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:

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

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

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

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 $|p_1(c\lambda)| \leq 1$ is imposed, we have $-2 \leq \alpha c\lambda \leq 0$, which gives that $|p_1(\lambda)| \geq 1 - 2/c$. 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: