PERGAMON Mathematical and Computer Modelling 36 (2002) 1161–1168

MATHEMATICAL AND COMPUTER MODELLING

www.elsevier.com/locate/mcm

Smooth and Adaptive Gradient Method with Retards

J.-L. LAMOTTE
Université Pierre et Marie Curie
Laboratoire LIP6-CNRS, 4 Place Jussieu
75252 Paris Cedex 05, France
lamotte@anp.lip6.fr

B. MOLINA AND M. RAYDAN
Departamento de Computación, Facultad de Ciencias
Universidad Central de Venezuela, UCV, Ap. 47002
Caracas 1041-A, Venezuela
<bmolina><mraydan>@reacciun.ve

(Received July 2001; revised and accepted October 2001)

Abstract—The gradient method with retards (GMR) is a nonmonotone iterative method recently developed to solve large, sparse, symmetric, and positive definite linear systems of equations. Its performance depends on the retard parameter $\bar{m}$. The larger the $\bar{m}$, the faster the convergence, but also the faster the loss of precision is observed in the intermediate computations of the algorithm. This loss of precision is mainly produced by the nonmonotone behavior of the norm of the gradient which also increases with $\bar{m}$. In this work, we first use a recently developed inexpensive technique to smooth down the nonmonotone behavior of the method. Then we show that it is possible to choose $\bar{m}$ adaptively during the process to avoid loss of precision. Our adaptive choice of $\bar{m}$ can be viewed as a compromise between numerical stability and speed of convergence. Numerical results on some classical test problems are presented to illustrate the good numerical properties. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords—Gradient method with retards, Polynomial preconditioners, Residual smoothing technique, Barzilai-Borwein method.

1. INTRODUCTION

In a recent paper, Friedlander *et al.* [1] introduced the gradient method with retards (GMR) to find the unique global minimizer of quadratic functions of the form

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

where $A \in \mathbb{R}^{n\times n}$ is large, sparse, symmetric, and positive definite (SPD). The equivalent problem of finding the unique solution $x^* = A^{-1}b$ of the linear system $Ax = b$ appears in a wide variety

This work was supported by CDCH-UCV Project 03-13-4598-2000, the Scientific Computing Center at UCV. and Laboratoire LIP6-CNRS, France.
The authors are indebted to two anonymous referees whose comments helped to improve this paper.

of applications. In particular, it arises frequently in problems that involve the discretization of partial differential equations.

The practical version of the GMR family can be written as

$$x_{k+1} = x_k - \frac{1}{\alpha_{\nu(k)}} g_k, \tag{2}$$

where g_k is the gradient vector of f evaluated at x_k, and

$$\alpha_{\nu(k)} = \frac{g^t_{\nu(k)} A g_{\nu(k)}}{g^t_{\nu(k)} g_{\nu(k)}}. \tag{3}$$

In (3), $\nu(k)$ is arbitrarily chosen in the set

$$\{k, k-1, \dots, \max\{0, k-\bar{m}\}\}, \tag{4}$$

where $\bar{m}$ is a given positive integer. For instance, when $\nu(k) = k$, then the GMR reduces to the classical steepest descent method. If $\nu(k) = k - 1$, then the GMR becomes the first Barzilai-Borwein method. See [1,2] for details. In [1], the above choices and some other different values of $\nu(k)$ are combined with a preconditioned version to solve well-known test problems. Notice that (3) corresponds to the practical case ($\rho_k = 1$) in [1].

The preconditioned GMR family has been already compared with the preconditioned conjugate gradient method for solving large-scale problems. It has been observed (see [1,3]) that some of the members in the GMR family are competitive with the CG method, and some times preferable.

It has also been observed that the performance of the sequence $\{x_k\}$ converging to x^* depends on the retard parameter $\bar{m}$. Indeed, the larger the $\bar{m}$, the faster the convergence, but also the faster the loss of precision is observed in the intermediate computations of the algorithm. This loss of precision is mainly produced by the nonmonotone behavior of the norm of the gradient which also increases with $\bar{m}$. To be precise, highly nonmonotone iterations yield $\alpha_{\nu(k)}$ very close to zero [1], and therefore, the calculations in (2) produce an inaccurate approximation of x_{k+1}. For more details on loss of precision and stability of algorithms, see [4]. The numerical behavior of iterative methods in the presence of rounding error propagation and loss of precision has been analyzed by Vignes and La Porte [5]. See also [6].

In this work, we are now interested in adding some new features to improve the numerical performance of the GMR method. We first combine the GMR with a recently developed inexpensive technique to smooth down the nonmonotone behavior of the method. Residual smoothing techniques were first introduced by Schonauer [7], and further analyzed by Brezinski and Zaglia [8], and also by Zhou and Walker [9].

Then we show that it is possible to choose $\bar{m}$ adaptively during the process to avoid the loss of precision and to stop the process correctly. Our adaptive choice of $\bar{m}$ can be viewed as a compromise between stability and speed of convergence.

The rest of the paper is divided into sections as follows. In Section 2, we introduce the smoothing technique directly on the preconditioned version of GMR. We illustrate the smoothing process with a simple numerical example. We also describe our strategy to dynamically adapt the parameter $\bar{m}$ during the process. In Section 3, the algorithm is implemented and numerical results on some classical test problems are presented to illustrate the good numerical properties. Finally, in Section 4, we present concluding remarks.

2. SMOOTH AND ADAPTIVE PRECONDITIONED GMR

In most practical applications, it is convenient to use a preconditioner to accelerate the convergence of the process. Given an SPD matrix C that approximates A, the preconditioned gradient method with retards (PGMR) can be written as follows (see [1]).

ALGORITHM 2.1. PGMR.

```
Given $x_0$: $\in \mathbb{R}^n$, $\alpha_0$ a nonzero real number,
$\bar{m}$ 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 $\nu(k) \in \{k, k-1, \ldots, \max\{0, k-\bar{m}\}\}$
    Solve $Ch_k = g_k$, for $h_k$
    Set $p_k = Ah_k$
    Set $x_{k+1} = x_k - \frac{1}{\alpha_{\nu(k)}} h_k$
    Set $g_{k+1} = g_k - \frac{1}{\alpha_{\nu(k)}} p_k$
    Set $\alpha_{k+1} = \frac{h_k^t p_k}{g_k^t h_k}$
End do
```

The following choices for $\nu(k)$ have been already used [1]:

(1) $\nu(k) =$ random integer between $\bar{k}$ and k [random (RA)];

(2) $\nu(k) = k - 1$ [Barzilai-Borwein (BB)];

(3)

$$\nu(k) = \begin{cases} k, & \text{if } \nu(k-1) < \bar{k} \text{ or } k = 0, \\ \nu(k-1), & \text{otherwise,} \end{cases} \qquad \text{[cyclic (CY)];}$$

(4) $\nu(k) = \bar{k}$ [maximum retard (MR)];

(5)

$$\nu(k) = \begin{cases} \bar{k}, & \text{if } k \text{ is even,} \\ k, & \text{if } k \text{ is odd,} \end{cases} \qquad \text{[maximum-minimum retard (MMR)];}$$

(6) $\nu(k) = \operatorname{argmax}\{\lambda(x_k), \ldots, \lambda(x_{\bar{k}})\}$ [maximum-lambda (MAXL)];

(7) $\nu(k) = \operatorname{argmin}\{\lambda(x_k), \ldots, \lambda(x_{\bar{k}})\}$ [minimum-lambda (MINL)];

where $\bar{k} = \max\{0, k - \bar{m}\}$.

2.1. Smoothing Technique

As in most iterative methods for linear systems, the sequence $\{\|r_k\|_2\}$, where $r_k = b - Ax$, plays an important role to monitor the quality of the iterates and also to stop the process. Since the GMR family has a nonmonotone behavior, then this sequence does not go "smoothly" to zero, and as a consequence, it leads to numerical instabilities. Nevertheless, the nonmonotone behavior of the GMR family plays an important role in the speed of convergence usually observed in practice. This situation motivates us to combine PGMR with the smoothing technique introduced by Schonauer [7] (see also [8,9]).

The idea is to generate an auxiliary sequence $\{y_k\}$ as follows:

$$y_k = x_k + \beta_k(y_{k-1} - x_k), \qquad \text{for } k = 1, 2, \ldots,$$

where $y_0 = x_0$ and β_k minimizes $\|b - A(x_k + \beta(y_{k-1} - x_k))\|_2^2$ over all $\beta \in \Re$, i.e.,

$$\beta_k = \frac{(r_k - \bar{r}_k)^t r_k}{\|r_k - \bar{r}_k\|_2^2},$$

where $\bar{r}_k = b - Ay_{k-1}$. These vectors can be obtained in a recursive and more stable way by

$$\bar{r}_k = r_{k-1} + \beta_{k-1}(\bar{r}_{k-1} - r_{k-1}).$$

By the minimization property of β_k, it is clear that

$$\|\bar{r}_k\|_2 \leq \min\{\|r_k\|_2, \|\bar{r}_{k-1}\|_2\},$$

and thus, the convergence of y_k is guaranteed and smoothed.

For completeness, we present the smoothed preconditioned gradient method with retards (SPGMR).

ALGORITHM 2.2. SPGMR.

```
Given $x_0$: $\in \mathbb{R}^n$, $\alpha_0$ a nonzero real number,
$\bar{m}$ a positive integer and $C$ an SPD matrix of order $n$.
Set $r_0 = b - Ax_0$, $y_0 = x_0$, and $\bar{r}_1 = r_0$.
Compute $x_1$, $r_1 = -g_1$ and $\alpha_1$ using Algorithm PGMR.
Set $\beta_1 = (r_1 - r_0)^t r_1 / \|r_1 - r_0\|_2^2$, and $y_1 = x_1 + \beta_1(y_0 - x_1)$.
For $k = 1, 2, \ldots$ (until convergence) do
    Choose $\nu(k) \in \{k, k-1, \ldots, \max\{0, k - \bar{m}\}\}$
    Solve $Ch_k = -r_k$, for $h_k$
    Set $p_k = Ah_k$
    Set $x_{k+1} = x_k - \frac{1}{\alpha_{\nu(k)}} h_k$
    Set $r_{k+1} = r_k + \frac{1}{\alpha_{\nu(k)}} p_k$
    Set $\bar{r}_{k+1} = r_k + \beta_k(\bar{r}_k - r_k)$
    Set $\beta_{k+1} = (r_{k+1} - \bar{r}_{k+1})^t r_{k+1} / \|r_{k+1} - \bar{r}_{k+1}\|_2^2$
    Set $y_{k+1} = x_{k+1} + \beta_{k+1}(y_k - x_{k+1})$
    Set $\alpha_{k+1} = h_k^t p_k / g_k^t h_k$
End do
```

To illustrate the behavior of the smoothing technique combined with the PGMR family, we now present a numerical experiment that will be described in a general setting in Section 3. In Figure 1, we report the norm of the residual r_k for the MMR, without the smoothing technique, and also the norm of the residual $\bar{r}_k$ associated with the sequence y_k of the MMR with the smoothing technique. We can observe that the sequence y_k converges as fast as the sequence x_k to the solution, but it does so monotonically and smoothly, therefore avoiding the dangerous jumps produced by the nonmonotone behavior of the method.

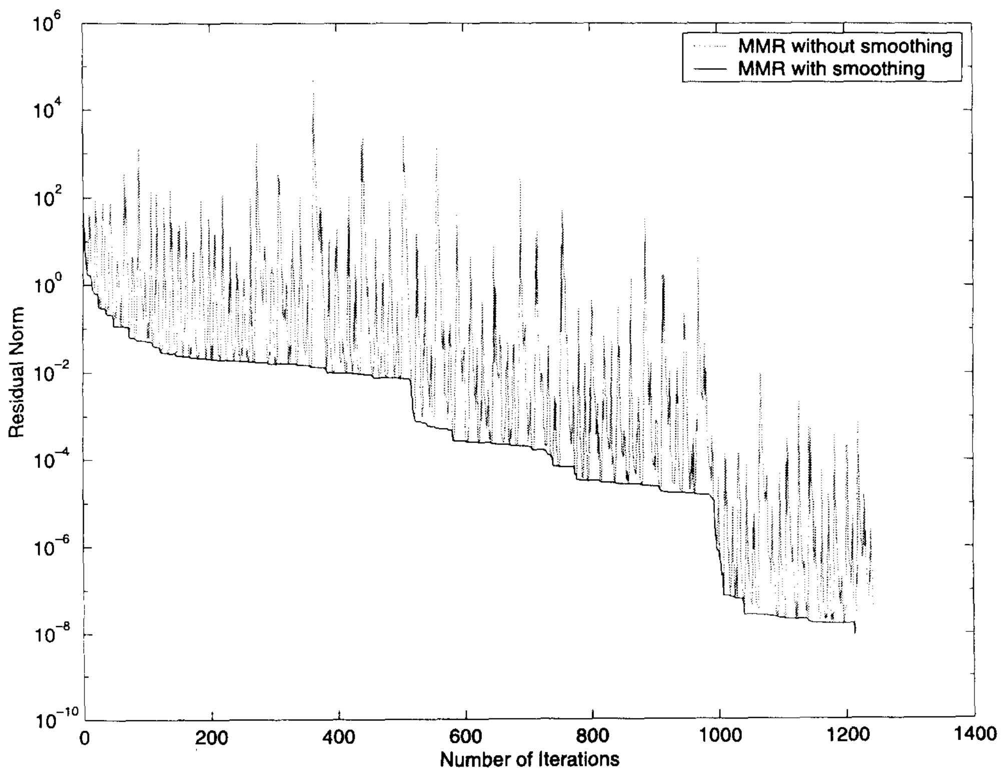

Figure 1. PGMR when $n = 500$ using MMR with $\bar{m} = 3$.

2.2. Adaptive Technique

We introduce an adaptive choice that combines different retards at different iterations. The convergence of the different options in the PGMR family depends on the retard parameter $\bar{m}$.

The results obtained with the BB ($\bar{m} = 1$) steplength are not as good as the ones obtained with the rest of the options, but it tends to reduce the norm of the residual at each iteration, avoiding the dangerous jumps that lead to the loss of precision observed when $\bar{m}$ is large. This behavior motivates us to combine the BB method with other schemes as follows: use a more aggressive scheme as frequently as possible and keep track of the number of times that the norm of the residual increases. If it increases INC consecutive times, then use the BB steplength for BBT consecutive iterations. After that, we return to the aggressive choice and repeat the process. This adaptive choice of $\bar{m}$ can be viewed as a compromise between stability and speed of convergence.

For our model problem, that will be fully described in Section 3, we tested this scheme with different positive integer values of INC and BBT.

3. NUMERICAL RESULTS

Consider the elliptic partial differential equation

$$-\frac{\partial^2(k_1 u(x,y))}{\partial x^2} - \frac{\partial^2(k_2 u(x,y))}{\partial y^2} + \tilde{\gamma} u(x,y) = f_1(x,y), \tag{5}$$

where $\tilde{\gamma} \geq 0$, k_1 and k_2 are real scalar parameters, and $f_1(x,y)$ is a given function. We solve (5) 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 (5) in the interior of the unit square, and equals zero on the boundary.

To obtain the numerical solution of this problem, we discretize (5) using the central finite difference scheme of five points in a uniform grid $r \times r$ with a step $h = 1/(r+1)$ and the natural ordering, producing a system of linear equations $Ax = b$ of order $n = r^2$. Matrix A is SPD and has block tridiagonal structure. Every one of the diagonal blocks of A is a square, symmetric, and tridiagonal matrix of order r, and there are exactly r blocks in the diagonal of A. The diagonal elements of A are $(4 + \gamma)$, where $\gamma = h^2\tilde{\gamma}$. For more details, see [3]. Note that for $\gamma = 0$, our model problem reduces to the classical Poisson equation for which A is ill-conditioned. When γ increases, the matrix A gradually improves its ill-conditioning.

The preconditioning step of Algorithm PGMR is solved using truncated Neumann series (TNS). We have chosen TNS because it only involves matrix-vector products to approximate the inverse of A. Some other preconditioning strategies (e.g., SSOR, incomplete Cholesky) have been used with PGMR [1,3] obtaining similar results. Solving the system $Ch_k = g_k$ by TNS is equivalent to m-steps of a basic iterative method. In this work, we have selected the classical iterative method of Jacobi. The main reason is that it is highly parallelizable, which is a nice feature for the PGMR algorithm (see [10,11]).

All experiments were run on Pentium III, 700 MHz, 128 MB RAM, using Fortran 77. The initial guess was always chosen as $x_0 = 0 \in \mathbb{R}^n$, and the stopping criterion is always $\|g_k\|_2 \leq 10^{-8}$. The function f_1 was chosen such that the right-hand side vector b yields the exact solution $x_i^* = 1/n$ for all i. We set $\gamma = 0$ and we use TNS with $m = 4$ Jacobi steps.

In our first experiment, we compare the performance of PGMR for the seven different choices of steplength described in Section 2 using the smoothing technique.

In Table 1, we report, for $\bar{m} = 2, 3, 4$, and 5, the number of iterations required to stop the process, the two-norm of the residual ($\|g\|$), and the two-norm of the error ($\|e\|$) for six choices of steplengths. In Table 2, we report the same results for the BB choice that does not depend on the retard parameter $\bar{m}$. In general, it can be observed that the best results are obtained by MR, RA, and CY. On the other hand, the worst results are obtained by BB and MMR. We also observe that, on average, the best choice for $\bar{m}$ is three. Indeed, when $\bar{m}$ increases, some of the options deteriorate the numerical performance. It is worth mentioning that during the convergence process, the most stable method is BB and the methods that show the most significant deterioration are MR, MINL, and CY.

Table 1. PGMR with the smoothing technique for all possible choices of steplength.

	n		MR	RA	CY	MAXL	MINL	MMR
$\bar{m} = 2$	200	Iter.	577	450	570	504	634	4801
		$\|\|g\|\|$	0.98×10^{-8}	0.98×10^{-8}	0.73×10^{-8}	0.79×10^{-8}	0.68×10^{-8}	0.97×10^{-8}
		$\|\|e\|\|$	0.32×10^{-9}	0.12×10^{-9}	0.30×10^{-11}	0.28×10^{-9}	0.29×10^{-9}	0.46×10^{-9}
	300	Iter.	740	634	804	660	806	10513
		$\|\|g\|\|$	0.94×10^{-8}	0.68×10^{-8}	0.47×10^{8}	0.76×10^{-8}	0.97×10^{-8}	0.99×10^{-8}
		$\|\|e\|\|$	0.44×10^{-9}	0.43×10^{-10}	0.41×10^{-10}	0.35×10^{-9}	0.14×10^{-11}	0.47×10^{-9}
	400	Iter.	1108	1526	1095	1088	1484	18337
		$\|\|g\|\|$	0.70×10^{-8}	0.99×10^{-8}	0.18×10^{-8}	0.34×10^{-8}	0.72×10^{-8}	0.99×10^{-8}
		$\|\|e\|\|$	0.27×10^{-9}	0.50×10^{-9}	0.11×10^{-12}	0.15×10^{-9}	0.33×10^{-9}	0.47×10^{-9}
$\bar{m} = 3$	200	Iter.	481	506	605	478	671	467
		$\|\|g\|\|$	0.90×10^{-8}	0.96×10^{-8}	0.81×10^{-8}	0.98×10^{-8}	0.81×10^{-8}	0.98×10^{-8}
		$\|\|e\|\|$	0.10×10^{-9}	0.24×10^{-10}	0.14×10^{-9}	0.16×10^{-9}	0.20×10^{-9}	0.44×10^{-9}
	300	Iter.	841	519	720	677	698	794
		$\|\|g\|\|$	0.65×10^{-8}	0.72×10^{-8}	0.76×10^{8}	0.80×10^{-8}	0.38×10^{-8}	0.42×10^{-8}
		$\|\|e\|\|$	0.10×10^{-9}	0.67×10^{-10}	0.17×10^{-10}	0.22×10^{-9}	0.29×10^{-9}	0.15×10^{-9}
	400	Iter.	916	815	1184	1031	1021	1173
		$\|\|g\|\|$	0.94×10^{-8}	0.12×10^{-8}	0.59×10^{-8}	0.90×10^{-8}	0.27×10^{-8}	0.37×10^{-8}
		$\|\|e\|\|$	0.19×10^{-9}	0.58×10^{-10}	0.73×10^{-9}	0.43×10^{-9}	0.12×10^{-9}	0.11×10^{-9}
$\bar{m} = 4$	200	Iter.	575	434	600	566	886	549
		$\|\|g\|\|$	0.66×10^{-8}	0.88×10^{-8}	0.93×10^{-8}	0.99×10^{-8}	0.99×10^{-8}	0.95×10^{-8}
		$\|\|e\|\|$	0.15×10^{-8}	0.68×10^{-10}	0.51×10^{-9}	0.49×10^{-9}	0.10×10^{-7}	0.10×10^{-10}
	300	Iter.	843	863	917	853	940	1205
		$\|\|g\|\|$	0.53×10^{-8}	0.94×10^{-8}	0.96×10^{-8}	0.84×10^{-8}	0.27×10^{-8}	0.23×10^{-8}
		$\|\|e\|\|$	0.38×10^{-6}	0.40×10^{-9}	0.13×10^{-6}	0.33×10^{-9}	0.10×10^{-6}	0.61×10^{-10}
	400	Iter.	1183	957	1492	1304	1581	1537
		$\|\|g\|\|$	0.70×10^{-8}	0.85×10^{-8}	0.95×10^{-8}	0.65×10^{-8}	0.82×10^{-8}	0.96×10^{-8}
		$\|\|e\|\|$	0.75×10^{-6}	0.50×10^{-10}	0.20×10^{-7}	0.43×10^{-10}	0.43×10^{-6}	0.48×10^{-9}
$\bar{m} = 5$	200	Iter.	685	788	546	756	471	951
		$\|\|g\|\|$	0.33×10^{-8}	0.73×10^{-8}	0.73×10^{-8}	0.76×10^{-8}	0.28×10^{-8}	0.74×10^{-8}
		$\|\|e\|\|$	0.48×10^{-7}	0.25×10^{-8}	0.24×10^{-9}	0.29×10^{-4}	0.28×10^{-10}	0.12×10^{-5}
	300	Iter.	1104	770	911	1422	648	1566
		$\|\|g\|\|$	0.96×10^{-8}	0.98×10^{-8}	0.92×10^{8}	0.84×10^{-8}	0.79×10^{-8}	0.44×10^{-8}
		$\|\|e\|\|$	0.89×10^{-4}	0.39×10^{-10}	0.39×10^{-9}	0.11×10^{-4}	0.28×10^{-10}	0.55×10^{-5}
	400	Iter.	2483	800	731	5184	831	3225
		$\|\|g\|\|$	0.86×10^{-8}	0.89×10^{-8}	0.90×10^{-8}	0.87×10^{-8}	0.75×10^{-8}	0.43×10^{-8}
		$\|\|e\|\|$	0.40×10^{-3}	0.46×10^{-8}	0.28×10^{-9}	0.13×10^{-2}	0.76×10^{-10}	0.49×10^{-3}

Table 2. PGMR with the smoothing technique for the BB method.

	n	200	300	400
BB	Iter.	826	854	1206
	$\|\|g\|\|$	0.86×10^{-8}	0.99×10^{-9}	0.70×10^{-8}
	$\|\|e\|\|$	0.93×10^{-10}	0.22×10^{-10}	0.27×10^{-9}

Table 3. PGMR with the smoothing scheme and the six possible combinations with BB of the adaptive technique with $\bar{m} = 3$.

n		MR-BB	RA-BB	CY-BB	MAXL-BB	MINL-BB	MMR-BB
200	Iter.	520	614	576	591	585	594
	$\|g\|$	0.82×10^{-8}	0.57×10^{-8}	0.68×10^{-8}	0.61×10^{-8}	0.58×10^{-8}	0.54×10^{-8}
	$\|e\|$	0.29×10^{-9}	0.39×10^{-10}	0.17×10^{-11}	0.28×10^{-9}	0.83×10^{-10}	0.55×10^{-11}
	NBB	86	120	122	68	144	58
	NOTH	434	494	454	523	441	536
300	Iter.	786	913	617	522	1367	1015
	$\|g\|$	0.36×10^{-8}	0.72×10^{-8}	0.71×10^{-8}	0.79×10^{-8}	0.83×10^{-8}	0.68×10^{-8}
	$\|e\|$	0.12×10^{-9}	0.19×10^{-12}	0.34×10^{-9}	0.34×10^{-9}	0.17×10^{-9}	0.28×10^{-10}
	NBB	136	164	133	56	358	128
	NOTH	650	749	484	466	1009	887
400	Iter.	912	1151	1337	1301	1818	899
	$\|g\|$	0.82×10^{-8}	0.77×10^{-8}	0.20×10^{-8}	0.79×10^{-8}	0.99×10^{-8}	0.29×10^{-8}
	$\|e\|$	0.44×10^{-10}	0.39×10^{-9}	0.54×10^{-11}	0.19×10^{-9}	0.47×10^{-9}	0.43×10^{-10}
	NBB	162	218	270	182	464	102
	NOTH	750	933	1067	1119	1354	797

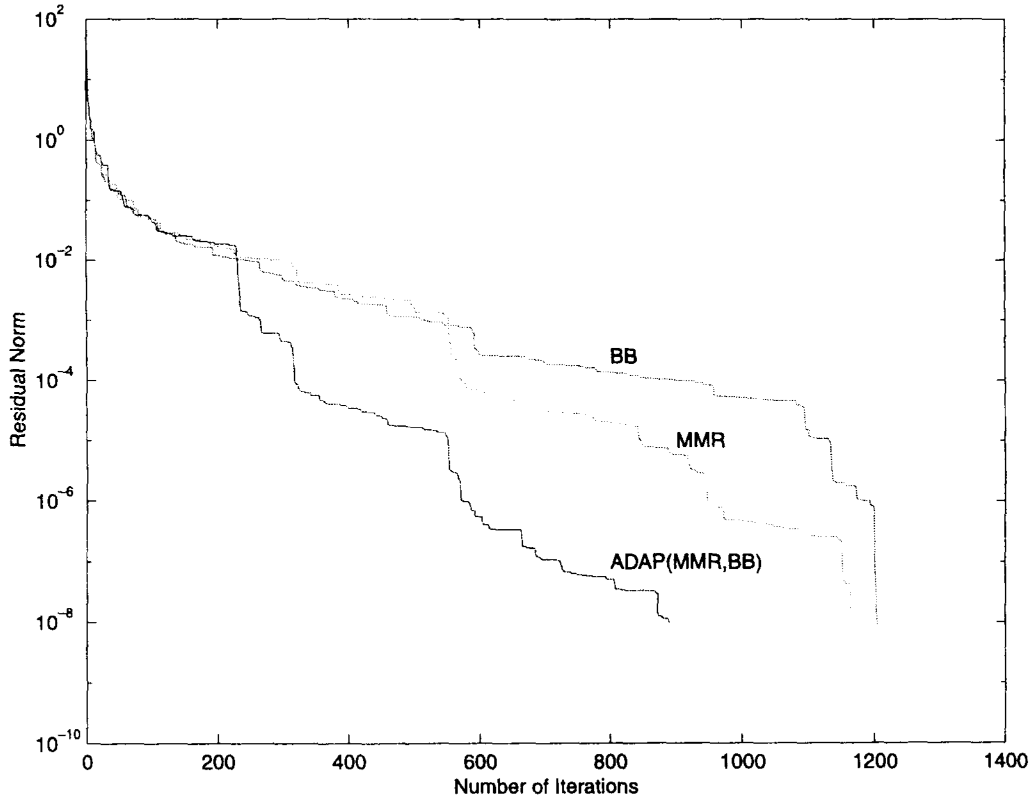

Figure 2. PGMR when $n = 400$ using the adaptive scheme with MMR and BB.

In our second experiment, we compare the performance of PGMR combining the six different choices of steplength with the BB method, using the adaptive technique described in Section 2. For this experiment, we also use the smoothing technique. We tested the adaptive scheme with different positive integer values of INC and BBT. The best results were obtained with INC = 3 and BBT = 2. In Table 3, we report, for these values of INC, BBT, and for $\bar{m} = 3$, the number of iterations required to stop the process, the two-norm of the residual ($\|g\|$), the two-norm of the error ($\|e\|$), the number of times that the BB method was used (NBB), and the number of times that the other choice was used (NOTH) for the six possible combinations.

Notice that, in general, the adaptive versions outperform the BB method and present a more stable numerical performance than the associated choice of steplength without using the adaptive technique. We observe that for $n = 300$, the best combination was BB with MAXL, and for $n = 400$, the best combination was BB with MMR. In particular, we show in Figure 2 the two-norm of the residual during the convergence process for the combination BB and MMR when $n = 400$. We also show, in the same figure, the behavior of BB and MMR without the adaptive technique. It is clear that the adaptive scheme outperforms both methods.

The parameters INC and BBT used in our experiments were obtained for our model problem (5). For using the adaptive scheme on other real applications, some previous testing is required to choose both parameters.

4. FINAL REMARKS

We present two ingredients for the PGMR that significantly help the numerical performance of the different options without increasing the computational cost. First, we present an inexpensive technique to smooth down the convergence process of the nonmonotone variants of the PGMR family. Second, using this smoothing technique, we describe an adaptive scheme that combines the BB method with any other choice of steplength to accelerate the convergence, and to improve the stability of the combined method. Encouraging numerical results are included on some classical PDE linear systems to illustrate the advantages of using the new ingredients associated with the PGMR family.

REFERENCES

1. A. Friedlander, J.M. Martinez, B. Molina and M. Raydan, Gradient method with retards and generalizations, *SIAM J. Numer. Anal.* **36**, 275–289, (1999).
2. J. Barzilai and J.M. Borwein, Two point step size gradient methods, *IMA J. Numer. Anal.* **8**, 141–148, (1988).
3. B. Molina and M. Raydan, The preconditioned Barzilai-Borwein method for the numerical solution of partial differential equations, *Numerical Algorithms* **13**, 45–60, (1996).
4. N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, PA, (1996).
5. J. Vignes and M. La Porte, *Error Analysis in Computing*, Information Processing 74, North-Holland, Amsterdam, (1974).
6. J. Asserrhine, J.M. Chesnaux and J.L. Lamotte, Estimation of round-off errors on several computer architectures, *Journal of Universal Computer Science* **1**, 450–464, (1995).
7. W. Schonauer, *Scientific Computing on Vector Computers*, North-Holland, Amsterdam, (1987).
8. C. Brezinski and M.R. Zaglia, Hybrid procedures for solving linear systems, *Numer. Math.* **67**, 1–19, (1994).
9. L. Zhou and H.F. Walker, Residual smoothing techniques for iterative methods, *SIAM J. Sci. Comput.* **15**, 297–312, (1994).
10. Y. Saad, Practical use of polynomial preconditionings for the conjugate gradient method, *SIAM J. Sci. Stat. Comp.* **6**, 865–881, (1985).
11. Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, MA, (1996).