A simple strategy for varying the restart parameter in GMRES(m)

A. H. Baker, E. R. Jessup, Tz. V. Kolev

October 4, 2007

Disclaimer

A simple strategy for varying the restart parameter in GMRES(m)

A. H. Baker [a,*,1], E. R. Jessup [b,2], Tz. V. Kolev [a,1]

[a]*Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Box 808 L-551, Livermore, CA 94551*

[b]*Department of Computer Science, University of Colorado, Boulder, CO 80309-0430*

Abstract

When solving a system of linear equations with the restarted GMRES method, a fixed restart parameter is typically chosen. We present numerical experiments that demonstrate the beneficial effects of changing the value of the restart parameter in each restart cycle on the total time to solution. We propose a simple strategy for varying the restart parameter and provide some heuristic explanations for its effectiveness based on analysis of the symmetric case.

Key words: GMRES, iterative methods, Krylov subspace, restart parameter
1991 MSC: 65F10

---

1 Introduction

The generalized minimum residual method (GMRES) [20] is a common choice for solving the large sparse linear system of equations

$$Ax = b, \tag{1}$$

when A is a nonsymmetric matrix. GMRES determines an approximate solution

$$x_1 \in x_0 + K_k(A, r_0), \tag{2}$$

where $K_k(A, r_0) \equiv span\{r_0, Ar_0, \dots, A^{k-1}r_0\}$ denotes a k-dimensional Krylov subspace, x_0 is the initial guess, and r_0 is the initial residual ($r_0 \equiv b - Ax_0$). Because the amount of storage and computational work required by GMRES increases with each iteration, the method is typically restarted as suggested in [20]. Restarted GMRES, denoted by GMRES(m), performs m iterations of GMRES, and then the resulting approximate solution is used as the initial guess to start another m iterations. This process repeats until the residual norm is small enough. The group of m iterations between successive restarts is referred to as a cycle, and m is referred to as the restart parameter. We indicate the restart cycle number with a subscript: x_i is the approximate solution after i cycles or $m \times i$ total iterations and r_i is the corresponding residual.

Our focus in this work is on the selection of the restart parameter for GMRES(m). Traditionally, it has been assumed that the larger the value of m, the fewer iterations are required for convergence because a large m improves the information in the GMRES residual polynomial (see, e.g., [15]). Moreover, a large enough m for GMRES(m) can to some extent reduce the impediment to super-linear convergence [23] and may be required to avoid stalling [20]. However, if m is too large, the goal of restarting as a means of reducing computational and storage costs is negated. Furthermore, it was recently shown that a smaller m can actually result in fewer iterations for some problems (see, e.g., [6,8]). This unexpected result highlights the practical difficulty in choosing an appropriate value of m.

In practice, one generally attempts to choose a value for m that balances the good convergence properties typically resulting from a large value with the reduction of computational work resulting from a smaller value. The value of the GMRES(m) restart parameter is typically chosen prior to the solve and remains fixed for the entire solve. However, several authors have proposed varying the restart parameter for a variety of reasons. In [15], Joubert aims to reduce the total time to solution by examining the effect of m on both convergence behavior and computational cost. He proposes an adaptive method that determines whether or not to restart based on a sophisticated test criterion that weighs the work requirements against the estimated residual norm reduction for both scenarios. In [22], the authors propose an adaptive implementation of GMRES(m) with the goal of avoiding stagnation. When their test criterion detects stagnation, the restart parameter is increased. Similarly, the adaptive method proposed in [12] is a modification of the method in [22] that also increases m to avoid stagnation, but then reduces it after a fixed number of cycles to better control the costs. The adaptive method in [25] also has the goal of preventing stagnation and chooses m based on a comparison

of the Ritz and harmonic Ritz values.

We are interested in the selection of the GMRES(m) restart parameter and its effect on the time to solution. Our investigation into using a non-fixed restart parameter to improve performance was largely motivated by two considerations. First, previously mentioned works demonstrating that a larger m does not necessarily result in fewer iterations, together with the fact that restart cycles with smaller m are cheaper per iteration, influenced us to experiment with reducing the restart parameter whenever possible. Second, an observation was made in [2] that the GMRES(m) residual vectors at the end of each restart cycle may alternate direction in a repetitive fashion, thereby slowing convergence. Thus, our investigation was also motivated by the possibility that varying the restart parameter could be beneficial for convergence by disrupting this repetitiveness. We restrict our examination of the convergence of GMRES(m) to the acceleration of time to solution via selection of restart parameters. We are not concerned at this time with the stagnation of GMRES(m); a number of other works address stagnation issues through a variety of approaches (see, e.g. [22,12,25,17,9,11]). Our approach to this investigation is largely experimental, though we do provide analysis for the case when A is symmetric or normal. Our primary contribution is the simplicity of our approach for modifying the restart parameter and its effectiveness on a variety of problems.

This paper is organized as follows. In Section 2, we present an algorithm for varying the restart parameter in GMRES(m). In Section 3, we demonstrate the usefulness of our method for general matrices with experimental results from a variety of problems. Then, in Section 4, we offer some insight as to the effectiveness of the method via heuristic explanations based on experimental evidence and some theory for symmetric matrices. Finally, in Section 5, we give some concluding remarks.

2   A method for varying the restart parameter

In this section, we present a simple method for varying the restart parameter in GMRES(m). We first define two terms. We refer to the angles between consecutive residual vectors as *sequential* angles, e.g., $\angle(r_{i+1}, r_i)$, and the angles between *every other* residual vector as *skip* angles, e.g., $\angle(r_{i+1}, r_{i-1})$. As shown in [2], sequential angles for GMRES(m) are related to the drop in residual norm from one cycle to the next by

$$\cos \angle(r_{i+1}, r_i) = \frac{\|r_{i+1}\|_2}{\|r_i\|_2}. \tag{3}$$

3

When consecutive residual vectors are nearly orthogonal, the drop in residual norm between two consecutive restart cycles is large. In fact, if $r_{i+1} \perp r_i$, then the exact solution has been found. Note that for the purposes of our method, we calculate the angles such that they are always between 0 and 90 degrees.

Our strategy for varying the restart parameter includes specifying minimum and maximum restart parameters, m_{min} and m_{max}, respectively, such that $m_{min} \leq m_i \leq m_{max}$ is satisfied for each cycle i with restart parameter m_i. We refer to our modification of GMRES(m) as αGMRES(m_{max}, m_{min}), and pseudo-code is given in Figure 1. The first cycle begins with restart parameter m_{max}. After that, we calculate the value of the sequential angle at the end of each restart cycle to determine the next m_i. In essence, we decrease the restart parameter by a small number d at each cycle until we reach m_{min}. At that point, we increase m_i to the maximum m_{max}. However, if the sequential angle is large (close to 90), indicating good convergence, then we keep the current restart parameter instead of adjusting it. Likewise, if the sequential angle is small (close to zero), we revert to m_{max}. In practice, as seen in Figure 1, we check the convergence rate, $\|r_{i+1}\|_2 / \|r_i\|_2$, at the end of each cycle instead of the sequential angle value, c.f. (3). Note that when a small sequential angle is detected, a strategy that addresses stagnation, such as [11] or several others mentioned in the previous section, could be employed if desired. We generally take $d = 3$ and m_{min} very small (1 or 3) as the cycles with the small restart parameters are cheap. The parameter m_{max} is typically chosen the same as one would chose a fixed m for GMRES(m), storage considerations generally being a priority. Furthermore, we consider "small" sequential angles to be those less than about 8 degrees and "large" angles to be those greater than 80 degrees. These particular parameter values work well in our experiments, but can generally be adjusted for a particular class of problems.

From Figure 1, it is apparent that the simplicity of the test criterion for αGMRES(m_{max}, m_{min}) results in an negligible amount of overhead. Furthermore, one can easily and quickly modify an existing implementation of GMRES(m) to vary the restart parameter in this manner.

3 Numerical Experiments

In this section, we provide experimental evidence that demonstrates the effectiveness of the αGMRES(m_{max}, m_{min}) algorithm presented in the previous section. First, we look at a variety of problems available from the Matrix Market Collection [18] and the University of Florida Sparse Matrix Collection [5]. These problems are listed in Table 1. If a right-hand side was not provided, we generated a random right-hand side. Next, we examine a convection-diffusion problem that we generated with a finite-element code. Finally, we discuss a

```
        cr = 1;                    /* convergence rate */
        max_cr = cos(8)            /* max conv. rate = cosine of 8 degrees ≈ .99 */
        min_cr = cos(80);          /* min conv. rate = cosine of 80 degrees ≈ .175 */
        d = 3;                     /* increment for adjusting */
        i = 0                      /* counter for restart cycles*/
        x_0 = 0                    /* initial guess */
        r_i = b - Ax_i             /* residual */

    while (not converged)
        /* calculate restart parameter m_i */
        if (cr > max_cr or i == 0 ) /* first cycle or near stagnation */
            m_i = m_max
        else if ( cr < min_cr ) /* converging well */
            m_i = m_{i-1}
        else /* adjust */
            if (m_{i-1} - d) ≥ m_min
                m_i = m_{i-1} - d
            else
                m_i = m_max
        end
        /* restart cycle */
        for j = 1 : m_i
            /* gmres iteration */
            /* (break if convergence criterion is met) */
        end
        i = i + 1
        /* calculate conv. rate ( = cosine of the sequential angle) */
        cr = ||r_i||_2/||r_{i-1}||_2
    end
```

Fig. 1. αGMRES(m_{max}, m_{min})

few experiments with preconditioned GMRES.

We use a modified implementation of GMRES(m) based on the version available in *hypre* 2.0 [10,13]. All tests were run until $\|r_i\|_2/\|r_0\|_2$ was less than 10^{-6}, and the times reported are the averages of 10 runs. A zero initial guess was used for all problems. For this comparison, we chose restart parameter $m = 30$ for GMRES(m) because it is a common choice and often the default in general linear solver packages such as PETSc [3]. For αGMRES(m_{max}, m_{min}), we chose $m_{max} = 30$ and $m_{min} = 3$ to keep the storage requirement less than or equal to that of GMRES(30).

Table 1

List of test problems together with the matrix order (n), number of nonzeros (nnz), source (MM = Matrix Market, UF = University of Florida Collection) and the application area.

	Problem	n	nnz	Source	Application Area
1	add20	2395	17,319	MM	computer component design
2	cdde1	961	4681	MM	2D convection-diffusion operator
3	circuit_2	4510	21,199	UF	circuit simulation
4	epb1	14,734	95,053	UF	heat exchanger simulation
5	FEM_3D_thermal1	17,880	430,740	UF	3D nonlinear thermal problems
6	fpga_trans_01	1220	7382	UF	circuit simulation
7	matrix-new_3	125,329	893,984	UF	semiconductor device
8	orsirr_1	1030	6858	MM	oil reservoir simulation
9	orsreg_1	2205	14,133	MM	oil reservoir simulation
10	pde2961	2961	14,585	UF	model PDE problem
11	raefsky1	3242	294,276	UF	incompressible fluid flow
12	raefsky2	3242	293,551	UF	incompressible fluid flow
13	rdb2048	2048	12,032	MM	reaction-diffusion Brusselator model
14	sherman4	1104	3786	MM	oil reservoir simulation
15	steam2	600	13,760	MM	injected steam oil recovery
16	stomach	213,360	3,021,648	UF	bioengineering
17	wang2	2903	19,093	UF	electron continuity equations
18	wang3	26,064	177,168	UF	electron continuity equations
19	watt_1	1856	11,360	MM	petroleum engineering
20	young3c	841	3988	UF	acoustic scattering

3.1 Matrices from test collections

The x-axis in Figure 2 corresponds to the 20 test problems in Table 1. The y-axis is the ratio of time to converge for GMRES(30) to the time to converge for αGMRES(30, 3). All of the bars extend above one, indicating that αGMRES(3, 30) requires less time to converge than GMRES(30). For example, for Problem 1, GMRES(30) takes roughly 1.7 times as long to converge as does αGMRES(30, 3). We compare convergence times here because our interest is in improving the time to solution. Note that for two of these problems (Problems 13 and 16), GMRES(30) actually requires fewer iterations than does αGMRES(30, 3). However, because iterations in restart cycles with smaller restart parameters are on average cheaper, the time to solution of αGMRES(30, 3) is faster.

Because of the effectiveness of αGMRES(30, 3) for these problems, a natural question is whether using a fixed parameter smaller than 30 would be more beneficial. We did not do an extensive parameter study, but instead additionally looked at parameters $m = 20$, 10, and 3. In general, for all of the test problems, the number of iterations increased as m decreased. In terms
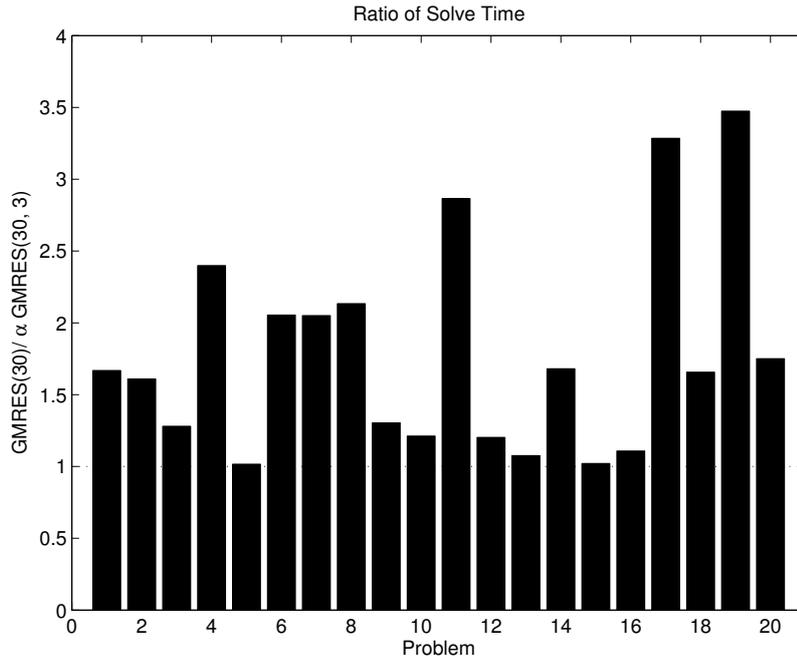
Fig. 2. Matrices from test collections: a comparison of the time to convergence for GMRES(30) and αGMRES(30, 3). Times are averages of 10 runs.

of the time to solution, GMRES(30) was only beat twice, and that was by GMRES(10) for Problems 16 and 18. When comparing GMRES(m), with $m = 30, 20, 10$, and 3, to αGMRES(30, 3), Problem 16 was the only problem for which αGMRES(30, 3) was not the fastest. For Problem 16, while GMRES(10) was faster than αGMRES(30, 3), αGMRES(10, 3) was still faster than GMRES(10). Therefore, our experimental results here indicate that varying the restart parameter in the manner of αGMRES(m_{max}, m_{min}) is typically an improvement compared to GMRES(m_{max}) and, at the very least, does not increase the time to solution.

3.2 Convection-diffusion problem

Now we look at a 2D convection-diffusion problem discretized with linear finite elements on an unstructured meshing of the unit square:

$$-\Delta u - c \cdot \nabla u = f. \tag{4}$$

The boundary conditions are Dirichlet ($u = 0$) on the bottom and top of the square and Neumann ($\partial u/\partial n = 0$) on the remainder of the boundary. The right-hand side is $f = 1$. The resulting matrix A is of size $n = 35,169$, with $242,843$ nonzeros. A coarse version of the unstructured mesh is shown in Figure 3.
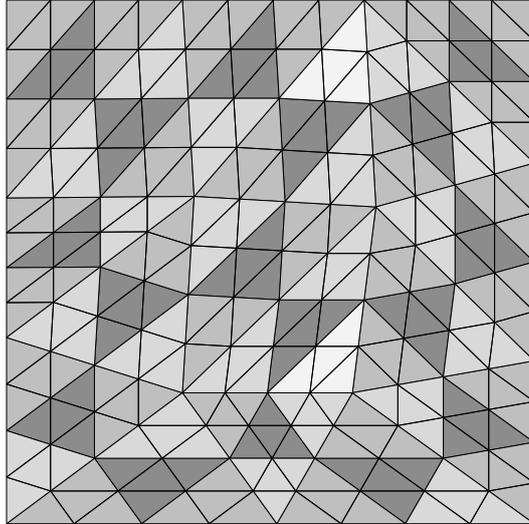
7

Fig. 3. A coarse version of the unstructured mesh used for the convection-diffusion problems.

This convection-diffusion equation makes an interesting test problem because, by varying vector c, we can easily affect the difficulty of the problem for GMRES(m) and αGMRES(m_{max}, m_{min}). We created 14 test problems by varying the components of c between 0 and 1550. For each test problem, the two components of c, corresponding to the x- and y-dimensions were chosen to be the same. The choice $c = 0$ corresponds to the symmetric case, and, as c increases, the matrix A becomes increasingly nonsymmetric. The choice $c = 1550$ corresponds approximately to the smallest value of c for which stagnation occurs. In Table 2, we list the chosen values for c along with the assigned problem number.

First we discuss the convergence of GMRES(m) for these problems. The numbers of GMRES(30) and GMRES(10) iterations required for convergence are listed in Table 2. If GMRES(m) did not converge in 10,000 iterations, then we say that the problem did not converge. Problem 1, which is symmetric, takes more GMRES(30) iterations to converge than any other problem, with the exception of Problem 14, which stagnates. The problems in the middle range of c take the fewest iterations to converge. Additionally, unlike the test problems in the previous section and most problems we have encountered in practice, many of these convection-diffusion problems (Problems 5-11) are unusual in that the number of iterations required for GMRES(m) convergence generally decreases with decreasing m. With fewer iterations required for a smaller restart parameter, GMRES(10) is clearly faster than GMRES(30) for Problems 5-11. GMRES(10) is also faster for Problems 4, 12, and 13, despite taking slightly more iterations. (Though we note that it is only beneficial to decrease m up to a point: GMRES(3) takes both more iterations and more time to converge than does GMRES(10) for all problems.)

8

Table 2

Numbered list of test problems generated by the convection-diffusion equation (3.2) together with the convection coefficients (c), and the numbers of GMRES(30) and GMRES(10) iterations (and time in seconds) required to achieve $\|r_i\|_2/\|r_0\|_2 < 10^{-6}$. (DNC = does not converge.)

	c	GMRES(30)	GMRES(10)
1	0	7473 (78.1)	DNC (–)
2	1	7130 (74.2)	DNC (–)
3	10	1728 (18.1)	5263 (32.1)
4	50	585 (6.2)	925 (5.7)
5	100	590 (6.1)	560 (3.4)
6	300	771 (8.0)	603 (3.7)
7	500	891 (9.3)	751 (4.6)
8	700	1004 (10.3)	903 (5.5)
9	900	1125 (12.0)	1036 (6.4)
10	1100	1255 (13.1)	1140 (7.0)
11	1300	1334 (13.8)	1279 (7.8)
12	1500	2007 (20.8)	2280 (14.0)
13	1525	2495 (26.0)	3030 (18.5)
14	1550	DNC (–)	DNC (–)

We now again compare the time to solution for GMRES(30) to αGMRES(30, 3). The x-axis in Figure 4 corresponds to the 14 test problems in Table 2. The y-axis is the ratio of time to converge for GMRES(30) to the time to converge for αGMRES(30, 3). For all of these problems, αGMRES(3, 30) requires less time to converge than GMRES(30). Problems 1 and 2, which are symmetric or nearly symmetric and require the most iterations, benefit the most from the adaptive scheme. (In the next section, we discuss in some detail the performance of GMRES(m) on symmetric problems.) On the other hand, Problem 14 stagnates for GMRES(m_{max}), and as mentioned previously, αGMRES(m_{max}, m_{min}) does not overcome stagnation. Overall, it appears that αGMRES(3, 30) provides the most improvement for problems that are close to symmetric.

Recall that for Problems 5-11, GMRES(10) requires fewer iterations and is faster than GMRES(30). Therefore, it is not surprising that GMRES(10) is also faster than αGMRES(30, 3) for these problems. However, for these problems, αGMRES(10, 3) is either slightly faster or about the same as GMRES(10). The trend in the results in Figure 4 could also be interpreted as αGMRES(m_{max}, m_{min}) is less beneficial for problems for which the GMRES(m) iteration count decreases with decreasing m.
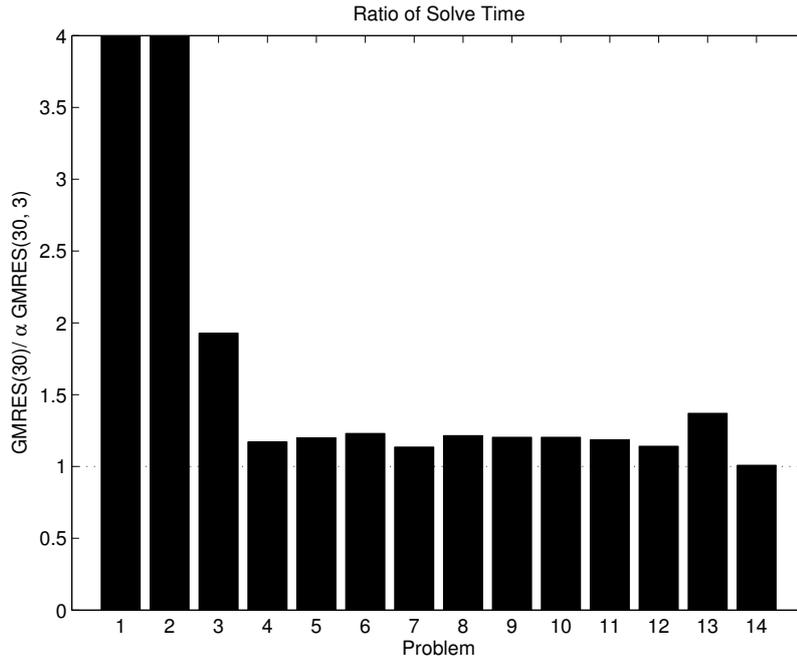
Fig. 4. Convection-diffusion problems: a comparison of the time to convergence for GMRES(30) and αGMRES(30, 3). Times are averages of 10 runs.

3.3   GMRES with preconditioning

As with most acceleration strategies, improving the time to solution for GMRES(m) via the αGMRES(m_{max}, m_{min}) method is not intended as an alternative to a good preconditioner. Therefore, while our focus in this paper is on GMRES(m) convergence without preconditioning, as a curiosity, we provide the results of limited testing of αGMRES(30, 3) with preconditioners in Table 3. The first five test problems are from the University of Florida Sparse Matrix Collection, and the sixth problem is Problem 3 from Table 2 in the previous section. The preconditioners used are an algebraic multigrid preconditioner, BoomerAMG, an ILU preconditioner, Euclid, and a sparse approximate inverse (SPAI) preconditioner, ParaSails, all available in the *hypre* software library. (See [13] for additional information on these preconditioners.) We do not claim to have found the optimal preconditioner for any of these problems, but simply chose a variety for illustrative purposes.

While a more extensive investigation of preconditioning is a subject for future work, these test problems seem to indicate that our restart parameter strategy can still be effective when combined with preconditioning. As in the case of no preconditioning, for problems that already converge quickly, the scheme does not have a negative effect. For example, Problem 3 converges in fewer than four GMRES(30) restarts, and the sequential angles are large enough that αGMRES(3, 30) never changes the restart from the initial $m = 30$.

10

Table 3
List of test problems together with the matrix order (n), preconditioner, application area, and percentage improvement (%) of the time to solution of αGMRES(3, 30) over that of GMRES(30).

	Problem	n	Preconditioner	Application Area	%
1	adder_dcop_01	1813	AMG	circuit simulation	28
2	bcircuit	68,902	ILUT(0.01, 10)	semiconductor simulation	50
3	epb3	84,617	ILU(0)	heat exchanger simulation	0
4	garon2	13,535	ILU(0)	computational fluid dynamics	18
5	xenon1	48,600	AMG	crystalline compound materials	9
6	Problem 3, Table 2	35,169	SPAI	convection-diffusion	15

4 Discussion

The purpose of this section is to provide a heuristic explanation as to why our simple strategy for varying the restart parameter is often effective. As mentioned in Section 1, the motivation for our new algorithm came from both the potential to disrupt repetitive behavior in GMRES(m) as well as the desire to use a smaller m whenever possible. Therefore, we first analyze the repetitive behavior of symmetric matrices in Section 4.1. Then, in Section 4.2, we explain our algorithm design in terms of the theory in Section 4.1 and the desire to reduce computational cost. To further support our strategy for changing the restart parameter, we discuss a work per gain in accuracy metric in Section 4.3. Finally, in Section 4.4, we provide additional comments on our algorithm related to the numerical results presented in the previous section.

4.1 Symmetric matrices: repetitive convergence behavior

Our motivation for exploring variable restart parameters came in part from an observation in [2]: GMRES(m) residual vectors often point in nearly the same direction after every other restart cycle. For example, for restart cycles i and $i+2$, one may observe that $r_{i+2} \approx \sigma \, r_i$, where $\sigma \leq 1$. Furthermore, it was observed in [2] that slow GMRES(m) convergence can often be attributed to this alternating behavior for both symmetric and non-symmetric problems. In the absence of small sequential angles (which indicate stagnating), small skip angles (defined at the start of Section 2) indicate that alternating behavior is occurring.

In [2], it is proved that alternating must occur for both symmetric and skew-symmetric problems for the special case of a restart parameter that is one less than the matrix order. Here we demonstrate more generally that alternating occurs in GMRES(m) for *any* nonincreasing restart parameter and symmetric or skew-symmetric A. Let ϕ_i be the skip angle between the residual vectors at

11

the end of restart cycles $i + 1$ and $i - 1$, r_{i+1} and r_{i-1}, respectively. Let $(\cdot, \cdot)$ and $\| \cdot \|_2$ be the usual Euclidean inner product and norm in $\mathbb{R}^n$. We expect

$$\cos \phi_i = \frac{(r_{i+1}, r_{i-1})}{\|r_{i+1}\|_2 \|r_{i-1}\|_2} \tag{5}$$

to approach one if alternating is occurring. (In this discussion, we tacitly ignore the case where $r_i = 0$, for any i, as this indicates that the method has converged.) Following from (2), we can write the GMRES(m) residual at restart cycle i in terms of a polynomial in A times the previous residual. In particular,

$$r_i = r_{i-1} - \sum_{k=1}^{m} \alpha_{ik} A^k r_{i-1}, \tag{6}$$

where α_{ik} are chosen such that $\|r_i\|_2$ is minimized, or equivalently, $r_i \perp AK_m(A, r_{i-1})$ (see, e.g., [19]).

Theorem 1 (Alternating residuals when m_i is nonincreasing) *When $A \in \mathbb{R}^{n \times n}$ is symmetric or skew-symmetric and m_i is nonincreasing for successive restart cycles i, GMRES(m) produces a sequence of residual vectors r_i at the end of each restart cycle i such that the skip angles approach zero, i.e.,*

$$\lim_{i \to \infty} \cos \phi_i = 1.$$

Proof.

Because we allow m_i to vary at each cycle, we have

$$r_i = r_{i-1} - \sum_{k=1}^{m_i} \alpha_{ik} A^k r_{i-1}. \tag{7}$$

From (7),

$$(r_{i+1}, r_{i+1}) = (r_{i+1}, r_i - \sum_{k=1}^{m_{i+1}} \alpha_{(i+1)k} A^k r_i).$$

By definition, $r_{i+1} \perp AK_m(A, r_i)$, so the above reduces to

$$(r_{i+1}, r_{i+1}) = (r_{i+1}, r_i). \tag{8}$$

Similarly,

$$(r_{i+1}, r_{i-1}) = (r_i - \sum_{k=1}^{m_{i+1}} \alpha_{(i+1)k} A^k r_i, r_{i-1}),$$

12

and since A is symmetric (or skew-symmetric), the above reduces to

$$(r_{i+1},\ r_{i-1}) = (r_i,\ r_{i-1}) - \sum_{k=m_i+1}^{m_{i+1}} (\alpha_{(i+1)k} A^k r_i,\ r_{i-1}). \tag{9}$$

Because m_i is nonincreasing, $m_{i+1} \le m_i$, (9) reduces to

$$(r_{i+1},\ r_{i-1}) = (r_i,\ r_{i-1}). \tag{10}$$

Therefore, from (8) and (10),

$$(r_{i+1},\ r_{i-1}) = (r_i,\ r_i).$$

Now, using (5), we can write

$$\cos\phi_i = \frac{\|r_i\|_2^2}{\|r_{i+1}\|_2 \|r_{i-1}\|_2}. \tag{11}$$

Let a_i be the sequence

$$a_i = \frac{\|r_{i+1}\|_2}{\|r_i\|_2}. \tag{12}$$

Substituting into (11),

$$\cos\phi_i = \frac{a_{i-1}}{a_i}. \tag{13}$$

Because $\cos\phi_i \le 1$, from (13), $a_{i-1} \le a_i$. Therefore, the sequence a_i is monotone increasing. (Note that $a_i > 0$ if $r_i \ne 0$.) By construction, $\|r_{i+1}\|_2 \le \|r_i\|_2$, and, therefore, the sequence a_i is bounded. As a result, there exists a β such that $a_i \to \beta$ as $i \to \infty$ which implies that

$$\lim_{i \to \infty} \cos\phi_i = \frac{\beta}{\beta} = 1.$$

☐

The above shows that for symmetric (or skew-symmetric) matrices, alternating is a limiting process for GMRES(m) in the case of a nonincreasing or a fixed restart parameter. This result corroborates the observation made in [2] that, for a fixed restart parameter, the skip angles tend to zero (alternating becomes more pronounced) as the iteration progresses. We note that GMRES(30) for Problem 1 in Table 2, which is symmetric, has average and median skip angles of .3 and .07 degrees, respectively. While the above proof shows that this convergence happens on an infinite scale, in practice, the limit may be reached quite rapidly. In fact, a_i may be thought of as the convergence rate of GMRES(m) and correlates to the limiting sequential angle as shown

13

in (3). It is important to observe that, when this limit is reached, GMRES(m) convergence continues at this rate and does not improve.

The particular result regarding the convergence rate can be extended to normal matrices, as shown in Theorem 2 below. In contrast to Theorem 1, the skip angle may not converge to zero.

Theorem 2 (Convergence for normal matrices) *Let $A \in \mathbb{R}^{n \times n}$ be normal and m_i nonincreasing for successive restart cycles i of GMRES(m). Then the sequence of convergence factors defined by a_i in (12) is nondecreasing and tends to a limiting value. In particular, the convergence of GMRES(m) gets worse with every restart cycle.*

Proof. As in the proof of Theorem 1, from (12) and (8), we have

$$\frac{a_{i-1}}{a_i} = \frac{(r_i,\ r_{i-1})}{\|r_{i+1}\|_2 \|r_{i-1}\|_2}.$$

Let P_{i+1} be the GMRES(m) polynomial at restart cycle $i+1$ defined by

$$P_{i+1} = I - \sum_{k=1}^{m_{i+1}} \alpha_{(i+1)k} A^k$$

such that $r_{i+1} = P_{i+1} r_i$, c.f.(6). Because m_i is nonincreasing, we have

$$(r_i,\ r_{i-1}) = (r_i,\ P_{i+1} r_{i-1}) = (P_{i+1}^T r_i,\ r_{i-1}).$$

Therefore,

$$\frac{a_{i-1}}{a_i} = \frac{(P_{i+1}^T r_i,\ r_{i-1})}{\|P_{i+1} r_i\|_2 \|r_{i-1}\|_2} \leq \frac{\|P_{i+1}^T r_i\|_2}{\|P_{i+1} r_i\|_2}. \tag{14}$$

Since A is normal, $P_{i+1} P_{i+1}^T = P_{i+1}^T P_{i+1}$, and the last term in (14) is equal to one. Thus, a_i is a nondecreasing sequence. As shown previously, $a_i \leq 1$. Note that if A has a positive definite symmetric part, then a_i may be bounded by a number strictly less than one (see, e.g., [7]).   □

Another confirmation of the repetitive behavior of GMRES(m) residual for symmetric matrices may be deduced from the work of Zavorin in [24]. This manuscript focuses on determining the worst-case behavior of (non-restarted) GMRES. In Section 5 of [24], Zavorin proves that A and A^T achieve the same worst-case behavior after an equivalent number of GMRES iterations and that this behavior happens at a "cross-equality point." He then provides an algorithm, called the CE algorithm, that determines vectors that satisfy cross-equality (though these are not necessarily the worst-case vectors). To link Zavorin's GMRES study to the alternating behavior for GMRES(m) restart

cycles, two key observations are required. First, Zavorin defines a vector c, quantifying GMRES performance, which is simply related to the residual r by $c = \frac{r}{\|r\|_2^2}$ (see the proof of Theorem 2.1 in [14]). Second, when $A = A^T$, one sweep of the CE algorithm is essentially two cycles of GMRES(m) [16]. In particular, if one introduces vectors c_i corresponding to each restart cycle, then they are related to the GMRES(m) residuals r_i by

$$c_i = \sigma_i \frac{r_i}{\|r_i\|_2}.$$

Here σ_i is a sequence of positive numbers defined by $\sigma_0 = \|r_0\|_2$ and $\sigma_{i+1}\sigma_i = \frac{\|r_i\|_2}{\|r_{i+1}\|_2}$. The CE algorithm is guaranteed to converge, and its stopping criterion has the form

$$\|c_{i+1} - c_{i-1}\| < \epsilon. \tag{15}$$

Recalling from the previous section that ϕ_i is the skip angle, (15) can be shown to be equivalent to

$$\sigma_{i-1} |\sin \phi_i| < \epsilon.$$

Note that $\sigma_{i+1} = (\cos \phi_i)\sigma_{i-1}$, and, therefore, $\sigma_i \geq \min\{\frac{1}{\|r_0\|_2}, \|r_1\|_2\}$. Thus, the convergence of the CE algorithm for any ϵ means that $\phi_i \to 0$ when $i \to \infty$.

4.2 Motivation for our strategy

While we have only shown above that alternating occurs for symmetric matrices, this alternating behavior is often observed for nonsymmetric matrices as well [2]. Intuitively, it seems that a diversity of approximation spaces is important for fast GMRES(m) convergence, and therefore, this repetitive behavior is potentially damaging. This conjecture is supported, for example, by the effectiveness of the LGMRES method [2], which prevents alternating and generally improves convergence over GMRES(m) for both symmetric and nonsymmetric problems. With this in mind, as mentioned in Section 1, our investigation into varying the restart parameter was partially motivated by the notion that convergence could be accelerated by "disrupting" repetitive behavior. The potential for disrupting repetitiveness by varying the restart parameter is supported by the results of experiments in [1]. There, experiments are performed with GMRES(m) whereby the restart parameter is chosen randomly at each cycle i such that $m_{min} \leq m_i \leq m_{max}$. This random selection technique is shown experimentally always to increase the median skip angle over that of GMRES(m_{max}) for the test problems in [1] and often to improve

15

the time to solution. However, randomly choosing restart parameters can also slow down convergence and does not make for a practical and reliable technique.

In addition, it is well known that minimizing the residual norm as much as possible within each cycle (for example, by choosing the largest m possible) is not always optimal and that the residual vector direction is more important that the norm (see, e.g., [21]). Furthermore, contrary to conventional wisdom, a smaller m can occasionally even reduce the number of iterations as required (see, e.g., [6,8]). Therefore, we feel that there is little motivation to use the largest possible m for each restart cycle, particularly when cycles with smaller m are computationally cheaper per iteration. (When stagnation is detected, a larger m may be needed for a few cycles, but, as noted in Section 1, that issue has been addressed by others.) We further discuss the potential benefit of reducing the restart parameter in Section 4.3.

Therefore, based on the above heuristic considerations, we arrived at the manner of varying the restart parameter presented in Section 2. First, recall from Theorem 1 that the skip angle converges for symmetric matrices when m_i is nonincreasing. Consistent with this theorem, we typically observe experimentally that, when the restart parameter is decreased incrementally, the skip angles also tend to get smaller for nonsymmetric matrices. In addition, when the restart parameter changes from a small to a large value, the increase in skip angle is often quite large. Therefore, our intent is to improve performance by choosing a pattern of restart parameters that includes periodic jumps from small to large restart parameters. To this end, we begin the method with the largest allowable restart parameter, m_{max}, which reduces the initial residual as much as possible in the first cycle. We then slowly decrease the restart parameter by a small number d and allow the skip angles to gradually get smaller while providing some diversity in restart parameter values. We recommend choosing $d = 3$, as this appears to be the most effective value. When we reach the minimum m_{min}, we increase m_i to the maximum m_{max} and benefit from a large decrease in the residual norm. Because cycles with small restart parameters are relatively cheap, it is reasonable to choose m_{min} very small; we typically choose a value of 1 or 3.

We monitor the value of the sequential angle (or convergence rate) at the end of each restart cycle to determine whether adjustments to this default pattern should be made. In particular, if convergence is very good, which is indicated by a large sequential angle, there is no need to modify the current restart parameter and risk convergence worsening. On the other hand, if the method is stagnating, we revert to m_{max}. If the use of m_{max} still leads to stagnation, it is highly unlikely that anything will be gained by varying the restart parameter further. In fact, as previously mentioned, to increase robustness, our method could invoke a separate stagnation strategy when stagnation is detected. In

essence, the monitoring of the sequential angle prevents our method from performing worse than GMRES(m_{max}) for problems for which GMRES(m_{max}) converges well.

4.3 Work per gain in accuracy considerations

Additional motivation for using a sequence of decreasing restart parameters can be derived by considering the work per gain in accuracy (WPGA) measurement (see, e.g., [4]). For example, for GMRES(m), the work for a restart cycle $i + 1$ with restart parameter $m = m_{i+1}$ is $O(m^2)$ due to the orthogonalization costs (we ignore the matrix-vector multiply costs as they do not depend on m). We then define our measure of gain in accuracy by $\frac{\|r_i\|_2}{\|r_{i+1}\|_2}$. In terms of the convergence rate $a_i = \frac{\|r_{i+1}\|_2}{\|r_i\|_2}$, then we have the WPGA for a single restart cycle defined as

$$WPGA = \frac{work}{gain\ in\ accuracy} = a_i m^2. \tag{16}$$

Consider a smaller restart parameter $\tilde{m} = m - d$ and let $\tilde{a}_i$ be the corresponding convergence rate. A natural question is whether there exists a reasonable d (i.e., positive and less than m) such that the WPGA is lower with $\tilde{m}$ than m. In other words, is there a d such that

$$\tilde{a}_i \tilde{m}^2 < a_i m^2 \tag{17}$$

holds? Assuming A is normal, from Theorem 2, we have $a_{i-1} \leq a_i \leq \tilde{a}_i \leq 1$. Using the these bounds, a straightforward calculation shows that if $a_{i-1} > .25$, then $d \geq \frac{(1 - \sqrt{a_{i-1}})m}{\sqrt{a_{i-1}}}$ satisfies (17). In other words, if the convergence rate is not small, then it makes sense to use a smaller restart parameter. This analysis can be extended to k consecutive restart cycles. In this case, the WPGA is

$$WPGA = (\prod_{i=1}^{k} a_i)(\sum_{i=1}^{k} m_i^2). \tag{18}$$

Now we define $m_i = m$ and $\tilde{m}_i = m - id$, and we want to determine whether a d exists such that $0 \leq d \leq m/(k+1)$ and

$$(\prod_{i=1}^{k} \tilde{a}_i)(\sum_{i=1}^{k} \tilde{m}_i^2) \leq (\prod_{i=1}^{k} a_i)(\sum_{i=1}^{k} m^2) \tag{19}$$

is satified. We let $\tilde{a} = \prod_{i=1}^{k} \tilde{a}_i$ and $a = \prod_{i=1}^{k} a_i$ and assume, without loss of generality, that $\tilde{a} \geq a$. In Table 4, we give estimates of pairs of a and d that satisfy (19) for $m = 30$ and multiple values of k. Because the value of a represents the convergence progress over several cycles, our estimates show

17

Table 4
For fixed values of k between 1 and 10, this table lists the value of a above which the WPGA can be decreased by reducing m by d for each of the k cycles. The corresponding average convergence factor per cycle ($\hat{a}_i$) is also listed.

k	a	d	$\hat{a}_i$
1	.25	15	.25
2	.28	10	.53
3	.29	7	.66
4	.30	6	.74
5	.31	5	.79
6	.31	4	.82
7	.31	3	.85
8	.31	3	.87
9	.32	3	.88
10	.32	2	.89

that if this convergence is less than $1/3$, we could have reduced the restart parameter without penalty in terms of the WPGA. The average convergence factor per cycle, $\hat{a}_i$, is simply $a^{1/k}$. We note that the estimates in our table are pessimistic (for example, we assume $\tilde{a} = 1$), and in practice it is beneficial to reduce m for much smaller values of the average convergence factor.

4.4   Additional comments on experimental results

Varying the restart parameter certainly appears to be very effective for the symmetric and nearly symmetric convection-diffusion equation test problems, Problems 1 and 2 in Table 2. For these two problems, the number of iterations required for convergence is reduced significantly by our adaptive scheme. In fact, for problems that are symmetric or very close to symmetric such as Problems 1 and 2, we find that using $max\_cr = 1.0$ (see Figure 1) typically yields the optimal results. In other words, we always adjust the restart parameter regardless of how close the convergence rate is to 1.0. Interestingly, we also note that Joubert states in [15] that his adaptive method for choosing m is much more effective than using a fixed m for symmetric problems.

In terms of the effect of varying the restart parameter according to αGMRES(m_{max}, m_{min}) on the skip angles, there is a mild correlation. We collected the skip angle at each restart cycle for the results shown in Section 3. For the problems in Table 1, the median skip angle for αGMRES(30, 3) was larger than that of GMRES(30) for 15 of the 20 problems (all but Problems 2, 9, 10, 12, and 16). For the convection-diffusion problems in Table 2, the median skip angle was increased by αGMRES(30, 3) only for Problems 1-3. The remaining problems had relatively high skip angles to begin with.

Disregarding any possible positive effect of varying the restart parameter on the repetitiveness of GMRES(m), much of the decrease in solution time can be attributed to the fact that restart cycles with smaller restart parameters are cheaper per iteration. Therefore, it is certainly beneficial to use smaller restart parameters when doing so does not negatively impact the convergence rate. We generally find that the benefit of αGMRES(m_{max}, m_{min}) is greater when the difference between m_{max} and m_{min} is enough to impact the orthogonalization costs and allow for more variation in the restart parameter. For example, αGMRES(30, 3) is typically more of an improvement over GMRES(30) than αGMRES(10, 3) is over GMRES(10). In the case of a smaller m_{max}, such as 10, it may be advantageous to lower d to 2 and max_cr to something on the order of .92. In general, we find that a higher m_{max} such as 30 or 20 is much less sensitive to the choice of d and max_cr than is $m_{max} = 10$, for example.

5 Concluding remarks

In this paper, we explore the feasibility of improving the time to solution of GMRES(m) by changing the restart parameter at each cycle. While our proposed strategy is certainly a heuristic technique, it appears to be an effective one. Our numerical experiments indicate that αGMRES(m_{max}, m_{min}) is typically an improvement over GMRES(m_{max}) and generally leads to a faster time to solution. While there are many sophisticated techniques for improving GMRES(m) convergence, the strengths of our adaptive strategy are its effectiveness, the ease with which it can be incorporated into an existing GMRES(m) code, and the potential for complementing other GMRES(m) acceleration or stagnation-avoidance techniques.

Acknowledgments

References

[1] A. H. Baker. *On improving the performance of the linear solver restarted GMRES*. PhD thesis, University of Colorado at Boulder, 2003.

[2] A. H. Baker, E. R. Jessup, and T. Manteuffel. A technique for accelerating the convergence of restarted GMRES. *SIAM Journal on Matrix Analysis and Applications*, 26(4):962–984, 2005.

[3] Satish Balay, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc Web page, 2001. http://www.mcs.anl.gov/petsc.

[4] M. Berndt, T. Manteuffel, and S. McCormick. Local error estimates and adaptive refinement for first-order system least squares (FOSLS). *Electronic Transactions on Numerical Analysis*, 6:35–43, 1998.

[5] T. Davis. University of Florida sparse matrix collection, http://www.cise.ufl.edu/research/sparse/matrices., 2007.

[6] M. Eiermann, O. G. Ernst, and O. Schneider. Analysis of acceleration strategies for restarted minimum residual methods. *Journal of Computational and Applied Mathematics*, 123:261–292, 2000.

[7] S. C. Eisenstat, H. C. Elman, and M. H. Schultz. Variational iterative methods for nonsymmetric systems of linear equations. *SIAM Journal on Numerical Analysis*, 20(2):345–357, 1983.

[8] M. Embree. The tortoise and the hare restart GMRES. *SIAM Review*, 45(2):259–256, 2003.

[9] Azeddine Essai. Weighted FOM and GMRES for solving nonsymmetric linear systems. *Numerical Algorithms*, 18:277–292, 1998.

[10] Robert D. Falgout and Ulrike Meier Yang. *hypre*: a library of high performance preconditioners. In P.M.A. Sloot, C.J.K. Tan., J.J. Dongarra, and A.G. Hoekstra, editors, *Computational Science - ICCS 2002 Part III*, volume 2331 of *Lecture Notes in Computer Science*, pages 632–641. Springer–Verlag, 2002.

[11] M. A. Gomes-Ruggiero, V. L. da Rocha Lopes, and J. V. Toledo-Benavides. Some applications for Newton-Krylov methods with a safeguard for GMRES(m). Technical report, DMA - Imecc - Unicamp, October 2006. Available from http://www.ime.unicamp.br/ marcia/homepage/artigos.htm.

[12] Mitsuru Habu and Takashi Nodera. GMRES(m) algorithm with changing the restart cycle adaptively. In *Proceedings of the Algorithmy 2000 Confererence on Scientific Computing*, pages 254–263, 2000. Available from www.emis.de/journals/AMUC/_contributed/algo2000.

[13] hypre: a library of high performance preconditioners, http://www.llnl.gov/casc/linear_solvers/, 2007.

[14] Ilse C. F. Ipsen. Expressions and bounds for the GMRES residual. *BIT*, 40(3):524–535, 2000.

[15] W. Joubert. On the convergence behavior of the restarted GMRES algorithm for solving nonsymmetric linear systems. *Numerical Linear Algebra with Applications*, 1(5):427–447, 1994.

[16] Julien Langou. University of Colorado at Denver, Department of Mathematics, personal communication, 2005.

[17] Ronald B. Morgan. GMRES with deflated restarting. *SIAM Journal on Scientific Computing*, 24(1):20–37, 2002.

[18] National Institute of Standards and Technology, Mathematical and Computational Sciences Division. Matrix Market. http://math.nist.gov/MatrixMarket, 2007.

[19] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, 1996.

[20] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[21] V. Simoncini. A new variant of restarted GMRES. *Numerical Linear Algebra with Applications*, 6:61–77, 1999.

[22] Maria Sosonkina, Layne T. Watson, Rakesh K. Kapania, and Homer F. Walker. A new adaptive GMRES algorithm for achieving high accuracy. *Numerical Linear Algebra with Applications*, 5:275–297, 1998.

[23] H. A. van der Vorst and C. Vuik. The superlinear convergence behavior of GMRES. *Journal of Computational and Applied Mathematics*, 48:327–341, 1993.

[24] Ilya Zavorin. Spectral factorization of the Krylov matrix and convergence of GMRES.
Technical Report CS-TR-4309, Department of Computer Science, University of Maryland, 2001. Available from http://www.cs.umd.edu/Library/TRs/.

[25] Linjie Zhang and Takashi Nodera. A new adaptive restart for GMRES(m) method. *The ANZIAM Journal*, 46:409–426, 2005.