Linear-System Solver for EFG-Type Saddle-Point Problem without Using *QR* Decomposition*)

Atsushi KAMITANI, Teruou TAKAYAMA, Ayumu SAITOH and Hiroaki NAKAMURA[1,2)]

Yamagata University, Yamagata 992-8510, Japan

[1)]*National Institute for Fusion Science, 322-6 Oroshi-cho, Toki 509-5292, Japan*

[2)]*Nagoya University, Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan*

(Received 19 December 2021 / Accepted 16 February 2022)

A novel method is proposed for solving an EFG-type Saddle-Point (EFG-SP) problem. Although the null-space method and the variable-reduction method (VRM) were developed as a solver of a saddle-point problem, both methods are extremely time-consuming in solving an EFG-SP problem. This is attributable to the *QR* decomposition that is indispensable for both methods. For the purpose of resolving this problem, the improved Variable-Reduction Method (iVRM) is formulated without using the *QR* decomposition. A numerical code has been developed for solving an EFG-SP problem with the iVRM, the VRM and the ICCG method. By means of the code, the performance of the three methods is investigated numerically. The results of computations show that, from the standpoint of convergence property and computational speed, the iVRM is even superior to either of the VRM and the ICCG method.

Keywords: computer simulation, element-free Galerkin method, Krylov space method, meshless approach, *QR* decomposition, saddle-point problem

DOI: 10.1585/pfr.17.2403014

1. Introduction

The Element-Free Galerkin (EFG) method [1] was developed as a meshless approach for solving a boundary-value problem. It has been widely applied to various fields in the simulation science: electromagnetic field computation, structural analysis, fracture analysis and so on.

If the EFG method is applied to a boundary-value problem, the following saddle-point problem [2] is obtained:

$$\begin{bmatrix} B & C \\ C^T & O \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{d} \end{bmatrix}. \tag{1}$$

Moreover, the submatrix B is singular and sparse. Throughout the present study, such a linear system is called an EFG-type Saddle-Point (EFG-SP) problem.

For the purpose of solving a saddle-point problem with a singular submatrix B, the null-space method [2, 3] and the Variable-Reduction Method (VRM) [4, 5] have been so far proposed. In the null-space method, the orthonormal basis of $\mathrm{Ker}\,(C^T)$ needs to be determined. On the other hand, in the VRM, the orthonormal basis of $\mathrm{Im}\,C$ needs to be determined. Therefore, costly *QR* decompositions [6] are indispensable for both methods.

The purpose of the present study is to reformulate the VRM without using the *QR* decomposition. In addition, we numerically investigate applicability of the resulting method to an EFG-SP problem.

author's e-mail: kamitani@yz.yamagata-u.ac.jp

*) This article is based on the presentation at the 30th International Toki Conference on Plasma and Fusion Research (ITC30).

2. EFG-Type Saddle-Point Problem

In this section, after an EFG-SP problem is mathematically defined, we investigate whether the ICCG method can be applied to the problem or not. In the following, an n-dimensional real vector space is denoted by $\mathbb{R}^n$.

If $B \in \mathbb{R}^{N\times N}$, $C \in \mathbb{R}^{N\times K}$ and $\boldsymbol{c} \in \mathbb{R}^N$, $\boldsymbol{d} \in \mathbb{R}^K$ are given matrices and given vectors, respectively, a linear system of the form (1) is generally called a saddle-point problem. Here, $\boldsymbol{u} \in \mathbb{R}^N$ and $\boldsymbol{\lambda} \in \mathbb{R}^K$ are both unknown vectors. Besides, the coefficient matrix in (1) is called a saddle-point matrix.

Let us consider the additional five conditions:

i) The submatrix B is symmetric and positive semidefinite.
ii) The submatrix C has a full-column rank.
iii) $\mathrm{Ker}\,B \cap \mathrm{Ker}\,(C^T) = \{\boldsymbol{0}\}$.
iv) There exists a real number q such that $0 \le q < 1$ and $K = O(N^q)$.
v) The numbers of nonzero elements in B and C are $O(N)$ and $O(K)$, respectively.

An EFG-SP problem is defined to be a saddle-point problem satisfying the above five conditions. Note that, when i) and ii) are fulfilled, the condition iii) is the necessary and sufficient condition for the saddle point matrix in (1) to be nonsingular.

As is well known, a saddle-point problem is difficult to solve numerically. In order to investigate this tendency, we apply the ICCG method directly to an EFG-SP problem

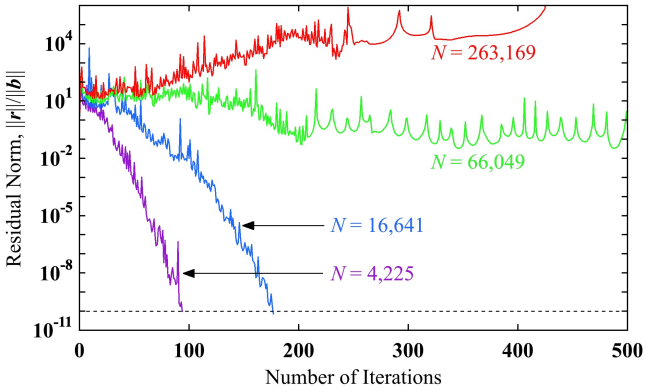

Fig. 1 Residual histories of the ICCG method. Here, not only the parameters of the EFG method but also the details of a 2D Poisson problem are described in Sec. 4.

that is obtained by discretizing a two-dimensional (2D) Poisson problem with the EFG method. In the EFG-SP problem, N denotes the number of nodes. The resulting residual histories[1] of the ICCG method are depicted in Fig. 1. The ICCG method shows convergence both for N = 4,225 and for N = 16,641, whereas any tendency of convergence cannot be found both for N = 66,049 and for N = 263,169. In this sense, an increase in the number N of nodes will remarkably degrade the convergence property of the ICCG method. This result suggests that a large-scale EFG-SP problem cannot be solved with the ICCG method.

3. Improvement of Variable-Reduction Method

As a method for solving a saddle-point problem satisfying only the conditions, i), ii), iii), and iv), the authors developed the VRM. In other words, sparsity of submatrices, B and C, is not taken into consideration in the VRM. In this section, the basic idea of the VRM is briefly explained and, subsequently, the VRM is reformulated without using the QR decomposition.

3.1 VRM

In the VRM, the vector $\boldsymbol{\lambda}$ is eliminated from a saddle-point problem (1) by means of the following 3 steps. The QR decomposition of submatrix C is first computed as

$$C = QRP^T, \tag{2}$$

where $Q \in \mathbb{R}^{N\times K}$ is a matrix such that $Q^TQ = I$. Here, I denotes an identity matrix. In addition, $R \in \mathbb{R}^{K\times K}$ and $P \in \mathbb{R}^{K\times K}$ are an upper triangular matrix and a permutation matrix, respectively. Next, by using Q, two orthogonal projection matrices, F and U, are determined by

$$F = QQ^T, \tag{3}$$

$$U = I - F. \tag{4}$$

Incidentally, matrices, F and U, express orthogonal projections onto $\mathrm{Im}\,C$ and $(\mathrm{Im}\,C)^\perp$, respectively. Finally, vector $\boldsymbol{\lambda}$ is eliminated from the saddle-point problem (1) by using these two projection matrices. Consequently, we obtain the following linear system:

$$B^\dagger \boldsymbol{u} = \boldsymbol{c}^\dagger. \tag{5}$$

Here, $B^\dagger \in \mathbb{R}^{N\times N}$ and $\boldsymbol{c}^\dagger \in \mathbb{R}^N$ are defined by

$$B^\dagger \equiv U^T B U + F, \tag{6}$$

$$\boldsymbol{c}^\dagger \equiv U^T[\boldsymbol{c} - B\boldsymbol{d}^\dagger] + \boldsymbol{d}^\dagger, \tag{7}$$

where $\boldsymbol{d}^\dagger$ is given by

$$\boldsymbol{d}^\dagger = QR^{-T}P^T\boldsymbol{d}. \tag{8}$$

The actual calculation of the VRM is composed of the following two steps:

Step A Vectors, $\boldsymbol{c}^\dagger$ and $\boldsymbol{d}^\dagger$, are calculated.
Step B The linear system (5) is solved with the conjugate gradient (CG) method to get $\boldsymbol{u}$.

Step A is called an overhead, hereafter.

As is described in [4, 5], the VRM is considerably effective to a saddle-point problem with a dense submatrix B. In contrast, it should not be applied to an EFG-SP problem in which B and C are both sparse. The reason for this can be explained by means of operation counts. For example, for the case with a 2D boundary-value problem, operation count of the QR decomposition and that of each iteration in the CG method are estimated as $O(N^2)$ and $O(N^{3/2})$, respectively (see Appendix A). Hence, the VRM becomes an inefficient N^m algorithm with $m \geq 2$ for this case[2]. This is why the VRM cannot be recommended as a solver of an EFG-SP problem.

3.2 Reformulation of VRM

As mentioned above, the VRM is inappropriate for a solver of an EFG-SP problem. The main reason for this is that the QR decomposition can be a rate-determining step in the VRM. In order to resolve this problem, we reformulate the VRM without using the QR decomposition. The resulting method is called the improved Variable-Reduction Method (iVRM).

In the iVRM, the orthogonal projection matrix F and the vector $\boldsymbol{d}^\dagger$ are calculated by using the following equations:

$$F = C(C^TC)^{-1}C^T, \tag{9}$$

[1] Let $\boldsymbol{x}'$ be an approximate solution of the linear system $A\boldsymbol{x} = \boldsymbol{b}$. Then, the vector $\boldsymbol{r} \equiv \boldsymbol{b} - A\boldsymbol{x}'$ and its norm $\|\boldsymbol{r}\|$ are called a residual vector and a residual norm, respectively. Especially when the linear system is solved with an iterative method such as the ICCG method and the CG method, dependence of a residual norm on the number of iterations is called a residual history. In such an iterative method, the iteration process is stopped if the termination condition, $\|\boldsymbol{r}\|/\|\boldsymbol{b}\| \leq \epsilon$, is fulfulled. Throughout the present study, ϵ is called a termination determinant and the minimum number of iterations satisfying the termination condition is called a convergent iteration number.

[2] Similarly, for the case with a 3D boundary-value problem, the VRM becomes an inefficient N^m algorithm with $m \geq 7/3$.

$$\boldsymbol{d}^{\dagger} = C(C^T C)^{-1}\boldsymbol{d}. \tag{10}$$

In other words, by replacing (3) and (8) with (9) and (10), respectively, the iVRM can be obtained. Note that any QR decompositions are not included in the iVRM.

The actual calculation of the iVRM is composed of the same two steps as that of the VRM. In Step B, each iteration in the CG method contains one multiplication of matrix $B^{\dagger}$ and an N-dimensional vector. In addition, the matrix-vector multiplication requires two multiplications of matrix F and an N-dimensional vector. As is apparent from (9) , the linear system:

$$C^T C\boldsymbol{z} = \boldsymbol{v}, \tag{11}$$

must be solved to perform the multiplication of matrix F and an N-dimensional vector. In the present study, the CG method is also applied to the solution of (11). In Appendix B, we describe how the computational cost for the iVRM is influenced by solving (11) with the CG method.

For the case with an EFG-SP problem obtained from a 2D boundary-value problem, let us compare operation count for the iVRM with that for the VRM. As mentioned above, in the VRM, operation counts for the overhead and for each iteration in the CG method are $O(N^2)$ and $O(N^{3/2})$, respectively. On the other hand, in the iVRM, operation counts for the overhead and for each iteration are both $O(N)$. In this sense, the computational cost for the iVRM is considerably reduced as compared with that for the VRM. A detailed explanation on operation counts for the iVRM/VRM is given in Appendix A.

4. Performance Evaluation

4.1 Test problem

As a test problem for evaluating the performance of the iVRM and the VRM, we adopt the following 2D Poisson problem on the domain Ω bounded by a Jordan curve $\partial\Omega$:

$$-\nabla^2 u = p \qquad \text{in} \quad \Omega, \tag{12}$$

$$u = \bar{u} \qquad \text{on} \quad \partial\Omega, \tag{13}$$

where p and $\bar{u}$ are known functions in Ω and on $\partial\Omega$, respectively.

After discretizing the above Poisson problem by means of the collocation EFG [7] with N nodes including K boundary nodes, we get a saddle-point problem (1). Here, vectors, $\boldsymbol{c}$ and $\boldsymbol{d}$, are given by

$$\boldsymbol{c} = \sum_{i=1}^{N}\left(\iint_{\Omega}\phi_i(\boldsymbol{x})\,p(\boldsymbol{x})\,d^2\boldsymbol{x}\right)\boldsymbol{e}_i, \tag{14}$$

$$\boldsymbol{d} = \sum_{k=1}^{K}\bar{u}(\boldsymbol{y}_k)\,\boldsymbol{e}_k^*, \tag{15}$$

whereas matrices, B and C, are given by

$$B = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\iint_{\Omega}\nabla\phi_i\cdot\nabla\phi_j\,d^2\boldsymbol{x}\right)\boldsymbol{e}_i\boldsymbol{e}_j^T, \tag{16}$$

$$C = \sum_{i=1}^{N}\sum_{k=1}^{K}\phi_i(\boldsymbol{y}_k)\,\boldsymbol{e}_i\boldsymbol{e}_k^{*T}. \tag{17}$$

In addition, $\{\phi_i(\boldsymbol{x})\}_{i=1}^{N}$ denotes shape functions of the MLS approximation [1]. Moreover, $\boldsymbol{y}_k$ is the kth boundary node. Furthermore, the standard bases of $\mathbb{R}^N$ and $\mathbb{R}^K$ are denoted by $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_N\}$ and $\{\boldsymbol{e}_1^*, \boldsymbol{e}_2^*, \cdots, \boldsymbol{e}_K^*\}$, respectively. After a straight forward calculation, we can easily prove that (16) and (17) satisfy all the conditions, i)–v). Hence, the obtained saddle-point problem is an EFG-SP problem.

4.2 Numerical experiments

Let us compare the performance of the iVRM with that of the VRM and the ICCG method. To this end, the iVRM, the VRM and the ICCG method are implemented to the numerical code in which the above Poisson problem is solved with the EFG method. By using the code, we investigate the performance of the iVRM, the VRM and the ICCG method as a solver of an EFG-SP problem. Numerical experiments are carried out under the computational environment listed in Table 1.

Throughout the present study, the domain Ω is assumed to be a square region $(0,1)\times(0,1)$. In addition, functions, p and $\bar{u}$, are assumed so that an analytic solution of the 2D Poisson problem may be $u = \exp[-(x^2+y^2)]$.

Parameters in the EFG method are assumed as follows. First, after the square region Ω is equally divided in the x- and y-directions, the resulting grid points in $\Omega\cup\partial\Omega$ are assumed as nodes. Second, a linear basis function and the exponential-type weight function are adopted in the MLS approximation. Furthermore, the support radius R of the shape functions is assumed as $R = 1.5\,h$, where h is a distance between the nearest two nodes. Incidentally, the value of the termination determinant ϵ is assumed as $\epsilon = 10^{-10}$ in solving not only (5) but also (11) with the CG method.

We first investigate the convergence property of the ICCG method, the VRM and the iVRM. Figure 2 shows residual histories for the three methods. We see from this figure that, from the standpoint of convergence property, both the iVRM and the VRM are even superior to the ICCG method. Incidentally, the residual history for the iVRM completely agrees with that for the VRM. This agreement is attributable to mathematical equivalence between the iVRM and the VRM.

Table 1 Computational Environment.

Item	Description
OS	Ubuntu ver. 16.04.7
CPU	Intel© Core™ i9-9900K CPU@3.60GHz
Memory	32 GB
Compiler	GNU Fortran (gfortran) ver. 5.4.0
Compiler Options	-O2

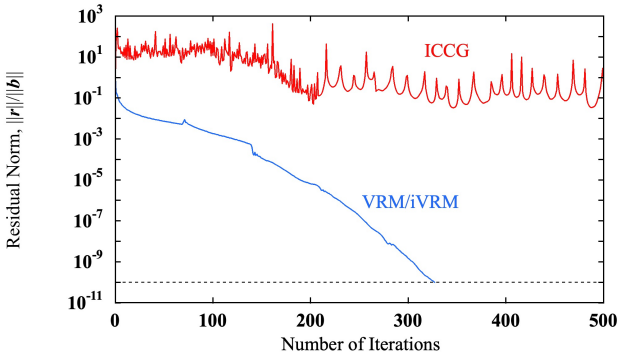

Fig. 2 Residual histories for the ICCG method, the VRM and the iVRM for the case with $N = 66{,}049$.

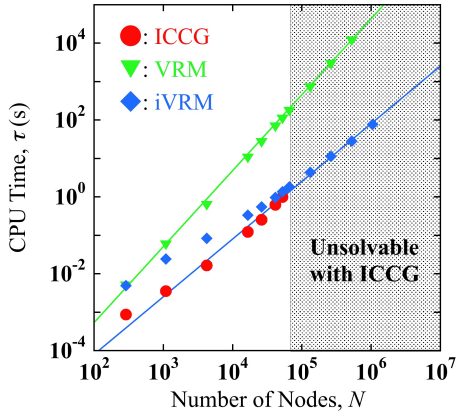

Fig. 3 Dependence of the CPU time τ on the number N of nodes.

Next, we investigate the computational speed of the ICCG method, the VRM and the iVRM. To this end, the CPU times required for the three methods are measured. The results of measurements are shown in Fig. 3. This figure indicates that, although the ICCG does not converge for the case with $N \gtrsim 6 \times 10^4$, both the VRM and the iVRM show convergence for the same case. In addition, the CPU times for the VRM and for the iVRM are roughly proportional to N^2 and $N^{1.5}$, respectively. Hence, from the standpoint of the computational speed, the iVRM is much faster than the VRM and the ICCG method.

The above results indicate that the iVRM can be a powerful tool for solving an EFG-SP problem.

5. Conclusion

We have proposed the iVRM as a solver of an EFG-SP problem. A numerical code is developed for solving an EFG-SP problem with either the iVRM, the VRM, or the ICCG method. By using the code, the performance of the iVRM is numerically evaluated as compared with not only the VRM but also the ICCG method.

Conclusions obtained in the present study are summarized as follows.

- From the viewpoint of convergence property, the iVRM/VRM is clearly superior to the ICCG method.
- From the viewpoint of the computational speed, the iVRM is even faster than either of the VRM and the ICCG method.

Therefore, we can conclude that the iVRM is highly effective to the numerical solution of an EFG-SP problem.

Acknowledgment

This work was supported in part by Japan Society for the Promotion of Science under a Grant-in-Aid for Scientific Research (C) No. 21K04016. A part of this work was also carried out with the support and under the auspices of the NIFS Collaboration Research program (NIFS21KNSS163, NIFS20KNTS069, NIFS20KKGS027).

Appendix A. Operation Counts for iVRM and VRM

In this appendix, we estimate operation counts for the iVRM and for the VRM. In the following, operation count for an algebraic operation $\alpha_{\rm op}$ is denoted by $c(\alpha_{\rm op})$.

Let us first estimate operation count for the VRM. As is apparent from (2), (7), and (8), operation count for the overhead is dominated by the following three operations: QR decomposition of C, $U^T[\boldsymbol{c} - B\,\boldsymbol{d}^\dagger]$, and $QR^{-T}P^T\,\boldsymbol{d}$. On the other hand, operation counts for the three operations are given by $c(QR \text{ decomposition of } C) = O(N^{2q+1})$, $c(U^T[\boldsymbol{c} - B\,\boldsymbol{d}^\dagger]) = O(N^{q+1})$, and $c(QR^{-T}P^T\,\boldsymbol{d}) = O(N^{q+1})$. Since operation count for the overhead is a sum of the three operations, it can be written as $c(\text{overhead}) = O(N^{2q+1})$. Similarly, operation count for each iteration in the CG method is determined as $c(\text{each iteration}) = O(N^{q+1})$.

Although operation count for the iVRM can be also estimated in the similar way, it contains the convergent iteration number m_C for solving (11). According to the theory on the Krylov space method, m_C generally satisfies $m_C = O(K^r)$, where $0 \le r \le 1$. Hence, operation counts for the overhead and for each iteration in the CG method are both given by $O(N^{\max[q(r+1),1]})$.

Especially for the case where an EFG-SP problem is obtained from a 2D boundary-value problem, $q = 1/2$ is fulfilled. In addition, the results of computations show that $r = 0$ is approximately satisfied for this case. Hence, in the VRM, operation counts for the overhead and for each iteration in the CG method are given by $O(N^2)$ and $O(N^{3/2})$, respectively. Furthermore, in the iVRM, operation counts for the overhead and for each iteration are both $O(N)$.

Appendix B. Influence of (11) on Computational Cost for iVRM

The computational cost for each iteration in the iVRM is almost occupied with the computational costs for the following two operations: (i) solution of (11) with the CG method and (ii) matrix-vector multiplication $B\boldsymbol{u}$. In this appendix, we investigate how the computational cost for

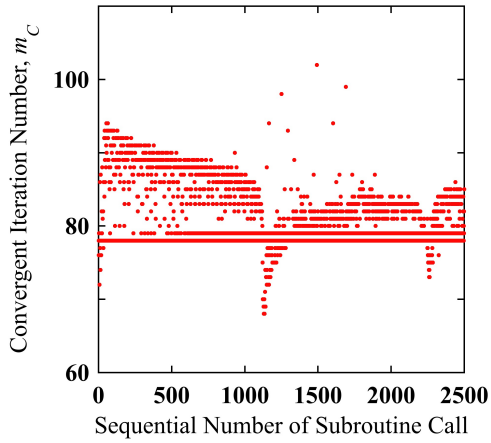

Fig. B1 Dependence of the convergence iteration number m_C on the sequential number of subroutine call for the case with N = 1,050,625.

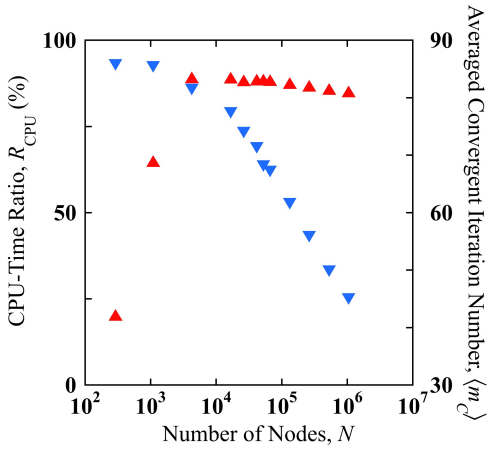

Fig. B2 The CPU-time ratio $R_{\rm CPU}$ and the averaged convergent iteration number $\langle m_C \rangle$ as functions of the number N of nodes. Here, the symbols, ▼ and ▲, indicate the values of $R_{\rm CPU}$ and $\langle m_C \rangle$, respectively.

the iVRM is affected by solving (11) with the CG method.

First, we investigate how a convergent iteration number m_C changes for each solution of (11). To this end, m_C is determined as a function of the sequential number of subroutine call and the results of computations are depicted in Fig. B1. This figure indicates that m_C varies from 68 to 102 with an average of 81.1. Thus, the convergent iteration number does not change drastically. In this sense, the computational cost for the solution of (11) with the CG method can be characterized by the average $\langle m_C \rangle$ of the convergent iteration number.

Next, dependence of the averaged convergent iteration number $\langle m_C \rangle$ on the number of nodes is numerically determined and the results of computations are depicted in Fig. B2. We see from this figure that $\langle m_C \rangle$ remains almost constant for the case with $N \gtrsim 4 \times 10^3$. This result means that, for this case, operation count for solving (11) with the CG method is given by $O(K) = O(N^{1/2})$. On the other hand, operation count for the matrix-vector multiplication $\boldsymbol{Bu}$ is given by $O(N)$. Therefore, an increase in N will weaken the influence of the solution of (11) with the CG method on the computational cost for the iVRM.

Finally, we quantitatively investigate the influence of the solution of (11) with the CG method on the computational cost for the iVRM. As a measure of the influence, we adopt the CPU-time ratio $R_{\rm CPU}$ defined by $R_{\rm CPU} \equiv \tau_{(11)}/\tau$. Here, $\tau_{(11)}$ denotes the accumulated CPU time for solving (11) with the CG method, whereas τ is the total CPU time for solving (5) with the iVRM. Dependence of the CPU-time ratio $R_{\rm CPU}$ on the number of nodes is numerically determined and it is also depicted in Fig. B2. The CPU-time ratio $R_{\rm CPU}$ decreases monotonously with an increase in N until it amounts down to 25% for $N = 1,050,625$. Thus, with an increase in N, the solution of (11) with the CG method have weaker effect on the computational cost for the iVRM. Therefore, we can conclude that the iVRM is suitable for a large-sized EFG-SP problem.

[1] T. Belytschko, Y.Y. Lu and L. Gu, Int. J. Numer. Meth. Eng. **37**, 229 (1994).
[2] M. Rozložník, *Saddle-Point Problems and Their Iterative Solution* (Birkhäuser, Switzerland, 2018) Chaps. 1 and 4.
[3] M. Benzi, G.H. Golub and J. Liesen, Acta Numer. **14**, 1 (2005).
[4] A. Kamitani, T. Takayama, A. Saitoh and S. Ikuno, J. Adv. Simulat. Sci. Eng. **4**, 117 (2018).
[5] A. Kamitani, T. Takayama and A. Saitoh, Int. J. Appl. Electromagn. Mech. **59**, 157 (2019).
[6] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 4th ed. (Johns Hopkins University Press, Maryland, 2013) p. 246.
[7] A. Kamitani, T. Takayama, T. Itoh and H. Nakamura, Plasma Fusion Res. **6**, 2401074 (2011).