Acceleration Techniques for Linear-System Solver in Shielding Current Analysis of Cracked HTS Film[*]

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

Yamagata University, 4-3-16 Jonan, Yonezawa 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*

Two types of acceleration techniques, $\mathcal{H}$-matrix arithmetics and an $\mathcal{H}$-matrix-based variable preconditioning (VP), as well as their combination are applied to a linear-system solver in the shielding current analysis of a cracked high-temperature superconducting film. Although the combination seems to be the most effective of three types of the acceleration techniques, the results of computations show that, from the standpoint of the acceleration performance, neither the $\mathcal{H}$-matrix-based VP nor the combination is superior to $\mathcal{H}$-matrix arithmetics. The reason for this unexpected result is explained from the standpoint of operation counts.

1. Introduction

A large-area high-temperature superconducting (HTS) film has been recently fabricated for various engineering applications such as electromagnets, energy storage, device sensor, and power cable. However, for the case where an HTS film contains cracks in it, its physical property is significantly deteriorated. Hence, contactless methods have been desired for detecting cracks in an HTS film. In order to conceptually design the contactless methods, the shielding current density must be accurately evaluated in a cracked HTS film. For this reason, the authors have developed numerical methods for analyzing the shielding current density on the basis of the current-vector-potential method [1–3].

After temporally discretized with the implicit method, an initial-boundary-value problem for the current-vector potential reduces to a nonlinear boundary-value problem at each time step. Especially for the case with a cracked HTS film, application of the Newton method to the nonlinear problem yields a linear system of special type at each iteration cycle of the Newton method. The linear system is known as a difficult problem to numerically solve and is called a saddle-point problem. In fact, even if the Krylov-space method such as GMRES is applied to the linear system, its residual history shows a degraded convergence property. In order to resolve this problem, the authors developed the variable-reduction method (VRM) [2, 3]. As a result, it is found that the VRM is effective in speeding up the shielding current analysis of a cracked HTS film.

The purpose of the present study is to propose acceleration techniques for the VRM. In addition, after implementing the proposed techniques to the shielding current analysis, we clarify which technique shows the highest performance.

2. Shielding Current Analysis

In the present study, the shielding current analysis is applied to the scanning permanent-magnet method (SPMM) [2, 4] that is one of contactless methods for measuring the spatial distribution of the critical current density in an HTS film. A schematic view of the SPMM is shown in Fig. 1. In the SPMM, while a cylindrical permanent magnet of height H_{PM} and radius R_{PM} is moved along the film surface at a constant speed v, an electromagnetic force acting on the film is measured. In the present study, we assume that an HTS film contains a single crack and that its cross section is a rectangle of length l and width w. In addition, the crack is assumed to have the same geometric center as the film and its direction is assumed to be parallel to the longitudinal direction of the film. Besides, the permanent magnet is assumed to move on a straight
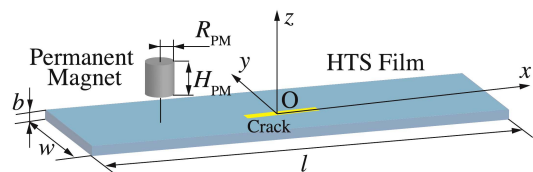

Fig. 1   A schematic view of the SPMM.

line that passes through the geometric center of the film and is along the longitudinal direction of the film.

In the following, $\boldsymbol{J}$ and $\boldsymbol{E}$ are a shielding current density and an electric field, respectively, and $\boldsymbol{B}$ is a magnetic flux density generated by the permanent magnet. Besides, Ω is a rectangular cross section of the film and $\boldsymbol{n}$ denotes an outward normal unit vector on the boundary $\partial\Omega$ of Ω. Moreover, $\boldsymbol{e}_z$ is a unit vector along the thickness direction. Also, $\boldsymbol{x}$ and $\boldsymbol{x}'$ are position vectors of two points in the xy plane.

Under the thin-plate approximation, there exists a scalar function $T(\boldsymbol{x},t)$ such that $\boldsymbol{J} = (2/b)\nabla \times (T\boldsymbol{e}_z)$, and its time evolution is governed by the following integro-differential equation [1–3]:

$$\mu_0 \frac{\partial}{\partial t}(\hat{W}T) + (\nabla \times \boldsymbol{E}) \cdot \boldsymbol{e}_z = -\frac{\partial}{\partial t}\langle \boldsymbol{B} \cdot \boldsymbol{e}_z \rangle, \qquad (1)$$

where μ_0 is a permeability of vacuum and b is the film thickness. Also, $\langle \ \rangle$ denotes an average operator over the thickness and the operator $\hat{W}$ is given by

$$\hat{W}T \equiv \frac{2T(\boldsymbol{x},t)}{b} + \iint_\Omega Q(|\boldsymbol{x} - \boldsymbol{x}'|)\, T(\boldsymbol{x}',t)\, d^2\boldsymbol{x}', \quad (2)$$

where $Q(r) = -(\pi b^2)^{-1}[r^{-1} - (r^2 + b^2)^{-1/2}]$.

The initial and boundary conditions to (1) are assumed as follows: $T = 0$ at $t = 0$ and $\boldsymbol{J} \cdot \boldsymbol{n} = 0$ on $\partial\Omega$. For the J-E constitutive relation, the following power law [1–3, 5, 6] is assumed:

$$\boldsymbol{E} = E(|\boldsymbol{J}|)\frac{\boldsymbol{J}}{|\boldsymbol{J}|}, \ E(J) = E_C \left(\frac{J}{J_C}\right)^N, \qquad (3)$$

where E_C and J_C are the critical electric field and the critical current density, respectively, and N is a positive constant. By solving (1) together with the initial and boundary conditions, we can analyze the time evolution of $\boldsymbol{J}$.

Throughout the present study, the physical and geometric parameters are fixed as follows: $J_C = 1$ MA/cm^2, $E_C = 1$ mV/m, $N = 20$, $w = 12$ mm, $l = 11w$, $b = 1$ μm, $H_{PM} = 2$ mm, $R_{PM} = 0.8$ mm, and $v = 10$ mm/s.

3. Convergence-Property Improvement for Linear System Solver

3.1 Linear system of special type

By applying the complete implicit scheme to the initial-boundary-value problem of (1), we get a nonlinear boundary-value problem at each time step that can be solved by means of the Newton method. In other words, a solution of the nonlinear problem is iteratively determined by solving a linear boundary-value problem at each iteration cycle of the Newton method.

If the linear boundary-value problem is discretized by using shape functions $\{\psi_i\}_{i=1}^n$ of the finite element method (FEM), we get the following linear system [2, 3]:

$$\begin{bmatrix} A & C \\ D^T & O \end{bmatrix}\begin{bmatrix} \delta\boldsymbol{T} \\ \delta\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{c} \end{bmatrix}. \qquad (4)$$

Here, $\delta\boldsymbol{T} \in \mathbb{R}^n$ and $\delta\boldsymbol{\lambda} \in \mathbb{R}^k$ are unknown vectors, whereas $\boldsymbol{b} \in \mathbb{R}^n$ and $\boldsymbol{c} \in \mathbb{R}^k$ are given ones. Furthermore, $A \in \mathbb{R}^{n\times n}$ and $C, D \in \mathbb{R}^{n\times k}$ are given matrices. Here, $2(k-1)$ and n denote the number of elements adjacent to crack surfaces and the number of nodes, respectively. Throughout the present study, k is called the number of constraints and it is assumed to satisfy $n \gg k$.

Specifically, A in (4) is determined by

$$A = W + \delta E_N, \qquad (5)$$

where the (i, j) entries of W and δE_N are given by

$$(W)_{ij} = \mu_0(\psi_i, \hat{W}\psi_j)_\Omega,$$
$$(\delta E_N)_{ij} = \frac{2\Delta t}{b}(1, (\nabla\psi_i \times \boldsymbol{e}_z) \cdot \overleftrightarrow{J}(\boldsymbol{J}) \cdot (\nabla\psi_j \times \boldsymbol{e}_z))_\Omega.$$

Here, Δt denotes a time step size and an inner product is defined by $(f, g)_\Omega \equiv \iint_\Omega f(\boldsymbol{x})g(\boldsymbol{x})\, d^2\boldsymbol{x}$. In addition, the 2nd-order tensor $\overleftrightarrow{J}(\boldsymbol{J})$ is given by

$$\overleftrightarrow{J}(\boldsymbol{J}) = \frac{d}{dJ}\left[\frac{E(J)}{J}\right]\frac{\boldsymbol{J} \otimes \boldsymbol{J}}{J} + \frac{E(J)}{J}\overleftrightarrow{1},$$

and $\overleftrightarrow{1}$ is a 2nd-order identity tensor. As is apparent from (5), the matrix A is calculated from an FEM matrix W. In other words, A is a function of W, i.e. $A = A(W)$. Hence, if an approximate matrix H is used instead of W, A in (4) is replaced with $A(H)$.

3.2 Variable-reduction method

If GMRES is applied to (4), its convergence property becomes degraded remarkably with an increase in k [2, 3]. For the purpose of improving such an unfavorable convergence property, the authors developed the VRM.

In the VRM, the QR factorizations [7] of C and D are first computed as follows: $C = Q_C R_C P_C^T$, $D = Q_D R_D P_D^T$, where $Q_C, Q_D \in \mathbb{R}^{n\times k}$ are matrices such that $Q_C^T Q_C = I$ and $Q_D^T Q_D = I$. Here, I denotes an identity matrix. Besides, $R_C, R_D \in \mathbb{R}^{k\times k}$ and $P_C, P_D \in \mathbb{R}^{k\times k}$ denote upper triangular matrices and permutation matrices, respectively. By using Q_C and Q_D, we get the following projection matrices: $U_C \equiv I - Q_C Q_C^T$, $F \equiv Q_C Q_D^T$ and $U \equiv I - F$.

Next, by using U_C, F and U, $\delta\boldsymbol{\lambda}$ can be eliminated from (4). As a result, (4) is equivalent to the following two linear systems :

$$A^\dagger \delta\boldsymbol{T} = \boldsymbol{b}^\dagger, \qquad (6)$$
$$\delta\boldsymbol{\lambda} = P_C R_C^{-1} Q_C^T (\boldsymbol{b} - A\,\delta\boldsymbol{T}). \qquad (7)$$

Here, for $\forall B \in \mathbb{R}^{n\times n}$ and $\forall \boldsymbol{f} \in \mathbb{R}^n$, $B^\dagger$ and $\boldsymbol{f}^\dagger$ are defined by $B^\dagger \equiv U_C^T B U + F$ and $\boldsymbol{f}^\dagger \equiv U_C^T[\boldsymbol{f} - B\boldsymbol{c}^*] + \boldsymbol{c}^*$, where $\boldsymbol{c}^* = Q_C R_D^{-T} P_D^T \boldsymbol{c}$.

After (6) is numerically solved for $\delta\boldsymbol{T}$ by means of GMRES, $\delta\boldsymbol{T}$ is substituted into (7) to get $\delta\boldsymbol{\lambda}$. Throughout the present study, this method is called the VRM.

4. $\mathcal{H}$-Matrix-Based Acceleration

In order to speed up the VRM, we employ two numerical techniques, $\mathcal{H}$-matrix arithmetics and $\mathcal{H}$-matrix-based variable preconditioning, that are both based on an $\mathcal{H}$-matrix of the FEM matrix W.

4.1 Determination of $\mathcal{H}$-matrix

After a cluster tree is generated on the basis of the information on node positions, W is transformed to an $\mathcal{H}$-matrix by using the tree [8, 9]. Specifically, for a cluster pair of σ and ω on the same level in the tree, we check whether the admissibility condition, $\min[\mathrm{diam}(\sigma), \mathrm{diam}(\omega)] \leq \eta \, \mathrm{dist}(\sigma, \omega)$, is satisfied or not. Here, $\mathrm{diam}(\sigma)$ denotes a diameter of σ, whereas $\mathrm{dist}(\sigma, \omega)$ is a distance between σ and ω. In addition, η is an admissibility parameter.

If the cluster pair, σ and ω, satisfies the admissibility condition, the submatrix $W_{(\sigma, \omega)}$, which corresponds to the cluster pair, is approximated as $W_{(\sigma, \omega)} \cong \sum_{i=1}^{r} \boldsymbol{u}_i \boldsymbol{v}_i^T (\equiv W_A)$. Otherwise, the condition is checked again for four pairs, each of which is composed of child clusters of σ and ω. Incidentally, W_A is called a low-rank submatrix.

By starting the above check for a pair of two root clusters, we can obtain an $\mathcal{H}$-matrix H for W.

4.2 Acceleration by $\mathcal{H}$-matrix arithmetics

Since matrix-vector multiplications are the most time-consuming at each iteration of GMRES, the VRM can be accelerated by high-speed matrix-vector multiplications $A^\dagger \boldsymbol{v}$. On the other hand, the operation count for $W\boldsymbol{v}$ fills a large portion of that for $A^\dagger \boldsymbol{v}$. Hence, the VRM can be accelerated by using fast matrix-vector multiplications $W\boldsymbol{v}$.

In the present study, fast matrix-vector multiplications $W\boldsymbol{v}$ are realized by means of $\mathcal{H}$-matrix arithmetics [8].

4.3 Acceleration by variable preconditioning

For the purpose of accelerating the VRM, the variable preconditioning (VP) [10] is implemented to the VRM. As is well known, $\boldsymbol{z} \equiv (A^\dagger)^{-1}\boldsymbol{v}$ is approximately calculated in the VP. In the present study, the VP is carried out by roughly solving the linear system, $A_{\mathrm{H0}}^\dagger \boldsymbol{z} = \boldsymbol{v}$, with GMRES. Here, an approximate matrix A_{H0} is defined by $A_{\mathrm{H0}} \equiv A(H_0)$. Also, H_0 is a matrix obtained by substituting zero matrices into all low-rank submatrices of H.

In the present study, the above method is called an $\mathcal{H}$-matrix-based VP and the iteration in solving $A_{\mathrm{H0}}^\dagger \boldsymbol{z} = \boldsymbol{v}$ with GMRES is called an inner loop. Also, the maximum number of iterations in the inner loop is denoted by N_s.

5. Numerical Results

As is apparent from Sec.3, the linear system (6) changes depending both on the iteration cycle of the Newton method and on time. Therefore, throughout Sec.5, acceleration techniques are applied to (6) for the 17th iter-ation cycle at $t = l/(1200v)$ and their performances are assessed. As the measure of the speedup effect of an accelerated VRM (A-VRM), we adopt the speedup ratio τ/τ_A, where τ and τ_A denote CPU times required for the VRM and for the A-VRM, respectively. Throughout Sec.5, values of n and k are fixed as $n = 11649$ and $k = 61$, respectively.

In the present section, only the speedup effect of A-VRMs is investigated. This is mainly because their accuracy is hardly affected by $\mathcal{H}$-matrix-based accelerations. The details of their accuracy are discussed in Appendix A.

5.1 $\mathcal{H}$-matrix arithmetics

Let us first investigate the influence of $\mathcal{H}$-matrix arithmetics on matrix-vector multiplications. As the measure of speed for matrix-vector multiplications, the acceleration rate S_H is defined by $S_H \equiv \tau^*/\tau_H^*$, where τ_H^* and τ^* are CPU times for matrix-vector multiplications with and without $\mathcal{H}$-matrix arithmetics, respectively. Moreover, as the measure of accuracy for $\mathcal{H}$-matrix arithmetics, the relative error e_H is defined by $e_H \equiv \|W\boldsymbol{v} - H\boldsymbol{v}\|/\|W\boldsymbol{v}\|$. The acceleration rate and the relative error are calculated as functions of the admissibility parameter η and they are depicted in Fig. 2. Both the acceleration rate and the relative error increase monotonously with η. For the purpose of ensuring both speed and accuracy of $\mathcal{H}$-matrix arithmetics, the value of η is fixed as $\eta = 2$, hereafter.

Next, we investigate how the A-VRM is affected by $\mathcal{H}$-matrix arithmetics. For this purpose, the speedup ratio is measured and is shown in Table 1. For the A-VRM in which only $\mathcal{H}$-matrix arithmetics are implemented, the speedup ratio τ/τ_A amounts up to 9.52. This result indicates the speedup effect due to fast matrix-vector multiplications. Hence, the implementation of $\mathcal{H}$-matrix arithmetics is extremely effective to acceleration of the VRM.

5.2 $\mathcal{H}$-matrix-based VP

First, we investigate the convergence property of the A-VRM in which an $\mathcal{H}$-matrix-based VP is implemented.
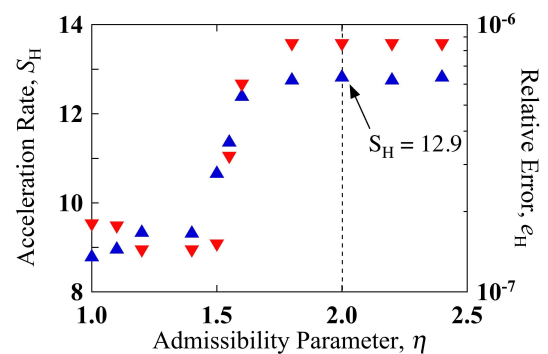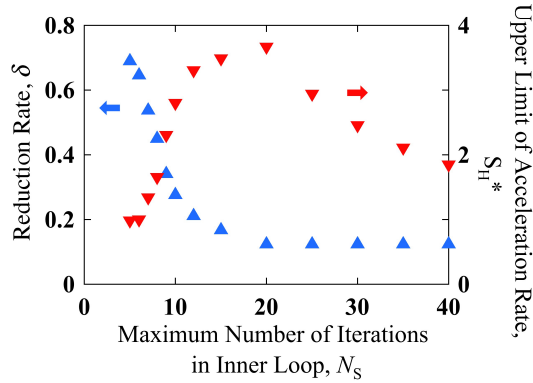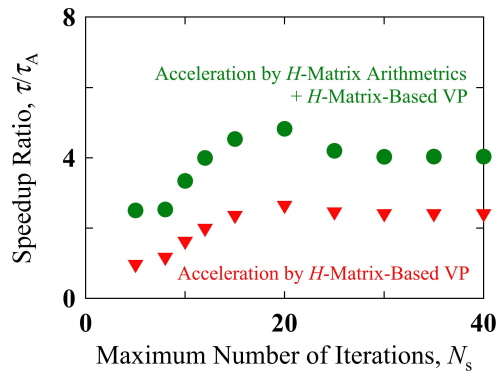

Fig. 2 Dependence of the acceleration rate S_H and the relative error e_H on the admissibility parameter η. Here, the symbols, ▲ and ▼, indicate the values of S_H and e_H, respectively.

Table 1　Speedup ratio τ/τ_A for two types of A-VRM.

acceleration technique	τ/τ_A
$\mathcal{H}$-matrix arithmetics	9.52
combination method ($N_s = 20$)	4.83

Fig. 3　The reduction rate δ of convergent iteration numbers and the upper limit of S_H^* of acceleration rate as functions of the maximum number N_s of iterations in the inner loop.

Fig. 4　Dependence of the speedup ratio τ/τ_A on the maximum number N_s of iterations.

As the measure of the convergence property, the reduction rate δ of convergent iteration numbers is defined by $\delta \equiv N_{\text{with}}/N_{\text{without}}$. Here, N_{with} and N_{without} are iteration numbers required for the convergence of GMRES with and without an $\mathcal{H}$-matrix-based VP, respectively. The dependence of δ on N_s is shown in Fig. 3. We see from this figure that the reduction rate δ decreases monotonously with N_s until to reach around 0.13 for $N_s \gtrsim 20$. In other words, the convergence property of the A-VRM is improved with an increase in N_s.

Next, we investigate the speedup effect by an $\mathcal{H}$-matrix-based VP. For this purpose, the speedup ratio is measured as a function of N_s and is depicted in Fig. 4. This figure indicates that τ/τ_A takes a maximum at $N_s \cong 20$. Moreover, the maximum is about 2.6.

The above results indicate that, although an $\mathcal{H}$-matrix-based VP improves the convergence property of the VRM,

its speedup effect is not remarkable as compared with $\mathcal{H}$-matrix arithmetics.

5.3　Combined acceleration techniques

In order to further accelerate the A-VRM with an $\mathcal{H}$-matrix-based VP, $\mathcal{H}$-matrix arithmetics are also applied to matrix-vector multiplications. We first compare the speedup effect by the combination of an $\mathcal{H}$-matrix-based VP and $\mathcal{H}$-matrix arithmetics with that by an $\mathcal{H}$-matrix-based VP only. To this end, the speedup ratio of the combination method is measured as a function of N_s and is also plotted in Fig. 4. We see from this figure that, by implementing $\mathcal{H}$-matrix arithmetics, the A-VRM with an $\mathcal{H}$-matrix-based VP can be sped up by a factor of 1.7 - 2.8.

Next, we compare acceleration by the combination method with that by $\mathcal{H}$-matrix arithmetics only. The speedup ratio of the combination method is determined and is also shown in Table 1. Apparently, the acceleration performance by the combination method is much inferior to that by $\mathcal{H}$-matrix arithmetics only.

Let us explain the reason for such an unexpected result. From a straightforward calculation of operation counts, the speedup ratio for the A-VRM with $\mathcal{H}$-matrix arithmetics only and that for the combination method are roughly estimated as $\tau/\tau_A \sim S_H$ and $\tau/\tau_A \gtrsim S_H/[\delta(1 + \gamma N_s S_H)]$, respectively. Here, γ denotes a ratio of the number of elements in the full-rank block to that of all elements in the $\mathcal{H}$-matrix. Hence, if the inequality $S_H < S_H^* (\equiv (1/\delta - 1)/(\gamma N_s))$ is satisfied, acceleration by the combination method is superior to that by $\mathcal{H}$-matrix arithmetics only. In other words, S_H^* is an upper limit of S_H above which the combination method shows more effective acceleration than $\mathcal{H}$-matrix arithmetics only. The dependence of S_H^* on N_s is also shown in Fig. 3. On the other hand, we get $S_H = 12.9$ for $\eta = 2$ (see Fig. 2). Therefore, the above inequality is not fulfilled at all. This is why the combination method is slower than the the A-VRM with $\mathcal{H}$-matrix arithmetics only.

From the above results, we can conclude that the implementation of $\mathcal{H}$-matrix arithmetics only is the most effective to acceleration of the VRM.

6.　Conclusion

We have investigated acceleration techniques for the VRM that is a linear-system solver in the shielding current analysis of a cracked HTS film. Not only two types of acceleration techniques, $\mathcal{H}$-matrix arithmetics and $\mathcal{H}$-matrix-based VP, but also the combination of both techniques is applied to the VRM and their performances are assessed.

Conclusions obtained in the present study are summarized as follows. $\mathcal{H}$-matrix arithmetics show the most effective performance of three types of the acceleration techniques. This is mainly because there must exist an upper limit S_H^* of the acceleration rate S_H for the combination

method to show a higher performance than $\mathcal{H}$-matrix arithmetics. However, speed of matrix-vector multiplications by $\mathcal{H}$-matrix arithmetics is so fast that S_H can easily exceed S_H^*.

Acknowledgment

Appendix A   Influence of $\mathcal{H}$-Matrix-Based Accelerations on Accuracy

In this appendix, we investigate how accuracy of A-VRMs is influenced by $\mathcal{H}$-matrix-based accelerations. As the measure of accuracy, we use the relative error defined by $e_\text{A} \equiv \|v_\text{N} - v_\text{A}\|/\|v_\text{A}\|$. Here, $v_\text{A} \in \mathbb{R}^n$ is a random vector whose components are determined by the random number generator. Besides, $v_\text{N} \in \mathbb{R}^n$ is a numerical solution obtained by solving the linear system, $A^\dagger v = b_\text{r}$, either with or without acceleration techniques. Here, b_r is determined by $b_\text{r} = A^\dagger v_\text{A}$.

The relative errors are calculated not only for the VRM without any acceleration but also for three types of A-VRMs, and the results of computations are listed in Table A1. We see from this table that accuracy is not at all affected by $\mathcal{H}$-matrix arithmetics. In contrast, it is slightly influenced by the $\mathcal{H}$-matrix-based VP. These results indi-

Table A1 Influence of $\mathcal{H}$-matrix arithmetics and the $\mathcal{H}$-matrix-based VP on the relative error e_A. Here, N_s is assumed as $N_\text{s} = 20$ in the $\mathcal{H}$-matrix-based VP. In addition, the symbols, $\bigcirc$ and $\times$, represent the case with and without the acceleration techniques, respectively.

$\mathcal{H}$-matrix arithmetics	$\mathcal{H}$-matrix-based VP	e_A
$\times$	$\times$	0.99×10^{-6}
$\bigcirc$	$\times$	0.99×10^{-6}
$\times$	$\bigcirc$	1.33×10^{-6}
$\bigcirc$	$\bigcirc$	1.33×10^{-6}

cate that accuracy of A-VRMs is hardly affected by $\mathcal{H}$-matrix-based accelerations.

[1] A. Kamitani, T. Takayama and S. Ikuno, IEEE Trans. Magn. **47**, 1138 (2011).
[2] A. Kamitani, T. Takayama, A. Saitoh and S. Ikuno, J. Adv. Simulat. Sci. Eng. **4**, 117 (2018).
[3] A. Kamitani, T. Takayama and A. Saitoh, Int. J. Appl. Electromagn. Mech. **59**, 157 (2019).
[4] K. Hattori, A. Saito, Y. Takano *et al.*, Physica C **471**, 1033 (2011).
[5] L. Makong, A. Kameni, L. Queval *et al.*, IEEE Trans. Magn. **54**, 7205404 (2018).
[6] F. Sirois, F. Grilli and A. Morandi, IEEE Trans. Appl. Supercond. **29**, 8000110 (2019).
[7] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 4th ed. (Johns Hopkins University Press, Maryland, 2013) p.246.
[8] M. Bebendorf, *Hierarchical Matrices* (Springer-Verlag, Berlin, 2008) p.49.
[9] S. Kurz, O. Rain and S. Rjasanow, IEEE Trans. Magn. **38**, 412 (2002).
[10] K. Abe and S. Zhang, Int. J. Numer. Anal. Model. **2**, 147 (2005).