187	**11 529**	0.25
bcsstk16	4 884	401	**401**	0.25	msc01440	1 440	807	**807**	0.2
bcsstk17	10 974	27 014	**14 841**	0.25	msc04515	4 515	8 066	**6 889**	0.2
bcsstk18	11 948	5 895	**4 332**	0.3	msc10848	10 848	3 356	**3 356**	0.2
bcsstk21	3 600	1 455	1 594	0.25	msc23052	23 052	19 088	**7 340**	0.2
bcsstk23	3 134	8 182	**5 619**	0.2	msdoor	415 863	8 113	**6 655**	0.25
bcsstk24	3 562	2 383	**1 537**	0.3	nasa1824	1 824	9 520	**6 515**	0.3
bcsstk25	15 439	8 369	8 971	0.25	nasa2146	2 146	355	**355**	0.2
bcsstk26	1 922	12 624	**8 761**	0.2	nasa2910	2 910	19 574	**13 683**	0.3
bcsstk27	1 224	863	887	0.3	nasa4704	4 704	43 448	**32 961**	0.2
bcsstk36	23 052	15 466	**12 001**	0.25	nasasrb	54 870	10 302	**10 223**	0.3
bcsstk38	8 032	1 584	**1 584**	0.25	nd3k	9 000	67 509	86 986	0.25
bcsstm08	1 074	4 183	**4 183**	0.2	nd6k	18 000	92 468	**41 133**	0.2
bcsstm11	1 473	623	**287**	0.3	nd24k	72000	84 165	**73 216**	0.3
bcsstm12	1 473	2 838	**2 375**	0.3	offshore	259 789	3 826	3 949	0.3
bcsstm23	3 134	2 143	**1 857**	0.25	oilpan	73 752	4 647	**3 899**	0.3
bcsstm24	3 562	2 102	**1 611**	0.25	olafu	16 146	69 575	80 804	0.3

It is necessary to emphasize that the stabilization was designed not to speed-up the BB method when it safely converges. In such cases, it may increase the number of iterations, which is a negative outcome. The main purpose of the stabilization is to prevent the BB method from making too long steps. This serves for decreasing the number of BB iterations in case of its poor convergence or even making the method convergent when it fails, which is a positive outcome. Outcomes of all these aforementioned types were observed in our numerical experiments with stabilizing the BB method. One can easily recognize them in the tables presented below.

We focus here on demonstrating the potentials of improving convergence for the BB method. Therefore, our stabilized version is not checked here against another optimization algorithms. Since the computational cost of one iteration for the BB algorithms are practically the same as for their stabilized versions, only the number of iterations are compared. Notice that the

Table 4.2: Numerical results for linear systems from the SuiteSparse Matrix Collection, Part II.

PROBLEM		BB1	BB1stab		PROBLEM		BB1	BB1stab	
name	n	it	it	c	name	n	it	it	c
bcsstm25	15 439	2 266	**2 119**	0.2	parabolic_fem	525 825	5 451	**2 989**	0.2
bcsstm26	1 922	1 614	**1 239**	0.2	plat1919	1 919	3 297	**2 804**	0.2
bcsstm39	46 772	575	**575**	0.2	plbuckle	1 282	5 601	**3 726**	0.3
BenElechi1	245 874	3 137	**3121**	0.3	Pres_Poisson	14 822	17 291	**13 461**	0.25
bloweybq	10 001	107	**107**	0.2	pwtk	21 7918	26 060	**21 798**	0.25
bmw7st_1	141 347	2 463	**2 463**	0.2	s1rmq4m1	5 489	9 043	**6 890**	0.2
bmwcra_1	148 770	86 966	123 528	0.25	s1rmt3m1	5 489	10 092	11 576	0.25
bodyy4	17 546	154	**154**	0.25	s2rmq4m1	5 489	5 371	8 958	0.2
bodyy5	18 589	405	**405**	0.3	s2rmt3m1	5 489	7 850	**6 039**	0.25
bodyy6	19 366	809	853	0.3	s3dkq4m2	90 449	16 169	**16 169**	0.2
bone010	986 703	55 659	**55 659**	0.25	s3dkt3m2	90 449	18 654	**10 739**	0.2
boneS01	127 224	7 688	**5 669**	0.2	s3rmq4m1	5 489	8 413	**7 848**	0.25
boneS10	914 898	28 584	**24 899**	0.2	s3rmt3m1	5 489	16 901	19 625	0.3
bundle1	10 581	244	**244**	0.2	s3rmt3m3	5 357	15 586	**6 737**	0.25
cant	62 451	19 609	22 895	0.2	Serena	1 391 349	47 765	**23 155**	0.25
cbuckle	13 681	6 963	10 770	0.25	ship_001	34 920	17 575	**17 499**	0.2
cfd1	70 656	4 475	**3 555**	0.2	ship_003	121 728	64 349	69 948	0.3
cfd2	123 440	5 515	8 145	0.25	shipsec1	140 874	8 730	**6 681**	0.2
Chem97ZtZ	2 541	125	**114**	0.25	shipsec5	179 860	2 565	3 113	0.3
consph	83 334	15 034	**11 232**	0.25	shipsec8	114 919	3 900	5 827	0.3
crankseg_1	52 804	4 012	**4 012**	0.2	smt	25 710	38 442	**24 695**	0.25
crankseg_2	63 838	4 914	**3 614**	0.3	sts4098	4 098	8 262	12 042	0.2
crystm01	4 875	100	**100**	0.2	t2dah_e	11 445	2 557	**1 612**	0.3
crystm02	13 965	114	**114**	0.2	t2dal_e	4 257	1 585	**1 171**	0.25
ct20stif	52 329	6 482	**6 482**	0.25	t3dl_e	20 360	503	**361**	0.2
cvxbqp1	50 000	383	**383**	0.2	thermal1	82 654	5 812	**5 812**	0.2
Dubcova1	16 129	181	**181**	0.2	thermal2	1 228 045	22 201	**7 170**	0.25
Dubcova2	65 025	372	**348**	0.3	tmt_sym	726 713	40 335	**40 335**	0.25
Dubcova3	146 689	520	**429**	0.2	Trefethen_2000	2 000	258	**258**	0.2
ex3	1 821	508	**387**	0.2	Trefethen_20000	20 000	358	**358**	0.2
ex9	3 363	1 202	**1 202**	0.3	Trefethen_20000b	19 999	404	**404**	0.2
ex10	2 410	3 038	**2 023**	0.25	vanbody	47 072	19 354	**19 133**	0.2
ex10hs	2 548	2 412	**1 628**	0.2	wathen100	30 401	238	**238**	0.25
ex13	2 568	2 972	**2 972**	0.2	wathen120	36 441	308	**308**	0.2
ex15	6 867	3 022	3 298	0.3	—	—	—	—	—

number of iterations is the same as the number of gradient evaluations.

In our numerical experiments, the BB1 algorithm was generating too long steps more frequently than the BB2 algorithm. This is often caused by relatively too small values of the scalar product $s_{k-1}^T y_{k-1}$ in the denominator of α_k^{BB1}. This explains why the stabilization is, in general, more important for the BB1 stepsize choice than for the BB2. Therefore, the numerical results presented here refer mainly to the BB1.

4.1. Quadratic test functions

A part of the numerical experiments was related to minimizing convex quadratic functions (3.13). This problem is equivalent to solving the system of linear equations

$$Ax = b.$$

The matrices in our set of test problems come from the SuiteSparse Matrix Collection [12, 22]. For generating the vector b, we assumed that the solution $x^* = e$, i.e., $b = Ae$, where $e = (1, 1, \dots, 1)^T$. The total number of problems in our test set is 141, where the problem size n varies from thousands to millions.

For the adaptive selection of Δ by formula (4.1), we tried just a few values of the parameter c, namely, 0.2, 0.25 and 0.3. In Tables 4.1 and 4.2, the number of iterations are reported for algorithms BB1 and BB1stab. For the latter, the best of the three results is presented along with the corresponding value of c. If the reported result is the same as for the BB1 algorithm, then it is obvious that the number of iterations remains the same for all values of c larger than the indicated one. The number of iterations, which is not worse than for the BB1 algorithm, are highlighted in this and other tables in this paper. One can see that, comparing with the BB1, its stabilized version is faster in solving 74 problems, while it is slower in 30 problems. Furthermore, the reduction in the number of iterations obtained by virtue of the stabilization was often substantial. We also tested the BB2 and BB2stab algorithms for these same 141 problems. We tried $c = 0.1$, 0.2, 0.25 and 0.3 in the adaptive selection of Δ by formula (4.1). Comparing with the BB2, BB2stab is faster in solving 58 problems, while for the given values of c, the stabilization is unable to decrease the number of BB2 iterations in 58 problems.

4.2. Nonquadratic test functions

For general functions, it is more difficult than for quadratic ones to avoid the cases, when x_1 is chosen too close to x_0 or too far away of it. In order to avoid such poor choices of these two points, our BBstab algorithms are initialized with only one point, namely, x_0. The point x_1 is produced in the algorithms by checking if the inequality $f(x_0 + s_0) < f(x_0)$ is satisfied for $s_0 = -\alpha_0 g_0$, where $\alpha_0 = 1/\|g_0\|_\infty$. Otherwise, a number, typically few, of backtracking steps are performed by dividing the current vector s_0 by 4, while the required inequality is violated.

We begin here by comparing the performance of the BB algorithms and their stabilized versions on the strongly convex Raydan function (2.1) for $n = 1000$. The point $x_0 = -10 \cdot e$ was used for starting the algorithms. The standard BB1 algorithm failed to solve the problem. After two iterations, an overflow in computing $s_k^T y_k$ was reported. If to introduce the bounds $[10^{-30}, 10^{30}]$ for α^{BB1}, like it is often done in practice, then it also fails, although after a larger number of iterations. Namely, at iteration 123 and all subsequent iterations, an underflow was observed in calculating x_{k+1} for $\|s_k\| < 10^{-26}$. In these two cases, the standard BB2 also failed. However, the same test problem for the same x_0 was successfully solved by BB1stab

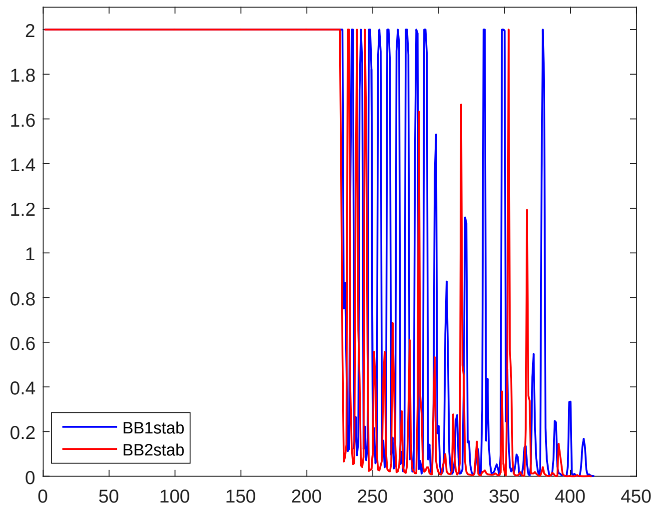

Fig. 4.1. Graphs of $\|s_k\|$ for BB1stab and BB2stab with $\Delta = 2$ for Raydan function (2.1).

and BB2stab with $\Delta = 2$ in 418 and 416 iterations, respectively. No bounds, like $[10^{-30}, 10^{30}]$, are used in our implementation of the BB algorithms and their stabilized versions. Figure 4.1 illustrates the stabilization effect. One can see that the BB1 was generating too long steps more frequently than the BB2. This observation is in general agreement with the other numerical experiments that we performed and also with the theory, which says that $\alpha_k^{BB1} \geq \alpha_k^{BB2}$.

The performance of our algorithms was compared also for unconstrained minimization problems from the CUTEst collection [16], which provides a standard starting point x_0 for each of them. We excluded from our comparison quadratic problems and those, in which the BB1/BB2 algorithm converged in less than 20 iterations. The results reported here concern only the

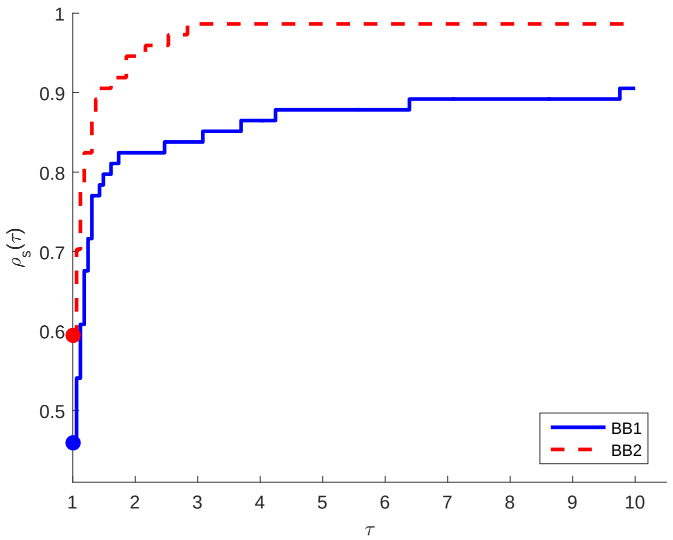

Fig. 4.2. Performance profiles of the BB1 and BB2 algorithms adapted to solving nonconvex unconstrained minimization problems (based on solving problems from the CUTEst collection).

problems, where at least one of the compared algorithms converged, and also those, where the both algorithms converged to the same point.

Table 4.3: Numerical results for unconstrained minimization problems from the CUTEst collection, adaptive selection of Δ.

PROBLEM		BB1	BB1stab		PROBLEM		BB1	BB1stab	
name	n	it	it	c	name	n	it	it	c
ALLINITU	4	21	**21**	0.1	EXTROSNB	1 000	23	**23**	0.5
ARGTRIGLS	200	626	**626**	0.5	FLETCBV2	5 000	30 225	98 735	1
BA-L1LS	57	34	**33**	1	FLETCHCR	1 000	1 892	1 964	1
BA-L16LS	66 462	64	66	0.5	FREUROTH	5 000	52	**52**	0.5
BA-L21LS	34 134	197	**179**	1	HEART8LS	8	44	**44**	0.5
BA-L49LS	23 769	65	**60**	1	HYDC20LS	99	35	**35**	1
BA-L52LS	192 627	280	**277**	0.1	LUKSAN11LS	100	31	32	1
BA-L73LS	33 753	65	69	0.5	LUKSAN12LS	98	40	**38**	0.5
BDQRTIC	5000	41	**41**	0.5	LUKSAN17LS	100	230	**187**	0.1
BROWNBS	2	4 110	**961**	0.1	LUKSAN21LS	100	6 284	28 255	1
BROYDN3DLS	5 000	21	**21**	0.1	LUKSAN22LS	100	64	**51**	0.1
BROYDN7D	5 000	29	**29**	0.1	MOREBV	5 000	54 926	$> 10^5$	1
BROYDNBDLS	5 000	58	**58**	1	MSQRTALS	1 024	71	**56**	0.5
CHAINWOO	4 000	96	**42**	1	MSQRTBLS	1 024	56	59	0.5
CHNROSNB	50	133	**133**	0.5	NCB20	5 010	23	**22**	0.1
CHNRSNBM	50	93	**93**	0.1	NONDQUAR	5 000	40 401	89 179	1
CRAGGLVY	5 000	56	**50**	1	NONMSQRT	4 900	54	**54**	0.1
CUBE	2	$> 10^5$	**61**	1	OSCIGRAD	100 000	81	**81**	1
CURLY10	10 000	64	**56**	0.1	OSCIPATH	10	30	**30**	0.1
CURLY20	10 000	56	**56**	0.5	PENALTY2	200	730	1 909	1
CURLY30	10 000	57	**57**	0.5	PENALTY3	200	21	**21**	0.5
DENSCHNF	2	122	**31**	0.5	POWELLSG	5 000	65	**47**	1
DIXMAANE	3 000	24	**23**	0.1	ROSENBR	2	$> 10^5$	**332**	1
DIXMAANF	3 000	24	**23**	0.1	ROSENBRTU	2	$> 10^5$	**85**	1
DIXMAANI	3 000	22	**22**	1	SCURLY30	10 000	252	**234**	0.1
DIXMAANJ	3 000	23	**23**	1	SPMSRTLS	4 999	335	**268**	0.1
DIXMAANM	3 000	773	**515**	1	SROSENBR	5 000	$> 10^5$	**55**	0.5
DIXMAANN	3 000	711	**502**	0.5	SSBRYBND	5 000	4 247	11 005	1
DIXMAANO	3 000	589	**417**	1	SSCOSINE	5 000	3 882	10 414	1
DIXMAANP	3 000	310	**305**	0.1	TOINTGOR	50	40	44	0.5
EDENSCH	2 000	48	**36**	1	TOINTGSS	5 000	5 006	**5 004**	1
EIGENALS	2 550	41	**41**	0.1	VAREIGVL	50	415	**323**	0.5
EIGENCLS	2 652	145	170	0.5	VESUVIALS	8	235	$> 10^5$	1
ERRINROS	50	2 920	**746**	1	VESUVIOULS	8	256	$> 10^5$	1
ERRINRSM	50	25 807	**7 366**	0.1	WATSON	12	120	217	0.1

Recall that the BB method was originally designed for solving convex problems in which case it is guaranteed that α_k^{BB} is nonnegative. Since the most of the unconstrained minimization test problems in the CUTEst collection are nonconvex, we had to adapt the BB method to solving this kind of problems. In our implementation of the BB method and its stabilized version, we follow paper [6] in setting

$$\alpha_k^{BB} \leftarrow \frac{\|s_k\|}{\|y_k\|}, \tag{4.2}$$

whenever $\alpha_k^{BB} \leq 0$. This makes our algorithms much more robust. Figure 4.2 presents results of solving 74 problems from the CUTEst collection. The BB1 and BB2 algorithms failed in 4 and 3 cases, respectively. The plots of the performance profiles introduced in [13] indicate that the BB2 algorithm is more robust than the BB1. Furthermore, the former algorithm required, on average, fewer iterations for solving problems. The main reason is that the BB1 algorithm generates too long steps more frequently. In what follows, we focus on presenting here results of stabilizing the BB1 algorithm, because it gains more from the stabilization than the BB2 algorithm.

Table 4.3 presents results of solving 70 nonquadratic test problems from the CUTEst collection. We tried only three values of the parameter c in the adaptive choice of Δ using (4.1), namely, 0.1, 0.5 and 1.0. The BB1 and BB1stab algorithms were not able to solve problems during 10^5 iterations in 4 and 3 cases, respectively. The BB1stab requires fewer number of iterations in 32 cases, while the BB1 performs better only in 17 cases. In 21 cases, the BB1stab with the indicated values of c requires the same number of iterations as the BB1.

We made experiments also with directly setting a certain value of Δ in the BB1stab. The trial values were 0.01, 0.1 and 1.0. For a few test problems, the results are better than for the aforementioned adaptive choice with $c = 0.1$, 0.5 and 1.0. For 22 of 71 problems, the number of iterations is smaller than in case of the BB1. These results are reported in Table 4.4. The preselected values of Δ allowed the BB1stab to solve five problems of those not solved by the BB1, including problems MOREBV and TQUARTIC, in which the adaptive choice of Δ failed. In case of TQUARTIC, the BB1 terminated because of producing NaN (Not a Number) in Matlab. The experiments with the preselected values of Δ indicate that there is plenty of room for improving the very simple adaptive strategy proposed in this paper.

Table 4.4: Numerical results for unconstrained minimization problems from the CUTEst collection, preselected Δ.

PROBLEM		BB1	BB1stab		PROBLEM		BB1	BB1stab	
name	n	it	it	Δ	name	n	it	it	Δ
BROWNBS	2	4 110	**80**	1	LUKSAN11LS	100	31	**23**	1
CHNROSNB	50	133	**50**	1	LUKSAN17LS	100	230	**166**	1
CHNRSNBM	50	93	**41**	1	MOREBV	5 000	54 926	**44 712**	0.01
CUBE	2	$> 10^5$	**94**	0.1	MSQRTALS	1 024	71	**66**	0.1
DENSCHNF	2	122	**31**	1	NONMSQRT	4 900	54	**51**	1
DIXMAANM	3 000	773	**715**	1	OSCIPATH	10	30	**27**	1
DIXMAANO	3 000	589	**514**	1	ROSENBR	2	$> 10^5$	**129**	0.1
ERRINROS	50	2 920	**923**	1	ROSENBRTU	2	$> 10^5$	**664**	0.1
ERRINRSM	50	25 807	**6 165**	0.1	SPMSRTLS	4 999	335	**294**	1
FLETCBV2	5 000	30 225	**25 325**	1	SROSENBR	5 000	$> 10^5$	**206**	1
FLETCHCR	1 000	1 892	**572**	1	TQUARTIC	5 000	F	**5 325**	0.1

For the BB2stab algorithm, we still tried the same three values of the parameter c in the adaptive choice of Δ using (4.1) as for BB1stab. In 77 test problems, the BB2stab performs better in 25 cases, while the BB2 performs better only in 15 cases. Table 4.5 presents results for all the cases when the BB2stab requires fewer number of iterations.

Table 4.5: Numerical results for unconstrained minimization problems from the CUTEst collection, adaptive selection of Δ.

PROBLEM		BB2	BB2stab		PROBLEM		BB2	BB2stab	
name	n	it	it	c	name	n	it	it	c
BA-L21LS	34134	191	**187**	1	EIGENALS	2550	44	**39**	0.5
BA-L52LS	192627	358	**316**	1	EIGENCLS	2652	201	**177**	0.1
BDQRTIC	5000	41	**37**	0.1	INDEFM	100000	23	**21**	1
BROWNBS	2	4110	**961**	0.1	LUKSAN17LS	100	198	**183**	0.1
CHNRSNBM	50	54	**45**	0.1	MSQRTALS	1024	77	**72**	0.5
DENSCHNF	2	29	**28**	1	MSQRTBLS	1024	59	**58**	0.5
DIXMAANE	3000	21	**20**	0.1	NONDIA	5000	-	**10599**	0.5
DIXMAANF	3000	24	**22**	0.1	OSCIPATH	10	26	**25**	0.1
DIXMAANI	3000	25	**20**	0.1	PENALTY3	200	22	**21**	0.1
DIXMAANJ	3000	24	**22**	0.1	POWELLSG	5000	44	**38**	0.5
DIXMAANM	3000	610	**425**	0.1	VAREIGVL	50	490	**407**	0.1
DIXMAANN	3000	611	**448**	0.1	WATSON	12	340	**170**	0.5
DIXMAANO	3000	464	**414**	0.1					

5. Conclusions

In the present paper, it was proposed to stabilize the conventional BB method by virtue of bounding the distance between sequential iterates. The purpose was to improve its convergence, when it is affected by too long steps $\|\alpha_k^{BB} g_k\|$, and also to make the BB method convergent, when it fails to converge. Both a theoretical and numerical study of the stabilized version was conducted. We have proved that the stabilization provides the BB method with a global convergence without recourse to using any line search. The numerical results presented here are highly encouraging. The proposed very simple adaptive selection of Δ was able to successfully trap a value which is appropriate for each specific problem. However, we hope that this paper will stimulate development of more efficient algorithms for adaptive selection of Δ.

Acknowledgments. Part of this work was done during Oleg Burdakovs visit to the Chinese Academy of Sciences, which was supported by the Visiting Scientist award under the Chinese Academy of Sciences President's International Fellowship Initiative for 2017. The second author was supported by the Chinese Natural Science Foundation (No. 11631013) and the National 973 Program of China (No. 2015CB856002).

References

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

[2] E.G. Birgin, J.M. Martínez and M. Raydan, Nonmonotone spectral projected gradient methods on convex sets, *SIAM J. Optim.*, **10**:4 (2000), 1196-1211.

[3] E.G. Birgin, J.M. Martínez and M. Raydan, Spectral projected gradient methods: Review and perspectives, *Journal of Statistical Software*, **60**:3 (2014), 1-21.

[4] F.E. Curtis and W. Guo, R-linear convergence of limited memory steepest descent, *IMA Journal of Numerical Analysis*, **38** (2018), 720-742.

[5] Y.H. Dai, Alternate step gradient method, *Optimization*, **52** (2003), 395-415.

[6] Y.H. Dai, M. Al-Baali and X. Yang, A positive Barzilai-Borwein-like stepsize and an extension for symmetric linear systems, in *Numerical Analysis and Optimization*, Springer, 2015, 59-75.

[7] Y.H. Dai and R. Fletcher, On the asymptotic behaviour of some new gradient methods, *Math. Program.*, **103** (2005), 541-559.

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

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

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

[11] Y.H. Dai, L.Z. Liao and D. Li, An analysis of the Barzilai and Borwein gradient method for unsymmetric linear equations, In *Optimization and Control with Applications*, Springer, 2005, 183-211.

[12] T.A. Davis and Y. Hu, The University of Florida sparse matrix collection, *ACM Transactions on Mathematical Software*, **38**:1 (2011), 1-25.

[13] E. D. Dolan and J.J. Moré, Benchmarking optimization software with performance profiles, *Math. Programming*, **91** (2002), 201-213.

[14] R. Fletcher, On the Barzilai-Borwein method, in *Optimization and control with applications*, Springer, 2005, 235-256.

[15] A. Friedlander, J. M. Martínez, B. Molina and M. Raydan, Gradient method with retards and generalizations, *SIAM J. Numer. Anal.*, **36**:1 (1998), 275-289.

[16] N.I.M. Gould, D.Orban and Ph.L. Toint, CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization, *Computational Optimization and Applications*, **60** (2015), 545-557.

[17] L. Grippo, F. Lampariello and S. Lucidi, A nonmonotone line search technique for Newton's method, *SIAM J. Numer. Anal.*, **23**:4, 707-716.

[18] L. Grippo and M. Sciandrone, Nonmonotone globalization techniques for the Barzilai-Borwein gradient method, *Computational Optimization and Applications*, **23**:2 (2002), 143-169.

[19] W. Liu and Y.H. Dai Minimization algorithms based on supervisor and searcher cooperation, *J. Optim. Theory Appl.*, **111**:2 (2001), 359-379.

[20] M. Raydan, On the Barzilai and Borwein choice of steplength for the gradient method, *IMA J. Numer. Anal.*, **13**:3 (1993), 321-326.

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

[22] *The SuiteSparse Matrix Collection*, https://sparse.tamu.edu/, 2019.

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