

Article

VRP-GMRES(m) Iteration Algorithm for Fast Multipole Boundary Element Method

Chunxiao Yu, Cuihuan Ren * and Xueting Bai

College of Science, Yanshan University, Qinhuangdao 066004, China; chxy@ysu.edu.cn

* Correspondence: 18713510209@163.com; Tel.: +1-871-351-0209

Abstract: To solve large scale linear equations involved in the Fast Multipole Boundary Element Method (FM-BEM) efficiently, an iterative method named the generalized minimal residual method (GMRES)(m) algorithm with Variable Restart Parameter (VRP-GMRES(m) algorithm) is proposed. By properly changing a variable restart parameter for the GMRES(m) algorithm, the iteration stagnation problem resulting from improper selection of the parameter is resolved efficiently. Based on the framework of the VRP-GMRES(m) algorithm and the relevant properties of generalized inverse matrix, the projection of the error vector r_{m+1} on r_m is deduced. The result proves that the proposed algorithm is not only rapidly convergent but also highly accurate. Numerical experiments further show that the new algorithm can significantly improve the computational efficiency and accuracy. Its superiorities will be much more remarkable when it is used to solve larger scale problems. Therefore, it has extensive prospects in the FM-BEM field and other scientific and engineering computing.

Keywords: FM-BEM; variable restart parameter; GMRES(m); error vector projection; convergence

1. Introduction

Mathematical models of partial differential equations are usually established for many problems in scientific and engineering fields. After discretization, the equations can be concluded as the solution of a large scale linear system of equations, which can be expressed as

$$Ax = b, A \in R^{n \times n}, x, b \in R^n, \quad (1)$$

where the coefficient matrix A is nonsingular. The generalized minimal residual (GMRES(m)) algorithm based on the Galerkin principle is the most successful method to solve Equation (1). At present, it has served as the fast solver for the Fast Multipole Boundary Element Method (FM-BEM). FM-BEM is a combination of the fast multipole expansion method (FMM) and the traditional boundary element method (BEM). FMM was first proposed by Greengard and Rokhlin [1,2] to accelerate the evaluation of interactions of large ensembles of particles governed by the Laplace's equation. The main idea behind the FMM is a multipole expansion of the kernel in which the connection between the collocation point and the source point is separated. Many research works have been published since then to improve and extend the applicability of the FMM [3–7]. Recently, Gu and Chen [8,9] applied the FMM to accelerate the solutions of the regularized meshless method (RMM) for large-scale three-dimensional elasticity problems and extended the singular boundary method (SBM) for the solution of 3D problems in linear elasticity. With the FMM and iterative solver GMRES(m), we are able to construct the FM-BEM. In the last several decades, the FM-BEM has been developed and used successfully for solving many large-scale applied mechanics problems. Shen and Liu put forward an adaptive fast multipole boundary element method for three-dimensional acoustic wave problems [10]; Wang and Yao came up with a new version of FMM for the numerical analysis of mechanical properties in 3D particle reinforced composites [11]; Gui and Huang presented an optimization FMM-BEM for a 3D Elastic Problem [12] and more work on these topics can be found in References [13–15]. In addition, the GMRES(m) algorithm has become the research focus in many fields [16–19]. Actual computations showed that it was very effective when the coefficient matrix A was well-conditioned or not severely ill-conditioned [20,21]. Otherwise preconditioned techniques must be applied [22–25].

For the GMRES(m) algorithm, the selected parameter m is fixed during the whole iterative process; therefore, the selection of m is one of the key factors for the algorithm implementation. Research shows that a small value of m may result in slow convergence or no convergence, while a large value of m will cause too much of a memory requirement. Therefore, the appropriate selection of parameter m is a trouble for many scholars. Recently, some researchers have tried to overcome these difficulties by changing the restart parameter m in the GMRES(m) algorithm. Baker tried to improve the convergence by selecting an arbitrary parameter m . In addition, he tried it by determining the parameter m according to the continuous two times of residual norms ratio [26,27]. However, the effect was not so good. Peairs determined the value of m by a reinforcement learning method, which indicated that proper changing of m could improve the computational efficiency. However, it was only a kind of machine learning method. Thus, it had certain limitations [28].

In this paper, the GMRES(m) algorithm combined with the FM-BEM is studied, and a new kind of GMRES(m) algorithm with Variable Restart Parameter (VRP-GMRES(m) algorithm) is presented. By appropriately changing a variable restart parameter, the new algorithm can effectively avoid many disadvantages caused by improper selection of parameter m in the GMRES(m) algorithm. At the same time, the computational efficiency and accuracy will be greatly improved.

2. Fundamental Theory for the GMRES(m) Algorithm

2.1. Galerkin Principle

Given an arbitrary initial value $x_0 \in R^n$. Let $x = x_0 + z$, and then Equation (1) is equivalent to

$$Az = r_0, \quad (2)$$

where $r_0 = b - Ax_0$ indicates the initial residual vector. Suppose that K_m and L_m are two m -dimensional Krylov subspaces in R^n , and they are formed by

$$K_m = \text{span} \{ r_0, Ar_0, \dots, A^{m-1}r_0 \},$$

$$L_m = \text{span} \{ Ar_0, A^2r_0, \dots, A^m r_0 \} = AK_m.$$

Let v_1, v_2, \dots, v_m and w_1, w_2, \dots, w_m be the bases of K_m and L_m , respectively; $V_m = \{v_i\}_{i=1}^m$ and $W_m = \{w_i\}_{i=1}^m$. For the solution of Equation (2), the Galerkin principle can be described as follows: give a fixed $m > 0$, and find an approximate solution $z_m \in K_m$, so that $(r_0 - Az_m) \perp L_m$, namely, $(r_0 - AV_m y_m) \perp L_m$ or $(r_0 - AV_m y_m, w_i) = 0, y_m \in R^m$.

2.2. Arnoldi Process

The Arnoldi process can be described as follows:

1. For the given $m > 0$ and $x_0 \in R^n$, compute $v_1 = r_0 / \|r_0\|, r_0 = b - Ax_0$.
2. When $k = 1, 2, \dots, m$, compute $\bar{v}_{k+1} = Av_k - \sum_{i=1}^k h_{i,k} v_i, h_{i,k} = (Av_k, v_i)$ and $h_{k+1,k} = \|\bar{v}_{k+1}\|$, respectively. It is obvious that $\bar{v}_{k+1} \perp v_i (i = 1, 2, \dots, k)$.
3. If $h_{k+1,k} = 0$, then $v_{k+1} = 0$ and stop. Otherwise, let $v_{k+1} = \bar{v}_{k+1} / h_{k+1,k}$.

The Arnoldi process has the following important property:

Theorem 1. In the Arnoldi process, let $V_{m+1} = (V_m | v_{m+1})$ and $\bar{H}_m = \begin{bmatrix} H_m \\ h_{m+1,m} e_m^T \end{bmatrix}$, and then we have the following relationship [29]:

$$\begin{aligned} AV_m &= V_m H_m + h_{m+1,m} v_{m+1} e_m^T, \\ AV_m &= V_{m+1} \bar{H}_m, \quad V_{m+1}^T V_{m+1} = I, \end{aligned} \quad (3)$$

where $H_m = (h_{ij})_m$ is an upper Hessenberg matrix, $e_m^T = (0, 0, \dots, 1) \in R^m$, and I is an $m + 1$ order identity matrix.

2.3. GMRES(m) Algorithm

Theorem 2. Suppose that $A \in R^{n \times n}$ and $L_m = AK_m$, $x = x_0 + z = x_0 + V_m y$. Then, the approximate solution z_m obtained from the Galerkin principle makes the residual $\|r_0 - Az\|$ be at a minimum in the Krylov subspace K_m , and the residual satisfies [30]

$$\|r\| = \|b - Ax\| = \|r_0 - Az\| = \|\beta e_1 - \bar{H}_m y\|, \quad (4)$$

where $\beta = \|r_0\|$, $e_1 = (1, 0, \dots, 0)^T \in R^{m+1}$.

Theorem 2 indicates that $\min_{z_m \in K_m} \|r_0 - Az_m\| = \min_{y \in R^m} \|\beta e_1 - \bar{H}_m y\|$. The basic thought of the GMRES(m) algorithm is to give a fixed restart parameter m and compute z_m by an iterative procedure so that $\|r_m\| = \|r_0 - Az_m\| < \varepsilon$, $\forall \varepsilon > 0$.

The GMRES(m) algorithm includes the following steps:

1. Give a fixed integer $m \ll n$ and an initial value $x^{(0)} \in R^n$, compute $r^{(0)} = b - Ax^{(0)}$, and let $r_0 = r^{(0)}$.
2. Obtain $\{v_i\}_{i=1}^m$ and \bar{H}_m through the Arnoldi process.
3. Solve the least squares problem

$$\min_{y \in R^m} \|\beta e_1 - \bar{H}_m y\|$$

to obtain y_m , where $e_1 = (1, 0, \dots, 0)^T \in R^{m+1}$. Then, compute $z_m = V_m y_m$.

4. Form an iterative process $x^{(k+1)} = x^{(k)} + z_m$ or $r^{(k+1)} = r^{(k)} - Az_m$, $k = 0, 1, 2, \dots$.
5. If $\|r^{(k+1)}\| < \varepsilon$, then $x^* \approx x^{(k+1)}$ and stop. Otherwise, let $r_0 = r^{(k+1)}$, and go to 2.

For the GMRES(m) algorithm, arbitrary selection of parameter m cannot guarantee convergence. In fact, appropriate change of m can effectively improve the convergence and avoid the phenomenon of slow convergence or even no convergence.

3. VRP-GMRES(m) Algorithm

3.1. VRP-GMRES(m) Algorithm

The basic thought of the VRP-GMRES(m) algorithm is to appropriately change the restart parameter m and carry out iterative procedures so that the residual vector satisfies $\|r_m\| = \|r_0 - Az_m\| < \varepsilon$, $\forall \varepsilon > 0$. It includes the following main steps:

1. Give an integer $m \ll n$ and an initial value $x^{(0)} \in R^n$, compute $r^{(0)} = b - Ax^{(0)}$, and let $r_0 = r^{(0)}$.
2. Carry out the Arnoldi process and obtain $\{v_i\}_{i=1}^m$ and \bar{H}_m .
3. Solve the least squares problem

$$\min_{y \in R^m} \|\beta e_1 - \bar{H}_m y\| \quad (5)$$

and get y_m . Then, compute $z_m = V_m y_m$.

4. Form an iterative process by $x^{(k+1)} = x^{(k)} + z_m$, $k = 0, 1, 2, \dots$. Compute $r^{(k+1)} = b - Ax^{(k+1)}$.
5. If $\|r^{(k+1)}\| < \varepsilon$. Then, $x^* \approx x^{(k+1)}$ and stop. If $\|r^{(k+1)}\| \geq \varepsilon$, then let $r_0 = r^{(k+1)}$, $m = m + 1$, and go to 2.

For the VRP-GMRES(m) algorithm, slow convergence or no convergence rarely occurs. However, the memory requirement will grow with the increase of parameter m , which can be cleverly avoided. When m increases to some extent and the residual norm reduces to a proper value, the parameter m can be fixed and the GMRES(m) algorithm will be carried out.

3.2. Convergence Analysis

Definition 1. Suppose that $A \in C^{m \times n}$, $X \in C^{m \times n}$, at the same time if

$$\begin{cases} AXA = A, \\ XAX = X, \\ (AX)^H = AX, \\ (XA)^H = XA, \end{cases}$$

then X is the pseudo inverse matrix of A , denoted A^+ , namely $X = A^+$ [31].

Theorem 3. Suppose that $A \in C^{m \times n}$, and $A = BC$ is the maximum rank decomposition of A , then $X = C^H(CC^H)^{-1}(B^HB)^{-1}B^H$ is the pseudo inverse matrix of A .

Proof of Theorem 3.

$$\begin{aligned} AXA &= BCC^H(CC^H)^{-1}(B^HB)^{-1}B^HBC = BC = A, \\ XAX &= C^H(CC^H)^{-1}(B^HB)^{-1}B^HBCC^H(CC^H)^{-1}(B^HB)^{-1}B^H \\ &= C^H(CC^H)^{-1}(B^HB)^{-1}B^H = X. \end{aligned}$$

Hereto, the first two Moore Penrose equations are established. The following is to verify the other two Moore Penrose equations:

$$\begin{aligned} (AX)^H &= [BCC^H(CC^H)^{-1}(B^HB)^{-1}B^H]^H = [B(B^HB)^{-1}B^H]^H \\ &= B(B^HB)^{-1}B^H = AX, \\ (XA)^H &= [C^H(CC^H)^{-1}(B^HB)^{-1}B^HBC]^H = [C^H(CC^H)^{-1}C]^H \\ &= C^H(CC^H)^{-1}C = XA, \end{aligned}$$

From Definition 1, the matrix X is a pseudo inverse matrix of A . \square

Corollary 1. Suppose that $A \in C^{m \times n}$ is a matrix with full column rank. Then, we have

$$A^+ = (A^HA)^{-1}A^H. \quad (6)$$

Theorem 4. Suppose that r_m and r_{m+1} are the error vector of the m cycle and $m + 1$, respectively. Then, the following relationship is established:

$$\cos \langle r_m, r_{m+1} \rangle = \frac{\|r_{m+1}\|_2}{\|r_m\|_2}.$$

Proof of Theorem 4. For the VRP-GMRES(m) algorithm, when $m = m + 1$, the subspace K_m becomes

$$K_{m+1} = \text{span} \{r_m, Ar_m, \dots, A^m r_m\}.$$

From the Arnoldi process, $\{v_i\}_{i=1}^{m+1}$ and \tilde{H}_{m+1} are obtained, where $v_1 = \frac{r_m}{\beta}$, $\beta = \|r_m\|_2$, $\cos \langle r_m, r_{m+1} \rangle = \cos \langle v_1, r_{m+1} \rangle = \frac{v_1^T r_{m+1}}{\|r_{m+1}\|_2}$.

From Equation (5), the least squares solution $y_{m+1} = \bar{H}_{m+1}^+ (\beta e_1)$, $e_1 \in R^{m+2}$ is obtained, where \bar{H}_{m+1}^+ is a real matrix with full column rank. According to Equation (6), we have

$$\bar{H}_{m+1}^+ = \left(\bar{H}_{m+1}^T \bar{H}_{m+1} \right)^{-1} \bar{H}_{m+1}^T.$$

Then,

$$y_{m+1} = \left(\bar{H}_{m+1}^T \bar{H}_{m+1} \right)^{-1} \bar{H}_{m+1}^T (\beta e_1), \quad e_1 \in R^{m+2}. \quad (7)$$

On the other hand, we have

$$r_{m+1} = b - Ax_{m+1} = b - A(x_m + V_{m+1}y_{m+1}) = r_m - AV_{m+1}y_{m+1}.$$

According to Equation (3), we have $AV_{m+1} = V_{m+2}\bar{H}_{m+1}$. Then, we have

$$r_{m+1} = r_m - V_{m+2}\bar{H}_{m+1}y_{m+1} = \beta v_1 - V_{m+2}\bar{H}_{m+1}y_{m+1}.$$

Thus,

$$\begin{aligned} \|r_{m+1}\|_2^2 &= (\beta v_1 - V_{m+2}\bar{H}_{m+1}y_{m+1})^T (\beta v_1 - V_{m+2}\bar{H}_{m+1}y_{m+1}) \\ &= \beta^2 - \beta v_1^T V_{m+2}\bar{H}_{m+1}y_{m+1} - \beta y_{m+1}^T \bar{H}_{m+1}^T V_{m+2}^T v_1 \\ &\quad + y_{m+1}^T \bar{H}_{m+1}^T V_{m+2}^T V_{m+2}\bar{H}_{m+1}y_{m+1} \\ &= \beta^2 - \beta h^T y_{m+1} - (\beta y_{m+1}^T \bar{H}_{m+1}^T V_{m+2}^T v_1 \\ &\quad - \beta y_{m+1}^T \bar{H}_{m+1}^T V_{m+2}^T V_{m+2}\bar{H}_{m+1} (\bar{H}_{m+1}^T \bar{H}_{m+1})^{-1} \bar{H}_{m+1}^T e_1) \\ &= \beta^2 - \beta h^T y_{m+1} = \beta(\beta - h^T y_{m+1}). \end{aligned}$$

At the same time, because

$$v_1^T r_{m+1} = v_1^T (\beta v_1 - V_{m+2}\bar{H}_{m+1}y_{m+1}) = \beta - h^T y_{m+1}$$

(h^T indicates a vector formed by the first line elements in \bar{H}_{m+1}), follows

$$\cos \langle r_m, r_{m+1} \rangle = \frac{\beta - h^T y_{m+1}}{\sqrt{\beta(\beta - h^T y_{m+1})}} = \frac{\|r_{m+1}\|_2}{\|r_m\|_2}. \quad (8)$$

□

Theorem 5. For the VRP-GMRES(m) algorithm, let $E = \|r_m\|_2$, $\bar{E} = \|r_{m+1}\|_2$, and therefore $\bar{E} < E$.

Proof of Theorem 5. In Equation (8), $\cos \langle r_m, r_{m+1} \rangle \leq 1$, namely,

$$\frac{\|r_{m+1}\|_2}{\|r_m\|_2} \leq 1.$$

Because $h^T y_{m+1} \neq 0$, it follows

$$\bar{E} = \|r_{m+1}\|_2 < \|r_m\|_2 = E. \quad (9)$$

□

4. Numerical Experiments

In this section, three different types of numerical experiments are given to illustrate the validity and feasibility of the VRP-GMRES(m) algorithm. In Example 1 and Example 2, take $x^{(0)} = (0, 0, \dots, 0)^T \in R^{(n-1)^2}$ as the initial value and $\varepsilon = 1 \times 10^{-8}$ as the convergence criterion.

Example 1. Consider the following one-dimensional Wave equation:

$$\begin{cases} u_{tt} = 4u_{xx}, & 0 < x < 1, \quad 0 < t < 0.5, \\ u(0, t) = u(1, t) = 0, & 0 \leq t \leq 0.5, \\ u(x, 0) = f(x) = \sin(\pi x) + \sin(2\pi x), & 0 \leq x \leq 1, \\ u_t(x, 0) = g(x) = 0, & 0 \leq x \leq 1. \end{cases} \quad (10)$$

As is shown in Figure 1a, the exact solution is $u(x, t) = \sin(\pi x) \cos(2\pi t) + \sin(2\pi x) \cos(4\pi t)$, where $u(x, t)$ indicates the amplitude. For the problem expressed by Equation (10), n equal divisions along the x -direction and t -direction can be obtained by the central difference method.

$$Ax = b, \quad A \in R^{(n-1)^2 \times (n-1)^2}, \quad x, b \in R^{(n-1)^2}. \quad (11)$$

Equation (11) is solved by the VRP-GMRES(m) algorithm, and the numerical results are shown in Figure 1b. From Figure 1, the numerical solution is consistent with the exact solution.

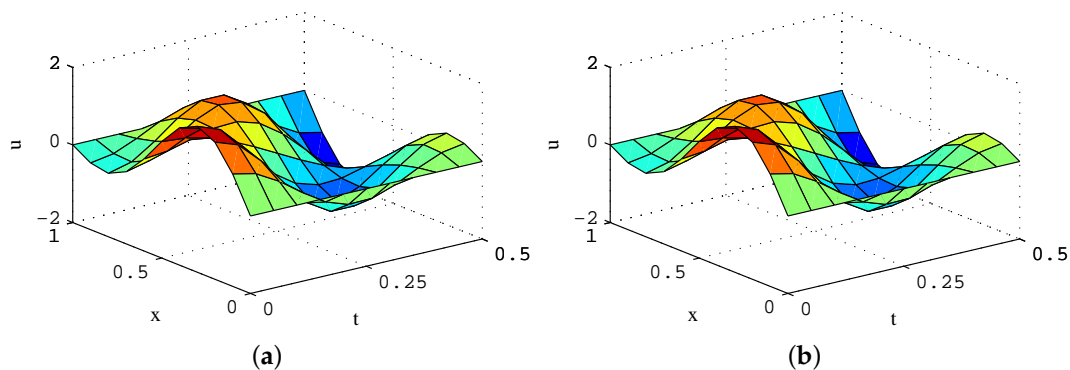


Figure 1. Amplitude distribution ($n = 10, m = 7$). (a) exact solution; and (b) numerical solution.

Equation (11) is solved by the GMRES(m) algorithm and VRP-GMRES(m) algorithm, respectively. When $n = 10, m = 7$, the condition number is 56.5079. For the GMRES(m) algorithm, iterative stagnation occurs after 9 iterations, and the residual norm is 1.4099. However, for the VRP-GMRES(m) algorithm, the residual norm has reduced to 1.4230×10^{-14} after only 3 iterations, as is shown in Figure 3. The comparison of absolute errors in the iterative process is shown in Figures 2 and 3.

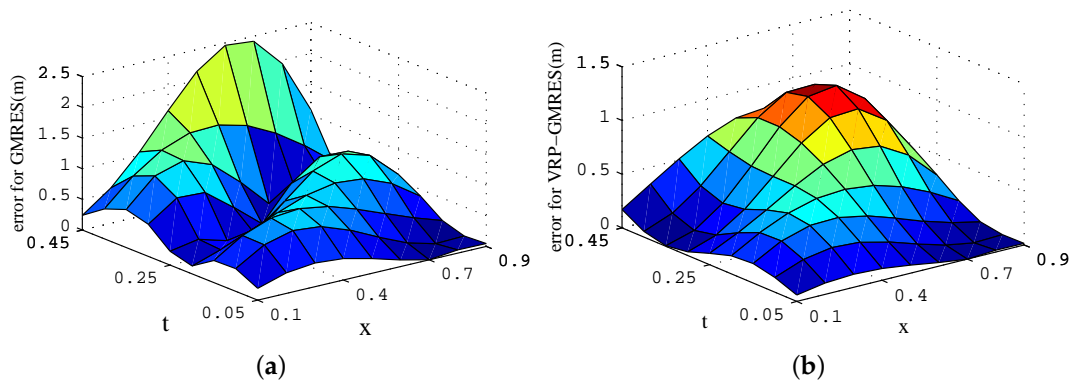


Figure 2. Absolute errors after two iterations. (a) generalized minimal residual method (GMRES)(m) algorithm; (b) Variable Restart Parameter(VRP)-GMRES(m) algorithm.

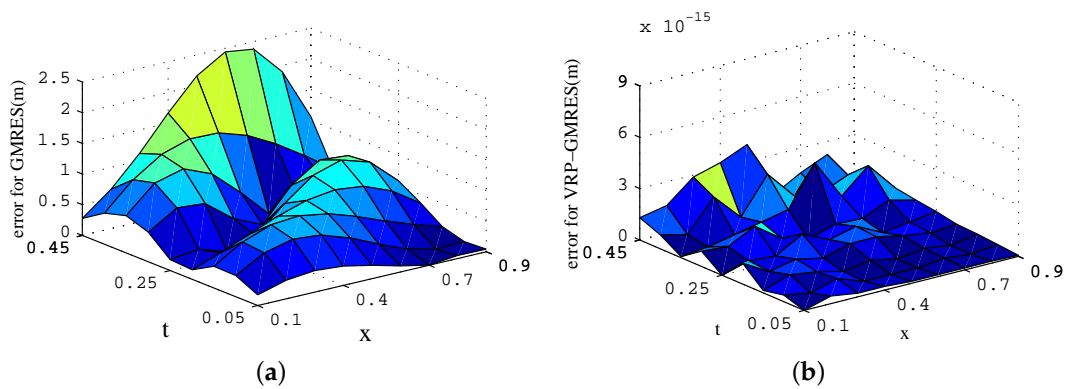


Figure 3. Absolute errors after three iterations. (a) GMRES(m) algorithm; (b) VRP-GMRES(m) algorithm.

From Figures 2 and 3, for the same number of iterations, we can see that the absolute errors generated from the VRP-GMRES(m) algorithm are much smaller than those from the GMRES(m) algorithm. With the increase in the number of iterations, the errors from the VRP-GMRES(m) algorithm reduces much faster. It shows that the new algorithm not only has higher computational accuracy, but can also speedily converge.

The parameters n and m are taken as different values and the calculation results from the two algorithms are compared. Tables 1–4 show the iteration times, computation time, computational accuracy and convergence, respectively. From Tables 1–3, we can see that the new algorithm is fast convergent, stable, highly accurate and effective. From Tables 1–4, the selection of parameter m is very important for the GMRES(m) algorithm, and the improper parameter m will result in an algorithm failure. However, it has no effect on the VRP-GMRES(m) algorithm because the variable restart parameter m can partly reduce the sensitivity of m .

Table 1. Comparison of iteration times for the two algorithms.

n	m	GMRES(m)	VRP-GMRES(m)
10	8	11	2
20	19	3	2
30	29	3	3
40	39	15	12
50	54	79	14
60	85	164	14

Table 2. Comparison of computation times for the two algorithms.

n	m	GMRES(m)	VRP-GMRES(m)
10	8	0.029415 s	0.018254 s
20	19	0.028982 s	0.013442 s
30	29	0.092144 s	0.086172 s
40	39	1.381825 s	0.847108 s
50	54	23.539635 s	4.683026 s
60	85	157.340531 s	14.662790 s

Table 3. Comparison of computational accuracy for the two algorithms.

n	m	GMRES(m)	VRP-GMRES(m)
10	8	2.1069×10^{-9}	2.9235×10^{-14}
20	19	1.1751×10^{-15}	1.1984×10^{-15}
30	29	2.0093×10^{-12}	5.4751×10^{-14}
40	39	2.4647×10^{-9}	$2.1327e \times 10^{-10}$
50	54	3.3092×10^{-11}	2.0783×10^{-11}
60	85	2.0593×10^{-9}	6.5913×10^{-13}

Table 4. Comparison of convergence for the two algorithms.

n	m	GMRES(m)	VRP-GMRES(m)
10	8	stagnated	Convergent
20	19	stagnated	Convergent
30	29	stagnated	Convergent
40	39	stagnated	Convergent
50	54	stagnated	Convergent
60	85	stagnated	Convergent

In fact, the condition number of coefficient matrix A keeps growing with the increase of n . When $n = 60$, it reaches 2.1098×10^3 , which makes Equation (11) become severely ill-conditioned. From Table 1, under the same accuracy, the number of iterations for the GMRES(m) algorithm is 11.7 times larger than that for the VRP-GMRES(m) algorithm, and the computation time is 10.7 times larger. With the increase of computing scale, the VRP-GMRES(m) algorithm will be much more effective, and its engineering application prospect is much wider.

Example 2. Consider the following two-dimensional Poisson equation:

$$\begin{cases} u_{xx} + u_{yy} = 2(3x + x^2 + y^2), & (x, y) \in \Omega, \\ u(x, y) = x^2(x + y^2) + 2, & (x, y) \in \partial\Omega, \end{cases} \quad (12)$$

where $\Omega = \{(x, y) | 0 \leq x \leq 1, 0 \leq y \leq 1\}$.

The exact solution is $u(x, y) = x^2(x + y^2) + 2$, which indicates a temperature distribution function. The temperature distribution is shown in Figure 4a. For the problem expressed by Equation (12), n equal divisions along the x -direction and t -direction can be obtained by a five-point difference scheme. After discretization, a linear equation can be obtained, which is expressed as follows:

$$Ax = b, \quad A \in R^{(n-1)^2 \times (n-1)^2}, \quad x, b \in R^{(n-1)^2}. \quad (13)$$

Take $n = 40, m = 10$, Equation (13) is solved by the VRP-GMRES(m) algorithm, and the temperature distribution is shown in Figure 4b. From Figure 4, the numerical solution is consistent with the exact solution.

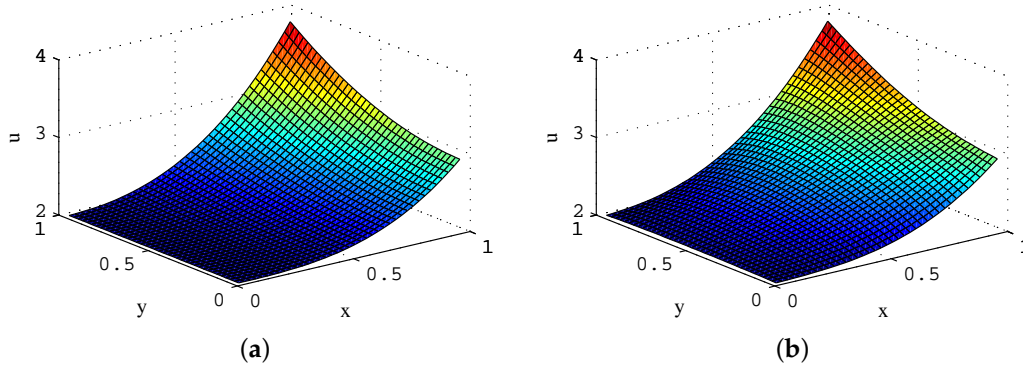


Figure 4. Temperature distribution ($n = 40, m = 10$). (a) exact solution; and (b) numerical solution.

When $m = 10$, Equation (13) is solved by the GMRES(m) algorithm and VRP-GMRES(m) algorithm, respectively. With the increase of the computational scale, there are few changes in the iteration times for the VRP-GMRES(m) algorithm, and it is much lower than that for the GMRES(m) algorithm, as shown in Figure 5a. Thus, the new algorithm has higher computational efficiency. At the same time, it has higher computational accuracy, as is shown in Figure 5b. In addition, the total computation time for the VRP-GMRES(m) algorithm is much less than that for the GMRES(m) algorithm, as is shown in Table 5.

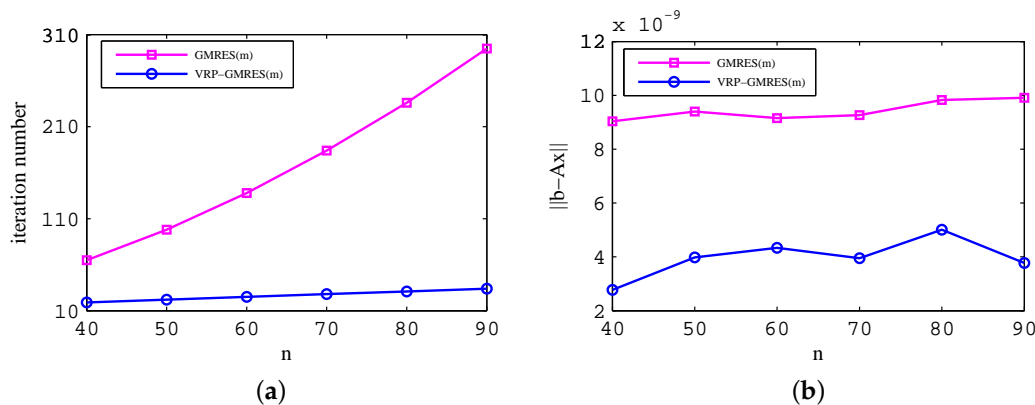


Figure 5. Comparison of the computational efficiency and accuracy under different meshes for the two algorithms ($m = 10$). (a) comparison of computational efficiency; and (b) comparison of computational accuracy.

Table 5. Comparison of computation times under different meshes for the two algorithms ($m = 10$).

n	GMRES(m)	VRP-GMRES(m)
40	1.580847 s	0.829866 s
50	5.702555 s	2.445235 s
60	16.965306 s	2.445235 s
70	43.204335 s	14.072318 s
80	100.077638 s	27.731674 s
90	189.671624 s	51.908418 s

Example 3. Consider an elastic body A, B, C (with sides of 5 mm, 4 mm, 3 mm) in contact with each other. The model and the discrete meshes are shown in Figure 6. The discrete data are shown in Table 6. Bodies A, B and C are of the same material with Young modulus $E = 210 \text{ Gpa}$ Poisson ratio $\nu = 0.3$, and the friction coefficient $f = 0.2$. For body C, a uniform load $P = 100 \text{ Mpa}$ is applied to the top surface. The total load is divided into six steps, and the contact tolerance is 0.001 mm.

In the example, the VRP-GMRES(m) algorithm is used as a fast solver for the FM-BEM, and some results are shown in Figure 7, which are consistent with those obtained by the GMRES(m) algorithm. The iteration times by the GMRES(m) algorithm and VRP-GMRES(m) algorithm are shown in Table 7. From Table 7, we can see that there are few changes in the iteration times for the VRP-GMRES(m) algorithm. However, the relative error is quite small for the pressure of the same node, as is shown in Figure 8.

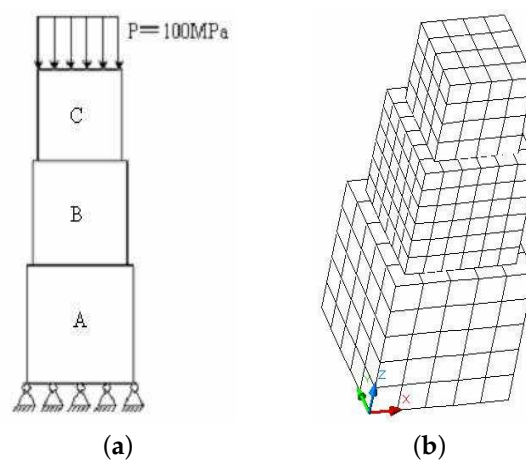


Figure 6. Calculation model and discrete meshes.

Table 6. Discrete data.

	Body A	Body B	Body C	Sum
Node number	152	218	98	468
Element number	150	216	96	462
Contact nodes	36	49	25	110
Contact elements	25	36	16	77
Degree number	648	801	369	1818

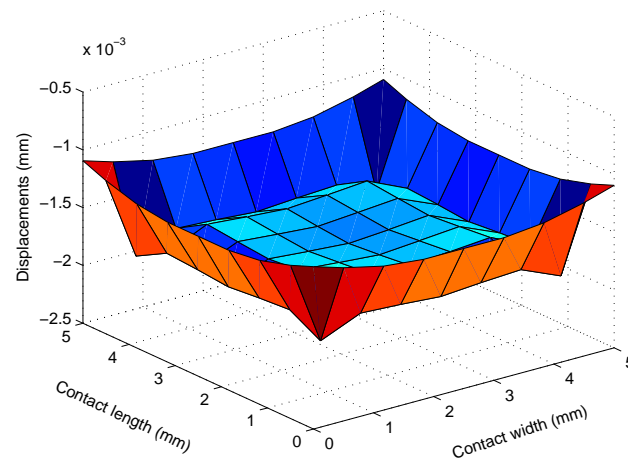


Figure 7. Displacements on the contact surface for body A ($z = 5$ mm).

Table 7. Comparison of iteration times for the two algorithms.

m	9	11	13	15	17	19
GMRES(m)	192	173	93	67	55	37
VRP-GMRES(m)	26	26	26	26	26	26

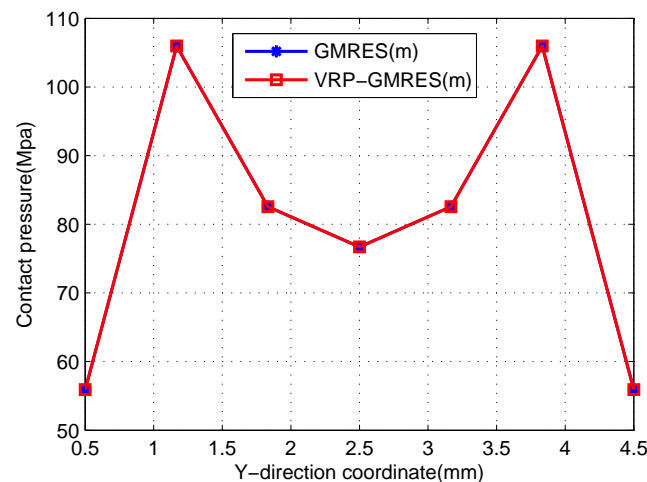


Figure 8. Comparison of pressure using two algorithms ($x = 2.5$ mm, $z = 5$ mm).

From Table 7 and Figure 8, the VRP-GMRES(m) algorithm is more rapidly and stably convergent than the GMRES(m) algorithm.

5. Conclusions

In this paper, an iterative VRP-GMRES(m) algorithm is proposed for the solution of linear equations, which can be used as an improved solver of the FM-BEM and can be used to solve many mathematical, mechanical, physical and engineering problems. Through theoretical analysis, the new algorithm is proved to be rapidly convergent with higher computational accuracy. Numerical experiments show that the presented algorithm not only has higher computational efficiency and accuracy, but also is much more stable. The results of the last example also prove that it is highly

efficient and rapidly convergent for the elastic problems. On the whole, the new algorithm has extensive prospects in the FM-BEM field and other large-scale scientific and engineering computing.

Acknowledgments: This work is supported by National Natural Science Foundation of China (No. 11301459) and Natural Science Foundation of Hebei Province (No. A2015203121).

Conflicts of Interest: The authors declare no conflict interest.

References

1. Greengard, L.; Rokhlin, V. A fast algorithm for particle simulations. *J. Comput. Phys.* **2001**, *73*, 325–348.
2. Rokhlin, V. A fast algorithm for the discrete Laplace transformation. *J. Complex.* **1988**, *4*, 12–32.
3. White, C.A.; Head-Gordon, M. Derivation and efficient implementation of the fast multipole method. *J. Chem. Phys.* **1994**, *101*, 6593–6605.
4. White, C.A.; Johnson, B.G.; Gill, P.M.W.; Head-Gordon, M. The continuous fast multipole method. *Chem. Phys. Lett.* **1994**, *230*, 8–16.
5. White, C.A.; Head-Gordon, M. Rotating around the quartic angular momentum barrier in fast multipole method calculations. *J. Chem. Phys.* **1996**, *105*, 5061–5067.
6. Beatson, R.; Greengard, L. *A Short Course on Fast Multipole Methods. Wavelets Multilevel Methods & Elliptic Pdes*; Oxford University Press: Oxford, UK, 1997.
7. Cheng, H.; Greengard, L.; Rokhlin, V. A Fast Adaptive Multipole Algorithm in Three Dimensions. *J. Comput. Phys.* **1999**, *155*, 468–498.
8. Gu, Y.; Chen, W.; Gao, H.; Zhang, C. A meshless singular boundary method for three-dimensional elasticity problems. *Int. J. Numer. Methods Eng.* **2016**, *107*, 109–126.
9. Wang, Z.; Gu, Y.; Chen, W. Fast-multipole accelerated regularized meshless method for large-scale isotropic heat conduction problems. *Int. J. Heat Mass Transf.* **2016**, *101*, 461–469.
10. Shen, L.; Liu, Y.J. An adaptive fast multipole boundary element method for three-dimensional acoustic wave problems based on the Burton-Miller formulation. *Comput. Mech.* **2007**, *12*, 554–561.
11. Wang, H.; Yao, Z. Large Scale Analysis of Mechanical Properties in 3-D Fiber-Reinforced Composites Using a New Fast Multipole Boundary Element Method. *J. Tsinghua Univ. (Sci. Technol.)* **2007**, *12*, 461–472.
12. Wang, H.; Yao, Z. Application of a new fast multipole BEM for simulation of 2D elastic solid with large number of inclusions. *Acta Mech. Sin.* **2004**, *20*, 613–622.
13. Liu, Y. *Fast Multipole Boundary Element Method: Theory and Applications in Engineering*; Cambridge University Press: Cambridge, UK, 2009.
14. Engheta, N.; Murphy, W.D.; Rokhlin, V.; Vassiliou, M.S. The fast multipole method (FMM) for electromagnetic scattering problems. *IEEE Trans. Antennas Propag.* **1992**, *40*, 634–641.
15. Liu, Y.J.; Nishimura, N.; Yao, Z.H. A fast multipole accelerated method of fundamental solutions for potential problems. *Eng. Anal. Bound. Elem.* **2005**, *29*, 1016–1024.
16. Greengard, L.; Rokhlin, V. A fast algorithm for particle simulations. *J. Comput. Phys.* **2001**, *73*, 325–348.
17. Philippeon, B.; Reichel, L. The generation of Krylov subspace bases. *Appl. Numer. Math.* **2012**, *62*, 1171–1186.
18. Luo, D.M.; Chen, P.J.; Wu, X.M. Application of GMRES algorithm to hovering rotor simulation. *Kongqi Donglixue Xuebao/Acta Aerodyn. Sin.* **2012**, *30*, 471–476.
19. Dai, Y.Z.; Song, J.Z.; Ren, H.L.; Li, H. Application of GMRES to the hydroelastic analysis of large offshore structure. *Ocean Eng.* **2003**, *21*, 15–22.
20. Pu, B.Y.; Huang, Y.Z.; Wen, C. A preconditioned and extrapolation-accelerated GMRES method for PageRank. *Appl. Math. Lett.* **2014**, *37*, 95–100.
21. Saad, Y.; Schultz, M.H. GMRES a Generalized Minimal Residual Algorithm for Solving Non-symmetric Linear Systems. *Siam J. Sci. Stat. Comput.* **2006**, *7*, 856–869.
22. Ayachour, E.H. A fast implementation for GMRES method. *J. Comput. Appl. Math.* **2003**, *159*, 269–283.
23. Jose, E.P.; Alex, A.P.; Ricardo, P.; Carlos, P.P. Making use of BDE-GMRES methods for solving short and long-term dynamics in power systems. *Int. J. Electr. Power Energy Syst.* **2013**, *45*, 293–302.
24. Yin, J.F.; Hayami, K. Preconditioned GMRES methods with incomplete Givens orthogonalization method for large sparse least-squares problems. *J. Comput. Appl. Math.* **2009**, *226*, 177–186.

25. Lin, F.R.; Yang, S.W.; Jin, X.Q. Preconditioned iterative methods for fractional diffusion equation. *J. Comput. Phys.* **2014**, *256*, 109–117.
26. Liang, Y.; Szularz, M.; Yang, L.T. Finite-element-wise domain decomposition iterative solvers with polynomial preconditioning. *Math. Comput. Model.* **2013**, *58*, 421–437.
27. Baker, A.H. *On Improving the Performance of the Linear Solver Restarted Gmres*; University of Colorado at Boulder: Boulder, CO, USA, 2003.
28. Baker, A.H.; Jessup, E.R.; Kolev, T.V. A simple strategy for varying the restart parameter in GMRES(m). *J. Comput. Appl. Math.* **2009**, *230*, 751–761.
29. Peairs, L.; Chen, T.Y. Using reinforcement learning to vary the m in GMRES(m). *Procedia Comput.* **2011**, *4*, 2257–2266.
30. Essai, A. Weighted FOM and GMRES for solving nonsymmetric linear systems. *Numer. Algorithms* **1998**, *18*, 227–292.
31. Cai, D.Y.; Bai, F.S. *Advanced Numerical Analysis*; Tsinghua University Press: Beijing, China, 1997.



© 2016 by the authors; licensee *Preprints*, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).