Journal of Computational Mathematics, Vol.23, No.1, 2005, 83–92.

A PRODUCT HYBRID GMRES ALGORITHM FOR NONSYMMETRIC LINEAR SYSTEMS *

Bao-jiang Zhong
(*Department of Mathematics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*)

Abstract

It has been observed that the residual polynomials resulted from successive restarting cycles of GMRES(m) may differ from one another meaningfully. In this paper, it is further shown that the polynomials can complement one another harmoniously in reducing the iterative residual. This characterization of GMRES(m) is exploited to formulate an efficient hybrid iterative scheme, which can be widely applied to existing hybrid algorithms for solving large nonsymmetric systems of linear equations. In particular, a variant of the hybrid GMRES algorithm of Nachtigal, Reichel and Trefethen (1992) is presented. It is described how the new algorithm may offer significant performance improvements over the original one.

Mathematics subject classification: 65F10.
Key words: Nonsymmetric linear systems, Iterative methods, GMRES, Hybrid, Harmonic Ritz values, Residual polynomials.

1. Introduction

We are interested in the polynomial methods (sometimes loosely referred to as Krylov subspace methods [11]) for solving linear systems of the form

$$Ax = b, \quad A \in R^{n\times n}; \quad x, b \in R^n.$$

Starting from an initial guess, x_0, the methods generate a sequence of iterates $\{x_i\}$ whose residuals $\{r_i = b - Ax_i\}$ satisfy

$$r_i = p_i(A)r_0.$$

Here $\{p_i(z)\}$, known as residual polynomials, satisfy that $\deg p_i \le i$ and $p_i(0) = 1$.

By requiring p_i be optimal in the sense that

$$\|r_i\| = \|p_i(A)r_0\| = \min_{\deg p\le i, p(0)=1} \|p(A)r_0\|, \tag{1}$$

the GMRES algorithm [10] is defined. Here and throughout $\|\cdot\|$ is used to refer to the standard 2-norm. To limit the average work per iteration, GMRES is often restarted every steps, leading to the GMRES(m) algorithm:

$$r_{km} = p_{m,k}(A)r_{(k-1)m},\ p_{m,k}\ \textit{selected by}\ (1)\ \textit{based on}\ r_{(k-1)m}\ (k = 1, 2, \cdots). \tag{2}$$

The average work per iteration for GMRES(m) applied to general matrices is proportional to mn; large values of m generally improve convergence but also increase the work per iteration.

Considerably cheaper algorithms are the hybrid iterative algorithms. These algorithms typically run GMRES until sufficient information is extracted from A, then construct a polynomial of degree m and re-apply it by means of a basic one-step iterative method; namely,

$$r_{km} = [p_m(A)]^k r_0, k = 1, 2, \cdots, \tag{3}$$

* Reviced April 4, 2003; final revised August 23, 2004.

which requires only order n work per iteration, independent of m. For a survey of the hybrid algorithms, see [7].

In order to minimize the risk of convergence failure, Joubert [5] proposed an adaptive hybrid algorithm. By the algorithm, the results from cycles of GMRES(m) are used to form an effective polynomial, which approximates the ideal GMRES polynomial [12] $p_*(z)$, i.e., the minimizer of the problem

$$\min_{\deg p \le m, p(0)=1} \|p(A)\|.$$

The performance of $p_*(z)$ is in some sense the best possible for the existing hybrid algorithms. However, the implementation of Joubert's algorithm is quite complicated. A much more economical algorithm to explore the ideal GMRES polynomial was studied by Zhong [16]. In Zhong's algorithm, the polynomial to be used by (3) is simply chosen from a few residual polynomials of GMRES(m). Since successive restarting cycles of GMRES(m) always differ from one another meaningfully, the flexibility in choosing the polynomial can improve convergence significantly in many cases.

On the other hand, the results of [1, 12] indicate that for nonsymmetric problems, using hybrid iteration for the sake of increased speed may mean sacrificing robustness. It is now known that matrices exist for which GMRES(m) converges but the iteration (3) with every polynomial of degree $m(< n)$ does not. Furthermore, we note that even in the case where the two methods both converge, the hybrid iteration may perform much more disappointingly than GMRES(m). For a trivial example, consider the following 2×2 linear system $Ax = b$ with

$$A = \begin{pmatrix} \lambda & \\ & c\lambda \end{pmatrix}, b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, x_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \tag{4}$$

where $\lambda \neq 0$ and $c \gg 1$. Since here $n = 2$, we take $m = 1$. The iteration (3) can yield rapid convergence only if the used polynomial $p_1(z) = 1 + \alpha z$ satisfies that $|p_1(\lambda)|$ and $|p_1(c\lambda)|$ both lie well enough below 1. However, when $|p_1(c\lambda)| \le 1$ is imposed, we have $-2 \le \alpha c\lambda \le 0$, which gives that $|p_1(\lambda)| \ge 1 - 2/c$. Inevitably, the iteration (3) will be very slowly convergent for a large c, e.g., 100. On the other hand, the convergence rate of GMRES(1) applied to the same problem is about $\|r_{km}\|/\|r_{(k-1)m}\| = \sqrt{2}/2$ per cycle.

The example is contrived, but similar phenomena occur frequently in scientific computing, especially when A has some extremely small eigenvalues. This problem appears to be an inherent limitation of the existing hybrid algorithms.

In this paper we propose a new hybrid iterative scheme, referred to as *product hybrid scheme*. When an existing hybrid algorithm is implemented with this scheme, the way of how the polynomial is constructed and how it is applied is maintained, but the iteration (3) now is based on a product of several polynomials, rather than a single polynomial of degree m. In particular, in Section 2 we introduce a product variant of the hybrid GMRES algorithm of Nachtigal, Reichel and Trefethen [7]. In Section 3 we derive an explicit polynomial characterization of GMRES(m), which provides the main motivation for developing the product hybrid scheme. In Section 4 we consider some implementation issues of the product hybrid GMRES algorithm. In Section 5 we present some numerical examples to illustrate the remarkable superiority of the new algorithm. Finally, in Section 6 we conclude the paper.

2. The Product Hybrid GMRES Algorithm

Unlike other hybrid algorithms, which first estimate eigenvalues and then apply this knowledge in further iterations, the hybrid GMRES algorithm proposed in [7] avoids eigenvalue estimates. Instead, it runs GMRES until the residual norm drops by a certain factor, and then re-applies the polynomial implicitly constructed by GMRES via a Richardson iteration. Correspondingly, its product variant can be schematically shown as:

Product hybrid GMRES algorithm

Phase I Run GMRES(m) until $\|r_{km}\|$ drops by a suitable amount. Set $s = k$ and construct the GMRES residual polynomials $\{p_{m,k}(z)\}_{k=1}^{s}$, which satisfy (2).

Phase II Re-apply the product polynomial

$$\pi_s(z) = p_{m,s}(z)p_{m,s-1}(z)\cdots p_{m,1}(z)$$

cyclically until convergence:

$$r_{ksm} = [\pi_s(A)]^k r_0, k = 2, 3, \cdots .$$

The structure of the algorithm is practically appealing. In Phase I the GMRES iteration produces iterates as by-product, and its cost is only slightly greater (about $m^2/3n$ SAXPY operation per step) than a standard GMRES(m) iteration, due to the calculation of $\pi_s(z)$.

Throughout this paper it is assumed that r_0 contains components in all eigenvector directions of the matrix A. In fact, if r_0 is deficient in some eigenvector components, then we may remove the corresponding eigenvalue from A and reduce the system $Ax = b$ to an equivalent one, which has a lower dimension. The product hybrid GMRES algorithm has the first advantage over the original one in that its convergence behavior is well understood, as stated in the following theorem.

Theorem 1. *The product hybrid GMRES algorithm converges if GMRES(m) converges.*

Proof. If GMRES(m) converges, it holds that

$$\lim_{s\to+\infty} \|\pi_s(A)r_0\| = 0.$$

Then with a suitable s, we must have $\|\pi_s(A)\| < 1$, which leads to convergence of the Richardson iteration in Phase II.

Since we may take $sm \geq n$, it can be easily proved that the product variants of other hybrid algorithms also have the same convergence bound as GMRES(m). This makes a significant breakthrough in the application of hybrid iteration as a technique, see [1].

The remainder of this paper is devoted to show that the product hybrid GMRES algorithm (essentially, the product hybrid scheme) is mathematically feasible, practical, and computationally superior to its original one. The subject is related to the asymptotic convergence rate of the Richardson iteration, in terms of the quantity

$$\upsilon = \max_{\lambda\in\sigma(A)} \{|\pi_s(\lambda)|\},$$

where $\sigma(A) = \{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ is the spectrum of A. Note that a large υ will hinder or preclude the convergence of the Richardson iteration, even though the modulus of $\pi_s(z)$ is very small on all the other eigenvalues of A.

In the next section we show theoretically that if GMRES(m) converges steadily, then $\pi_s(z)$ can seek a kind of equilibrium in residual reduction among all the eigenvector components, in some sense that

$$|\pi_s(\lambda_1)| \approx |\pi_s(\lambda_2)| \approx \cdots \approx |\pi_s(\lambda_n)|. \tag{5}$$

Such an equilibrium is of great significance: the Richardson iteration with this product polynomial will then yield a convergence rate, in an asymptotic sense, as good as that of the previous GMRES(m) iteration, resulting in greatly decreased work requirements. The theoretical results are numerically illustrated in Section 5.

3. A Polynomial Characterization of GMRES(m)

Harmonic Ritz values are employed to study GMRES(m). Let $V_m = [v_1, v_2, \cdots, v_m]$ be an orthonomal basis of $K_m(r_0, A)$, where $K_m(r_0, A) = span\{A^i r_0\}_{i=0}^{m-1}$ is the Krylov subspace.

Define $G_m = (V_m^T A^T V_m)^{-1} V_m^T A^T A V_m$. The harmonic Ritz values of A are the eigenvalues of G_m [6]. The following lemma reveals a fundamental relation between GMRES and the harmonic Ritz values, see [2, 3].

Lemma 2. *Denote by $\{\lambda_i^{(m)}\}_{i=1}^m$ the m harmonic Ritz values of A. The GMRES polynomial $p_m(z)$ satisfies*

$$p_m(z) = \frac{(\lambda_1^{(m)} - z)(\lambda_2^{(m)} - z)\cdots(\lambda_m^{(m)} - z)}{\lambda_1^{(m)}\lambda_2^{(m)}\cdots\lambda_m^{(m)}}.$$

Here it is assumed that GMRES(m) converges, so that $\lambda_i^{(m)} \neq \infty (i = 1, 2, \cdots, m)$ [17]. Otherwise we have $p_m(z) \equiv 1$. Assume for simplicity that the eigenvalues of A and G_m are all simple. Denote by P_m the orthogonal projector onto $K_m(r_0, A)$, and Q_m the oblique projector onto $K_m(r_0, A)$ and orthogonal to $AK_m(r_0, A)$. The following lemma is established as a generalization of Theorem 3.7 of [4], see [16].

Lemma 3. *Let $S_m^{-1} G_m S_m = diag(\lambda_1^{(m)}, \cdots, \lambda_m^{(m)})$ and $cond(S_m) = \|S_m\|\|S_m^{-1}\|$. Denote by $\varphi_i (i = 1, 2, \cdots, m)$ the ith normalized eigenvector of A. There exists a harmonic Ritz value $\lambda_i^{(m)}$ such that*

$$|\lambda_i^{(m)} - \lambda_i| \leq 2\gamma_{m,i} cond(S_m) \frac{\|(I - P_m)\varphi_i\|}{\|P_m \varphi_i\|},$$

in which $\gamma_{m,i} = \|Q_m(A - \lambda_i I)(I - P_m)\|$.

Write the initial vector r_0 as $r_0 = \sum_{j=1}^n \alpha_j \varphi_j$. The following result is established in [8].

Lemma 4. *Assume that $\alpha_i \neq 0$ and let $\xi_i = \sum_{j=1, j\neq i}^n |\alpha_j|/|\alpha_i|$. Then*

$$\|(I - P_m)\varphi_i\| \leq \xi_i \min_{\deg p \leq m-1; p(\lambda_i)=1} \max_{j=1,2,\cdots,n; j\neq i} |p(\lambda_j)| \stackrel{def}{=} \xi_i \varepsilon_i^{(m)}.$$

With the previous preliminaries, we now give the main result below, which is an explicit polynomial characterization of GMRES(m).

Theorem 5. *Assuming GMRES(m) starts with $r_0 = \sum_{j=1}^n \alpha_j \varphi_j$ and $\alpha_i \neq 0$, with a proper order of $\{\lambda_i\}_{i=1}^n$ we will have*

$$|\pi_s(\lambda_i)| \leq F_{m,i} \frac{\sum_{j=1, j\neq i}^n |\alpha_j \pi_{s-1}(\lambda_j)|}{|\alpha_i|} \tag{6}$$

with

$$F_{m,i} = 2\frac{\prod_{j=1, j\neq i}^m |\lambda_j^{(m)} - \lambda_i|}{\prod_{j=1}^m |\lambda_j^{(m)}|} \gamma_{m,i} cond(S_m) \frac{\varepsilon_i^{(m)}}{\|P_m \varphi_i\|},$$

where the scalars with the subscript or superscript (m), except $\varepsilon_i^{(m)}$, are all associated to the sth restarting cycle of GMRES(m).

Proof. It follows from $r_{(s-1)m} = \pi_{s-1}(A) r_0$ that

$$r_{(s-1)m} = \sum_{j=1}^n \alpha_j \pi_{s-1}(\lambda_j)\varphi_j.$$

Then

$$\begin{aligned}
|\pi_s(\lambda_i)| &= |p_{m,s}(\lambda_i)\pi_{s-1}(\lambda_i)| \\
&\stackrel{Lemma\ 2}{=} \frac{\prod_{j=1, j\neq i}^m |\lambda_j^{(m)} - \lambda_i|}{\prod_{j=1}^m |\lambda_j^{(m)}|} |\lambda_i^{(m)} - \lambda_i| |\pi_{s-1}(\lambda_i)| \\
&\stackrel{Lemma\ 3}{\leq} 2\frac{\prod_{j=1, j\neq i}^m |\lambda_j^{(m)} - \lambda_i|}{\prod_{j=1}^m |\lambda_j^{(m)}|} \gamma_{m,i} cond(S_m) \frac{\|(I - P_m)\varphi_i\|}{\|P_m \varphi_i\|} |\pi_{s-1}(\lambda_i)|
\end{aligned}$$

$$\begin{aligned} &\overset{Lemma\ 4}{\leq} && 2\frac{\prod_{j=1,j\neq i}^{m}|\lambda_j^{(m)}-\lambda_i|}{\prod_{j=1}^{m}|\lambda_j^{(m)}|}\gamma_{m,i}cond(S_m) \\ & && \cdot\frac{\varepsilon_i^{(m)}}{\|P_m\varphi_i\|}\frac{\sum_{j=1,j\neq i}^{n}|\alpha_j\pi_{s-1}(\lambda_j)|}{|\alpha_i\pi_{s-1}(\lambda_i)|}|\pi_{s-1}(\lambda_i)| \\ &= && F_{m,i}\frac{\sum_{j=1,j\neq i}^{n}|\alpha_j\pi_{s-1}(\lambda_j)|}{|\alpha_i|}. \end{aligned}$$

The superiority of the product hybrid GMRES algorithm is interpreted by the following two remarks of Theorem 5.

Remark 1. When $s = 1$, we have

$$|p_{m,1}(\lambda_i)| \leq F_{m,i}\frac{\sum_{j=1,j\neq i}^{n}|\alpha_j|}{|\alpha_i|}.$$

If r_0 is nearly deficient in the ith eigenvector component, i.e., $|\alpha_i| \ll |\alpha_j|$ for $j \neq i$, then $|p_{m,1}(\lambda_i)|$ is likely to be considerably large, which is fatal to the Richardson iteration. On the other hand, since r_m becomes rich in this component, $|p_{m,2}(\lambda_i)|$ will be correspondingly small in the next GMRES cycle. Then $p_{m,2}(z)$ can act as an equilibrium to $p_{m,1}(z)$, and a product of the two polynomials will improve the performance of the Richardson iteration significantly.

Remark 2. If r_0 is (nearly) equally rich in all the eigenvector components, i.e., $|\alpha_1| = |\alpha_2| = \cdots = |\alpha_n|$, then it is obtained from (6) that

$$|\pi_s(\lambda_i)| \leq F_{m,i}\sum_{j=1,j\neq i}^{n}|\pi_{s-1}(\lambda_j)| \ (i = 1, 2, \cdots, n),$$

which implies that the product polynomial $\pi_s(z)$ will be simultaneously reduced on the spectrum of A. On the other hand, it is well observed that a single polynomial $p_m(z)$ always has little reduction on extremely small eigenvalues. This comparison motivates the product hybrid GMRES algorithm.

4. Implementation Issues of the Product Hybrid GMRES Algorithm

The crucial element for successful application of the product hybrid GMRES algorithm revolves around the choices of m and s. The same strategy for choosing m for the hybrid GMRES algorithm is suggested here:

$$Goal: \ equal\ amounts\ of\ work\ in\ Phase\ I\ and\ Phase\ II, \tag{7}$$

where the work is measured by vector operations and a vector operation is defined to be the cost of an "axpy" operation $ax + y$ involving a scalar a and n-vector x and y.

The condition (7) has some solid justifications. In particular, we have the following theorem, which is an immediate generalization of Theorem 1 of [7].

Theorem 6. *Assume that the GMRES(m) iteration of Phase I converges steadily, and the Richardson iteration of Phase II converges at exactly the same rate as in Phase I, i.e.,*

$$\frac{\|r_{ksm}\|}{\|r_0\|} = \left(\frac{\|r_{sm}\|}{\|r_0\|}\right)^k \quad for\ all\ k \geq 0. \tag{8}$$

Let m be determined by the condition (7). Then the product hybrid GMRES algorithm converges, and no other choice of m could have reduced the computing time by more than a factor of two.

Proof. See [7].

So long as the assumptions made in Theorem 6 hold, this strategy for choosing m is actually optimal with respect to work requirement in the sense that any other choice might lead to a penalty of a factor greater than two, see [7].

Now we work out the algebra requirement to implementation (7). Suppose in Phase I the residual has been reduced by the factor of $\|r_{sm}\|/\|r_0\| = \tau^s$ and our desired accuracy is

$\|r_{final}\|/\|r_0\| = \varepsilon$. Assume that one matrix-vector multiplication costs δ vector operations for some $\delta \geq 0$. According to estimates in [10], the work performed so far is

$$Phase\ I\ work:\ \ sm(m+3+\delta)\ vector\ operations.$$

In the Richardson iteration of Phase II the work per step will be $1+\delta$ vector operations. By (8), the total number of steps to convergence will be $m\log\varepsilon/\log\tau$, hence in Phase II, $sm[\log\varepsilon/(s\log\tau)-1]$. This implies that

$$Phase\ II\ work:\ \ sm(1+\delta)[\log\varepsilon/(s\log\tau)-1]\ vector\ operations.$$

The condition (7) can be realized by equating the work of Phase I and Phase II:

$$m+3+\delta = (1+\delta)[\log\varepsilon/(s\log\tau)-1]. \tag{9}$$

To summarize, here is how we decide when to restart GMRES. During the first GMRES cycle the left-hand side of (9) increases monotonically and the right-hand side decreases monotonically (because τ is decreasing). We restart GMRES as soon as the left-hand side exceeds the right-hand side.

Note that the optimal choice of m for the product hybrid algorithm depends on the choice of s. Then we need to choose s beforehand. This is a mater of experience. It is well observed that if GMRES(m) converges steadily, in most cases only a couple of GMRES cycles are required in order for an effective product polynomial. Generally, we take $s = 2$. The effect of such a decision is numerically illustrated in the next Section.

The choices of m and s presented above are based on some idealized assumptions which do not always hold in practice. For ill-conditioned problems both m and s should be larger. To a certain extent users of the product hybrid GMRES algorithm will inevitably have to make some decisions themselves, just as in using GMRES(m) or some other iterative methods. More fundamentally, since there are no definite rules governing such choices, we suggest an adaptive procedure be developed to make decisions automatically. The mechanism of the procedure is that if the residual decrease of the Phase II iteration is not adequate, a decision should be made to return to Phase I and restart a new GMRES cycle from the current best available solution, maybe with an increased m.

The structure of the practical variant is shown below. The details of how to compute the GMRES residual polynomials and how to apply them via the Richardson iteration can be seen in [7]. For a more stable way to implement GMRES(m) and construct the residual polynomials, see [15].

(1) Initialize. Take $s = 2$ and let m be decided by (9). Set L=TURE, which is used to decide whether GMRES should be restarted with the same m or an increased m.

(2) Run GMRES(m) within two cycles and compute the GMRES polynomials.

(3) Reapply the computed product polynomial via the Richardson iteration and perform stopping test.

(4) If the convergence proves unsatisfactory and L=TURE, restart GMRES another couple of cycles from the current best available solution with the same m. Set L=FALSE and GOTO (3).

(5) If the convergence proves unsatisfactory and L=FALSE, restart GMRES another couple of cycles from the current best available solution with an increased m. Set L=TURE and GOTO (3).

λ_i	-10	-1	-0.1	0.1	1	10
$\lvert p_{4,1}(\lambda_i)\rvert$	9.898E-7	9.997E-3	0.9898	0.9898	9.997E-3	9.898E-7
$\lvert p_{4,2}(\lambda_i)\rvert$	3.300E+5	32.67	0.3300	0.3300	32.67	3.300E+5
$\lvert \pi_2(\lambda_i)\rvert$	0.3266	0.3266	0.3266	0.3266	0.3266	0.3266

Table 1: Example 1

5. Numerical Experiments

The product hybrid GMRES algorithm is tested and compared with other iterative methods. The first experiment is designed to illustrate the theoretical results of Sections 3. The other experiments are designed to compare the product hybrid GMRES algorithm (PH-GMRES(s, m)) with the hybrid GMRES algorithm (H-GMRES(m)), GMRES(m), and for a special choice, GMRESR(m) (using GMRES(m) as the inner iteration method) [14], which is known as another hybrid variant of GMRES. In each experiment the right-hand side is chosen as $b = (1, 1, \cdots, 1)^T$ and the initial guess x_0 is taken to be zero. Except the first example, the convergence tolerance is $\varepsilon = 10^{-10}$ and for each example we present a plot shows $\log_{10} \|r_{sm}\|$ as a function of work measured by vector operations, defined in Section 4.

Example 1. Take $A = diag(-10, -1, -0.1, 0.1, 1, 10)$ and run GMRES(4) within two cycles. In the first cycle $\{\lambda_i^{(4)}\}_{i=1}^4 = \{-9.999, -0.995, 0.995, 9.999\}$, which are far form the small eigenvalues $\{-0.1, 0.1\}$ but well close to the large eigenvalues $\{-10, -1, 1, 10\}$. Correspondingly, $p_{4,1}(z)$ has little reduction on the eigenvalues $\{-0.1, 0.1\}$(see Table 1), which will yield very slow convergence in the Richardson iteration if used. In the second GMRES cycle, since r_4 becomes very rich in the eigenvector components of $\{-0.1, 0.1\}$ but nearly deficient in the direction of $\{-10, -1, 1, 10\}$, we get $\{\lambda_i^{(4)}\}_{i=1}^4 = \{-1.4089, -0.1223, 0.1223, 1.4089\}$, which do a good job in locating the small eigenvalues $\{-0.1, 0.1\}$. Correspondingly, $p_{4,2}(z)$ is significantly reduced on $\{-0.1, 0.1\}$ but becomes considerable large on $\{-10, -1, 1, 10\}$(see Remark 1 of Theorem 5). As a result, the product polynomial $\pi_2(z)$ is (nearly) equally reduced in all the eigenvector components, which takes advantage of an equilibrium between $p_{4,1}(z)$ and $p_{4,2}(z)$(see Remark 2 of Theorem 5). Using $\pi_2(z)$ can then yield a convergence rate as good as GMRES(4), however, at significantly less cost.

Note that the same observation was made in solving the problem (4) of Section 1: PH-GMRES(2,1) can converge rapidly at the same rate as GMRES(1).

Example 2. This problem is taken from [13], for which GMRES is superlinear convergent [13, 18]. The matrix is of the form $A = SBS^{-1}$ with $S, B \in R^{1000\times 1000}$ selected to be

$$S = \begin{pmatrix} 1 & 0.1 & & \\ & 1 & \ddots & \\ & & \ddots & 0.1 \\ & & & 1 \end{pmatrix}; B = \begin{pmatrix} 1 & & & \\ & 2 & & \\ & & \ddots & \\ & & & 1000 \end{pmatrix}.$$

Since A has so many different eigenvalues, H-GMRES(20) goes to considerable trouble in making its iteration polynomial, to some extent, be equally small on $\sigma(A)$. Nevertheless, it beats GMRES(20) by a small factor. Meanwhile, a surprising improvement is offered by PH-GMRES(2, 20), see Figure 1. This illustrates the remarkable superiority of the product hybrid scheme.

We consider in a little more detail the performance of GMRESR(20), which is fastest among the four algorithms. As a sophisticated hybrid variant of GMRES, GMRESR(m) constructs a different polynomial preconditioner in each iteration step. Furthermore, because this method does not restart, it tends to be superlinear convergent [14]. This appealing property appears to play an important role in the rapid convergence of GMRESR(m) here.

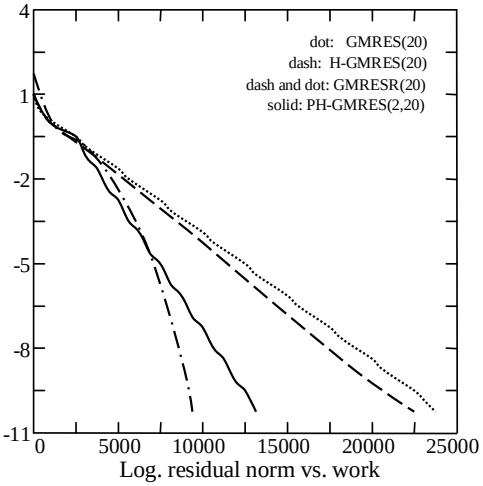

Figure 1: Example 2

Example 3. This problem is taken from [7]. Let A be a large upper-triangular Toeplitz matrix of the form

$$A = \begin{pmatrix} 1 & 1 & 0.5 & & & \\ & 1 & 1 & 0.5 & & \\ & & 1 & 1 & 0.5 & \\ & & & \ddots & \ddots & \ddots \end{pmatrix} (1000 \times 1000).$$

This matrix has just the single eigenvalue $\{1\}$. However, for practical purposes it behaves much more nearly as if its spectrum were $\sigma(A)_{practical} = f(D)$, where D is the closed unit disk and $f(z) = 1 + z + 0.5z^2$ is the symbol of this Toeplitz matrix.

We see in Figure 2 that H-GMRES(5) performs ideally well, but PH-GMRES(5) is still further ahead. The observation indicates that the product hybrid scheme can also improve convergence for matrices whose effective spectrum are quite different from their exact spectrum. This time GMRESR(5) lags behind GMRES(5) due to the additional cost of its inner iteration process.

Example 4. Finally, we consider a realistic test problem taken from the Harwell-Boeing collection. The matrix (GRE115) was produced from runs of the package QNAP written by CII-HB for simulation modeling of computer systems and used as a test bed for ordering codes.

It is observed that H-GMRES(m) diverges for all possible choices of m, whereas PH-GMRES(s,20) converges rapidly when $s \geq 5$. The behavior of H-GMRES(33) and PH-GMRES (5,20) is shown in Figure 3. GMRESR(20) is superlinear convergent and does much better. However, by choosing another set of the parameters m and s, we see PH-GMRES(2,33) is the fastest.

6. Concluding Remarks

We have given some insights into the convergence characteristics of GMRES(m), and as a natural outgrowth, defined an effective hybrid iterative scheme which appears quite promising for solving large nonsymmetric linear systems of equations.

The most general hybrid algorithms are based on Arnoldi eigenvalue estimates. Since both the Arnoldi and GMRES iterations make use of a Hessenberg matrix obtained by the orthogonalization of a sequence of Krylov vectors, these computations are usually carried out simultaneously. However, the Arnoldi eigenvalue estimates are likely to be far from small eigen-

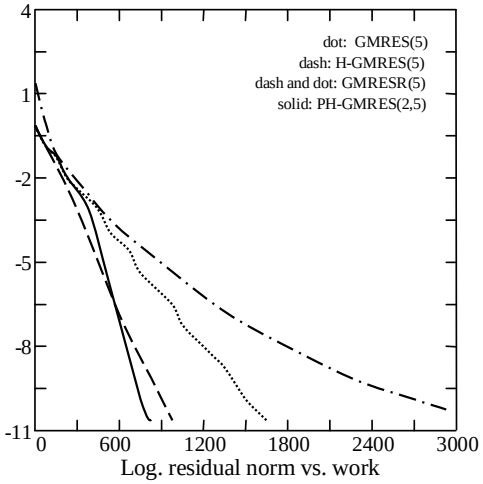

Figure 2: Example 3

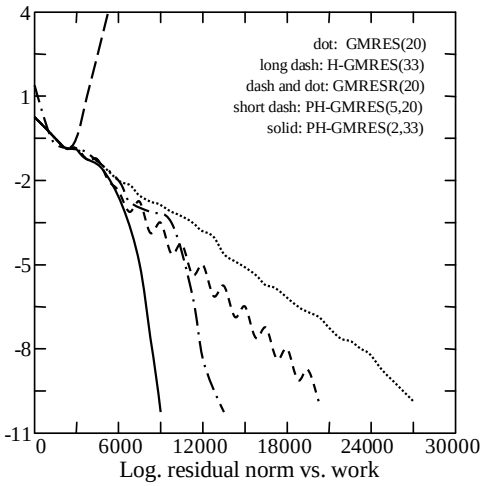

Figure 3: Example 4

values for the first "Arnoldi/GMRES" cycle. Consequently, the resulted polynomial may be considerably large on the small eigenvalues. With this consideration, we suggest hybrid algorithms of the "Arnoldi/GMRES" type also be implemented with the product hybrid scheme. According to our theoretical results presented above, polynomials resulted from the following "Arnoldi/GMRES" cycles will contain enough information on the location of the small eigenvalues, so that a product of these polynomials will also be potentially advantageous.

Acknowledgments. The author wishes to thank Professor Dai Hua for many helpful comments on this subject. The author is also indebt to Professor H. A. Van der Vorst for reading an earlier version of this paper and suggesting the comparison between PH-GMRES and GMRESR.

References

[1] V. Faber, W. Joubert, E. Knill and T. Manteuffel, Minimal residual method stronger than poly-

nomial preconditioning, *SIAM J. Matrix Anal. Appl.*, **17** (1996), 707-729.

[2] R.W. Freund, Quasi-kernel polynomials and their use in non-Hermitian matrix iterations, *J. Comput. Appl. Math.*, **43** (1992), 135-158.

[3] S. Goossens, D. Roose, Ritz and harmonic Ritz values and the convergence of FOM and GMRES, *Numer. Linear Algebra Appl.*, **6** (1999), 281-293.

[4] Z.X. Jia, The convergence of generalized Lanczos methods for large unsymmetric eigenproblems, *SIAM J. Matrix Anal. Appl.*, **16** (1995), 843-862.

[5] W. Joubert, A robust GMRES-based adaptive polynomial preconditioning algorithm for nonsymmetric linear systems, *SIAM J. Sci. Comput.*, **15** (1994), 427-439.

[6] R.B. Morgan, Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations, *SIAM J. Matrix Anal. Appl.*, **21** (2000), 1112-1135.

[7] N.M. Nachtigal, L. Reichel, L.N. Trefethen, A hybrid GMRES algorithm for nonsymmetric linear systems, *SIAM J. Matrix Anal. Appl.*, **13** (1992), 796-825.

[8] Y. Saad, Variations on Arnoldi's method for computing eigenelements of large nonsymmetric matrices, *L. A. A.*, **34** (1980), 269-295.

[9] Y. Saad, Projection methods for solving large sparse eigenvalues problems, Lecture Notes in Math. 973 (Springer, Berlin, 1983)

[10] Y. Saad, M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.*, **7** (1986), 856-869.

[11] Y. Saad, H.A. Van der Vorst, Iterative solution of linear systems in the 20th century, *J. Comput. Appl. Math.*, **123** (2000), 1-33.

[12] K.C. Toh, GMRES vs. ideal GMRES, *SIAM J. Matrix Anal. Appl.*, **18** (1997), 30-36.

[13] H.A. Van der Vorst, C. Vuik, The superlinear convergence behavior of GMRES, *J. Comput. Appl. Math.*, **48** (1993), 327-341.

[14] H.A. Van der Vorst, C. Vuik, GMRESR: A family of nested GMRES methods, *Numer. Linear Algebra Appl.*, **1** (1994), 369-386.

[15] B.J. Zhong, Implementation of the hybrid GMRES algorithm using Householder transformations, *Transactions of NUAA*, **14** (1997), 146-152.

[16] B.J. Zhong, A flexible hybrid GMRES algorithm (in Chinese), *Numer. Math. -A J. of Chinese Universities*, **23** (2001), 261-272.

[17] B.J. Zhong, On the breakdowns of the Galerkin and least-squares methods, *Numer. Math. -A J. of Chinese Universities*, **11** (2002), 137-148.

[18] B.J. Zhong, The convergence rate of GMRES (in Chinese), *Numer. Math. -A Journal of Chinese Universities*, **25** (2003), 253-260.