Optimization
Vol. 52, Nos. 4–5, August–October 2003, pp. 395–415

Taylor & Francis
Taylor & Francis Group

ALTERNATE STEP GRADIENT METHOD*

YU-HONG DAI†

State Key Laboratory of Scientific and Engineering Computing,
Institute of Computational Mathematics and Scientific/Engineering Computing,
Academy of Mathematics and System Sciences, Chinese Academy of Sciences,
P.O. Box 2719, Beijing 100080, P.R. China

(Received 26 November 2002; In final form 23 July 2003)

The Barzilai and Borwein (BB) gradient method does not guarantee a descent in the objective function at each iteration, but performs better than the classical steepest descent (SD) method in practice. So far, the BB method has found many successful applications and generalizations in linear systems, unconstrained optimization, convex-constrained optimization, stochastic optimization, etc. In this article, we propose a new gradient method that uses the SD and the BB steps alternately. Hence the name "alternate step (AS) gradient method." Our theoretical and numerical analyses show that the AS method is a promising alternative to the BB method for linear systems. Unconstrained optimization algorithms related to the AS method are also discussed. Particularly, a more efficient gradient algorithm is provided by exploring the idea of the AS method in the GBB algorithm by Raydan (1997).

To establish a general *R*-linear convergence result for gradient methods, an important property of the stepsize is drawn in this article. Consequently, *R*-linear convergence result is established for a large collection of gradient methods, including the AS method. Some interesting insights into gradient methods and discussion about monotonicity and nonmonotonicity are also given.

Keywords: Linear system; Unconstrained optimization; Gradient method; Convergence rate; Nonmonotone line search

Mathematics Subject Classifications 2000: 65K05; 90C52

1. INTRODUCTION

It is well known that the solution of the linear system

$$Ax = b, \quad \text{where } A \in R^{n\times n}, \ b \in R^n, \tag{1.1}$$

also solves the minimization problem

$$q(x) := \frac{1}{2}x^T Ax - b^T x \tag{1.2}$$

*This work was also reported on the International Workshop on "Optimization and Control with Applications", Erice, Italy, July 9–17, 2001.

†E-mail: dyh@lsec.cc.ac.cn

ISSN 0233-1934 print: ISSN 1029-4945 online © 2003 Taylor & Francis Ltd
DOI: 10.1080/0233193031000161154 7

if A is a real symmetric positive definite (SPD) matrix. It is just due to this fact that Fletcher and Reeves [11] extended the linear conjugate gradient method [15] to solve the general unconstrained optimization problem

$$\min f(x), \quad x \in R^n. \tag{1.3}$$

The purpose of this article is to develop a new gradient method for (1.1) or (1.2), and then to discuss related algorithms for unconstrained optimization.

Denote $g_k = Ax_k - b$ to be the residual of (1.1) at x_k, or equivalently the gradient of the $q(x)$ in (1.2) at x_k. The well-known steepest descent (SD) method [5] is defined by

$$x_{k+1} = x_k - \alpha_k g_k, \tag{1.4}$$

where α_k is given by

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

The stepsize (1.5) minimizes $f(x_k - \alpha g_k)$ along the line. In practice, however, the SD method performs poorly, and is badly affected by ill-conditioning (see [1]). A well-known disadvantage of the method is that it may produce zigzags. Akaike [1] also proved that the directions generated tend to two different directions:

$$\lim_{k\to\infty} \frac{g_{2k}}{\|g_{2k}\|} = \bar{d}, \qquad \lim_{k\to\infty} \frac{g_{2k+1}}{\|g_{2k+1}\|} = \hat{d}, \tag{1.6}$$

where and below $\|\cdot\|$ is the two norm if it is not prescribed. The SD method is a simple but basic method, and has played an important role in the development of nonlinear optimization and many other fields. Reference [19] provides some new properties of the SD method.

Denote $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$. Since the matrix $D_k = (1/\alpha_k)I$ is an approximation to the Hessian of $q(x)$ at x_k, Barzilai and Borwein [4] chose the stepsize α_k such that D_k has a certain quasi-Newton property, i.e., $D_k = \arg\min\{\|Ds_{k-1} - y_{k-1}\|: D = \alpha^{-1}I\}$, yielding

$$\alpha_k^{\mathrm{BB}} = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}. \tag{1.7}$$

Noting that $s_{k-1} = -\alpha_{k-1} g_{k-1}$ and $y_{k-1} = As_{k-1}$, an equivalent formula of (1.7) is

$$\alpha_k^{\mathrm{BB}} = \frac{g_{k-1}^T g_{k-1}}{g_{k-1}^T A g_{k-1}}. \tag{1.8}$$

In this article, we call the method (1.4)–(1.7) as Barzilai and Borwein (BB) gradient method. Fletcher [9] shows how the stepsize (1.7) is related inversely to the eigenvalues of the matrix A. Comparing with the SD method, the BB method cannot

guarantee a descent in the objective function at each iteration, but often requires less computational work. A direct application of the BB method in chemistry can be found in [13].

An important work on the BB method was due to Raydan [21], where the method is extended to unconstrained optimization by incorporating the nonmonotone line search in [14]. The resulting algorithm, GBB, proves competitive to some standard codes. A successful application of the algorithm can be found in [2]. The idea of Raydan was also extended by [3] for minimizing differentiable functions on closed convex sets. Reference [17] provides a powerful scheme for unconstrained optimization problems with strong noises by combining the BB method and the stochastic approximation method.

Regarding convergence analyses, it is known that the SD method is Q-linearly convergent [1]. If $q(x)$ in (1.2) is a two-dimensional convex quadratic, Barzilai and Borwein [4] showed that the BB method is R-superlinearly convergent for almost all the starting points. For the higher-dimensional case, the analysis of the BB method is relatively difficult due to the nonmonotonic behavior. Nevertheless, Raydan [20] was able to prove the global convergence of the BB method. Dai and Liao [7] refined the analyses of Raydan and established the R-linear convergence of the method. One corollary of this result is that the BB method is locally R-linearly convergent for general functions [17]. Consequently, under suitable assumptions, the BB stepsize can be accepted [6] by the nonmonotone line search when the iterate is close to the solution, as first observed in Raydan [21].

Other work related to the BB method can be found in [3,12], and a recent report [22]. Specifically, Friedlander [12] investigated gradient methods with retards and generalizations for linear systems. To sum up, due to its simplicity and numerical efficiency, the BB method has now received much attention in the optimization community.

In this article, motivated by how to avoid producing zigzags, we propose a new gradient method. The new method uses the SD and BB steps alternately. Hence the name of alternate step (AS) gradient method. Our theoretical and numerical analyses show that the AS method is a promising alternative to the BB method for linear systems. Specifically, for two-dimensional SPD linear systems and any $\epsilon > 0$, the AS method is proved to be two-step $(3-\epsilon)$-Q-superlinearly convergent for almost all the starting points, whereas only R-superlinear convergence is achieved for the BB method. We will also show that the introduction of the AS method is very useful in designing new gradient algorithms for unconstrained optimization.

This article is organized as follows. In the next section, we describe our motivation and propose the AS method. Some properties of the method are given and other gradient methods related to the method are discussed. In Section 3, we analyze the AS method for SPD linear systems in the case when $n = 2$. Two-step $(3-\epsilon)$-Q-superlinear convergence result is established for the method. Section 4 analyzes the method for any-dimensional linear systems. To establish a general R-linear convergence result for gradient methods, one property called Property (A) of the stepsize is drawn. Consequently, R-linear convergence is proved for a large collection of gradient methods, including the AS method. In Section 5, we discuss unconstrained optimization algorithms related to the AS method. Particularly, a more efficient gradient algorithm is provided by exploring the idea of the AS method in the GBB algorithm. Conclusions and discussion are made in the last section.

2. ALTERNATE STEP GRADIENT METHOD

We consider the problem (1.1) or equivalently (1.2), where A is an SPD matrix. As mentioned in Section 1, the SD method produces zigzags. If all SD steps are replaced with BB steps, then the zigzagging phenomenon will not occur any more since the BB method is R-superlinearly convergent for two-dimensional convex quadratics. Note that each zigzag includes two steps. Intuitively, the zigzagging phenomenon will not occur either if one of the two SD steps is replaced with a BB step. This motivates us to consider the gradient method that chooses its stepsize as follows:

$$\alpha_k = \begin{cases} \alpha_k^{\mathrm{SD}}, & \text{for odd } k; \\ \alpha_k^{\mathrm{BB}}, & \text{for even } k. \end{cases} \tag{2.1}$$

In other words, the method uses an SD step and a BB step alternately. Hence, we call such method as alternate step gradient method, and abbreviate it to AS method.

Now we discuss the properties of the AS method. First, the AS method uses more recent information than the BB method. This is because, the AS method takes an SD step every two iterations, and the SD stepsize (1.5) and the BB stepsize (1.8) are the inverse Rayleigh quotients of A with respect to g_k and g_{k-1}, respectively. Following this line, we consider the method that takes a BB step after $m-1$ SD steps:

$$\alpha_{mk+i} = \begin{cases} \alpha_{mk+i}^{\mathrm{SD}}, & \text{for } i = 1, \ldots, m-1; \\ \alpha_{mk+m}^{\mathrm{BB}}, & \text{for } i = m, \end{cases} \tag{2.2}$$

where $m \geq 1$. It is obvious that (2.1) is corresponding to (2.2) with $m = 2$. We will see that the above method even with a big value of m performs significantly better than the SD method. This shows that the introduction of some delays in calculating the stepsize can speed up the SD method greatly. Although we do not recommend the method (2.3) with a large m in real applications and one may also consider for example the method that takes an SD step after $m-1$ BB steps, the study explains the importance of introducing some delays in calculating the stepsizes.

Another property of the AS method is that the stepsizes in the AS method are identical every two iterations, i.e., $\alpha_k^{\mathrm{BB}} = \alpha_{k-1}^{\mathrm{SD}}$. Consequently, the AS method requires less computational work than the SD method and the BB method in calculating the stepsizes. This saving is worthwhile even when A is a sparse matrix. Following this line, one may be interested in the method (1.4) where

$$\alpha_{mk+i} = \alpha_{mk+1}^{\mathrm{SD}}, \quad \text{for } \ i = 1, \ldots, m \tag{2.3}$$

and the method (1.4) where

$$\alpha_{mk+i} = \alpha_{mk+1}^{\mathrm{BB}}, \quad \text{for } \ i = 1, \ldots, m. \tag{2.4}$$

In (2.3) and (2.4), m is a positive integer. If $m = 2$, then (2.3) reduces to (2.1). The method (2.3) with relatively large values of m was also studied in [12]. In this article,

we call the methods (2.3) and (2.4) as cyclic steepest descent step (CSDS) method and cyclic Barzilai–Borwein step (CBBS) method, respectively. Note that

$$\alpha_{mk+1}^{\mathrm{BB}} = \frac{s_{mk}^T s_{mk}}{s_{mk}^T y_{mk}}. \tag{2.5}$$

One advantage of the CBBS method over the CSDS method is that, its stepsize can be computed without the matrix A explicitly. Thus like the BB method, it is easy to extend the CBBS method to unconstrained optimization by combining the nonmonotone line search. Further researches on this topic are required.

To have an immediate comparison of the AS and BB methods, we test the four-dimensional example in [4] in an SGI indigo workstation with MATLAB. Here we should note that, since our main purpose is to compare the AS and BB methods, only small dimensional linear systems are tested in this article. These examples are typical and facilitate the observation of the relationship between the stepsizes and the eigenvalues of A. We refer to [10,12] for comparisons of the BB method and the conjugate gradient method for some large-scale SPD linear systems.

The example in [4] is to minimize the function $q(x) = (1/2)x^T Ax - b^T x$, where

$$A = \mathrm{diag}\,(20,\ 10,\ 2,\ 1), \qquad b = e_4. \tag{2.6}$$

Here e_i means the i-dimensional vector whose elements are all 1. For both the BB and AS methods, the starting points are $x_1 = x_2 = 0$ and the stepsize $\alpha_2 = 1$. The stopping condition is $\|g_k\|_2 \le 10^{-9}$. The numerical results are listed in Table I. From Table I, we

TABLE I Comparing BB and AS for the example in [4]

k	*BB*		*AS*	
	$\|g_k\|$	α_k	$\|g_k\|$	α_k
2	2.000000000e+00	1.000000000e+00	2.000000000e+00	1.000000000e+00
3	2.104756518e+01	1.212121212e−01	2.104756518e+01	5.515438247e−02
4	2.713844044e+01	5.515438247e−02	4.573627514e+00	5.515438247e−02
5	2.994865127e+00	5.015928785e−02	1.985820021e+00	1.132205353e−01
6	7.415329742e−01	5.473128024e−02	7.052415295e−01	1.132205353e−01
7	5.735245384e−01	2.149779845e−01	5.740712412e−01	1.296041404e−01
8	3.795997585e−01	3.439341351e−01	6.223633511e−01	1.296041404e−01
9	5.504678760e−01	2.109907996e−01	8.585273865e−01	5.449617282e−02
10	6.061557888e−01	1.024061516e−01	2.430102830e−01	5.449617282e−02
11	7.204225765e−02	9.992090956e−02	2.064644743e−01	4.954058547e−01
12	6.534149118e−02	7.792830276e−02	5.901365954e−02	4.954058547e−01
13	4.280539524e−02	6.786426829e−02	5.251166367e−01	5.000730516e−02
14	2.801762984e−02	8.882203072e−02	4.487869368e−03	5.000730516e−02
15	2.284800386e−02	2.101416069e−01	2.243309753e−03	1.000031655e−01
16	2.991780903e−02	2.221805587e−01	1.128902045e−05	1.000031655e−01
17	9.417777814e−02	5.895504423e−02	9.030968717e−06	4.999930678e−01
18	1.786895454e−02	5.023775240e−02	1.008796076e−07	4.999930678e−01
19	5.419356357e−03	5.569706724e−02	9.079017800e−07	5.000000004e−02
20	4.815155825e−03	4.990214595e−01	1.798117219e−11	5.000000004e−02
21	8.239370279e−05	4.999838767e−01		
22	7.366507283e−04	5.059567565e−02		
23	8.776530117e−06	5.000000143e−02		
24	4.355755920e−08	5.000246318e−02		
25	2.177848363e−08	1.000025480e−01		
26	1.769866292e−10	1.000082561e−01		

see that the AS and BB methods require 20 and 26 iterations, respectively. Meanwhile, the AS and BB methods need to compute 9 and 24 stepsizes, respectively (the stepsize $\alpha_2 = 1$ is not considered here). This small example shows that the AS method may be a promising alternative to the BB method.

3. CONVERGENCE ANALYSES: TWO-DIMENSIONAL CASE

In the analyses of any gradient method (1.4) for SPD linear systems, by the invariance property under any orthogonal transformation we can assume without loss of generality that the matrix A is of the form

$$A = \mathrm{diag}\,(\lambda_1, \lambda_2, \ldots, \lambda_n), \tag{3.1}$$

where $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$. Further, we assume that $\lambda_1 = 1$, otherwise we consider the matrix $(1/\lambda_1)A$.

In this section, we analyze the two-dimensional case. The purposes of considering this special case are to see whether the AS method can avoid producing zigzags and to compare different gradient methods. Up to now, there is no other way in theory to compare different gradient methods.

Let $\lambda = \lambda_2$ and $g_k^{(i)}$ be ith component of g_k. The following theorem shows that for almost all the starting points, the AS method is two-step Q-superlinearly convergent for two-dimensional SPD linear systems. Hence like BB method, the AS method can avoid producing zigzags.

THEOREM 3.1 *Consider the linear system* (1.1), *where* A *is a two-dimensional SPD matrix. Consider the AS method* (1.4)–(2.1). *If the initial gradient* g_1 *can be expressed as*

$$g_1 = t\,(1, \pm 1)^T, \tag{3.2}$$

we have that

$$g_k = t\left(\frac{\lambda - 1}{\lambda + 1}\right)^{k-1}(1, \pm(-1)^{k-1}), \quad \textit{for } k \ge 1, \tag{3.3}$$

which means that $\|g_k\|$ *converges to zero linearly. Otherwise, if* (3.2) *does not hold for any* t, *there exist positive constants* c_1 and c_2 such that

$$\|g_k\| \le c_1(\lambda - 1)^k \lambda^{-c_2\sqrt{3}^k}, \tag{3.4}$$

which means that the convergence rate of $\|g_k\|$ *is R-superlinear. Further, for any* $\epsilon > 0$, $\|g_k\|$ *is two-step* $(3 - \epsilon)$*-Q-superlinearly convergent to zero, namely,*

$$\|g_{k+2}\| = O(\|g_k\|^{3-\epsilon}). \tag{3.5}$$

Proof If (3.2) is true for some t, we have from (2.1) that

$$\alpha_1 = \frac{g_1^T g_1}{g_1^T A g_1} = \frac{2}{1+\lambda}. \tag{3.6}$$

It follows from (1.4) and the definition of g_k that

$$g_{k+1} = (I - \alpha_k A) g_k. \tag{3.7}$$

Hence,

$$g_2 = (I - \alpha_1 A) g_1 = t\left(\frac{\lambda - 1}{\lambda + 1}\right)(1, \mp 1)^T. \tag{3.8}$$

Further, by the induction principle, we can prove that (3.3) holds for all $k \geq 1$.

Now we assume that (3.2) is false for any t. If at least one component of g_1 is zero, the AS method will give the solution in at most two iterations. Thus we can assume that

$$g_1 = \xi_1 \left(\lambda^{m_1}, (-1)^{I_1}\right)^T, \quad \text{where } m_1 \neq 0 \text{ and } I_1 \in \{0, 1\}. \tag{3.9}$$

Generally, suppose that

$$g_k = \xi_k \left(\lambda^{m_k}, (-1)^{I_k}\right)^T, \quad \text{for } k \geq 1. \tag{3.10}$$

It follows from (1.4), (1.5), (1.7), and (2.1) that

$$g_{2k} = (I - \alpha_{2k-1} A) g_{2k-1}, \qquad g_{2k+1} = (I - \alpha_{2k-1} A)^2 g_{2k-1}. \tag{3.11}$$

By (3.10), (3.11), and direct calculations, we can obtain

$$\begin{cases} m_{2k} = -m_{2k-1}, & I_{2k} = I_{2k-1} + 1, \quad \xi_{2k} = \xi_{2k-1} \dfrac{(\lambda - 1)\lambda^{2m_{2k-1}}}{\lambda^{2m_{2k-1}} + \lambda}; \\ m_{2k+1} = -3m_{2k-1}, & I_{2k+1} = I_{2k-1}, \quad \xi_{2k+1} = \xi_{2k-1} \dfrac{(\lambda - 1)^2 \lambda^{4m_{2k-1}}}{(\lambda^{2m_{2k-1}} + \lambda)^2}. \end{cases} \tag{3.12}$$

Thus

$$m_{2k+1} = (-3)^k m_1. \tag{3.13}$$

Assume without loss of generality that $m_1 > 0$, otherwise we can consider x_3 as the new starting point. If $m_{2i-1} \geq 0$, we have by $\lambda = \lambda_2 \geq 1$ that

$$1 \geq \frac{\lambda^{4m_{2i-1}}}{(\lambda^{2m_{2i-1}} + \lambda)^2} \geq \frac{1}{(\lambda + 1)^2}. \tag{3.14}$$

It follows by (3.12), the first inequality in (3.14), $\lambda \geq 1$, and (3.13) that

$$\begin{aligned}|\xi_{4k-1}| &= |\xi_1|\left[\prod_{i=1}^{2k-1}(\lambda-1)^2\frac{\lambda^{4m_{2i-1}}}{(\lambda^{2m_{2i-1}}+\lambda)^2}\right] \leq |\xi_1|(\lambda-1)^{4k-2}\prod_{i=1}^{k-1}\frac{\lambda^{4m_{4i-1}}}{(\lambda^{2m_{4i-1}}+\lambda)^2}\\ &\leq |\xi_1|(\lambda-1)^{4k-2}\prod_{i=1}^{k-1}\lambda^{4m_{4i-1}} = |\xi_1|(\lambda-1)^{4k-2}\prod_{i=1}^{k-1}\lambda^{-4m_1 3^{2i-1}}\\ &= |\xi_1|(\lambda-1)^{4k-2}\lambda^{-1.5m_1[3^{2k-2}-1]}.\end{aligned} \tag{3.15}$$

Relation (3.10) gives

$$\|g_k\| = |\xi_k|\sqrt{1+\lambda^{2m_k}} \tag{3.16}$$

In addition, it is obvious that $\lambda^{-1} \leq \alpha_k \leq 1$, which with (3.7) implies that

$$\|g_{k+1}\| \leq (\lambda-1)\|g_k\|. \tag{3.17}$$

Thus for any k, denoting $k = 4\bar{k} - 1 + j$ where $\bar{k}$ and $j \in [0,3]$ are some integers, we can obtain by (3.15)–(3.17) and $m_{4\bar{k}-1} < 0$ that

$$\begin{aligned}\|g_k\| &\leq (\lambda-1)^j\|g_{4\bar{k}-1}\| \leq \sqrt{2}(\lambda-1)^j|\xi_{4\bar{k}-1}|\\ &\leq \sqrt{2}|\xi_1|(\lambda-1)^{j+4\bar{k}-2}\lambda^{-1.5m_1[3^{2\bar{k}-2}-1]}\\ &\leq \sqrt{2}|\xi_1|(\lambda-1)^{k-1}\lambda^{-1.5m_1[\sqrt{3}^{k-6}-1]}.\end{aligned} \tag{3.18}$$

Therefore (3.4) holds with some $c_1 > 0$ and $c_2 > 0$.

By the first equality in (3.15), (3.12), the second inequality in (3.14), and (3.13), we can also show that

$$|\xi_{4k-1}| \geq |\xi_1|\left(\frac{\lambda-1}{\lambda+1}\right)^{4k-2}\lambda^{-1.5m_1[3^{2k-2}-1]}. \tag{3.19}$$

It follows from (3.10), (3.12), (3.15), and (3.19) that

$$\|g_{4k-1}\| \in |\xi_1|\lambda^{0.5m_{4k-1}+1.5}\left[\left(\frac{\lambda-1}{\lambda+1}\right)^{4k-2}, \sqrt{2}\,(\lambda-1)^{4k-2}\right]. \tag{3.20}$$

Using (3.12) and (3.16), we can obtain without difficulties that

$$\frac{1}{2}\|g_{4k-1}\|\left(\frac{\lambda-1}{\lambda+1}\right)^i\lambda^{\Gamma_i m_{4k-1}} \leq \|g_{4k+i-1}\| \leq \|g_{4k-1}\|(\lambda-1)^i\lambda^{\Gamma_i m_{4k-1}} \tag{3.21}$$

for $i = 1, \ldots, 5$, where

$$\Gamma_1 = \Gamma_2 = 1, \qquad \Gamma_3 = \Gamma_4 = 4, \qquad \Gamma_5 = 13. \tag{3.22}$$

Since by (3.13), m_{4k-1} tends to $-\infty$ exponentially, we know by (3.20)–(3.22) that the following relation holds for any $\epsilon > 0$:

$$\|g_{4k+i+1}\| = O(\|g_{4k+i-1}\|^{3-\epsilon}), \quad i = 1, \ldots, 4. \tag{3.23}$$

Therefore (3.5) holds. This completes our proof. ■

From the above theorem, we know that the convergence properties of the AS method are superior to those of the BB method in the two-dimensional case: (1) For almost all the starting points, the AS method is shown to be two-step $(3-\epsilon)$-Q-superlinearly convergent, whereas only R-superlinear convergence is achieved in [4] for the BB method; (2) As indicated by (3.4), the AS method is R-superlinearly convergent with the R-order of $\sqrt{3}$. This order is greater than the order $\sqrt{2}$ of the BB method.

By the same technique, we can establish the R-superlinear convergence result for the method (2.2), the CSDS method and the CBBS method for two-dimensional SPD linear systems, and obtain their R-orders as follows. For $m \geq 2$, the R-orders of the method (2.2) and the CSDS method are $\sqrt[m]{3}$ and $\sqrt[m]{2m-1}$, respectively. The R-order of the CBBS method is $\sqrt[4]{2}$ if $m = 2$, and

$$\sqrt[m]{m - 1.5 + \sqrt{m^2 - 3m + 0.25}} \quad \text{if } m \geq 3.$$

For clarity, we list the approximate R-orders of all these methods with $m \in [1, 8]$ into Table II. Note that if $m = 1$, both the method (2.2) and the CBBS method reduce to the BB method whereas the CSDS method gives rise to the SD method. Since the SD method is only Q-linearly convergent even for the two-dimensional case, its R-order is only 1. From Table II, we can see that among all these methods, the AS method (namely, (2.2) or (2.3) with $m = 2$) has the largest R-order, whereas the SD method has the smallest order. Meanwhile, we see that the R-orders of the method (2.2) with $m \in [2, 3]$, the CSDS method with $m \in [2, 7]$ and the CBBS method with $m \in [4, 6]$ are greater than the order of the BB method.

To demonstrate the above results, we tested an example for all the methods. Consider the minimization of $q(x) = (1/2)x^T Ax - b^T x$, where

$$A = \text{diag}(20, 10), \quad b = e_2. \tag{3.24}$$

For the purpose of comparison, we assume for all the methods that the starting points are $x_1 = x_2 = 0$ and $\alpha_2 = 1$ although for some methods only one starting point is required and their first stepsizes are well defined. The stopping condition is $\|g_k\| \leq 10^{-16}$. See Table III for the number of iterations required by all the methods. From Table III, we can see that (1) among all the methods, the AS method ((2.2) or

TABLE II Approximate R-orders for the two-dimensional case

m	1	2	3	4	5	6	7	8
(2.2)	1.4142	1.7321	1.4422	1.3161	1.2457	1.2009	1.1699	1.1472
(2.3)	1.0000	1.7321	1.7100	1.6266	1.5518	1.4913	1.4426	1.4029
(2.4)	1.4142	1.1892	1.2599	1.4614	1.4630	1.4361	1.4051	1.3759

TABLE III Numerical comparisons for a two-dimensional example

m	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
(2.2)	16	12	14	16	17	19	20	22
(2.3)	33	12	13	13	15	16	18	20
(2.4)	16	23	17	17	20	22	25	28

(2.3) with $m = 2$) requires the least iterations; (2) the method (2.2), the CSDS method, and the CBBS method with some values of m other than 2 perform better than or similarly to the BB method; and (3) the SD method ((2.3) with $m = 1$) is the slowest method. However, the SD method will significantly be improved if a BB step is done even after every eight SD steps. Table III basically demonstrates our theoretical analyses.

4. CONVERGENCE ANALYSES: ANY-DIMENSIONAL CASE

In this section, we establish the *R*-linear convergence of the AS method applied to any-dimensional SPD linear systems. To make our result more general, however, we define the following property for the stepsize α_k, where $g_k^{(i)}$ is the ith component of g_k and

$$G(k,l) = \sum_{i=1}^{l} \left(g_k^{(i)}\right)^2. \tag{4.1}$$

PROPERTY (A) *Suppose that the matrix* A *has the form of* (3.1), *with* $1 = \lambda_1 \le \cdots \le \lambda_n$. *We say that the stepsize* α_k *has Property* (A) *if there exist an integer* m *and positive constants* $M_1(\ge \lambda_1)$ *and* M_2 *such that* (i) $\lambda_1 \le \alpha_k^{-1} \le M_1$; (ii) *for any integer* $l \in [1, n-1]$ *and real number* $\epsilon > 0$, *if* $G(k-j,l) \le \epsilon$ *and* $(g_{k-j}^{(l+1)})^2 \ge M_2\epsilon$ *hold for* $j \in [0,\ \min\{k,m\} - 1]$, *then* $\alpha_k^{-1} \ge (2/3)\lambda_{l+1}$.

The item (i) means that the inverse stepsize is bounded below and above. By this, we will be able to show that the first eigencomponent of g_k tends to zero *Q*-linearly. The item (ii) requires that for any integer l, the inverse stepsize is so large that the $(l+1)$th eigencomponent of g_k will decrease if the first l eigencomponents of g_k are much smaller. Combining (i) and (ii), we can extend the result in [7] and establish the following *R*-linear convergence theorem. As will be seen later, there are many formulae of stepsize such that Property (A) holds.

THEOREM 4.1 *Consider the linear system* (1.1), *where* A *has the form* (3.1) *with* $1 = \lambda_1 \le \cdots \le \lambda_n$. *Consider the gradient method* (1.4), *where the stepsize* α_k *has Property* (A). *Then either* $g_k = 0$ *for some finite* k, *or the sequence* $\{\|g_k\|\}$ *converges to zero R-linearly*.

Proof It follows from (1.4) and (3.1) that

$$g_{k+1}^{(i)} = (1 - \alpha_k\lambda_i)g_k^{(i)}. \tag{4.2}$$

Denote $\delta_1 = \max\{(1-(\lambda_1/M_1))^2, 1/4\} \in (0,1)$ and $\delta_2 = \max\{(1-(M_1/\lambda_1))^2, 2\}$. Then by (4.2) and the definition of $G(k,l)$, it is easy to get that for all $k \geq 1$,

$$G(k+1,1) \leq \delta_1 G(k,1), \tag{4.3}$$

$$\left(g_{k+1}^{(i)}\right)^2 \leq \delta_2 \left(g_k^{(i)}\right)^2, \quad \text{for } i = 1, 2, \ldots, n, \tag{4.4}$$

$$\|g_{k+1}\|^2 \leq \delta_2 \|g_k\|^2. \tag{4.5}$$

The rest of our proof is divided into the following three parts.

(I) We prove that, for any integer $1 \leq l < n$ and $k \geq 1$, if there exist some $\epsilon_l \in (0, M_2^{-1})$ and integer m_l such that

$$G(k+j,l) \leq \epsilon_l \|g_k\|^2, \quad \text{for all } j \geq m_l, \tag{4.6}$$

then we must have

$$\left(g_{k+j_0}^{(l+1)}\right)^2 \leq M_2 \epsilon_l \|g_k\|^2, \quad \text{for some } j_0 \in [m_l, m_l + m + \Delta_l + 1], \tag{4.7}$$

where

$$\Delta_l = \left\lceil \frac{\log(M_2 \epsilon_l \delta_2^{-(m_l+m)})}{\log \delta_1} \right\rceil. \tag{4.8}$$

In fact, suppose that

$$\left(g_{k+j}^{(l+1)}\right)^2 > M_2 \epsilon_l \|g_k\|^2, \quad \text{for } j \in [m_l, m_l + m + \Delta_l]. \tag{4.9}$$

Then we have from (4.6), (4.9), and Property (A) (ii) that

$$\alpha_{k+j}^{-1} \geq \frac{2}{3}\lambda_{l+1}, \quad \text{for } j \in [m_l + m, m_l + m + \Delta_l]. \tag{4.10}$$

By (4.2), (4.10), and Property (A) (i) , we can get that

$$\left(g_{k+j+1}^{(l+1)}\right)^2 \leq \delta_1 \left(g_{k+j}^{(l+1)}\right)^2, \quad \text{for } j \in [m_l + m, m_l + m + \Delta_l]. \tag{4.11}$$

Thus we obtain from (4.4), (4.11), and the definition of Δ_l that

$$\left(g_{k+m_l+m+\Delta_l+1}^{(l+1)}\right)^2 \leq \delta_1^{\Delta_l+1} \left(g_{k+m_l+m}^{(l+1)}\right)^2 \leq \delta_1^{\Delta_l+1} \delta_2^{m_l+m} \left(g_k^{(l+1)}\right)^2 \leq M_2 \epsilon_l \|g_k\|^2.$$

So the relation (4.7) must hold.

(II) Denoting $m_{l+1} = m_l + m + \Delta_l + 1$ and $\epsilon_{l+1} = (1 + M_2 \delta_2^m)\epsilon_l$, we show that if (4.6) holds, we can further have

$$G(k+j, l+1) \leq \epsilon_{l+1} \|g_k\|^2, \quad \text{for all } j \geq m_{l+1}. \tag{4.12}$$

In fact, by (I), we know that there are infinitely many integers j_1 and j_2 with $j_2 > j_1 \geq j_0$ such that

$$\left(g_{k+j}^{(l+1)}\right)^2 \leq M_2 \epsilon_l \|g_k\|^2, \quad \text{for } j = j_1, j_2 \tag{4.13}$$

and

$$\left(g_{k+j}^{(l+1)}\right)^2 > M_2 \epsilon_l \|g_k\|^2, \quad \text{for } j \in [j_1 + 1, j_2 - 1]. \tag{4.14}$$

Then we have by (4.4) and (4.13) that

$$\left(g_{k+j}^{(l+1)}\right)^2 \leq \delta_2^m \left(g_{k+j_1}^{(l+1)}\right)^2 \leq M_2 \delta_2^m \epsilon_l \|g_k\|^2, \quad \text{for } j \in [j_1 + 1, j_1 + m]. \tag{4.15}$$

If $j_2 > j_1 + m$, then we have by Property (A), (4.2), (4.6), and (4.14) that

$$\alpha_{k+j}^{-1} \geq \frac{2}{3}\lambda_{l+1} \quad \text{and} \quad \left(g_{k+j+1}^{(l+1)}\right)^2 \leq \delta_1 \left(g_{k+j}^{(l+1)}\right)^2, \quad \text{for } j \in [j_1 + m, j_2 - 1]. \tag{4.16}$$

It follows from (4.5), (4.13), and (4.16) that

$$\left(g_{k+j}^{(l+1)}\right)^2 < \left(g_{k+j_1+m}^{(l+1)}\right)^2 \leq M_2 \delta_2^m \epsilon_l \|g_k\|^2, \quad \text{for } j \in [j_1 + m + 1, j_2]. \tag{4.17}$$

Due to the arbitrariness of j_1 and j_2, (4.15) and (4.17), we know that the following relation holds for any $j \geq j_0$:

$$\left(g_{k+j}^{(l+1)}\right)^2 \leq M_2 \delta_2^m \epsilon_l \|g_k\|^2. \tag{4.18}$$

Since $j_0 \leq m_{l+1}$, we then know by (4.6), (4.18) and the definition of $G(k, l)$ that (4.12) holds.

(III) Denoting for any $1 \leq l \leq n$,

$$\epsilon_l = \frac{1}{4}(1 + M_2 \delta_2^m)^{(l-n)}, \tag{4.19}$$

and letting $m_1 = \lceil \log \varepsilon_1 / \log \delta_1 \rceil$, $m_{l+1} = m_l + m + \Delta_l + 1$ for $l = 1, \ldots, n-1$ and $M = m_n$, we prove by induction that for all $1 \leq l \leq n$,

$$G(k + j, l) \leq \epsilon_l \|g_k\|^2, \quad \text{for all } j \geq m_l. \tag{4.20}$$

In fact, by (4.3) and the definition of m_1, relation (4.20) clearly holds for $l = 1$. Suppose that (4.20) is true for some $1 \leq l \leq n-1$. Then by the statement in (II), we know that (4.20) holds for $l + 1$. Thus by induction, we know that (4.20) holds for all $1 \leq l \leq n$. Note that $\epsilon_n = 1/4$ and $G(k, n) = \|g_k\|^2$. It follows from (4.20) that

$$\|g_{k+M}\|^2 \leq \frac{1}{4}\|g_k\|^2. \tag{4.21}$$

Since $M = m_n$ depends only on λ_1, M_1, and M_2, we can then obtain by (4.5) and (4.21) that the sequence $\{\|g_k\|\}$ converges to zero and the convergence rate is R-linear. ■

Assume that m is some positive integer and that $\zeta_j \geq 0, j = 1, \ldots, m$ are a set of real numbers. Also assume that the stepsize α_k can be expressed as the following form

$$\alpha_k = \frac{g_{\nu(k)}^T A^{\rho(k)} g_{\nu(k)}}{g_{\nu(k)}^T A^{(\rho(k)+1)} g_{\nu(k)}}, \tag{4.22}$$

where

$$\nu(k) \in \{k, k-1, \max\{k-m+1, 1\}\} \tag{4.23}$$

and

$$\rho(k) \in \{\zeta_1, \ldots, \zeta_m\}. \tag{4.24}$$

The global convergence of general gradient method (1.4) and (4.22) is shown in [12]. For such method, we can prove that Property (A) holds (see the following proof). Hence by Theorem 4.1, the method is R-linearly convergent.

COROLLARY 4.2 *Consider the linear system* (1.1), *where* A *has the form* (3.1) *with* $1 = \lambda_1 \leq \cdots \leq \lambda_n$. *Consider the gradient method* (1.4) *and* (4.22). *Then we have either* $g_k = 0$ *for some finite* k, *or the sequence* $\{\|g_k\|\}$ *converges to zero* R-*linearly*.

Proof Noting that α_k^{-1} is the Rayleigh quotient of A at the vector $\sqrt{A^{\rho(k)}} g_{\nu(k)}$, Property (A) (i) holds with $M_1 = \lambda_n$. Denote

$$\zeta_{\max} = \max_{i \in [1,m]} \zeta_i \quad \text{and} \quad M_2 = 2\lambda_l^{\zeta_{\max}}. \tag{4.25}$$

Then for any integer $l \in [1, n-1]$ and real number $\epsilon > 0$, if both $G(k-j, l) \leq \epsilon$ and $(g_{k-j}^{(l+1)})^2 \geq M_2\epsilon$ hold for $j \in [0, \min\{k, m\} - 1]$, we have by (4.22), $\zeta_j \geq 0$, and the choice of M_2 that

$$\begin{aligned} \alpha_k^{-1} &\geq \frac{\sum_{i=l+1}^n \lambda_i^{\rho(k)+1} \left(g_{\nu(k)}^{(i)}\right)^2}{\sum_{i=1}^l \lambda_i^{\rho(k)} \left(g_{\nu(k)}^{(i)}\right)^2 + \sum_{i=l+1}^n \lambda_i^{\rho(k)} \left(g_{\nu(k)}^{(i)}\right)^2} \\ &\geq \frac{\lambda_{l+1} \sum_{i=l+1}^n \lambda_i^{\rho(k)} \left(g_{\nu(k)}^{(i)}\right)^2}{\lambda_l^{\zeta_{\max}} G(\nu(k), l) + \sum_{i=l+1}^n \lambda_i^{\rho(k)} \left(g_{\nu(k)}^{(i)}\right)^2} \\ &\geq \frac{M_2\epsilon}{\lambda_l^{\zeta_{\max}}\epsilon + M_2\epsilon} \lambda_{l+1} = \frac{2}{3}\lambda_{l+1}. \end{aligned} \tag{4.26}$$

So, the second part of Property (A) holds. By Theorem 4.1, this corollary is true.

It is easy to see that the AS method, the method (2.2), the CSDS method, and the CBBS method can be written into the form (4.22). Consequently, by Corollary 4.2,

all these methods are *R*-linearly convergent for SPD linear systems. In addition to (4.22), there are also many other choices of α_k having Property (A). For example,

$$\alpha_k = \left(\frac{g_{\nu(k)}^T A^{\rho(k)} g_{\nu(k)}}{g_{\nu(k)}^T A^{(\rho(k)+\mu)} g_{\nu(k)}} \right)^{1/\mu}, \tag{4.27}$$

where μ is any positive number. One application of (4.27) with $\rho(k) = 0$ and $\mu = 2$ can be seen in [8]. More exactly, noting that $s_k = -\alpha_{k-1} g_{k-1}$ and $y_k = A s_k$, the formula 4.27 with $\rho(k) = 0$ and $\mu = 2$ reduces to

$$\alpha_k = \frac{\|s_{\nu(k)+1}\|}{\|y_{\nu(k)+1}\|}, \tag{4.28}$$

which can be regarded as some approximation of the inverse Lipschitz constant. In addition, interestingly enough, one may consider the following convex combination of the SD stepsize and the AS stepsize:

$$\alpha_k = \omega_k \alpha_k^{\mathrm{SD}} + (1 - \omega_k)\alpha_k^{\mathrm{AS}}, \quad \text{where } \omega_k \in [0, 1]. \tag{4.29}$$

The above includes the SD stepsize and the AS stepsize as its special cases. It is not difficult to see that the formula (4.29) has Property (A). Hence we know by Theorem 4.1 that the class of gradient methods (4.29) is *R*-linearly convergent.

An eight-dimensional example is used for demonstrating the convergence result. Consider the function $q(x) = (1/2)x^T A x - b^T x$, where

$$A = \mathrm{diag}\,(2000, 1000, 200, 100, 20, 10, 2, 1), \qquad b = e_8. \tag{4.30}$$

For the purpose of comparison, we assume for all the methods that the starting points are $x_1 = x_2 = 0$ and $\alpha_2 = 1$ as before. The stopping condition is $\|g_k\| \le 10^{-9}$. For this example, the BB and AS methods require 307 and 180 iterations, respectively. Figures 1 and 2 plot the sequences $\{\|g_k\|\}$ generated by the BB and AS

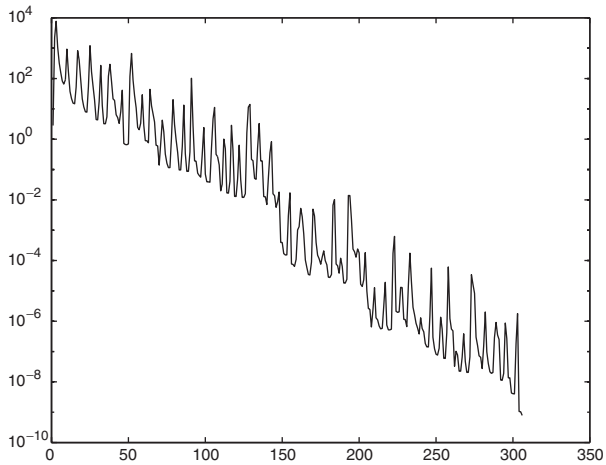

FIGURE 1 $\{\|g_k\|\}$ generated by BB.

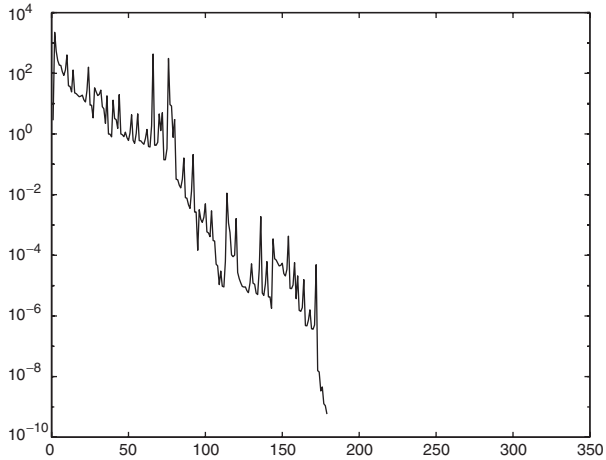

FIGURE 2 $\{\|g_k\|\}$ generated by AS.

TABLE IV Numerical comparisons for an eight-dimensional example

m	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*51*	*101*
(2.2)	307	180	344	372	431	444	524	352	313	414
(2.3)	$>10^4$	180	142	72	84	88	88	76	259	408
(2.4)	307	4881	133	96	79	94	95	92	310	307

methods, respectively. The figures demonstrate the nonmonotone behavior of the BB-like methods and confirm the *R*-linear convergence result to some extent.

We also tested the methods (2.2)–(2.4) with different values of m. See Table IV for the numbers of the iterations required by all the methods. Note again that the SD and BB methods are corresponding to the CSDS and CBBS methods with $m = 1$, respectively. From Table IV, we can see that the SD method is still the slowest, but will significantly be accelerated if a BB step is done even after every 100 SD steps. The CBBS method with $m = 2$ is also very slow. This can partly be explained by Table II, from which we can see that the method has the smallest *R*-order for the two-dimensional case except the SD method.

From Table IV, we also see that the CSDS and CBBS methods with $m \in [4, 8]$ outperform the method (2.2). The AS method performs better than the BB method, but the CSDS and CBBS methods with $m \in [4, 8]$ only require not more than 100 iterations. A possible explanation is given as follows. First we note that, like in the conjugate gradient field, there is also a quadratic termination result in gradient methods (this result can be dated back to at least [16]). More exactly, if $\{\alpha_1^{-1}, \ldots, \alpha_n^{-1}\} = \{\lambda_1, \ldots, \lambda_n\}$ which is the set of all the eigenvalues of A, the gradient method (1.4) gives the exact solution in at most $n + 1$ iterations. As the author observed, the stepsizes generated by the cyclic methods are usually closer to the inverse eigenvalues of A than those by the BB method. This can partly be seen from Table II since the AS method is corresponding to the CSDS method with $m = 2$. Theoretically, we can show that, under suitable assumptions, the Rayleigh

quotient $(g_{k+l}^T A g_{k+l}/g_{k+l}^T g_{k+l})$ generated by the gradient method with constant stepsizes

$$x_{k+l} = x_{k+l-1} - \alpha g_{k+l-1}, \quad \text{for } l = 1, 2, \ldots \tag{4.31}$$

tends to some eigenvalue of the matrix A if $l \to \infty$. Thus the cyclic methods are expected to perform better than the BB method. In a forthcoming paper, we will propose an adaptive strategy of choosing the scalar m so that the stepsizes $\{\alpha_k\}$ can capture the inverse eigenvalues of A better.

5. ALGORITHMS FOR UNCONSTRAINED OPTIMIZATION

In this section, we discuss unconstrained optimization algorithms related to the AS method. Two gradient algorithms will be proposed that are mainly based on the SD and GBB algorithms, respectively.

For convenience in statements, we first describe the SD and GBB algorithms for unconstrained optimization. For the SD algorithm, the following strong Wolfe line search is used while computing the stepsize α_k :

$$f(x_k) - f(x_k - \alpha_k g_k) \geq \delta \alpha_k \|g_k\|^2, \tag{5.1}$$

$$\left| g(x_k - \alpha_k g_k)^T g_k \right| \leq \sigma \|g_k\|^2, \tag{5.2}$$

where the parameters δ and σ are chosen to be 10^{-4} and 0.1, respectively. The unit initial stepsize is also used. The GBB algorithm is the global BB algorithm provided by Raydan [21], in which the nonmonotone line search [14] is used:

$$f(x_k - \alpha_k g_k) \leq \max_{0 \leq j \leq \min(k, M)} \{f(x_{k-j})\} - \delta \alpha_k \|g_k\|^2, \tag{5.3}$$

where $M = 10$ and $\delta = 10^{-4}$. For the GBB algorithm, the BB stepsize (1.4) is used as the initial stepsize, whereas some guaranteeing procedure is used to treat the case that $s_{k-1}^T y_{k-1} \leq 0$. See Algorithm 5.2 for more details in the calculation of the initial stepsize.

Our first algorithm related to the AS method is mainly based on the SD algorithm. Its basic idea is to try to carry out a strong Wolfe line search and a nonmonotone line search alternately. More exactly, if a nonmonotone step is used at the $(k-1)$th iteration, then at the kth iteration a strong Wolfe line search with the unit initial stepsize will be done. On the other hand, if the stepsize α_{k-1} is obtained by a strong Wolfe line search, we test whether α_{k-1} satisfies the nonmonotone line search condition (5.3). If yes, set $\alpha_k = \alpha_{k-1}$; otherwise, carry out a strong Wolfe line search with its initial stepsize being α_{k-1}. A detailed description of this algorithm will be given in the following, where $\bar{\alpha}_k$ stands for the initial stepsize used at the kth iteration. As suggested by one referee, another choice for the initial stepsize in the strong Wolfe line search is the BB stepsize. However, we found that the numerical results are similar to and sometimes much worse than those by the unit initial stepsize. In addition, the calculation of the

BB stepsize needs one vector–vector multiplication. Therefore the unit initial stepsize is used in our tests.

Algorithm 5.1

Step 0 Given $x_1 \in R^n$, $\delta = 10^{-4}$, $\sigma = 0.1$, $\epsilon \geq 0$, and any $M \geq 2$. Let $k := 1$ and $j := 1$.
Step 1 If $\|g_k\|_\infty \leq \epsilon$, stop.
Step 2 If $j = 1$, $\bar{\alpha}_k = 1$. Carry out a strong Wolfe line search, obtaining α_k. Go to Step 4.
Step 3 If $j = -1$ and (5.3) holds for α_{k-1}, set $\alpha_k = \alpha_{k-1}$ and go to Step 4. Otherwise, let $j := 1$, $\bar{\alpha}_k = \alpha_{k-1}$ and carry out a strong Wolfe line search.
Step 4 $x_{k+1} = x_k - \alpha_k g_k$, $k := k + 1$, $j := -j$, go to Step 1.

The second algorithm related to the AS method is mainly based on the GBB algorithm. In the GBB algorithm, a nonmonotone line search is always done except at the first iteration. However, it is possible that some nonmonotone line searches in the algorithm are relatively exact. Assuming that the kth line search is relatively exact, α_k can be regarded as an SD stepsize at the kth iteration and hence as a BB stepsize at the $(k+1)$th iteration. Thus α_k can be used as the initial stepsize of the next nonmonotone line search (we observed that this initial stepsize is usually accepted in our numerical experiments) and the calculation of a new BB stepsize is not necessary. The only problem with this idea of incorporating the AS method and the GBB algorithm is then how to evaluate if a line search is relatively exact. We may use the quantity $(g_{k+1}^T g_k / g_k^T g_k)$ since the previous search direction is $-g_k$. However, this requires the computation of the quantity $g_{k+1}^T g_k$, which is not ready if $\bar{\alpha}_{k+1}$ is not calculated. Instead we use the testing condition

$$\left| \frac{f(x_k) - f(x_{k+1})}{0.5\alpha_k g_k^T g_k} - 1 \right| \leq \eta_1, \tag{5.4}$$

where $\eta_1 \in (0, 1)$ is given. If (5.4) holds, it is reasonable to believe that the function is close to be quadratic along the line $\{x_k - \alpha g_k : \alpha \in R^1\}$. Further, since the new trial stepsize is usually obtained via a quadratic interpolation, we may regard that the line search is relatively exact. The algorithm that uses the testing condition (5.4) is stated as follows:

Algorithm 5.2

Step 0 Given $x_1 \in R^n$, $\delta = 10^{-4}$, $\alpha_{\min} = 10^{-30}$, $\alpha_{\max} = 10^{30}$, $\epsilon \geq 0$, $M = 10$; Set $\bar{\alpha}_1 = \max\{\alpha_{\min}, \min\{1/\|g_1\|_\infty, \alpha_{\max}\}\}$, $k := 1$.
Step 1 If $\|g_k\|_\infty \leq \epsilon$, stop.
Step 2 Carry out a nonmonotone line search, obtaining α_k; $x_{k+1} = x_k - \alpha_k g_k$.
Step 3 If (5.4) holds, $\bar{\alpha}_{k+1} = \alpha_k$ and go to Step 4. If $s_k^T y_k \leq 0$, $\bar{\alpha}_{k+1} = \alpha_{\max}$; otherwise, $\bar{\alpha}_{k+1} = \max\{\alpha_{\min}, \min\{(s_k^T s_k / s_k^T y_k), \alpha_{\max}\}\}$.
Step 4 Set $k := k + 1$ and go to Step 1.

Under the assumption that f is twice continuously differentiable and has bounded level sets, we can prove that both Algorithms 5.1 and 5.2 with $\epsilon = 0$ converge in the sense that

$$\lim_{k \to \infty} \|g_k\| = 0, \tag{5.5}$$

which indicates that every limit point of $\{x_k\}$ is a stationary point of f. To do so, it suffices to notice that both the initial trial stepsize $\bar{\alpha}_k$ and the real stepsize α_k at

every iteration are no less than some positive constant. This can easily be achieved by using the induction principle and the fact that if $\bar{\alpha}_k \geq \eta_2$, then we also have $\alpha_k \geq \eta_3$ no matter whether α_k is obtained by the strong Wolfe line search (5.1), (5.2) or by the nonmonotone line search (5.3).

We have tested the SD algorithm, the GBB algorithm, Algorithms 5.1 and 5.2 on an SGI Indigo workstation with the machine error of 2×10^{-16}. All algorithms are written with the FORTRAN language. The code of the GBB algorithm, that was kindly provided to us by Professor Marcos Raydan, is based on those for the SPG2 algorithm in [3]. Our test problems were taken from [18], except "Strictly Convex 1" and "Strictly Convex 2" that are provided in [21]. The test problems from [18] are numbered in the following way: "MGHi" means the ith problem in [18]. The total number of the test problems is 26.

For all the algorithms, the stopping condition is

$$\|g_k\|_\infty \leq 10^{-6}.$$

The maximal number of function evaluations is set to 9999, and the maximal times of calculating α_k in each line search is set to 20. We mark it with "Failed" if either of these limits is exceeded. The parameter M in Algorithm 5.1 can be set to any integer not less than 2, because we found that the choice of M does not influence the numerical performance provided that $M \geq 2$. For Algorithm 5.2, the parameter c_1 in (5.4) is set to be 0.001. The SD algorithm performs very poorly and can solve only 14 of the problems. Hence its numerical results are not listed. The results of the other algorithms are reported in Table V, where n denotes the dimension of the problem, and I, F, G are number of iterations, number of function evaluations, and number of gradient evaluations, respectively. For the GBB algorithm and Algorithm 5.2, the number of gradient evaluations is equal to that of iterations, since no gradient evaluation is required in the line search procedure. If a problem is successfully solved by the algorithm, then the CPU time (in second) is given; otherwise, we do not give the CPU time and use the symbol '*'. The symbol '-' means that the performance of Algorithm 5.2 is the same as that of the GBB algorithm. For Algorithm 5.2, the number of iterations at which the condition (5.4) holds are also listed in the brackets of Table V.

Algorithm 5.1 solves almost all of the test problems and is much faster than the SD algorithm. Comparing Algorithm 5.1 with the GBB algorithm, we see that the GBB algorithm is robuster and solves all the problems. However, for the three problems Algorithm 5.1 failed, the GBB algorithm also takes a large number of iterations. Among the rest 23 test problems, Algorithm 5.1 and the GBB algorithm win 11 and 12 respectively from the CPU time. Further, for the 11 problems with $n \geq 1000$, Algorithm 5.1 and the GBB algorithm win 7 and 4, respectively. For the very difficult problem "Strictly Convex 2" with $n = 10000$, Algorithm 5.1 only requires 61.20 s, that are significantly less than the CPU time 109.7 s required by the GBB algorithm. The comparison of Algorithm 5.1 and the GBB algorithm might indicate that the gradient method for unconstrained optimization is far from maturity.

Comparing the GBB algorithm and Algorithm 5.2, we found that the strategy of testing (5.4) influenced only 8 of the 26 test problems. For the eight problems, Algorithm 5.2 is uniformly much better than the GBB algorithm. This further shows that the introduction of the AS method is worthwhile although Algorithm 5.2 is

TABLE V Numerical comparisons of gradient algorithms

Method		*Algorithm 5.1*		*GBB*		*Algorithm 5.2*	
Problem	*n*	*I*/*F*/*G*	*Time*	*I*/*F*	*Time*	*I*/*F*	*Time*
MGH11	3	*Failed*	*	949/2507	2.960	778/2051(9)	2.461
MGH14	4	282/646/355	8.164e−03	163/329	3.345e−3	–	–
MGH18	6	1061/2068/1512	0.3590	1091/2042	0.3010	744/1456(21)	0.2105
MGH22	16	94/211/130	5.6000e−3	466/776	2.364e−2	233/393(9)	1.260e−2
MGH24	20	474/1184/762	0.1149	708/1939	0.1594	377/1046(6)	8.512e−2
	40	223/523/337	9.743e−2	258/527	9.600e−2	–	–
MGH28	20	*Failed*	*	907/923	0.3095	–	–
	50	*Failed*	*	6967/7018	12.17	–	–
MGH30	50	35/75/42	6.457e−3	38/39	6.226e−3	–	
	500	41/89/50	7.355e−2	36/37	6.107e−2	–	–
MGH31	50	34/84/53	2.626e−2	30/31	1.437e−2	–	
	500	26/59/33	0.1791	29/30	0.1451	–	
MGH22	100	94/211/130	2.482e−2	272/468	7.663e−02	219/362(4)	6.088e−2
	500	94/211/130	0.1331	425/755	0.6456	221/384(6)	0.3306
MGH25	100	11/50/22	5.770e−3	1/2	8.010e−4	–	
	1000	17/98/39	0.1090	1/2	5.318e−3	–	–
MGH21	1000	101/295/212	0.4182	53/279	0.2386	–	
	10 000	69/239/160	3.406	53/279	2.532	–	–
MGH23	1000	49/134/86	0.1469	56/251	0.2805	–	
	10 000	49/134/86	1.559	64/163	2.612	–	–
MGH26	1000	80/148/122	0.9051	89/205	1.161	–	
	10 000	38/66/57	4.235	83/107	7.408	–	–
Strictly	1000	4/6/6	1.951e−2	5/6	2.517e−2	–	
convex 1	10 000	4/6/6	0.1967	5/6	0.2561	–	–
Strictly	1000	459/1272/976	3.574	533/786	2.702	451/620(4)	1.813
convex 2	10 000	961/2274/1356	61.20	2091/3205	109.7	1516/2278(41)	90.97

mainly based on the GBB algorithm, and that it still remains to study how to design more efficient gradient methods for unconstrained optimization.

6. CONCLUSIONS AND DISCUSSION

In this article, motivated by how to avoid producing zigzags, we have proposed the alternate step (AS) gradient method. The AS method uses an SD step and a BB step alternately, and is superior to the BB method for symmetric and positive definite linear systems. *R*-linear convergence is established for a collection of gradient methods with Property (A). Gradient methods related to the AS method and an extension of the AS method to unconstrained optimization have carefully been discussed in this article. A more efficient gradient algorithm for unconstrained optimization is also provided by exploring the idea of the AS method in the GBB algorithm of Raydan [21].

Up to now, it seems difficult to compare different gradient methods in the higher-dimensional case even when $n = 3$ and hence the analyses of two-dimensional SPD linear systems are important. We have proved that if $n = 2$, the AS method is two-step $(3 - \epsilon)$-*Q*-superlinearly convergent for almost all the starting points. This result is superior to that for the BB method, for which only the *R*-linear convergence result is established. Our two-dimensional analyses have also shown that some other gradient methods such as the method (2.2), the CSDS method, and the CBBS method with suitable values of m are faster than the BB method.

In the any-dimensional case, we have drawn a property, namely, Property (A), of the stepsize with which a gradient method can be proved R-linearly convergent. As a result, a large collection of gradient methods including the AS, the CSDS, and the CBBS methods are proved to be R-linearly convergent. Our numerical experiments show that the cyclic methods, either CSDS or CBBS, are very efficient. Although we still do not know yet how to understand why these R-linear gradient methods are better than the SD method that is Q-linearly convergent, we have observed that the cyclic methods generate stepsizes that are much closer to the inverse eigenvalues of the coefficient matrix A. However, further research is required on how to choose the stepsizes in gradient methods efficiently under the supervision of the quadratic termination property of the gradient method.

This article has also provided some interesting insights into the use of nonmonotone steps in the gradient methods. To minimize a smooth function, it is natural to require the function value to be monotonically decreasing at each iteration. The SD method is just this type of method, but it is very slow. As seen from Tables III and IV, however, the SD method can significantly be accelerated if a BB step, that is normally a nonmonotone step, is done even after many SD steps. Another interesting point is about the choice of M in the nonmonotone line search. We know by [21] that the GBB algorithm works well for unconstrained optimization, but its numerical performance sometimes is very sensitive to the choice of the parameter M in (5.3). On the other hand, our numerical experiments showed that Algorithm 5.1 is independent of the value of M provided that $M \geq 2$. Comparing the GBB algorithm and Algorithm 5.1, the former is robuster and the latter provides better performance for large-scale test problems. Therefore it still remains to study how to design robuster and more efficient gradient algorithms for unconstrained optimization.

Acknowledgments

The author thanks Prof. Shao-lin Xi for providing the reference [16], Prof. Ya-xiang Yuan for his interesting discussions with the author about monotonicity and nonmonotonicity, and Dr. Da-chuan Xu for his careful reading of an early version of this article. Many thanks are also due to the referees, whose comments and suggestions improved this article greatly. This research was supported by the Chinese NSF grants 19801033 and 10171104 and a youth innovation fund of Chinese Academy of Sciences.

References

[1] H. Akaike (1959). On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method. *Ann. Inst. Statist. Math. Tokyo*, **11**, 1–17.
[2] E.G. Birgin, I. Chambouleyron and J.M. Martínez, (1999). Estimation of the optical constants and the thickness of thin films using unconstrained optimization. *J. Comput. Phys.*, **151**, 862–880.
[3] E.G. Birgin, J.M. Martínez and M. Raydan (2000). Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optim.*, **10**, 1196–1211.
[4] J. Barzilai and J.M. Borwein (1988). Two-point step size gradient methods. *IMA J. Numer. Anal.*, **8**, 141–148.
[5] A. Cauchy (1847). Méthode générale pour la résolution des systéms d'équations simultanées. *Comp. Rend. Sci. Paris*, **25**, 536–538.
[6] Y.H. Dai (2002). On the nonmonotone line search. *Journal of Optimization Theory and Applications*, **112**(2), 315–330.

[7] Y.H. Dai and L.-Z. Liao (2002). *R*-linear convergence of the Barzilai and Borwein gradient method. *IMA J. Numer. Anal.*, **22**, 1–10.
[8] Y.H. Dai and J.Y. Yuan (2001). On a class of gradient algorithms for unconstrained optimization (Preprint).
[9] R. Fletcher (1999). Low storage methods for unconstrained optimization. *Lectures in Applied Mathematics* (*AMS*), **26**, 165–179.
[10] R. Fletcher (2001). The Barzilai Borwein Method – Steepest Descent Resurgent? *Report in the International Workshop on* "*Optimization and Control with Applications*", July 9–17. Erice, Italy.
[11] R. Fletcher and C. Reeves, (1964). Function minimization by conjugate gradients. *Comput. J.*, **7**, 149–154.
[12] A. Friedlander, J.M. Martínez, B. Molina and M. Raydan (1999). Gradient method with retards and generalizations. *SIAM J. Numer. Anal.*, **36**, 275–289.
[13] W. Glunt, T.L. Hayden and M. Raydan (1993). Molecular conformations from distance matrices. *J. Comput. Chem.*, **14**, 114–120.
[14] L. Grippo, F. Lampariello and S. Lucidi (1986). A nonmonotone line search technique for Newton's method. *SIAM J. Numer. Anal.*, **23**, 707–716.
[15] M.R. Hestenes and E.L. Stiefel (1952). Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Standards Sect.*, **5**(49), 409–436.
[16] Y.L. Lai, (1981). Some properties of the steepest descent method. *Acta Mathematicae Applicatae Sinica*, **4**(2), 106–116.
[17] W.B. Liu and Y.H. Dai (2001). Minimization algorithms based on supervisor and searcher cooperation. *Journal of Optimization Theory and Applications*, **111**(2), 359–379.
[18] J.J. Moré, B.S. Garbow and K.E. Hillstrom (1981). Testing unconstrained optimization software. *ACM Transactions on Mathematical Software*, **7**, 17–41.
[19] J. Nocedal, A. Sartentaer C. Zhu (2002). On the behavior of the gradient norm in the steepest descent method. *Computational Optimization and Applications*, **22**(1), 5–35.
[20] M. Raydan (1993). On the Barzilai and Borwein choice of steplength for the gradient method. *IMA J. Numer. Anal.*, **13**, 321–326.
[21] M. Raydan (1997). The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM J. Optim.*, **7**, 26–33.
[22] M. Raydan (2001). Nonmonotone spectral methods for large-scale nonlinear systems. *Report in the International Workshop on* "*Optimization and Control with Applications*", July 9–17. Erice, Italy.