SIAM J. NUMER. ANAL.
Vol. 36, No. 1, pp. 275–289

GRADIENT METHOD WITH RETARDS AND GENERALIZATIONS*

A. FRIEDLANDER†, J. M. MARTÍNEZ†, B. MOLINA‡, AND M. RAYDAN‡

Abstract. A generalization of the steepest descent and other methods for solving a large scale symmetric positive definitive system $Ax = b$ is presented. Given a positive integer m, the new iteration is given by $x_{k+1} = x_k - \lambda(x_{\nu(k)})(Ax_k - b)$, where $\lambda(x_{\nu(k)})$ is the steepest descent step at a previous iteration $\nu(k) \in \{k, k-1, \ldots, \max\{0, k-m\}\}$. The global convergence to the solution of the problem is established under a more general framework, and numerical experiments are performed that suggest that some strategies for the choice of $\nu(k)$ give rise to efficient methods for obtaining approximate solutions of the system.

Key words. gradient method, Barzilai–Borwein method, Rayleigh quotient, conjugate gradient method, symmetric successive overrelaxation (SSOR) preconditioning strategy

AMS subject classifications. 65F10, 49M07, 65F15

PII. S003614299427315X

1. Introduction.

We are interested in the problem

$$\text{Minimize } f(x) \equiv \frac{1}{2}x^t Ax - b^t x, \tag{1.1}$$

where A is a real symmetric positive definite (SPD) $n \times n$ matrix. The global solution of (1.1) is the unique solution of $Ax = b$. The well-known steepest descent (or gradient) method is defined by the iteration

$$x_{k+1} = x_k - \lambda(x_k)g(x_k),$$

where $\lambda(x)$ is the minimizer of $\varphi(\lambda) \equiv f(x - \lambda g(x))$ and $g(x) = Ax - b$. Hence,

$$\lambda(x) = \frac{g(x)^t g(x)}{g(x)^t Ag(x)} \tag{1.2}$$

for all $x \in R^n$ such that $g(x) \neq 0$. This classical method is globally convergent and its storage requirements are minimal, but its rate of convergence is very low in critical cases (see [11]).

Barzilai and Borwein [3] proposed the iteration

$$x_{k+1} = x_k - \lambda(x_{k-1})g(x_k)$$

for solving $Ax = b$, where the first steplength "$\lambda(x_{-1})$" is arbitrary. Raydan [18] proved that the Barzilai–Borwein iteration is globally convergent and showed [17] that it is much more efficient than the steepest descent method for solving large scale positive definite linear systems of equations.

*Received by the editors August 22, 1994; accepted for publication (in revised form) January 26, 1998; published electronically, January 5, 1999.
http://www.siam.org/journals/sinum/36-1/27315.html

†Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil (friedlan@ime.unicamp.br, martinez@ime.unicamp.br). The first and second authors were supported by FAPESP grant 90-3724-6, FINEP, and FAEP-UNICAMP.

‡Departamento de Computación, Facultad de Ciencias, Universidad Central de Venezuela, Ap. 47002, Caracas 1041-A, Venezuela (bmolina@reacciun.ve, mraydan@reacciun.ve). The third author was supported by the Parallel and Distributed Computing Center at UCV. The fourth author was supported by BID-CONICIT project M-51940 and FAEP-UNICAMP.

The gradient method with retards, introduced in this paper, is a generalization of the steepest descent and the Barzilai–Borwein methods. Given a positive integer m, the new iteration is

$$x_{k+1} = x_k - \lambda(x_{\nu(k)})g(x_k), \tag{1.3}$$

where $\nu(k)$ is arbitrarily chosen in the set $\{k, k-1, \ldots, \max\{0, k-m\}\}$. A practical motivation for the introduction of this iteration is that it allows us to compute $\lambda(x_k)$ simultaneously in parallel with x_{k+1}. Observe that, even in the efficient conjugate gradient method, the computation of the steplength associated to the search direction necessarily precedes the computation of the new iterate. In this paper, we prove that the iteration (1.3) is globally convergent to the solution of the problem. From the practical point-of-view, we give numerical experiments that suggest that some strategies for the choice of $\nu(k)$ give rise to efficient methods for obtaining approximate solutions of $Ax = b$.

In the convergence analysis we state the iteration (1.3) under a more general framework. Observe that, by (1.2), the recurrence (1.3) is a particular case of

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

where $g_k = g(x_k)$ and the scalar α_k is the Rayleigh quotient of the matrix A at an n-dimensional vector, whose choice depends on the index k. For example, Barzilai and Borwein gave a second choice for the steplength that is covered by (1.4) but not by (1.3). This choice is

$$\alpha_k = \frac{y_{k-1}^t y_{k-1}}{s_{k-1}^t y_{k-1}} = \frac{s_{k-1}^t A^2 s_{k-1}}{s_{k-1}^t A s_{k-1}}, \tag{1.5}$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$. In this case, α_k is the Rayleigh quotient of A at the vector $\sqrt{A}s_{k-1}$. Our analysis corresponds to the iteration (1.4) for a family of choices of α_k that include (1.3) and (1.5).

This paper is organized as follows. In section 2 we establish the convergence results. The analysis that we present is an extension of the analysis presented in [18] for the particular choice of the Barzilai–Borwein method. The structure of the proofs will be the same as in [18]. However, important differences come from the fact that now the scalars α_k might be arbitrarily chosen from a finite collection of candidates. In section 3, we discuss this collection of possible practical choices, and we establish that some well-known methods can now be viewed as particular cases of this new scheme. We present some numerical experiments to demonstrate the potential of this new method. In section 4, we discuss preconditioning strategies. We present numerical results for elliptic partial differential equations to illustrate the behavior of this method when compared to the preconditioned conjugate gradient (PCG) method. Finally, in section 5 we present concluding remarks.

2. Convergence analysis. Let x_* be the unique minimizer of f. Assume that $\{x_k\}$ is the sequence generated by (1.4) from a given vector x_0, where α_k is an arbitrary Rayleigh quotient, that is $\alpha_k = u_k^t A u_k / u_k^t u_k$ for some $u_k \neq 0$. We define $e_k = x_* - x_k$ for all k. Using (1.4) and the fact that $g_k = Ax_k - b$, we have

$$Ae_k = \alpha_k s_k \qquad \text{for all } k. \tag{2.1}$$

Substituting $s_k = e_k - e_{k+1}$ in (2.1) we obtain for any k

$$e_{k+1} = \frac{1}{\alpha_k}(\alpha_k I - A)e_k. \tag{2.2}$$

Now for any initial error e_0, there exist constants $d_1^0, d_2^0, \ldots, d_n^0$ such that

$$e_0 = \sum_{i=1}^{n} d_i^0 v_i,$$

where $\{v_1, v_2, \ldots, v_n\}$ are orthonormal eigenvectors of A associated with the eigenvalues $\{\sigma_1, \sigma_2, \ldots, \sigma_n\}$. Throughout this work we assume that

$$0 < \sigma_{min} = \sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_n = \sigma_{max} .$$

Using (2.2) we obtain for any integer k,

$$e_{k+1} = \sum_{i=1}^{n} d_i^{k+1} v_i, \tag{2.3}$$

where

$$d_i^{k+1} = \left(\frac{\alpha_k - \sigma_i}{\alpha_k}\right) d_i^k = \prod_{j=0}^{k} \left(\frac{\alpha_j - \sigma_i}{\alpha_j}\right) d_i^0. \tag{2.4}$$

The convergence properties of the sequence $\{e_k\}$ depend on the behavior of each one of the sequences $\{d_i^k\}$, $1 \leq i \leq n$. In general, these sequences increase at some iterations. However, the following lemma shows that the sequence $\{d_1^k\}$ decreases Q-linearly to zero independently of the Rayleigh quotient of A that is being used, at every k, to obtain the scalar α_k.

LEMMA 2.1. *If the scalar α_k in (1.4) is always chosen as the Rayleigh quotient of A at any arbitrary nonzero vector, then the sequence $\{d_1^k\}$ converges to zero Q-linearly with convergence factor $\hat{c} = 1 - (\sigma_{min}/\sigma_{max})$.*

Proof. For any positive integer k,

$$d_1^{k+1} = \left(\frac{\alpha_k - \sigma_{min}}{\alpha_k}\right) d_1^k.$$

Since α_k satisfies

$$0 < \sigma_{min} \leq \alpha_k \leq \sigma_{max} \tag{2.5}$$

for all k, we have

$$|d_1^{k+1}| = \left(1 - \frac{\sigma_{min}}{\alpha_k}\right)|d_1^k| \leq \hat{c}|d_1^k|,$$

where $\hat{c} = 1 - \frac{\sigma_{min}}{\sigma_{max}} < 1$. ∎

We now describe the collection of possible choices of the scalar α_k that allows us to obtain global convergence of the iteration (1.4). Given a positive integer m, and $q_j \geq 1, j = 1, \ldots, m$ a set of real numbers, we set

$$\alpha_k = \frac{e_{\nu(k)}^t A^{\rho(k)} e_{\nu(k)}}{e_{\nu(k)}^t A^{(\rho(k)-1)} e_{\nu(k)}}, \tag{2.6}$$

where

$$\nu(k) \in \{k, k-1, \ldots, \max\{0, k-m\}\}$$

and

$$\rho(k) \in \{q_1, \ldots, q_m\}$$

for all $k = 0, 1, 2, \ldots$.

Notice that the scalar α_k given by (2.6) is the Rayleigh quotient of A at the vector $\sqrt{A^{(\rho(k)-1)}}\, e_{\nu(k)}$, and so it satisfies (2.5) for all k. Later, in section 3, we discuss some well-known methods and some new methods that can be embedded into the scheme given by (1.4) and (2.6). In the proof of our convergence theorem, we will use the following result.

LEMMA 2.2. *Let $\{e_k\}$ be the sequence of errors generated by the method given by* (1.4) *and* (2.6). *If the sequences $\{d_1^k\}, \{d_2^k\}, \ldots, \{d_l^k\}$, defined in* (2.4) *converge to zero for a fixed integer l, $1 \le l < n$, then,*

$$\liminf_{k\to\infty} |d_{l+1}^k| = 0\ .$$

Proof. Suppose, by way of contradiction, that there exists a constant $\varepsilon > 0$ such that

$$(d_{l+1}^k)^2 \min_{1\le j\le m} \sigma_{l+1}^{(q_j-1)} \ge \varepsilon \qquad \text{for all } k. \tag{2.7}$$

By (2.3), (2.6), and the orthonormality of the eigenvectors $\{v_1, v_2, \ldots, v_n\}$, the Rayleigh quotient α_k can be written as

$$\alpha_k = \frac{\sum_{r=1}^n (d_r^{\nu(k)})^2 \sigma_r^{\rho(k)}}{\sum_{r=1}^n (d_r^{\nu(k)})^2 \sigma_r^{(\rho(k)-1)}}. \tag{2.8}$$

Since the sequences $\{d_1^k\}, \ldots, \{d_l^k\}$ all converge to zero, there exists $\hat{k}$ sufficiently large such that

$$\max_{1\le j\le m} \left(\sum_{r=1}^{l} (d_r^k)^2 \sigma_r^{(q_j-1)} \right) \le \frac{\varepsilon}{2} \qquad \text{for all } k \ge \hat{k}. \tag{2.9}$$

By (2.8) and (2.9), we obtain for any $k \ge \hat{k} + m$,

$$\frac{(\sum_{r=l+1}^n (d_r^{\nu(k)})^2 \sigma_r^{(\rho(k)-1)})\sigma_{l+1}}{\frac{\varepsilon}{2} + (\sum_{r=l+1}^n (d_r^{\nu(k)})^2 \sigma_r^{(\rho(k)-1)})} \le \alpha_k \le \sigma_{max}. \tag{2.10}$$

Since

$$\sum_{r=l+1}^{n} (d_r^{\nu(k)})^2 \sigma_r^{(\rho(k)-1)} \ge (d_{l+1}^{\nu(k)})^2 \sigma_{l+1}^{(\rho(k)-1)} \ge \varepsilon,$$

then it follows from (2.10) that

$$\frac{2}{3}\sigma_{l+1} \le \alpha_k \le \sigma_{max} \qquad \text{for all } k \ge \hat{k} + m,$$

which implies the bound

$$\left|1-\frac{\sigma_{l+1}}{\alpha_k}\right| \le \max\left(\frac{1}{2}, 1-\frac{\sigma_{l+1}}{\sigma_{max}}\right) \quad \text{for all } k \ge \hat{k}+m. \tag{2.11}$$

Finally, using (2.11) and the first part of (2.4), we obtain for all $k \ge \hat{k}+m$,

$$|d_{l+1}^{k+1}| = \left|1-\frac{\sigma_{l+1}}{\alpha_k}\right| |d_{l+1}^k| \le \hat{c}|d_{l+1}^k|,$$

where

$$\hat{c} = \max\left(\frac{1}{2}, 1-\frac{\sigma_{min}}{\sigma_{max}}\right) < 1. \tag{2.12}$$

Because this conclusion contradicts the hypothesis (2.7), we find that the lemma is true. ∎

Theorem 2.1 establishes the convergence of the method defined by (1.4) and (2.6) when applied to a quadratic function with a symmetric positive definite Hessian.

THEOREM 2.1. *Let $f(x)$ be a strictly convex quadratic function. Let $\{x_k\}$ be the sequence generated by* (1.4) *and* (2.6) *and x_* the unique minimizer of f. Then, either $x_j = x_*$ for some finite j, or the sequence $\{x_k\}$ converges to x_*.*

Proof. We need only consider the case in which there is no finite integer j such that $x_j = x_*$. Hence, it suffices to prove that the sequence $\{e_k\}$ converges to zero. From (2.3) and the orthonormality of the eigenvectors we have

$$\|e_k\|_2^2 = \sum_{r=1}^{n} (d_r^k)^2.$$

Therefore, the sequence of errors $\{e_k\}$ converges to zero if and only if each one of the sequences $\{d_i^k\}$ for $i = 1, 2, \ldots, n$ converges to zero.

Lemma 2.1 shows that $\{d_1^k\}$ converges to zero. We prove that $\{d_p^k\}$ converges to zero for $2 \le p \le n$ by induction on p. Therefore, we let p be any integer from this interval, and we assume that $\{d_1^k\}, \ldots, \{d_{p-1}^k\}$ all tend to zero. Then for any given $\varepsilon > 0$ there exists $\hat{k}$ sufficiently large such that

$$\max_{1\le j\le m} \sum_{r=1}^{p-1} (d_r^k)^2 \sigma_r^{(q_j-1)} < \frac{\varepsilon}{2} \quad \text{for all } k \ge \hat{k}. \tag{2.13}$$

From (2.8) and (2.13), we obtain

$$\frac{(\sum_{r=p}^{n} (d_r^{\nu(k)})^2 \sigma_r^{(\rho(k)-1)})\sigma_p}{\frac{\varepsilon}{2} + (\sum_{r=p}^{n} (d_r^{\nu(k)})^2 \sigma_r^{(\rho(k)-1)})} \le \alpha_k \le \sigma_{max}, \tag{2.14}$$

for all integers $k \ge \hat{k}+m$. Moreover, by Lemma 2.2, there exists $k_p \ge \hat{k}+m$ such that

$$(d_p^{k_p})^2 \max_{1\le j\le m} \sigma_p^{(q_j-1)} < \varepsilon.$$

Now, consider any integer $k_0 > k_p$ such that $\max_j (d_p^{k_0-1})^2 \sigma_p^{(q_j-1)} < \varepsilon$ and $\max_j (d_p^{k_0})^2 \sigma_p^{(q_j-1)} \geq \varepsilon$. Clearly, for $k_0 \leq k \leq j_0 - 1$,

$$\max_{1\leq j\leq m} \sum_{r=p}^{n} (d_r^k)^2 \sigma_r^{(q_j-1)} \geq \max_{1\leq j\leq m} (d_p^k)^2 \sigma_p^{(q_j-1)} \geq \varepsilon, \tag{2.15}$$

where j_0 is the first integer greater than k_0 for which $\max_j (d_p^{j_0})^2 \sigma_p^{(q_j-1)} < \varepsilon$. Then, by (2.14) and (2.15), and using an argument similar to the one used in the proof of Lemma 2.2, we have

$$\frac{2}{3}\sigma_p \leq \alpha_k \leq \sigma_{max} \qquad \text{for } k_0 \leq k \leq j_0 - 1. \tag{2.16}$$

Thus, using (2.16) and the first part of (2.4), we obtain

$$|d_p^{k+2}| \leq \hat{c} |d_p^{k+1}| \qquad \text{for } k_0 \leq k \leq j_0 - 1,$$

where $\hat{c}$ is the constant (2.12), which satisfies $\hat{c} < 1$. Finally, using the bound

$$|d_p^{k_0+1}| \leq \left(\frac{\sigma_{max} - \sigma_{min}}{\sigma_{min}} \right)^2 |d_p^{k_0-1}| \, ,$$

which is implied by expression (2.5) and the first part of (2.4), we conclude that

$$(d_p^k)^2 \leq \left(\frac{\sigma_{max} - \sigma_{min}}{\sigma_{min}} \right)^4 (d_p^{k_0-1})^2 \leq \left(\frac{\sigma_{max} - \sigma_{min}}{\sigma_{min}} \right)^4 \frac{\varepsilon}{(\max_j \sigma_p^{(q_j-1)})}$$

for all $k_0 + 1 \leq k \leq j_0 + 1$. Further, (2.4) provides the inequality $(d_p^{k_0})^2 \leq ((\sigma_{max} - \sigma_{min})/\sigma_{min})^2 (d_p^{k_0-1})^2$. It follows from the conditions on k_0, k_p, and j_0 that $(d_p^k)^2$ is bounded above by a constant multiple of ε for all $k \geq k_p$. Hence, since $\varepsilon > 0$ can be chosen to be arbitrarily small, we deduce $\lim_{k\to\infty} |d_p^k| = 0$ as required, which completes the proof. ∎

It is worth mentioning that a straightforward extension of Theorem 2.3 can be made for the (not necessarily strictly) convex case. This extension has been done in [9] for the Barzilai–Borwein method.

3. Classical and new choices of α_k. If $\rho(k) = 3$ for all k, we see, using (2.1) that formula (2.6) takes the form

$$\alpha_k = \frac{s_{\nu(k)}^t A s_{\nu(k)}}{s_{\nu(k)}^t s_{\nu(k)}} = \frac{g_{\nu(k)}^t A g_{\nu(k)}}{g_{\nu(k)}^t g_{\nu(k)}}. \tag{3.1}$$

Therefore, in this case the method defined by (1.4) and (2.6) is the gradient method with retards introduced in section 1. On the other hand, the second method of Barzilai and Borwein, given by (1.5), corresponds to $\rho(k) \equiv 4$ and $\nu(k) \equiv k-1$. Consequently, the methods defined by $\rho(k) \equiv 4$ are generalizations of the second Barzilai–Borwein method.

We can define the following gradient methods with retards, with the same cost of the steepest descent method, both in terms of storage and work per iteration. We define first $\overline{k} = \max\{0, k - m\}$.

$$\nu(k) = \text{ random integer between } \overline{k} \text{ and } k; \tag{3.2}$$

$$\nu(k) = k \text{ if } \nu(k-1) < \overline{k} \text{ or } k = 0, \quad \nu(k) = \nu(k-1) \text{ otherwise;} \tag{3.3}$$

$$\nu(k) = \overline{k}; \tag{3.4}$$

$$\nu(k) = \text{ argmax } \{\lambda(x_k), \ldots, \lambda(x_{\overline{k}})\}; \tag{3.5}$$

$$\nu(k) = \text{ argmin } \{\lambda(x_k), \ldots, \lambda(x_{\overline{k}})\}; \tag{3.6}$$

$$\nu(k) = \overline{k} \text{ if } k \text{ is even, } \nu(k) = k \text{ if } k \text{ is odd.} \tag{3.7}$$

$$\nu(k) = \text{ random integer between } \overline{k} \text{ and } k-1. \tag{3.8}$$

From now on, we consider only gradient methods with retards as defined by (1.3). The following result should help us to decide which strategies, among those defined above, could be efficient.

THEOREM 3.1. *Let $\{s_k\}$ be the sequence generated by* (1.4) *and* (3.1). *Assume that the sequence $\{s_k/\|s_k\|\}$ is convergent, that is, there exists $s \in R^n$, $\|s\| = 1$, such that*

$$\lim_{k\to\infty} \frac{s_k}{\|s_k\|} = s. \tag{3.9}$$

Then

$$\lim_{k\to\infty} \alpha_k = s^t A s, \tag{3.10}$$

s is an eigenvector of A with eigenvalue s^tAs and the convergence of $\{x_k\}$ is Q-superlinear.

Proof. By (3.1) we have that

$$\alpha_k = \frac{s_{\nu(k)}^t}{\|s_{\nu(k)}\|} A \frac{s_{\nu(k)}}{\|s_{\nu(k)}\|}, \tag{3.11}$$

where $\nu(k) \geq k - m$. By (3.9),

$$\lim_{k\to\infty} \frac{s_{\nu(k)}}{\|s_{\nu(k)}\|} = s,$$

so (3.10) follows taking limits on both sides of (3.11).

Now, combining (2.1) and (2.2) it follows that

$$s_{k+1} = -\frac{1}{\alpha_{k+1}}(A - \alpha_k I)s_k.$$

Therefore,

$$\frac{s_{k+1}}{\|s_{k+1}\|} = -\frac{(A - \alpha_k I)s_k/\|s_k\|}{\|(A - \alpha_k I)s_k/\|s_k\|\|}. \tag{3.12}$$

Define $\alpha = s^t As$ and

$$c = \|(A - \alpha I)s\| = \lim_{k\to\infty} \|(A - \alpha_k I)s_k/\|s_k\|\|.$$

Suppose for a moment that $c \neq 0$. Taking limits on both sides of (3.12) we obtain

$$s = -\frac{1}{c}(A - \alpha I)s.$$

Therefore,

$$(A - (\alpha - c)I)s = 0.$$

This means that s is an eigenvector of A, with eigenvalue $\alpha - c$. Therefore, $\alpha = s^t As = \alpha - c$ and, thus, $c = 0$. So, $c = \|(A - \alpha I)s\| = 0$ and, consequently, s is an eigenvector of A with eigenvalue α, as we wanted to prove. Hence,

$$\lim_{k\to\infty} \frac{(A - \alpha_k I)s_k}{\|s_k\|} = (A - \alpha I)s = 0. \tag{3.13}$$

But (3.13) is the Dennis–Moré sufficient condition (see [5]) for superlinear convergence of the quasi-Newton iteration $x_{k+1} = x_k - B_k^{-1} g(x_k)$. In this case, $B_k = \alpha_k I$. Thus, the sequence $\{x_k\}$ converges Q-superlinearly, as we wanted to prove. □

3.1. Numerical experiments. In addition to the gradient methods with retards, we used the classical conjugate gradient method for solving our test problems (see [11], [14]). The conjugate gradient method uses one n-dimensional vector more than the gradient methods with retards, and requires a little more computational work per iteration. However, as it is well known, the conjugate gradient method has excellent convergence properties, including finite termination in n iterations and is the method of choice for solving large-scale symmetric positive definite linear systems. See, for example [2].

Iterative linear methods are very important in the context of iterative resolution of large-scale nonlinear systems of equations. In this case the "Newtonian linear system" must be solved approximately at each iteration of the nonlinear solver. See for instance, [4], [6], [7], [15], and [20]. A very high precision is neither necessary nor recommendable in the resolution of these linear systems and the danger of "oversolving" these subproblems may lead to unacceptable computer times in the resolution of the main problems. Thus, mild stopping criteria are used. The best-known stopping criterion (see [4]) imposes that the residual at a given iterate should be a fraction of the residual at the initial iteration. In our notation, this turns out to be

$$\|Ax_k - b\| \leq \theta \|Ax_0 - b\|. \tag{3.14}$$

Other situations where mild stopping criteria like (3.14) are used, when the main problem is more general than the linear system, may be found in [8], [9], and [13].

In our tests we used the stopping criterion (3.14) with $\theta = 10^{-1}, 10^{-2}, 10^{-3}$, and 10^{-4}. All our tests were done on an HP Apollo 135/125 workstation in double precision Fortran. The methods tested are listed below.

"0" Conjugate gradient method;
"1" Steepest descent method;

TABLE 3.1
Average of (iterations)/(flops $\times 10^{-3}$) using five initial points.

Method	$\theta = 10^{-1}$	$\theta = 10^{-2}$	$\theta = 10^{-3}$	$\theta = 10^{-4}$
0	3/18	9/54	28/168	92/552
1	4/20	16/80	83/415	414/2070
2	3/15	9/45	32/160	72/360
3	3/15	15/75	33/165	110/550
4	3/11	15/51	38/128	140/476
5	3/11	16/70	42/200	100/490
6	4/20	18/90	86/430	503/2515
7	3/15	11/55	35/175	210/1050
8	3/15	13/65	27/135	76/380
9	3/15	12/60	40/200	82/410

"2" Barzilai–Borwein method (method based on (3.4) with $m = 1$);
"3" Random choice of $\nu(k)$ (3.2);
"4" Cyclic choice of $\nu(k)$ (3.3);
"5" Maximum retard (method based on (3.4) with $m = 5$);
"6" Maximum λ (method based on (3.5);
"7" Minimum λ (method based on (3.6);
"8" Maximum-minimum retard (method based on (3.7);
"9" Random choice of $\nu(k)$, excluding steepest descent (3.8).

We now describe two different test problems and the corresponding numerical results.

Two point boundary value problems: Consider the matrix $A \equiv (a_{ij})$ given by

$$a_{ii} = 2/h^2, \;\; a_{i,i-1} = -1/h^2 \text{ if } i \neq 1, \;\; a_{i,i+1} = -1/h^2 \text{ if } i \neq n,$$

for $i = 1, \ldots, n$, where $h = 11/n$ and $n = 1000$. Linear systems of this kind appear frequently in the numerical solution of two point boundary value problems.

We generated a random solution x_* with components between -10 and 10 and we computed $b = Ax_*$. We used five initial points x_0, originated by different seeds of the random number generator. In Table 3.1 we report the average number of iterations and the average computational work required by each method using the five initial points, and $m = 5$, to achieve the stopping criterion with the $\| \; . \; \|_\infty$ norm. The computational work is reported in floating point operations (flops) divided by 10^3.

The preliminary numerical experiments reported in Table 3.1 seem to indicate that, among the different strategies for the gradient method with retards, the most efficient are "2" (Barzilai–Borwein), "3" (random), "4" (cyclic), "5" (maximum retard), "8" (maximum-minimum retard), and "9" (random excluding steepest descent). On the other hand, the steepest descent method, together with the strategies based on (3.5) and (3.6), are clearly less efficient. It is interesting to observe that the successful gradient methods with retards appear to use about the same number of iterations as the conjugate gradient method for obtaining the precision (3.14), for the tested values of θ. This can be a very useful feature of these methods since, as we mentioned before, they can be implemented using less computer time and storage than the conjugate gradient method. (For this particular problem, the computer time of one iteration of the gradient methods with retards is about 0.8 times the computer time of one iteration of the conjugate gradient method). Moreover, for some of the gradient methods

with retards, a steplength does not have to be computed at every iteration, and that could represent a considerable reduction in the computational work required during the process. For example, the cyclic choice of $\nu(k)$ (method 4) only needs to compute one steplength every $m+1$ iterations.

Random problems: We take $A = QDQ^t$, where

$$Q = (I - 2w_3w_3^T)(I - 2w_2w_2^T)(I - 2w_1w_1^T),$$

w_1, w_2, and w_3 are unitary random vectors, $D = Diag(\sigma_1, \ldots, \sigma_n)$ is a diagonal matrix where $\sigma_1 = 1$, $\sigma_n = cond$, and σ_j is randomly generated between 1 and $cond$ for $j = 2, \ldots, n-1$. The entries of the right-hand-side b are randomly generated between -10 and 10. The initial point is the null vector of R^n and, in these tests, we used $n = 5000$ and we allowed a maximum of $1,000$ iterations.

In Table 3.2 we report the number of iterations used to obtain $\|Ax_k - b\|_2 \leq \theta\|b\|_2$ by the CG method and some of the most efficient gradient methods with retards. We observe that the difference between the number of iterations used by different methods is not great. For large condition numbers, the performance of gradient methods with retards deteriorates. However, it is very interesting to observe that, for $cond = 10^7$ all the gradient methods with retards achieved the precision 0.1 in about 40 iterations, while the method of conjugate gradients used more than 400 iterations for the same purpose. This behavior favors the application of gradient methods with retards in the inexact-Newton context, as we mentioned before.

Since, in principle, all the gradient methods with retards, including steepest descent, satisfy the same convergence theorem, a more careful analysis is necessary to understand the numerical experiments. The following remarks are based on the available theory and tend to explain the numerical behavior observed.

(a) The hypothesis of Theorem 3.1 (superlinear convergence) is the convergence of the normalized gradients. This hypothesis cannot be verified if we choose the steepest descent method or the strategy based on (3.5), since, in both cases, the angle between successive gradients is at least 90 degrees.

(b) The "ideal" behavior of an iteration based on (1.4) could be obtained choosing $\{\alpha_0, \alpha_1, \alpha_2, \ldots\}$ as different eigenvalues of A. By (2.2), if (say) $\alpha_k = \sigma_i$, we have that $d_i^{k+j} = 0$ for all $j \geq 1$. As a result, this ideal method should find the solution in a number of iterations equal to the number of different eigenvalues of A. It is interesting to observe that this property is independent of the choice of the initial point. This is precisely the main termination result of the conjugate gradient method. An efficient gradient method with retards should approximate, in some sense, the ideal iteration, and numerical results suggest that the efficient strategies are successful in doing that. In particular, Glunt, Hayden, and Raydan [10] established a relationship with the shifted power method that adds understanding to the practical behavior of strategy "2" (Barzilai–Borwein).

4. Preconditioned version. We present the preconditioned version of the gradient method with retards (PGMR), and compare its performance with the PCG method on some large and sparse SPD linear systems arising from the discretization of elliptic PDE problems. Our presentation follows the recent work by Molina and Raydan [16] for the preconditioned Barzilai–Borwein method.

For practical purposes, we only consider the case $\rho(k) = 3$ for all k. In that case, one auxiliary linear system of equations needs to be solved per iteration, and the PGMR can be written as (see [16]):

TABLE 3.2
Iterations required by CG, Barzilai–Borwein "2", cyclic "4", maximum retard "5", and max-min retard "8", for random problems.

cond	θ	*CG*	2	4	5	8
10^2	10^{-1}	9	19	17	20	17
	10^{-2}	22	30	27	34	26
	10^{-3}	33	50	48	53	47
	10^{-4}	45	62	57	66	57
10^3	10^{-1}	25	21	23	30	22
	10^{-2}	63	66	70	78	70
	10^{-3}	102	124	123	140	121
	10^{-4}	132	166	160	185	162
10^4	10^{-1}	64	38	30	35	32
	10^{-2}	185	159	169	190	173
	10^{-3}	267	358	346	389	342
	10^{-4}	340	616	546	724	629
10^5	10^{-1}	235	41	33	37	33
	10^{-2}	338	386	406	670	398
	10^{-3}	412	> 1000	> 1000	> 1000	> 1000
	10^{-4}	447				
10^6	10^{-1}	342	43	34	42	35
	10^{-2}	416	> 1000	> 1000	> 1000	> 1000
	10^{-3}	453				
	10^{-4}	485				
10^7	10^{-1}	417	43	34	43	37
	10^{-2}	453	> 1000	> 1000	> 1000	> 1000
	10^{-3}	485				
	10^{-4}	519				

ALGORITHM 4.1. **: PGMR**

Given $x_0 \in R^n$, α_0 a nonzero real number,
m a positive integer and C an SPD matrix of order n.
Set $g_0 = Ax_0 - b$.
For $k = 0, 1, \ldots$ (until convergence) do
Choose $\nu(k) \in \{k, k-1, \ldots, max\{0, k-m\}\}$
Solve $Ch_k = g_k$, for h_k
Set $p_k = Ah_k$
Set $g_{k+1} = g_k - \dfrac{1}{\tilde{\alpha}_{\nu(k)}} p_k$
Set $x_{k+1} = x_k - \dfrac{1}{\tilde{\alpha}_{\nu(k)}} h_k$
Set $\tilde{\alpha}_{k+1} = \dfrac{h_k^t p_k}{g_k^t h_k}$
End do

Notice that every iteration of the PGMR algorithm requires two inner products, two scalar-vector multiplications, two vector additions, one matrix-vector multiplication, and the solution of a linear system of equations with the preconditioning matrix C. Since C is an SPD matrix, it follows from a straightforward application of Theorem 2.3 that the sequence $\{x_k\}$ converges to $x_\star$.

Consider now the elliptic partial differential equation

$$-\frac{\partial^2 u(x,y)}{\partial x^2} - \frac{\partial^2 u(x,y)}{\partial y^2} + \tilde{\gamma} u(x,y) = f_1(x,y), \tag{4.1}$$

TABLE 4.1
(iterations/flops $\times 10^{-6}$) when $\gamma = 0$ and $n = 25 \times 10^4$.

Method	$\theta = 10^{-1}$	$\theta = 10^{-4}$	$\theta = 10^{-8}$
PCG	55/207	89/334	128/480
BarBor	60/210	100/350	157/549
cyclic ($m = 3$)	70/218	101/315	170/530
cyclic ($m = 4$)	78/241	115/356	160/496
maximum ($m = 3$)	59/206	98/343	162/567
maximum ($m = 4$)	62/217	100/349	160/560
max-min ($m = 3$)	66/230	108/377	145/507
max-min ($m = 4$)	68/237	120/438	178/621

where $\tilde{\gamma} \geq 0$ and the function $f_1(x, y)$ are given. We solve (4.1) on the unit square $0 \leq x \leq 1$, $0 \leq y \leq 1$, with homogeneous Dirichlet boundary conditions, i.e., we wish to obtain a function $u(x, y)$ that is continuous on the unit square, satisfies (4.1) in the interior of the unit square, and equals zero on the boundary.

To obtain the numerical solution of this problem, we discretize (4.1) using the central finite difference scheme of five points in a uniform grid $p \times p$ with a step $h = 1/(p + 1)$ and the natural ordering, producing a system of linear equations $Ax = b$ of order $n = p^2$. The matrix A is SPD and has block tridiagonal structure. All the diagonal blocks of A are square, symmetric, and tridiagonal matrices of order p, and there are exactly p blocks in the diagonal of A. The diagonal elements of A are $(4 + \gamma)$, where $\gamma = h^2\tilde{\gamma}$. For more details see [16].

The ith position of the vector b is equal to $(f_1)_i$, where $(f_1)_i$ is the evaluation of f_1 in the ith node of the grid in the natural ordering. Finally, note that for $\gamma = 0$ our model problem reduces to the classical Poisson equation for which A is ill-conditioned. When γ increases, the condition number of A is gradually reduced.

We ran an implementation of Algorithms PGMR and PCG, to solve problem (4.1), for different values of the parameter γ and different step sizes h. We report results with the efficient PCG version described in Golub and Van Loan [11], and the well-known symmetric successive overrelaxation (SSOR) preconditioning strategy for both algorithms (see, for instance, [1]). The parameter ω associated with the SSOR technique was chosen as in [16], for both methods. We also ran some experiments with the modified incomplete Cholesky preconditioning strategy, introduced by Gustafsson [12]. However, the results were quite similar to the ones obtained with the SSOR strategy, and we do not report them.

We start the process in both algorithms at $x_0 = (0, 0, \ldots 0)^t$ and we set $\alpha_0 = 1$ in Algorithm PGMR. We stop the process, in both algorithms whenever $\|g_k\|_2 \leq \theta \, \|g_0\|_2$ for different values of θ.

In our first experiment we fix $\gamma = 0$ and $p = 500$, i.e., $n = 25 \times 10^4$. Table 4.1 shows the number of iterations and the computational work required by the PCG method and the PGMR with the following strategies (see section 3): Barzilai–Borwein (BarBor), cyclic retard (cyclic), maximum retard (maximum), and maximum-minimum retard (max-min). The computational work is reported in number of flops divided by 10^6. We can see that the different strategies in the PGMR family are competitive with the PCG method. The PCG method shows a moderate improvement over the PGMR family in number of iterations. Nevertheless, when low accuracy is required, the winner in computational work by a slight difference is a member of the PGMR family. In particular, the case $m = 3$ seems to perform better than $m = 4$ in the three strategies that involve the parameter m.

TABLE 4.2
(iterations/flops $\times 10^{-6}$) when $\gamma = 0.1$, $m = 3$ and $n = 25 \times 10^4$.

Method	$\theta = 10^{-1}$	$\theta = 10^{-4}$	$\theta = 10^{-8}$
PCG	8/30	17/64	30/113
BarBor	8/28	19/66	32/112
cyclic	9/28	18/56	32/99
maximum	9/31	19/66	32/112
max-min	8/28	18/63	31/108

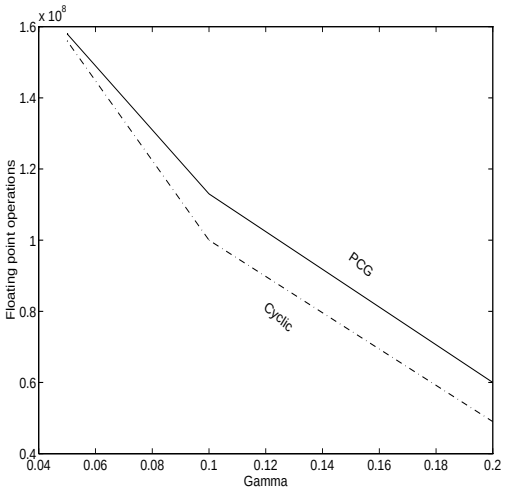

FIG. 1. *PCG and cyclic for $n = 25 \times 10^4$ and $\theta = 10^{-8}$.*

In our second experiment we study the effect of reducing the condition number of A by increasing γ. Table 4.2 shows the number of iterations and the computational work required by the PCG method and the PGMR when $n = 25 \times 10^4$, $\gamma = 0.1$, and $m = 3$. We observe that when the condition number of A is reduced, the PCG method and the PGMR tend to require the same number of iterations, and this number of iterations decreases. As a consequence, some strategies in the PGMR show a moderate but consistent improvement over the PCG method in number of flops. In particular, Figure 1 shows the number of flops required by PCG and cyclic retard ($m = 3$) for $n = 25 \times 10^4$, $\theta = 10^{-8}$, and different values of γ.

5. Concluding remarks. We have introduced the gradient method with retards to solve symmetric and positive definite linear systems of equations. This method is a generalization of the steepest descent and the Barzilai and Borwein methods, and covers a wide range of possible strategies to choose steplengths that are defined by Rayleigh quotients.

Based on our numerical results, we conclude that some of the new strategies are very efficient and, in general, compete favorably with the classical conjugate gradient (CG) method in storage requirements and computational work, when low precision is required. In particular, we have no theoretical explanation for the outstanding behavior of the GMR for ill-conditioned problems when $\theta = 10^{-1}$ (see Table 3.2). However, if the coefficient matrix is very ill-conditioned and high precision is required then the CG method is clearly a better option.

We also present a preconditioned version of the GMR and test it on large and sparse linear systems arising from the discretization of elliptic PDE problems. In this case, some of the PGMR strategies show a moderate but consistent improvement over the PCG method. In fact, the GMR is a gradient method for the minimization of quadratic functions. Therefore, as any gradient method for unconstrained minimization, its performance improves if the coefficient (Hessian) matrix is conditioned to approximate the identity, in which case it behaves more like the Newton method.

Finally, the possibility of computing the steplengths and the points in different processors offers an additional advantage in terms of overall computer time. For example, when we consider the extension to minimize nonquadratic functions (see Raydan [19]), the steplengths can be expensive to compute, and so, it would be interesting to have the chance of using "old ones" while waiting for the computation of "new ones." Furthermore, even in the sequential case, the possibility of repeating steplengths can be very time-saving. For instance, a gradient method that repeats the previous steplength whenever a nonmonotone criterion is satisfied should be developed, and deserves careful experimentation.

Acknowledgments. We are grateful to the associate editor, Prof. P. L. Toint, and the referees for their constructive comments and suggestions.

REFERENCES

[1] O. Axelsson, *A survey of preconditioning iterative methods for linear systems of algebraic equations*, BIT, 25 (1985), pp. 166–187.
[2] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, Ch. Romine, and H. van der Vorst, *Templates for the Solution of Linear Systems*: *Building Blocks for Iterative Methods*, SIAM, Philadelphia, PA, 1994.
[3] J. Barzilai and J. M. Borwein, *Two point step size gradient methods*, IMA J. Numer. Anal., 8 (1988), pp. 141–148.
[4] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 14 (1982), pp. 400–408.
[5] J. E. Dennis and J. J. Moré, *A characterization of superlinear convergence and its application to quasi-Newton methods*, Math. Comp., 28 (1974), pp. 549–560.
[6] P. Deuflhard, *Global inexact Newton methods for very large scale nonlinear problems*, Impact Comput. Sci. Engrg., 3 (1991), pp. 366–393.
[7] S. C. Eisenstat and H. F. Walker, *Globally convergent inexact Newton methods*, SIAM J. Optim., 4 (1994), pp. 393–422.
[8] A. Friedlander and J. M. Martínez, *On the maximization of a concave quadratic function with box constraints*, SIAM J. Optim., 4 (1994), pp. 177–192.
[9] A. Friedlander, J. M. Martínez, and M. Raydan, *A new method for large-scale box constrained quadratic minimization problems*, Optim. Methods Softw., 5 (1995), pp. 57–74.
[10] W. Glunt, T. L. Hayden, and M. Raydan, *Molecular conformations from distance matrices*, J. Comput. Chem., 14 (1993), pp. 114–120.
[11] G. H. Golub and Ch. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, London, 1989.
[12] I. Gustafsson, *A class of first order factorization methods*, BIT, 18 (1978), pp. 142–156.
[13] G. T. Herman, *Image Reconstruction from Projections*: *The Fundamentals of Computerized Tomography*, Academic Press, New York, 1980.
[14] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research National Bureau of Standards, B49 (1952), pp. 409–436.
[15] J. M. Martínez, *A theory of secant preconditioners*, Math. Comp., 60 (1993), pp. 681–698.
[16] B. Molina and M. Raydan, *Preconditioned Barzilai–Borwein method for the numerical solution of partial differential equations*, Numer. Algorithms, 13 (1996), pp. 45–60.
[17] M. Raydan, *Convergence Properties of the Barzilai and Borwein Gradient Method*, Ph.D. thesis, Dept. of Mathematical Sciences, Rice University, Houston, TX, 1991.
[18] M. Raydan, *On the Barzilai and Borwein choice of steplength for the gradient method*, IMA J. Numer. Anal., 13 (1993), pp. 321–326.

[19] M. Raydan, *The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem*, SIAM J. Optim., 7 (1997), pp. 26–33.
[20] T. J. Ypma, *Local convergence of inexact Newton methods*, SIAM J. Numer. Anal., 21 (1984), pp. 583–590.