East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.280611.030911a

Vol. **2**, No. 1, pp. 19-32
February 2012

An Efficient Variant of the GMRES(m) Method Based on the Error Equations

Akira Imakura[1,*], Tomohiro Sogabe[2] and Shao-Liang Zhang[1]

[1] *Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan.*
[2] *Aichi Prefectural University, 1522-3 Ibaragabasama, Kumabari, Nagakute-cho, Aichi-gun, Aichi, 480-1198, Japan.*

Received 28 June 2011; Accepted (in revised version) 3 September 2011

Available online 10 February 2012

Abstract. The GMRES(m) method proposed by Saad and Schultz is one of the most successful Krylov subspace methods for solving nonsymmetric linear systems. In this paper, we investigate how to update the initial guess to make it converge faster, and in particular propose an efficient variant of the method that exploits an *unfixed update*. The mathematical background of the unfixed update variant is based on the error equations, and its potential for efficient convergence is explored in some numerical experiments.

AMS subject classifications: 65F10

Key words: Nonsymmetric linear systems, GMRES(m) method, restart, error equations.

1. Introduction

In recent years, there has been extensive research on Krylov subspace methods for solving large and sparse linear systems of the form

$$A\boldsymbol{x} = \boldsymbol{b}, \quad A \in \mathbb{R}^{n\times n}, \quad \boldsymbol{x}, \boldsymbol{b} \in \mathbb{R}^n , \tag{1.1}$$

where the coefficient matrix A is assumed to be nonsymmetric and nonsingular. These linear systems often arise from the discretization of partial differential equations in computational science and engineering. The CG method [10] and the MINRES method [16] are two well known Krylov subspace methods for solving symmetric linear systems, but for general nonsymmetric linear systems the GMRES method [17] and the Bi-CGSTAB method [25] (and its variants [9, 20, 28]) are the most widely used. The IDR(s) method [22] proposed by Sonneveld and van Gijzen has recently also attracted considerable attention [21, 24]. Further details may be found in several surveys [8, 18, 19].

*Corresponding author. *Email addresses:* `a-imakura@na.cse.nagoya-u.ac.jp` (A. Imakura), `sogabe@ist.aichi-pu.ac.jp` (T. Sogabe), `zhang@na.cse.nagoya-u.ac.jp` (S.-L. Zhang)

In this paper, we focus on further developments in the GMRES method context. Let us first note that the original GMRES method [17] has shown good convergence, but it has considerable computational cost and storage requirements due to long-term recurrence. This is avoided in the GMRES(m) method [17] now widely used in practice, which involves a so-called *restart* that can be described as follows. In the first restart cycle, the restart frequency m is chosen and an initial guess $\boldsymbol{x}_0^{(1)}$ is made. For any lth restart cycle, the GMRES method using the initial guess $\boldsymbol{x}_0^{(l)}$ is applied with m iterations to the linear system (1.1), to produce the approximate solution $\boldsymbol{x}_m^{(l)}$ that may then be used to update the initial guess (i.e. $\boldsymbol{x}_0^{(l+1)} := \boldsymbol{x}_m^{(l)}$) in the $(l+1)$th restart cycle. This process is repeated until there is satisfactory convergence.

However, although the restart procedure avoids the computational cost and storage drawbacks of the GMRES method, it usually slows down the convergence. To improve the convergence of the GMRES(m) method, several techniques have recently been proposed for the "inside part" of m iterations [1–5, 7, 12–15, 23, 27]. When the initial guess of each restart cycle is updated via $\boldsymbol{x}_0^{(l+1)} := \boldsymbol{x}_m^{(l)}$ as outlined above, we call this the *fixed update* procedure. To further improve the convergence, in this paper we propose *variants of the GMRES(m) method* with *unfixed update*, mathematically based on the error equations and iterative refinement scheme.

The paper is organized as follows. In Section 2, we briefly discuss the GMRES(m) method. The proposed GMRES(m) method with unfixed update, and its mathematical background based on the error equations and iterative refinement scheme, is then presented in Section 3. An example variant of the GMRES(m) method with unfixed update is considered in Section 4, and in particular its convergence is explored in some numerical experiments in Section 5. Our conclusions are summarized in Section 6.

2. The GMRES(m) Method

Let $\boldsymbol{x}_0$ denote an initial guess for the solution of system (1.1), and $\boldsymbol{r}_0 := \boldsymbol{b} - A\boldsymbol{x}_0$ the corresponding initial residual. The Krylov subspace methods form a family of projection methods that extract an approximate solution $\boldsymbol{x}_k$ from an affine space spanned by the initial guess $\boldsymbol{x}_0$ and the Krylov subspace $\mathscr{K}_k(A, \boldsymbol{r}_0) \equiv \mathrm{span}\{\boldsymbol{r}_0, A\boldsymbol{r}_0, \cdots, A^{k-1}\boldsymbol{r}_0\}$ such that

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{z}_k, \qquad \boldsymbol{z}_k \in \mathscr{K}_k(A, \boldsymbol{r}_0).$$

The GMRES method [17] constructs the approximate solution $\boldsymbol{x}_k$ by the minimum residual condition as follows:

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + V_k \boldsymbol{s}_k, \qquad \boldsymbol{s}_k = \arg\min_{\boldsymbol{s}\in\mathbb{R}^k} \|\boldsymbol{r}_0 - AV_k\boldsymbol{s}\|_2, \tag{2.1}$$

where V_k is the $n \times k$ matrix with columns the orthogonal basis of the Krylov subspace $\mathscr{K}_k(A, \boldsymbol{r}_0)$ often obtained by the Arnoldi procedure. Thus in the GMRES algorithm, the minimization problem (2.1) is transformed by using the matrix formula of the Arnoldi procedure, and solved by QR factorization based on the Givens rotation. However, the computational cost of the GMRES method grows by at least $\mathrm{O}(k^2 n)$ and storage requirements by O(kn) as the number of iterations k increases, under a long-term recurrence based on the Arnoldi procedure. As we discussed above, the restarted version of the GMRES method known as the GMRES(m) method [17] has therefore been used. The GMRES(m) algorithm is as follows.

Algorithm 2.1 The GMRES(m) method

1: Choose the restart frequency m and the initial guess $\boldsymbol{x}_0$
2: Compute $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$ and set $\beta = \|\boldsymbol{r}_0\|_2, \boldsymbol{v}_1 = \boldsymbol{r}_0/\beta$
3: **for** $j = 1, 2, \cdots, m$, **do**
4: Compute $\boldsymbol{w}_j = A\boldsymbol{v}_j$
5: **for** $i = 1, 2, \cdots, j$, **do**
6: $h_{i,j} = (\boldsymbol{w}_j, \boldsymbol{v}_i)$
7: $\boldsymbol{w}_j = \boldsymbol{w}_j - h_{i,j}\boldsymbol{v}_i$
8: **end for**
9: $h_{j+1,j} = \|\boldsymbol{w}_j\|_2$
10: $\boldsymbol{v}_{j+1} = \boldsymbol{w}_j / h_{j+1,j}$
11: **end for**
12: Define the $(m+1) \times m$ Hessenberg matrix $\widetilde{H}_m = \{h_{i,j}\}_{1 \le i \le m+1, 1 \le j \le m}$
13: $\boldsymbol{x}_m = \boldsymbol{x}_0 + V_m \boldsymbol{s}_m$, where $\boldsymbol{s}_m = \arg\min_{\boldsymbol{s} \in \mathbb{R}^m} \|\beta \boldsymbol{e}_1 - \widetilde{H}_m \boldsymbol{s}\|_2$
14: Compute $\boldsymbol{r}_m = \boldsymbol{b} - A\boldsymbol{x}_m$. **If** convergence **then** Stop
15: Update $\boldsymbol{x}_0 := \boldsymbol{x}_m$, and go to 2

3. Updating the Initial Guess

In this section, we reconsider the update of the initial guess in the GMRES(m) method, with focus on the restart in Section 3.1 and the unfixed update in Section 3.2.

3.1. The restart

As noted in Section 1, the restart of the GMRES(m) method can be decomposed into the three main parts — viz.

Part 1. Choose the restart frequency m and the initial guess $\boldsymbol{x}_0^{(1)}$ in the 1st restart cycle.

Part 2. Solve $A\boldsymbol{x} = \boldsymbol{b}$ by m iterations of the GMRES method with the initial guess $\boldsymbol{x}_0^{(l)}$, to obtain the approximate solution $\boldsymbol{x}_m^{(l)}$.

Part 3. Update the initial guess of the next restart cycle — i.e. adopt $\boldsymbol{x}_0^{(l+1)} := \boldsymbol{x}_m^{(l)}$ (before repeating the iterative procedure).

Then, based on these three main parts, Algorithm 2.1 can be simplified as shown in Algorithm 3.1 below.

As previously mentioned, the restart avoids the long-term recurrence of the Arnoldi procedure but usually slows down the convergence, so several improvement techniques

Algorithm 3.1 The GMRES(m) method as a simplified description

1: Choose the restart frequency m and the initial guess $\boldsymbol{x}_0^{(1)}$
2: **for** $l = 1, 2, \cdots$, until convergence **do**
3: Solve (approximately) $A\boldsymbol{x} = \boldsymbol{b}$ by m iterations of the GMRES method with the initial guess $\boldsymbol{x}_0^{(l)}$, and get the approximate solution $\boldsymbol{x}_m^{(l)}$
4: Update the initial guess $\underline{\boldsymbol{x}_0^{(l+1)} := \boldsymbol{x}_m^{(l)}}$
5: **end for**

for the GMRES(m) method have been proposed. These improvements can be classified according to techniques for any one or two of the three parts. Thus the adaptive preconditioning techniques based on deflation [1, 5, 7] and those based on the augmented Krylov subspace [2, 4, 12–14] are improvements for Part 2 (Algorithm 3.1, Step 3). Techniques based on adaptively determining the restart frequency m [3, 23, 27] are improvements for Part 1 (Algorithm 3.1, Step 1); and a deflation technique with adaptively determined restart frequency m [15] is an improvement for both Part 1 and Part 2. On the other hand, until now Part 3 has been regarded as a connection that must remain the same — i.e. as in the original GMRES algorithm [17].

3.2. The GMRES(m) method with unfixed update

In this paper, we turn the focus onto Part 3 in Section 3.1, and investigate how to better update the initial guess to make the GMRES(m) convergence even faster.

In Algorithm 2.1, the initial guess of each restart cycle is updated such that

$$\boldsymbol{x}_0^{(l+1)} := \boldsymbol{x}_m^{(l)}. \tag{3.1}$$

This fixed update has not only been used in the traditional GMRES(m) method but also when any improvement technique for Part 1 and Part 2 has been included [1–5, 7, 12–15, 23, 27]. Instead of the fixed update (3.1), we introduce the following unfixed update:

$$\boldsymbol{x}_0^{(l+1)} := \boldsymbol{x}_m^{(l)} + \boldsymbol{y}^{(l+1)}, \tag{3.2}$$

where $\boldsymbol{y}^{(l+1)} \in \mathbb{R}^n$ is set by a certain strategy.

Our strategy will be discussed in Section 4, where we consider the GMRES(m) variant with unfixed update in detail. The algorithm of our GMRES(m) method with unfixed update is as follows:

4. A Variant of the GMRES(m) Method

In this section, we first discuss the mathematical background of the unfixed update (3.2) from analysis based on the error equations and the iterative refinement scheme. Efficient methods in the context of the GMRES(m) method with unfixed update (Algorithm

Algorithm 3.2 The GMRES(m) method with an unfixed update

Choose the restart frequency m and the initial guess $\boldsymbol{x}_0^{(1)}$
for $l = 1, 2, \cdots$, until convergence **do**
  Solve (approximately) $A\boldsymbol{x} = \boldsymbol{b}$ by m iterations of the GMRES method with the initial guess $\boldsymbol{x}_0^{(l)}$, and get the approximate solution $\boldsymbol{x}_m^{(l)}$
  Set the vector $\boldsymbol{y}^{(l+1)}$ based on a certain strategy
  Update the initial guess $\underline{\boldsymbol{x}_0^{(l+1)} := \boldsymbol{x}_m^{(l)} + \boldsymbol{y}^{(l+1)}}$
end for

3.2) are called variants of the GMRES(m) method; and by setting a strategy to define the vector $\boldsymbol{y}^{(l+1)}$, we indeed provide an example that has more efficient convergence than the GMRES(m) method. In Section 4.1, the error equations and the iterative refinement scheme are briefly introduced. In Section 4.2, we analyze the mathematical background of the unfixed update based on the error equations and the iterative refinement scheme. Then in Section 4.3 we provide an example of how to define $\boldsymbol{y}^{(l+1)}$ for a variant of the GMRES(m) method.

4.1. The error equations and iterative refinement scheme

Let $\boldsymbol{x}$ be the exact solution $\boldsymbol{x} := A^{-1}\boldsymbol{b}$ of the linear system (1.1), and $\widehat{\boldsymbol{x}}$ an approximate numerical solution. Let $\boldsymbol{e} := \boldsymbol{x} - \widehat{\boldsymbol{x}}$ denote the error vector, and $\widehat{\boldsymbol{r}} := \boldsymbol{b} - A\widehat{\boldsymbol{x}}$ the residual vector. Then from the relation

$$\boldsymbol{e} := \boldsymbol{x} - \widehat{\boldsymbol{x}} = A^{-1}(\boldsymbol{b} - A\widehat{\boldsymbol{x}}) = A^{-1}\widehat{\boldsymbol{r}} \,,$$

the error vector $\boldsymbol{e}$ can be computed by solving the *error equation*

$$A\boldsymbol{e} = \widehat{\boldsymbol{r}} \,. \tag{4.1}$$

Solving the error equations (4.1) recursively to improve the accuracy of the numerical solution constitutes the *iterative refinement scheme*, originally introduced by Wilkinson in [26] to eliminate the effect of round-off error in direct methods such as LU factorization.

4.2. Mathematical background of the unfixed update

In this section, we analyze the mathematical background of the unfixed update based on the error equations and the iterative refinement scheme. For the following analysis, we introduce two kinds of iterative refinement schemes — viz. Algorithm 4.1 and Algorithm 4.2 as shown, where each error equation is solved by m iterations of the GMRES method to achieve higher accuracy.

In Algorithm 4.1, the initial guess for each error equation is fixed as $\boldsymbol{e}_0^{(l)} := \boldsymbol{0}$. On the other hand, in Algorithm 4.2 each initial guess is set as $\boldsymbol{e}_0^{(l)} := \boldsymbol{y}^{(l)}$, where $\boldsymbol{y}^{(l)} \in \mathbb{R}^n$ is decided using a certain strategy. From these Algorithms, we analyze the mathematical background of the unfixed update. The analysis involves three steps (cf. also Fig. 1) — viz.

Algorithm 4.1 The iterative refinement scheme based on m iterations of the GMRES method with the initial guess $\boldsymbol{e}_0^{(l)} := \boldsymbol{0}$

1: Choose the initial guess $\boldsymbol{x}_0$ and set $\boldsymbol{x}_m^{(0)} := \boldsymbol{0}, \boldsymbol{r}_m^{(0)} := \boldsymbol{b}, \boldsymbol{e}_0^{(1)} := \boldsymbol{x}_0$
2: **for** $l = 1, 2, \cdots$, until convergence **do**
3: Solve (approximately) $A\boldsymbol{e} = \boldsymbol{r}_m^{(l-1)}$ by m iterations of the GMRES method with the initial guess $\boldsymbol{e}_0^{(l)}$, and get the approximate solution $\boldsymbol{e}_m^{(l)}$
4: Update $\boldsymbol{x}_m^{(l)} := \boldsymbol{x}_m^{(l-1)} + \boldsymbol{e}_m^{(l)}, \boldsymbol{r}_m^{(l)} := \boldsymbol{b} - A\boldsymbol{x}_m^{(l)}, \underline{\boldsymbol{e}_0^{(l+1)} := \boldsymbol{0}}$
5: **end for**

Algorithm 4.2 The iterative refinement scheme based on m iterations of the GMRES method with the initial guess $\boldsymbol{e}_0^{(l)} := \boldsymbol{y}^{(l)}$

1: Choose the initial guess $\boldsymbol{x}_0$ and set $\boldsymbol{x}_m^{(0)} := \boldsymbol{0}, \boldsymbol{r}_m^{(0)} := \boldsymbol{b}, \boldsymbol{e}_0^{(1)} := \boldsymbol{x}_0$
2: **for** $l = 1, 2, \cdots$, until convergence **do**
3: Solve (approximately) $A\boldsymbol{e} = \boldsymbol{r}_m^{(l-1)}$ by m iterations of the GMRES method with the initial guess $\boldsymbol{e}_0^{(l)}$, and get the approximate solution $\boldsymbol{e}_m^{(l)}$
4: Set the vector $\boldsymbol{y}^{(l+1)}$ based on a certain strategy
5: Update $\boldsymbol{x}_m^{(l)} := \boldsymbol{x}_m^{(l-1)} + \boldsymbol{e}_m^{(l)}, \boldsymbol{r}_m^{(l)} := \boldsymbol{b} - A\boldsymbol{x}_m^{(l)}, \underline{\boldsymbol{e}_0^{(l+1)} := \boldsymbol{y}^{(l+1)}}$
6: **end for**

Step 1. Algorithms 3.1 and 4.1 are shown to be mathematically equivalent.

Step 2. Algorithm 4.2 is shown to be a natural extension of Algorithm 4.1.

Step 3. Algorithms 3.2 and 4.2 are shown to be mathematically equivalent.

Step 1. Comparison between Algorithm 3.1 and Algorithm 4.1.

The mathematical equivalence of the GMRES(m) under Algorithm 3.1 and the iterative refinement scheme based on m iterations in the GMRES (Algorithm 4.1) is covered by the following proposition.

Proposition 4.1. *Let $\boldsymbol{x}_m^{(l)} = \boldsymbol{x}_0^{(l)} + \boldsymbol{z}_m^{(l)}$ be the approximate solution of the linear system $A\boldsymbol{x} = \boldsymbol{b}$ obtained by m iterations of GMRES with the initial guess $\boldsymbol{x}_0^{(l)} := \boldsymbol{x}_m^{(l-1)}$; and let $\boldsymbol{e}_m^{(l)} = \boldsymbol{e}_0^{(l)} + \tilde{\boldsymbol{z}}_m^{(l)}$ be the approximate solution of the linear system $A\boldsymbol{e} = \boldsymbol{r}_m^{(l-1)}$ obtained by m iterations of GMRES with the initial guess $\boldsymbol{e}_0^{(l)} := \boldsymbol{0}$. Then we have the equality*

$$\boldsymbol{z}_m^{(l)} = \tilde{\boldsymbol{z}}_m^{(l)},$$

where $\boldsymbol{r}_m^{(l-1)} := \boldsymbol{b} - A\boldsymbol{x}_m^{(l-1)}$.

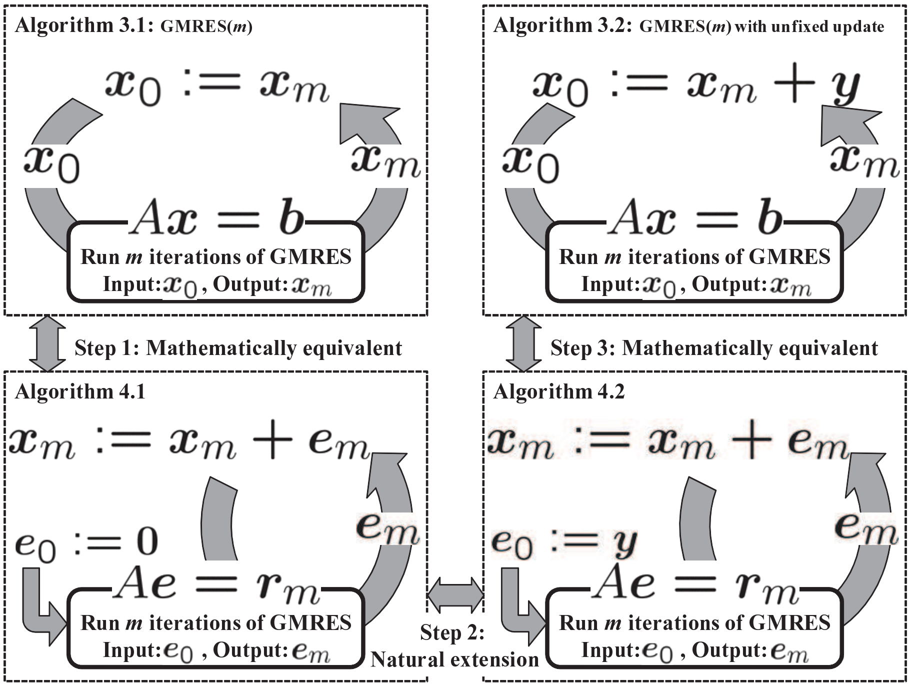

Figure 1: The analysis procedure on the mathematical background of the unfixed update.

Proof. From the minimum residual condition (2.1) for m iterations of GMRES, the vectors $\boldsymbol{z}_m^{(l)}$ and $\widetilde{\boldsymbol{z}}_m^{(l)}$ can be written as

$$\boldsymbol{z}_m^{(l)} = \arg \min_{\boldsymbol{z} \in \mathscr{K}_m(A, \boldsymbol{r}_0^{(l)})} \|\boldsymbol{b} - A(\boldsymbol{x}_0^{(l)} + \boldsymbol{z})\|_2,$$

$$\widetilde{\boldsymbol{z}}_m^{(l)} = \arg \min_{\boldsymbol{z} \in \mathscr{K}_m(A, \boldsymbol{r}_m^{(l-1)})} \|\boldsymbol{r}_m^{(l-1)} - A(\boldsymbol{0} + \boldsymbol{z})\|_2\,,$$

where from $\boldsymbol{x}_0^{(l)} := \boldsymbol{x}_m^{(l-1)}$ and $\boldsymbol{r}_m^{(l-1)} := \boldsymbol{b} - A\boldsymbol{x}_m^{(l-1)}$ we have

$$\begin{aligned} \arg \min_{\boldsymbol{z} \in \mathscr{K}_m(A, \boldsymbol{r}_0^{(l)})} \|\boldsymbol{b} - A(\boldsymbol{x}_0^{(l)} + \boldsymbol{z})\|_2 &= \arg \min_{\boldsymbol{z} \in \mathscr{K}_m(A, \boldsymbol{r}_0^{(l)})} \|(\boldsymbol{b} - A\boldsymbol{x}_m^{(l-1)}) - A\boldsymbol{z}\|_2 \\ &= \arg \min_{\boldsymbol{z} \in \mathscr{K}_m(A, \boldsymbol{r}_m^{(l-1)})} \|\boldsymbol{r}_m^{(l-1)} - A(\boldsymbol{0} + \boldsymbol{z})\|_2\,. \quad \square \end{aligned}$$

Step 2. Comparison between Algorithms 4.1 and 4.2

In general, when one solves the linear systems (1.1) by some Krylov subspace method (CG, GMRES etc.), it is not necessary to fix the initial guess at $\boldsymbol{x}_0 := \boldsymbol{0}$. From this viewpoint, Algorithm 4.2 can be regarded as a natural extension of Algorithm 4.1, in the sense that one can use any initial guess for each error equation.

Step 3. Comparison between Algorithms 3.2 and 4.2

The mathematical equivalence of the GMRES(m) with unfixed update (Algorithm 3.2) and the iterative refinement scheme based on m iterations in the GMRES (Algorithm 4.2) is covered by the following proposition.

Proposition 4.2. *Let $\boldsymbol{x}_m^{(l)} = \boldsymbol{x}_0^{(l)} + \boldsymbol{z}_m^{(l)}$ be the approximate solution for the linear system $A\boldsymbol{x} = \boldsymbol{b}$ obtained by m iterations of GMRES with the initial guess $\boldsymbol{x}_0^{(l)} := \boldsymbol{x}_m^{(l-1)} + \boldsymbol{y}^{(l)}$; and let $\boldsymbol{e}_m^{(l)} = \boldsymbol{e}_0^{(l)} + \widetilde{\boldsymbol{z}}_m^{(l)}$ be the approximate solution for the linear system $A\boldsymbol{e} = \boldsymbol{r}_m^{(l-1)}$ obtained by m iterations of GMRES with the initial guess $\boldsymbol{e}_0^{(l)} := \boldsymbol{y}^{(l)}$. Then we have the equality*

$$\boldsymbol{z}_m^{(l)} = \widetilde{\boldsymbol{z}}_m^{(l)} ,$$

where $\boldsymbol{r}_m^{(l-1)} := \boldsymbol{b} - A\boldsymbol{x}_m^{(l-1)}$.

Proof. Proposition 4.2 can be proved in the same way as Proposition 4.1. □

Short remark on the analysis

From Steps 1, 2 and 3, the vector $\boldsymbol{y}^{(l+1)}$ of the unfixed update (3.2) corresponds to the initial guess of each error equation in the iterative refinement scheme. Thus GMRES(m) with unfixed update (Algorithm 3.2) can be regarded as a natural extension of GMRES(m) (Algorithm 3.1), in reference to the iterative refinement scheme (cf. also Fig. 1).

4.3. An example of how to define $\boldsymbol{y}^{(l+1)}$ for efficient convergence

As noted in Section 4.2, the GMRES(m) method with unfixed update (Algorithm 3.2) can be regarded as a natural extension of the GMRES(m) method (Algorithm 3.1). However, if we do not have any strategy to define $\boldsymbol{y}^{(l+1)}$, then the unfixed update may lead to poor convergence. In this section, we illustrate an example where the vector $\boldsymbol{y}^{(l+1)}$ is set without any strategy (randomly), and then we provide another example to suggest how to define the vector $\boldsymbol{y}^{(l+1)}$ for more efficient convergence.

Fig. 2(a) shows the relative residual 2-norm history of the GMRES(m) method and the GMRES(m) method with unfixed update under no strategy. (The vector $\boldsymbol{y}^{(l+1)}$ is set randomly.) One can see that the GMRES(m) method with unfixed update does not guarantee a monotonic decrease in the residual 2-norm, and there is slower convergence than with the GMRES(m) method. Thus in general, an arbitrary vector $\boldsymbol{y}^{(l+1)}$ cannot guarantee good convergence — so we require a suitable strategy to define $\boldsymbol{y}^{(l+1)}$ for efficient convergence compared with the GMRES(m) method, and we now provide one example: i.e.

$$\boldsymbol{y}^{(l+1)} = \begin{cases} \boldsymbol{0} & (l = 1) \\ \alpha^{(l+1)}(\boldsymbol{z}_m^{(l)} + \boldsymbol{y}^{(l)} + \boldsymbol{z}_m^{(l-1)}) & (l \geq 2) \end{cases} , \tag{4.2}$$

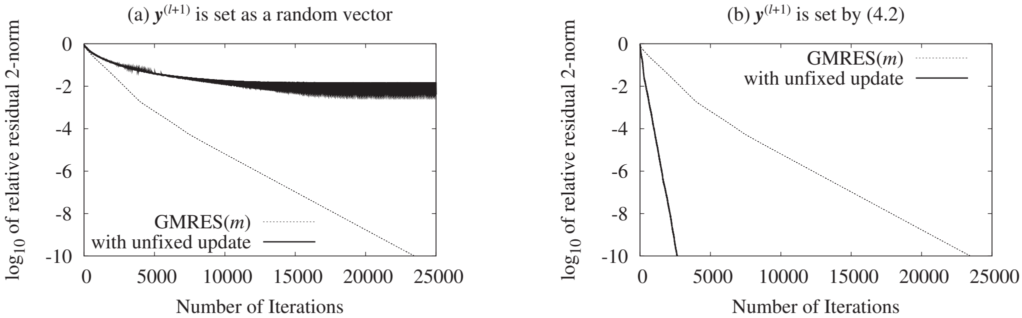

Figure 2: The relative residual 2-norm history for MEMPLUS, where $m = 10$.

where $\alpha^{(l+1)} = \arg\min_{\alpha\in\mathbb{R}} ||\boldsymbol{r}_m^{(l)} - \alpha A(\boldsymbol{z}_m^{(l)} + \boldsymbol{y}^{(l)} + \boldsymbol{z}_m^{(l-1)})||_2$. In our current implementation, the computational cost to obtain vector $\boldsymbol{y}^{(l+1)}$ based on (4.2) is one matrix-vector multiplication and some AXPY[†] and inner products. Strategy (4.2) guarantees the monotonic decrease in the residual 2-norm as well as the GMRES(m) method — i.e. we have

$$||\boldsymbol{r}_m^{(l+1)}||_2 \leq ||\boldsymbol{r}_0^{(l+1)}||_2 = ||\boldsymbol{r}_m^{(l)} - A\boldsymbol{y}^{(l+1)}||_2 \leq ||\boldsymbol{r}_m^{(l)}||_2.$$

Fig. 2(b) shows the relative residual 2-norm history of the GMRES(m) method and the GMRES(m) method with unfixed update, based on (4.2). In this result, the GMRES(m) method with unfixed update shows good convergence compared with the GMRES(m) method.

The GMRES(m) method with unfixed update based on (4.2) is the specific variant we consider further, and in the next section we present some numerical experiments to show the potential for efficient convergence.

5. Numerical Experiments and Results

We have proposed an efficient variant of the GMRES(m) method by reconsidering the update to the initial guess, but a suitable strategy to define $\boldsymbol{y}^{(l+1)}$ for efficient convergence in general requires further investigation. However, to show the potential for efficient convergence, we now present some results from numerical experiments.

5.1. Numerical experiments

We test the performance of the GMRES(m) method Algorithm 3.1, and the variant of the GMRES(m) method Algorithm 3.2 where $\boldsymbol{y}^{(l+1)}$ is set by (4.2). Their performance is evaluated using test problems from the Matrix Market [11] and UF Sparse Matrix Collection [6] that came from Finite element modeling (CAVITY), Circuit simulation problem (CIRCUIT, COUPLED, RAJAT), Optimization problem (CRASHBASIS), Thermal problem (FEM_3D_THERMAL), Partial differential equations (PDE), Chemical engineering (RDB),

[†]Addition of scaled vectors.

Structural problem (T2D_Q4), Petroleum engineering (WATT) and Materials problem (XENON). Since our purpose is to show a convergence potential within the specific variant of the GMRES(m) method, we set $m = 10$ and 50 for the restart frequency and did not use preconditioners. We set $\boldsymbol{b} = [1, 1, \cdots, 1]^{\mathrm{T}}$ as the right-hand side vector, $\boldsymbol{x}_0 = [0, 0, \cdots, 0]^{\mathrm{T}}$ for the initial guess, and the stopping criterion as $||\boldsymbol{r}_k||_2/||\boldsymbol{b}||_2 \leq 10^{-10}$. The notation † denotes that the methods did not converge within 50000 iterations. The numerical experiments were implemented with standard Fortran 77 in double precision arithmetic, on an Intel Xeon (2.8GHz).

5.2. Numerical results

The numerical results for $m = 10$ and 50 are presented in Tables 1 and 2, respectively. We analyze the results in terms of three aspects: the number of iterations; computation time per one restart cycle (m iterations); and total computation time.

First, let us consider the number of iterations (Iter) of both methods. In most cases, the variant of the GMRES(m) method shows almost the same or lower Iter than the GMRES(m) method. In particular, for CAVITY05 ($m = 10$), CAVITY10 ($m = 50$), COUPLED ($m = 50$) and RAJAT03 ($m = 10, 50$), the GMRES(m) method did not converge within 50000; on the other hand, the variant of the GMRES(m) method converged to the solution satisfying the required accuracy $||\boldsymbol{r}_k||_2/||\boldsymbol{b}||_2 \leq 10^{-10}$. The variant of the GMRES(m) method also converged within 1/20 iterations of the GMRES(m) method for WATT__2, XENON1 and XENON2 ($m = 10$). Moreover, from the comparison between Table 1 for $m = 10$ and Table 2 for $m = 50$, we can see that the smaller restart frequency m leads to a larger difference in Iter between both methods.

Next, we consider the computation time per restart cycle (t_{Restart}). In terms of t_{Restart}, for $m = 10$ the variant of the GMRES(m) method requires at most 10% more time than the GMRES(m) method. When $m = 50$, both methods require almost the same t_{Restart}, because in the current implementation the additional operations for the variant of the GMRES(m) method are one matrix-vector multiplication and some AXPY and inner-products per restart cycle for computing the vector $\boldsymbol{y}^{(l+1)}$ by (4.2).

In terms of the total computation time (t_{Total}), from the smaller Iter and almost the same t_{Restart}, we can see that the variant of the GMRES(m) method can converg within a much smaller computation time, except for the cases RDB2048L ($m = 10$) and PDE2961 ($m = 50$).

The relative residual 2-norm histories for CAVITY05, PDE2961, RDB2048L and XENON1 are shown in Fig. 3. We can see that the variant of the GMRES(m) method shows a monotonic decrease in the residual, as does the GMRES(m) method. In Fig. 3 for PDE2961 and RDB2048L, both methods show the same level of convergence throughout the whole iteration. On the other hand, for CAVITY05 and XENON1, the variant of the GMRES(m) method shows a better convergence than the GMRES(m) method throughout the whole iteration.

From these results, it appears that the variant of the GMRES(m) method may have a high potential for efficient convergence.

Table 1: Test problems (n: order of matrix, Nnz: number of nonzeros in matrix) and convergence results (Iter: the number of iterations, t_{Total}: total computation time, t_{Restart}: computation time per one restart cycle) of the GMRES(m) method and the variant of the GMRES(m) method, where $m = 10$.

Matrix		Solver	Iter	Time[sec.]	
n	Nnz			t_{Total}	t_{Restart}
CAVITY05		GMRES(m)	†	†	3.98×10^{-3}
1182	32747	Variant	11791	5.03×10^{0}	4.26×10^{-3}
CAVITY10		GMRES(m)	†	†	9.23×10^{-3}
2597	76367	Variant	†	†	9.89×10^{-3}
CIRCUIT_1		GMRES(m)	938	4.30×10^{-1}	4.57×10^{-3}
2597	35823	Variant	394	1.92×10^{-1}	4.91×10^{-3}
CIRCUIT_2		GMRES(m)	8830	5.06×10^{0}	5.71×10^{-3}
4510	21199	Variant	4862	2.88×10^{0}	5.93×10^{-3}
COUPLED		GMRES(m)	†	†	2.05×10^{-2}
11341	98523	Variant	†	†	2.15×10^{-2}
CRASHBASIS		GMRES(m)	819	2.63×10^{1}	3.21×10^{-1}
160000	1750416	Variant	678	2.27×10^{1}	3.35×10^{-1}
FEM_3D_THERMAL1		GMRES(m)	711	4.14×10^{0}	5.82×10^{-2}
17880	430740	Variant	281	1.74×10^{0}	6.18×10^{-2}
FEM_3D_THERMAL2		GMRES(m)	2660	1.33×10^{2}	5.01×10^{-1}
147900	3489300	Variant	591	3.16×10^{1}	5.34×10^{-1}
PDE2961		GMRES(m)	641	2.38×10^{-1}	3.77×10^{-3}
2961	14585	Variant	491	1.86×10^{-1}	3.82×10^{-3}
RAJAT03		GMRES(m)	†	†	9.19×10^{-3}
7602	32653	Variant	38701	3.65×10^{1}	9.44×10^{-3}
RDB2048L		GMRES(m)	741	1.98×10^{-1}	2.70×10^{-3}
2048	12032	Variant	898	2.44×10^{-1}	2.75×10^{-3}
RDB3200L		GMRES(m)	1294	5.44×10^{-1}	4.21×10^{-3}
3200	18880	Variant	1072	4.64×10^{-1}	4.33×10^{-3}
T2D_Q4		GMRES(m)	2520	3.88×10^{0}	1.54×10^{-2}
9801	87025	Variant	356	5.70×10^{-1}	1.61×10^{-2}
WATT__2		GMRES(m)	29129	7.38×10^{0}	2.51×10^{-3}
1856	11550	Variant	1630	4.22×10^{-1}	2.61×10^{-3}
XENON1		GMRES(m)	34112	5.40×10^{2}	1.57×10^{-1}
48600	1181120	Variant	1891	3.22×10^{1}	1.70×10^{-1}
XENON2		GMRES(m)	47602	2.60×10^{3}	5.45×10^{-1}
157464	3866688	Variant	2415	1.41×10^{2}	5.84×10^{-1}

6. Conclusion

In this paper, we considered the algorithm of the GMRES(m) method with unfixed update. From analysis based on the error equations and the iterative refinement scheme, we

Table 2: Test problems (n: order of matrix, Nnz: number of nonzeros in matrix) and convergence results (Iter: number of iterations, t_{Total}: total computation time, t_{Restart}: computation time per one restart cycle) of the GMRES(m) method and the variant of the GMRES(m) method, where $m = 50$.

Matrix		Solver	Iter	Time[sec.]	
n	Nnz			t_{Total}	t_{Restart}
CAVITY05		GMRES(m)	45027	3.00×10^{1}	3.33×10^{-2}
1182	32747	Variant	4801	3.23×10^{0}	3.37×10^{-2}
CAVITY10		GMRES(m)	†	†	7.65×10^{-2}
2597	76367	Variant	10251	1.58×10^{1}	7.71×10^{-2}
CIRCUIT_1		GMRES(m)	310	3.30×10^{-1}	5.33×10^{-2}
2597	35823	Variant	289	3.06×10^{-1}	5.40×10^{-2}
CIRCUIT_2		GMRES(m)	441	7.08×10^{-1}	8.12×10^{-2}
4510	21199	Variant	384	6.22×10^{-1}	8.28×10^{-2}
COUPLED		GMRES(m)	†	†	2.38×10^{-1}
11341	98523	Variant	26953	1.28×10^{2}	2.39×10^{-1}
CRASHBASIS		GMRES(m)	431	3.16×10^{1}	3.73×10^{0}
160000	1750416	Variant	422	3.10×10^{1}	3.78×10^{0}
FEM_3D_THERMAL1		GMRES(m)	318	3.09×10^{0}	4.93×10^{-1}
17880	430740	Variant	276	2.68×10^{0}	4.99×10^{-1}
FEM_3D_THERMAL2		GMRES(m)	775	6.83×10^{1}	4.44×10^{0}
147900	3489300	Variant	559	4.96×10^{1}	4.47×10^{0}
PDE2961		GMRES(m)	483	5.12×10^{-1}	5.43×10^{-2}
2961	14585	Variant	572	6.04×10^{-1}	5.40×10^{-2}
RAJAT03		GMRES(m)	†	†	1.37×10^{-1}
7602	32653	Variant	12701	3.49×10^{1}	1.37×10^{-1}
RDB2048L		GMRES(m)	337	2.48×10^{-1}	3.60×10^{-2}
2048	12032	Variant	300	2.26×10^{-1}	3.87×10^{-2}
RDB3200L		GMRES(m)	477	5.42×10^{-1}	5.83×10^{-2}
3200	18880	Variant	470	5.40×10^{-1}	5.83×10^{-2}
T2D_Q4		GMRES(m)	517	1.98×10^{0}	1.94×10^{-1}
9801	87025	Variant	455	1.77×10^{0}	1.96×10^{-1}
WATT__2		GMRES(m)	4606	3.14×10^{0}	3.40×10^{-2}
1856	11550	Variant	1452	9.90×10^{-1}	3.40×10^{-2}
XENON1		GMRES(m)	7047	1.90×10^{2}	1.35×10^{0}
48600	1181120	Variant	1892	5.13×10^{1}	1.36×10^{0}
XENON2		GMRES(m)	9456	9.10×10^{2}	4.82×10^{0}
157464	3866688	Variant	2341	2.27×10^{2}	4.86×10^{0}

found the GMRES(m) method with unfixed update is a natural extension of the GMRES(m) method. The variant of the GMRES(m) method based on (4.2) was examined, and some numerical experiments showed more efficient convergence than the GMRES(m) method

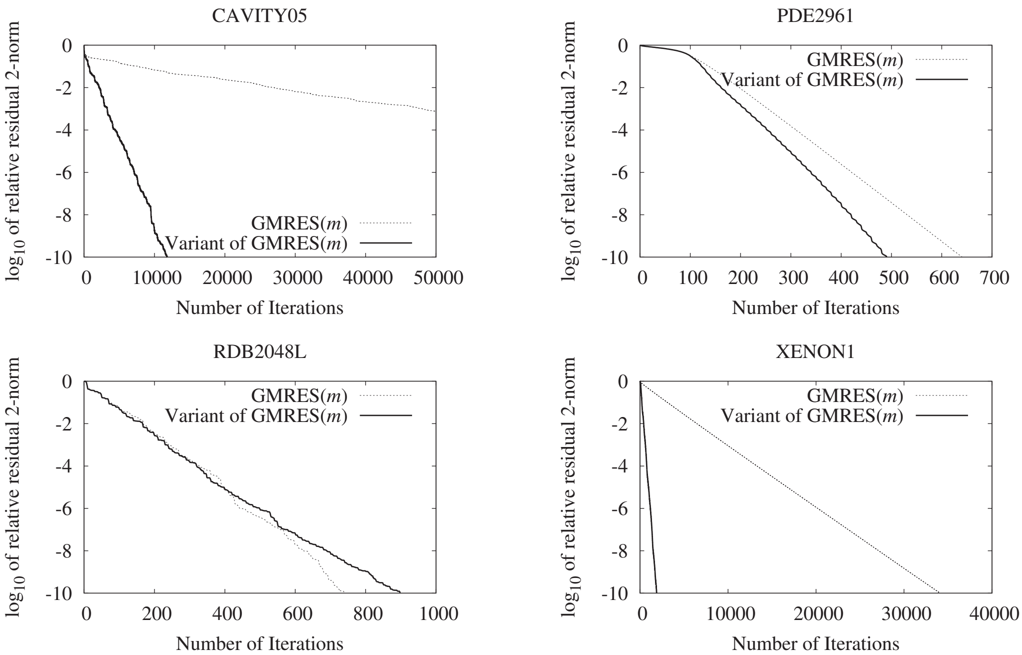

Figure 3: The relative residual 2-norm history for CAVITY05, PDE2961, RDB2048L and XENON1, where $m = 10$.

widely used to solve large sparse nonsymmetric linear systems. In future work, the convergence behavior of the variant of the GMRES(m) method should be analysed, to design the most suitable strategy to define the vector $\boldsymbol{y}^{(l+1)}$.

Acknowledgments

The authors thank an anonymous referee for useful comments. This work was supported by Grant-in-Aid for JSPS Fellows (Grant No. 21·8663) and Grant-in-Aid for Scientific Research (Grant No. 19560065, 21760058, 22104004).

References

[1] J. Baglama, D. Calvetti, G. H. Golub, and L. Reichel, *Adaptively preconditioned GMRES algorithms*, SIAM J. Sci. Comput., 20 (1999), pp. 17–29.

[2] J. Baglama and L. Reichel, *Augmented GMRES-type methods*, Numer. Linear Alg. Appl., 14 (2007), pp. 337–350.

[3] A. H. Baker, E. R. Jessup, and T. V. Kolev, *A simple strategy for varying the parameter in GMRES(m)*, J. Comput. Appl. Math., 230 (2009), pp. 751–761.

[4] A. H. Baker, E. R. Jessup, and T. Manteuffel, *A technique for accelerating the convergence of restarted GMRES*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 962–984.

[5] K. Burrage and J. Erhel, *On the performance of various adaptive preconditioned GMRES strategies*, Numer. Linear Alg. Appl., 5 (1998), pp. 101–121.

[6] T. A. Davis, *The University of Florida Sparse Matrix Collection*, Available from http://www.cise.ufl.edu/research/sparse/matrices/.

[7] J. Erhel, K. Burrage, and B. Pohl, *Restarted GMRES preconditioned by deflation*, J. Comput. Appl. Math., 69 (1996), pp. 303–318.

[8] G. H. Golub and H. A. van der Vorst, *Closer to the solution: Iterative linear solvers*, in The State of the Art in Numerical Analysis, I. S. Duff and G. A. Watros eds. Clarendon Press, Oxford, 1997, pp. 63–92.

[9] M. H. Gutknecht, *Variants of BiCGStab for matrices with complex spectrum*, SIAM J. Sci. Stat. Comput., 14 (1993), pp. 1020–1033.

[10] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Natl. Bur. Stand., 49 (1952), pp. 409–436.

[11] Matrix Market, Available from http://math.nist.gov/MatrixMarket/.

[12] R. B. Morgan, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171.

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

[14] R. B. Morgan, *GMRES with deflated restarting*, SIAM J. Sci. Comput., 24 (2002), pp. 20–37.

[15] K. Moriya and T. Nodera, *The DEFLATED-GMRES(m,k) method with switching the restart frequency dynamically*, Numer. Linear Alg. Appl., 7 (2000), pp. 569–584.

[16] C. C. Paige and M. A. Saunders, *Solution of Sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–729.

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

[18] Y. Saad and H. A. van der Vorst, *Iterative solution of linear systems in the 20th century*, J. Comput. Appl. Math., 123 (2000), pp. 1–33.

[19] V. Simoncini and D. B. Szyld, *Recent computational developments in Krylov subspace methods for linear systems*, Numer. Linear Alg. Appl., 14 (2007), pp. 1–59.

[20] G. L. G. Sleijpen and D. R. Fokkema, *BiCGStab(l) for linear equations involving unsymmetric matrices with complex spectrum*, Electronic Trans. Numer. Anal., 1 (1993), pp. 11–32.

[21] G. L. G. Sleijpen, P. Sonneveld, and M. B. van Gijzen, *Bi-CGSTAB as an induced dimension reduction method*, Appl. Numer. Math. 60 (2010), pp. 1100–1114.

[22] P. Sonneveld and M. B. van Gijzen, *IDR(s): A family of simple and fast algorithms for solving large nonsymmetric systems of linear equations*, SIAM J. Sci. Comput., 31 (2008), pp. 1035–1062.

[23] M. Sosonkina, L. T. Watson, R. K. Kapania, and H. F. Walker, *A new adaptive GMRES algorithm for achieving high accuracy*, Numer. Linear Alg. Appl., 5 (1998), pp. 275–297.

[24] M. Tanio and M. Sugihara, *GBi-CGSTAB(s,L): IDR(s) with higher-order stabilization polynomials*, J. Comput. Appl. Math., 235 (2010), pp. 765–784.

[25] H. A. van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 631–644.

[26] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice Hall, Englewood Cliffs, NJ, 1963.

[27] L. Zhang and T. Nodera, *A new adaptive restart for GMRES(m) method*, ANZIAM J., 46 (2005), pp. 409–426.

[28] S.-L. Zhang, *GPBi-CG: Generalized product-type methods based on Bi-CG for solving nonsymmetric linear systems*, SIAM J. Sci. Comput., 18 (1997), pp. 537–551.