Improved Variable-Reduction Method and Its Variant for Solving Asymmetric EFG-Type Saddle-Point Problem*)

Atsushi KAMITANI, Yuito SHINDO, Teruou TAKAYAMA and Hiroaki NAKAMURA[1,2)]

Yamagata University, Yonezawa, 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 13 December 2022 / Accepted 10 April 2023)

If a boundary-value problem is discretized with the extended Element-Free Galerkin (EFG) method, an asymmetric EFG-type Saddle-Point (EFG-SP) problem is obtained. Although the improved Variable-Reduction Method (iVRM) was originally developed as a solver for symmetric EFG-SP problems, it is extended so as to be applicable to asymmetric EFG-SP problems. As a result, not only the Asymmetric-version iVRM (AiVRM) but also its variant AiVRM2 is developed. A numerical code is developed for solving an asymmetric EFG-SP problem with the AiVRM/AiVRM2 and, by means of the code, performances of the two methods are investigated numerically. Consequently, it is found that, especially for a large-scale asymmetric EFG-SP problem, both the AiVRM and the AiVRM2 are more effective than the preconditioned Krylov subspace method.

Keywords: computer simulation, element-free Galerkin method, Krylov subspace method, meshless approach, preconditioning, saddle-point problem

DOI: 10.1585/pfr.18.2403039

1. Introduction

As meshless approaches for solving a boundary-value problem, the Element-Free Galerkin (EFG) method [1] and the eXtended Element-Free Galerkin (X-EFG) method [2] were developed. If a boundary-value problem of a certain class of partial differential equations such as the generalized Poisson equation and the Grad-Shafranov equation is discretized with the EFG method, we get a symmetric EFG-type Saddle-Point (EFG-SP) problem [1, 3] that is among symmetric saddle-point problems. However, it is difficult to solve a symmetric EFG-SP problem numerically. Although the null-space method [4, 5] was proposed for solving a symmetric saddle-point problem, it requires the costly *QR* decomposition [6, 7].

In order to resolve this problem, the authors formulated the improved Variable Reduction Method (iVRM) [3] without using any *QR* decompositions. In addition, they applied it to a symmetric EFG-SP problem originating from a two-dimensional (2D) Poisson problem. Consequently, it is found that, from the standpoint of both convergence property and computational cost, the iVRM is superior to ICCG and that the iVRM is applicable even to a large-scale problem for which ICCG does not yield a convergence solution.

On the other hand, if the X-EFG method is applied to a boundary-value problem, we obtain the following linear system [2, 8]:

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

where $B \in \mathbb{R}^{N\times N}$ and $C, D \in \mathbb{R}^{N\times K}$ are given matrices. Also, $\boldsymbol{c} \in \mathbb{R}^N$ and $\boldsymbol{d} \in \mathbb{R}^K$ are both given vectors, whereas $\boldsymbol{u} \in \mathbb{R}^N$ and $\boldsymbol{\lambda} \in \mathbb{R}^K$ are both unknown vectors. Here, N and K are two natural numbers satisfying $K < N$, and $\mathbb{R}$ denotes a set of all real numbers. Furthermore, submatrices, B, C and D, fulfill the following 6 conditions:

i) A submatrix B is singular.
ii) Both C and D have full-column ranks and satisfy $\mathrm{Im}\, C \cap \mathrm{Ker}\, D^T = \{\boldsymbol{0}\}$.
iii) The coefficient matrix in (1) is invertible.
iv) There exists a real number q such that $0 \le q < 1$ and $K = O(N^q)$.
v) Both C and D have the same nonzero-element pattern.
vi) If the numbers of nonzero elements in B, C and D are denoted by l_B, l_C and l_D, respectively, they are estimated as $l_B = O(N)$ and $l_C = l_D = O(K)$.

Throughout the present study, (1) is called an asymmetric EFG-SP problem. Since the iVRM was originally proposed as a solver for symmetric EFG-SP problems, it cannot be applied to asymmetric EFG-SP problems as it is. Moreover, any numerical methods have not been so far proposed for solving this type of a linear system.

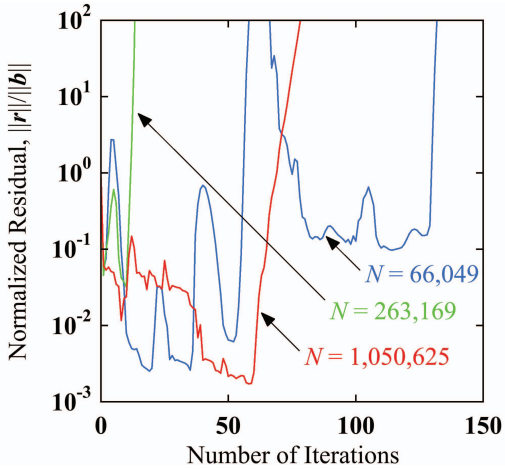
The purpose of the present study is to develop linear-system solvers for asymmetric EFG-SP problems by extending the basic idea of the iVRM. Moreover, we numerically investigate performances of the resulting solvers.

author's e-mail: kamitani@yz.yamagata-u.ac.jp
*) This article is based on the presentation at the 31st International Toki Conference on Plasma and Fusion Research (ITC31).

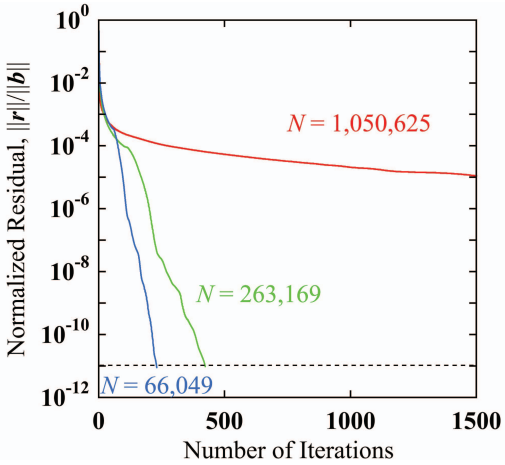
2. Direct Application of Preconditioned Krylov Subspace Method

In general, an asymmetric EFG-SP problem is difficult to solve numerically like a symmetric EFG-SP problem. For the purpose of investigating this tendency, we apply the preconditioned Krylov subspace method directly to an asymmetric EFG-SP problem. In the present study, an Incomplete LDU (ILDU) decomposition[1] is adopted as a preconditioner, whereas either BiCGSTAB or GMRES is used as the Krylov subspace method. In the following, BiCGSTAB and GMRES with an ILDU preconditioner are abbreviated as ILDU-BiCGSTAB and ILDU-GMRES, respectively.

Let us investigate a convergence property of ILDU-BiCGSTAB and ILDU-GMRES. To this end, both solvers are applied to asymmetric EFG-SP problems originating from a 2D Poisson problem. Their residual histories are depicted in Figs. 1 (a) and 1 (b). Apparently, ILDU-BiCGSTAB does not show a convergence behavior for any values of the number N of nodes. On the other hand, ILDU-GMRES is such a robust method that it can always yield a convergence solution regardless of values of N. However, its convergence property is considerably degraded with an increase in N. These results indicate that both ILDU-GMRES and ILDU-BiCGSTAB are inappropriate as solvers for large-scale asymmetric EFG-SP problems. For this reason, a high-performance solver has been desired for an asymmetric EFG-SP problem.



(a)



(b)

Fig. 1 Residual histories of (a) ILDU-BiCGSTAB and (b) ILDU-GMRES for asymmetric EFG-SP problems obtained from a 2D Poisson problem. Both the parameters used in the X-EFG method and the details of the Poisson problem are described in Section 4.1.

3. Asymmetric-Version iVRM

The iVRM is a linear-system solver for symmetric EFG-SP problems. Its basic idea is to eliminate a vector corresponding to the Lagrange multiplier from a saddle-point problem by means of orthogonal projectors [3]. In this section, we apply the basic idea to the development of the Asymmetric-version iVRM (AiVRM). To this end, vector $\boldsymbol{\lambda}$ is eliminated from (1) by using projectors. Such projectors are hereafter referred to as $\boldsymbol{\lambda}$-eliminators. In the present study, we use two types of $\boldsymbol{\lambda}$-eliminators so that two different solvers, AiVRM and AiVRM2, are obtained depending on the adopted $\boldsymbol{\lambda}$-eliminators.

3.1 AiVRM and AiVRM2

If two projection matrices, $F \equiv C(D^T C)^{-1} D^T$ and $U \equiv I - F$, are employed as $\boldsymbol{\lambda}$-eliminators, we get the following linear system:

$$B^{\dagger}\,\boldsymbol{u} = \boldsymbol{c}^{\dagger}, \tag{2}$$

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

$$B^{\dagger} \equiv U\,B\,U + F, \tag{3}$$

$$\boldsymbol{c}^{\dagger} \equiv U(\boldsymbol{c} - B\,\boldsymbol{d}^{\dagger}) + \boldsymbol{d}^{\dagger}, \tag{4}$$

$$\boldsymbol{d}^{\dagger} \equiv C(D^T C)^{-1}\,\boldsymbol{d}. \tag{5}$$

Here, F and U are projection matrices onto $\mathrm{Im}\,C$ along $(\mathrm{Im}\,D)^{\perp}$ and onto $(\mathrm{Im}\,D)^{\perp}$ along $\mathrm{Im}\,C$, respectively. Note that neither F nor U is a symmetric matrix.

On the other hand, if four orthogonal projection matrices, $F_C \equiv C(C^T C)^{-1} C^T$, $U_C \equiv I - F_C$, $F_D \equiv D(D^T D)^{-1} D^T$ and $U_D \equiv I - F_D$, are adopted as $\boldsymbol{\lambda}$-eliminators, the following linear system is obtained:

$$B^{\ddagger}\,\boldsymbol{u} = \boldsymbol{c}^{\ddagger}, \tag{6}$$

where $B^{\ddagger} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{c}^{\ddagger} \in \mathbb{R}^{N}$ are defined by

$$B^{\ddagger} \equiv U_C\,B\,U_D + F_D, \tag{7}$$

[1] Let $G(A) = \{(i,j) \in \mathbb{N}^2 : a_{ij} \neq 0\}$ be a nonzero-element pattern in a square matrix A of order n. Here, $\mathbb{N}$ denotes a set of all natural numbers. For the case where the Incomplete LU (ILU) decomposition is applied to a sparsity pattern $S = G(A) \cup \{(i,i) : i = 1, 2, \cdots, n\}$, the preconditioner is called an incomplete LDU decomposition.

$$\boldsymbol{c}^{\ddagger} \equiv U_C(\boldsymbol{c} - B\,\boldsymbol{d}^{\ddagger}) + \boldsymbol{d}^{\ddagger}, \tag{8}$$

$$\boldsymbol{d}^{\ddagger} \equiv D(D^T D)^{-1}\,\boldsymbol{d}. \tag{9}$$

Here, F_C and U_C are orthogonal projection matrices onto $\mathrm{Im}\,C$ and $(\mathrm{Im}\,C)^{\perp}$, respectively, whereas F_D and U_D are orthogonal projection matrices onto $\mathrm{Im}\,D$ and $(\mathrm{Im}\,D)^{\perp}$, respectively.

Throughout the present study, the numerical methods for solving (2) and (6) are called AiVRM and AiVRM2, respectively. In the actual calculation of both methods, after calculating $\boldsymbol{c}^{\dagger}$ and $\boldsymbol{c}^{\ddagger}$, (2) and (6) are solved for $\boldsymbol{u}$ with the Krylov subspace method. Note that, not only in calculating $\boldsymbol{d}^{\dagger}$ and $\boldsymbol{d}^{\ddagger}$ but also in evaluating the product of $\boldsymbol{\lambda}$-eliminators and a vector, linear systems must be solved for K unknowns. For example, a linear system $(D^T C)\boldsymbol{z} = \boldsymbol{v}$ needs to be solved for $\boldsymbol{z} \in \mathbb{R}^K$ in the AiVRM, whereas linear systems, $(C^T C)\boldsymbol{z} = \boldsymbol{v}$ and $(D^T D)\boldsymbol{z} = \boldsymbol{v}$, have to be solved for $\boldsymbol{z} \in \mathbb{R}^K$ in the AiVRM2. In the following, such linear systems are called inner linear systems, whereas (2) and (6) are called outer linear systems. In addition, the Krylov subspace method for solving inner and outer linear systems are called inner and outer solvers, respectively. Besides, the numerical determination of $\boldsymbol{c}^{\dagger}$ and $\boldsymbol{c}^{\ddagger}$ is referred to as an overhead.

In the AiVRM, BiCGSTAB is used for both the inner and outer solvers. On the other hand, two different methods are applied to the inner and outer solvers in the AiVRM2. Specifically, BiCGSTAB is used as an outer solver, whereas the conjugate gradient (CG) method is adopted as an inner solver. The choice of the CG method as an inner solver is based on the fact that inner linear systems of the AiVRM2 have symmetric coefficient matrices, $C^T C$ and $D^T D$. Incidentally, zero vectors are always assumed as initial guesses for solutions in both the inner and outer solvers.

3.2 Computational cost for inner solvers

Since an inner solver is the Krylov subspace method for solving a linear system with K unknowns, the number m_{in} of iterations required for its convergence can be written as $m_{\mathrm{in}} = O(K^r)$, where $0 \le r \le 1$. If this fact is taken into consideration, operation counts for the AiVRM/AiVRM2 are estimated in a similar way to Appendix A in [3]. Operation counts for an overhead and for each iteration in the outer solver are both given by $O(N^{\max[q(r+1),1]})$. Especially for the case with an asymmetric EFG-SP problem originating from a 2D Poisson problem, we have $q = 1/2$. Hence, operation counts both for an overhead and for each iteration in the outer solver are estimated as $O(N)$ for this case. This estimation suggests that, in the AiVRM/AiVRM2, inner solvers hardly affect the total computational cost for a sufficiently large value of N.

Let us quantitatively investigate the influence of inner solvers on the total computational cost. As the measure of the influence, we use the CPU-time ratio defined by $R_{\mathrm{CPU}} \equiv \tau_{\mathrm{in}}/\tau$. Here, τ_{in} denotes an accumulated CPU time for solving inner linear systems, whereas τ is a CPU time for solving (2) and (6) with the AiVRM and the AiVRM2, respectively. Dependences of R_{CPU} on N are numerically determined for the AiVRM and for the AiVRM2, and they are depicted in Fig. 2. The CPU-time ratio R_{CPU} decreases monotonously with an increase in N until it amounts down to about 35% for $N = 1,050,625$. This result indicates that an increase in N will weaken the influence of inner solvers on the total computational cost.

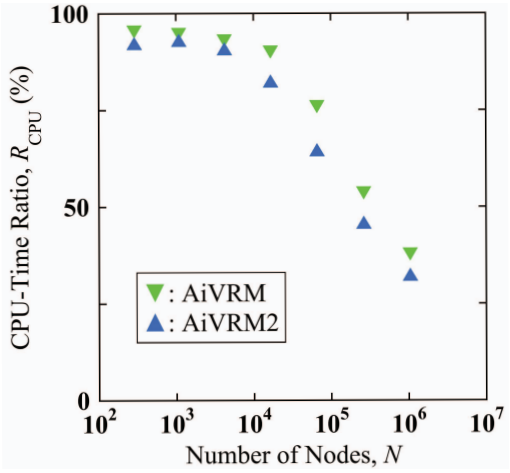


Fig. 2 Dependences of the CPU-time ratio R_{CPU} on the number N of nodes. Here, the same EFG-SP problems as in Fig. 1 are solved with the AiVRM/AiVRM2.

4. Numerical Experiments

4.1 Test problem

As a test problem, we consider the following 2D Poisson problem on the domain Ω bounded by a simple closed curve $\partial\Omega$: $-\nabla^2 u = p$ in Ω and $u = \bar{u}$ on $\partial\Omega$, where p and $\bar{u}$ are given functions in Ω and on $\partial\Omega$, respectively.

In this section, two kinds of inner products are defined by

$$(f, g)_{\Omega} \equiv \iint_{\Omega} f(\boldsymbol{x}) g(\boldsymbol{x})\, d^2\boldsymbol{x}, \tag{10}$$

$$[\zeta, \eta]_{\partial\Omega} \equiv \oint_{\partial\Omega} \zeta(s)\eta(s)\, ds. \tag{11}$$

Here, $\boldsymbol{x}$ denotes a position vector of a point in Ω, whereas s is an arc length along $\partial\Omega$. In addition, the boundary $\partial\Omega$ is specified by $\boldsymbol{x} = \boldsymbol{x}(s)$ $(\alpha \le s \le \beta)$, where α and β are both constants.

After discretizing the above Poisson problem by using the X-EFG method with N nodes including K boundary nodes, we get (1). If the (i, j)th entry of matrix A and the ith component of vector $\boldsymbol{a}$ are denoted by $(A)_{i,j}$ and $(\boldsymbol{a})_i$, respectively, entries of submatrices, B, C and D, and components of vectors, $\boldsymbol{c}$ and $\boldsymbol{d}$, are given by

$$(B)_{ij} = (1, \nabla\psi_i \cdot \nabla\phi_j)_{\Omega},$$

$$(C)_{ik} = [\psi_i, N_k]_{\partial\Omega}, \qquad (D)_{ik} = [\phi_i, N_k]_{\partial\Omega},$$
$$(\boldsymbol{c})_i = (\psi_i, p)_{\Omega}, \qquad (\boldsymbol{d})_k = [N_k, \bar{u}]_{\partial\Omega}.$$

Here, $\{\psi_i(\boldsymbol{x})\}_{i=1}^N$ and $\{\phi_i(\boldsymbol{x})\}_{i=1}^N$ are weight functions and shape functions of the MLS approximation [1, 2], respectively, and $\{N_k(s)\}_{k=1}^K$ are shape functions of the boundary element method.

4.2 Performance evaluation

First of all, we briefly explain the analysis conditions of numerical experiments. A domain Ω of the 2D Poisson problem is assumed to be $\Omega = (0, 1) \times (0, 1)$. Furthermore, parameters in the X-EFG method are assumed as follows. All nodes are homogeneously distributed in Ω and on $\partial\Omega$. Also, a support radius R of weight functions $\{\psi_i(\boldsymbol{x})\}_{i=1}^N$ is assumed as $R = 1.5\,h$, where h denotes a distance between the nearest two nodes. Moreover, exponential weight functions [1] are adopted as $\{\psi_i(\boldsymbol{x})\}_{i=1}^N$. In the numerical experiments, $\boldsymbol{c}$ and $\boldsymbol{d}$ are not calculated by using the formula described in Section 4.1. Instead, they are chosen so that the ith component of a solution of (1) may be given by $\mathrm{mod}(i, 5)/5$.

All numerical computations were performed on a Linux (Ubuntu ver. 20.04.4 LTS) machine with Intel Core i7-11700KF CPU (3.60 GHz) and 64 GB RAM. Furthermore, GNU Fortran (gfortran) ver. 9.4.0 was employed as a compiler and "`-O2`" was set as its option.

Let us first compare a convergence property of the AiVRM and the AiVRM2 with that of ILDU-GMRES. To this end, residual histories for the three methods are numerically determined and are depicted in Fig. 3. We see from this figure that both the AiVRM and the AiVRM2 are even superior to ILDU-GMRES in terms of a convergence property.

Next, we investigate the computational cost for the three methods. For this purpose, CPU times for the methods are measured as functions of the number N of nodes and they are depicted in Fig. 4. For $N \lesssim 10^4$, ILDU-GMRES is the fastest among the three methods. On the other hand, for $N \gtrsim 10^4$, the AiVRM2 becomes the fastest. Especially for the case with $N = 1,050,625$, the AiVRM2 is around 175 times faster than ILDU-GMRES. Moreover, the CPU time for the AiVRM/AiVRM2 is roughly proportional to $N^{1.3}$.

As is apparent from Fig. 3, the number of iterations required for convergence of the AiVRM is almost the same as that of the AiVRM2. However, Fig. 4 indicates that the CPU time for the AiVRM2 is always less than that for the AiVRM. The reason for this can be explained in terms of numbers of matrix-vector products in the inner solvers. As explained in Sec. 3.1, BiCGSTAB and the CG method are used as the inner solvers in the AiVRM and the AiVRM2, respectively. On the other hand, in each iteration of BiCGSTAB and the CG method, matrix-vector products are performed twice and once, respectively. Hence, operation counts in each iteration of the inner solver in the

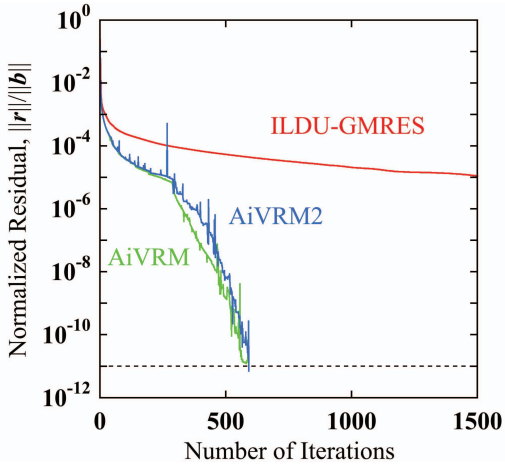


Fig. 3 Residual histories of ILDU-GMRES, AiVRM and AiVRM2 for the case with $N = 1{,}050{,}625$.

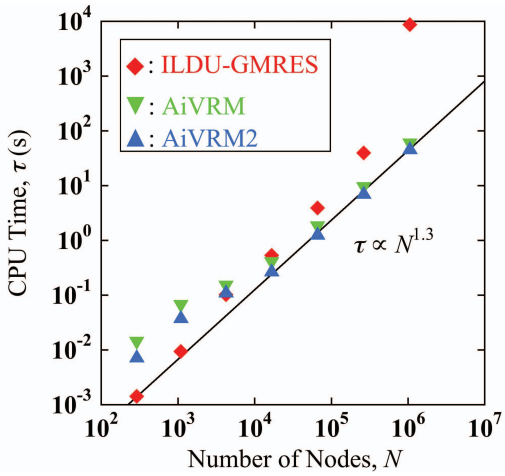


Fig. 4 Dependences of the CPU time τ on the number N of nodes.

AiVRM2 are about half of those in the AiVRM. This is why the CPU time for the AiVRM2 is less than that for the AiVRM.

Finally, we investigate the accuracy of the three methods. As a measure of the accuracy, we adopt the relative error defined by $\epsilon \equiv \|\boldsymbol{u}_{\mathrm{N}} - \boldsymbol{u}_{\mathrm{A}}\|/\|\boldsymbol{u}_{\mathrm{A}}\|$. Here, $\boldsymbol{u}_{\mathrm{A}}$ and $\boldsymbol{u}_{\mathrm{N}}$ are analytic and numerical solutions of an asymmetric EFG-SP problem corresponding the test problem, respectively. Furthermore, $\|\ \|$ denotes the Euclidean norm of a vector. The dependence of the relative error on the number N of nodes is shown in Fig. 5. We see from this figure that there is no significant difference in the accuracy of the three methods.

The above results indicate that both the AiVRM and the AiVRM2 are suitable for solving large-scale asymmetric EFG-SP problems.

5. Conclusion

In the present study, we have developed not only the

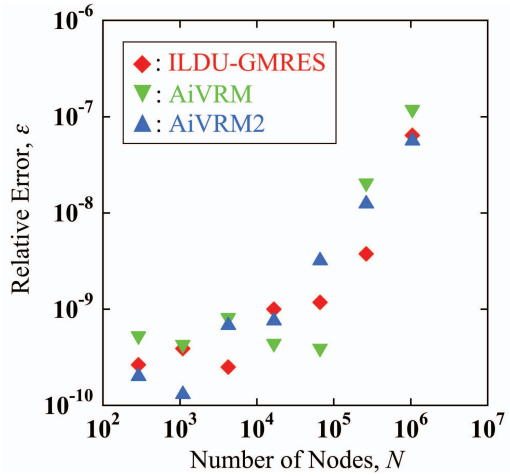


Fig. 5 Dependences of the relative error ϵ on the number N of nodes.

AiVRM but also the AiVRM2 as linear-system solvers for asymmetric EFG-SP problems. Furthermore, performances of the two methods are investigated numerically. As a result, the AiVRM and the AiVRM2 show the following two properties:

- From the standpoint of a convergence property, the two methods are even superior to ILDU-GMRES.
- From the standpoint of a computational speed, both of the methods are much faster than ILDU-GMRES in solving a large-scale problem.

From these results, we can conclude that both the AiVRM and the AiVRM2 are particularly effective in solving large-scale asymmetric EFG-SP problems.

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 (NIFS22 KISS018, NIFS22KIES001, NIFS22KIGS004).

[1] T. Belytschko, Y.Y. Lu and L. Gu, Int. J. Numer. Meth. Eng. **37**, 229 (1994).
[2] A. Kamitani, T. Takayama, T. Itoh and H. Nakamura, Plasma Fusion Res. **6**, 2401074 (2011).
[3] A. Kamitani, T. Takayama, A. Saitoh and H. Nakamura, Plasma Fusion Res. **17**, 2403014 (2022).
[4] M. Rozložník, *Saddle-Point Problems and Their Iterative Solution* (Birkhäuser, Switzerland, 2018) Chaps. 1 and 4.
[5] M. Benzi, G.H. Golub and J. Liesen, Acta Numer. **14**, 1 (2005).
[6] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 4th ed. (Johns Hopkins University Press, Maryland, 2013) p. 246.
[7] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in Fortran 77*, 2nd ed. (Cambridge Univ. Press, New York, 1992) p. 708.
[8] T. Itoh, A. Saitoh, S. Ikuno and A. Kamitani, J. Adv. Simulat. Sci. Eng. **3**, 188 (2017).