Journal of Computational Mathematics
Vol.40, No.5, 2022, 715–731.
http://www.global-sci.org/jcm
doi:10.4208/jcm.2101-m2019-0243

HEAVY BALL FLEXIBLE GMRES METHOD FOR NONSYMMETRIC LINEAR SYSTEMS*

Mei Yang[1)]
School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai 200240, China
Email: ymsjtu@sjtu.edu.cn
Ren-Cang Li
Department of Mathematics, University of Texas at Arlington, Arlington, TX, USA
Email: rcli@uta.edu

Abstract

Flexible GMRES (FGMRES) is a variant of preconditioned GMRES, which changes preconditioners at every Arnoldi step. GMRES often has to be restarted in order to save storage and reduce orthogonalization cost in the Arnoldi process. Like restarted GMRES, FGMRES may also have to be restarted for the same reason. A major disadvantage of restarting is the loss of convergence speed. In this paper, we present a heavy ball flexible GMRES method, aiming to recoup some of the loss in convergence speed in the restarted flexible GMRES while keep the benefit of limiting memory usage and controlling orthogonalization cost. Numerical tests often demonstrate superior performance of the proposed heavy ball FGMRES to the restarted FGMRES.

Mathematics subject classification: 65F10.
Key words: GMRES, Flexible GMRES, Heavy ball GMRES, Preconditioner, Linear system.

1. Introduction

The Generalized Minimal Residual (GMRES) method [12] is a well-established Krylov subspace method for solving a large and sparse nonsymmetric linear system of equations

$$Ax = b, \tag{1.1}$$

where $A \in \mathbb{C}^{n\times n}, b \in \mathbb{C}^n$, and $x \in \mathbb{C}^n$ is the unknown. Given an initial approximation x_0, the k-th GMRES approximation x_k is sought so that the k-th residual $r_k = b - Ax_k$ satisfies

$$\|r_k\|_2 = \min_{z\in\mathcal{K}_k(A,r_0)} \|b - A(x_0 + z)\|_2, \tag{1.2}$$

where $r_0 = b - Ax_0$, $\|\cdot\|_2$ is the Euclidian norm, and $\mathcal{K}_k(A, r_0)$ is the k-th Krylov subspace of A on r_0 defined by

$$\mathcal{K}_k(A, r_0) = \operatorname{span}\left\{r_0, Ar_0, A^2r_0, \cdots, A^{k-1}r_0\right\}. \tag{1.3}$$

Algorithmically, GMRES first builds an orthonormal basis of $\mathcal{K}_k(A, r_0)$ via the Arnoldi process and, along the way, A is projected onto the Krylov subspace to turn (1.2) into a much smaller $(k + 1) \times k$ least squares problem.

* Received October 30, 2019 / Revised version received June 12, 2020 / Accepted January 19, 2021 / Published online March 8, 2021 /
1) Corresponding author

Ideally, it is hoped that for a modest k relative to n, $\|r_k\|_2$ is sufficient tiny so that x_k can be regarded as a sufficiently accurate approximation to the exact solution of (1.1). But that is not always the case. When that happens, GMRES can become very expensive because of the heavy burden in memory for storing orthonormal basis vectors and for generating them in the Arnoldi process. A popular and the simplest remedy is the so-called restarted GMRES (REGMRES) [6] which sets an upper bound $k_{\max}$ on k and starts over as soon as k reaches the upper bound $k_{\max}$ but $\|r_{k_{\max}}\|_2$ is not yet tiny enough, using the latest approximation $x_{k_{\max}}$ as the initial guess for the next GMRES run. Doing so effectively put the memory requirement and orthogonalization cost under control, but not without tradeoff, which is slow convergence and potentially increases overall computational time in solving (1.1). In recognizing this tradeoff, researchers have made efforts address the issue.

The loss in convergence speed by REGMRES is due to its control on the largest possible number of Arnoldi steps that one GMRES run can use. Besides the use of the latest approximation as the initial guess for the next GMRES run, REGMRES completely throws away the Krylov subspaces built thus far. To partially compensate the throw-away, two main types of improvements are discussed, which include hybrid iterative methods and acceleration techniques. Our discussion mainly focuses on acceleration techniques. One natural idea to accelerate GMRES is to augment the Krylov subspace of REGMRES according to spectral information at the restart, named augmented Krylov subspace techniques, such as GMRES-E [8] , GMRES-IR [9] and GMRES-DR [10]. This kind of methods keeps the form of Krylov subspace. Another technique is to approximate the search space with non-Krylov subspace, i.e., approximation space. In [1], Baker, Jessup and Manteuffel presented the Loose GMRES (LGMRES) method. At the ℓ-th restart of LGMRES, the Krylov subspace $\mathcal{K}_k(A, r_0^{(\ell)})$ is generated and augmented with the t most recent error vectors, which are defined to be the differences between every two sequential solutions.

In [7], from the optimization perspective, Imakura, Li and Zhang proposed two comparable methods, the locally optimal GMRES (LOGMRES) and the heavy ball GMRES methods (H-BGMRES). LOGMRES augments the search space by adding the most previous solution vector. The latter one incorporates the idea in the heavy ball method from optimization [11, p.65] into REGMRES by adding a new vector – the difference of approximations from the previous two cycles of HBGMRES, i.e., the most previous error vector in LGMRES, to the next searching space. Numerical experiments show that HBGMRES often converges significantly faster than REGRMES. However, there are cases where the gain of HBGMRES over REGMRES is not so significant.

The preconditioning technique is often very effective in enhancing the performance and reliability of Krylov subspace methods, provided a reasonably good preconditioner can be found. Instead of (1.1), its right preconditioned linear system takes the form

$$(AM^{-1})y = b, \quad Mx = y. \tag{1.4}$$

The matrix M is called a preconditioner and it may exist in form in such a way that linear system $Mz = c$ is cheap to solve. However, it is usually hard to find a suitable preconditioner M for the problem at hand. Saad [13] proposed an inner-outer iteration method called the flexible GMRES method (FGMRES), in which GMRES is used as the outer iteration. In the inner iteration some linear system $Az = c$ is approximately solved and thus each inner iteration can be viewed as applying some preconditioner M, not known explicitly but implicitly in the action $M^{-1}c \approx A^{-1}c$. GMRESR [15], another similar inner-outer iteration, uses GCR [3] instead