Speedup of Shielding Current Analysis in High-Temperature Superconducting Film: Implementation of $\mathcal{H}$-Matrix Method*)

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

Yamagata University, 4-3-16 Johnan, Yonezawa, Yamagata 992-8510, Japan

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

(Received 28 November 2015 / Accepted 3 March 2016)

Numerical techniques are proposed for accelerating a linear-system solver appearing in the virtual-voltage method. After the proposed techniques are implemented to a numerical code for analyzing the shielding current density in a high-temperature superconducting film, their performance is numerically evaluated. The results of computations show that, if GMRES is incorporated as a linear-system solver, the speed of the virtual-voltage method will be remarkably enhanced. Moreover, it is also found that the implementation of the $\mathcal{H}$-matrix method to matrix-vector multiplications in GMRES will further improve the speed of the virtual voltage method.

Keywords: high-temperature superconductor, $\mathcal{H}$-matrix method, Newton method

DOI: 10.1585/pfr.11.2405041

1. Introduction

Recently, a high-temperature superconducting (HTS) film has been used for numerous engineering applications: magnet, energy storage system, power cable and so on. Since evaluation of the shielding current density is indispensable for the design of such engineering applications, several numerical methods [1–3] have been so far developed for analyzing the shielding current density. These methods are classified into two categories, according to the order of time- and space-discretizations.

After spatially discretized, an initial-boundary-value problem of the shielding current density is transformed to an initial-value problem of semi-explicit differential algebraic equations (DAEs) especially for the case where cracks are contained in an HTS film. The authors proposed a fast and stable algorithm [2] for solving the DAEs with the block LU decomposition. Throughout the present study, this method is called the DAE method.

On the other hand, time-discretization of the initial-boundary-value problem yields a problem in which a nonlinear boundary-value problem is solved at each time step. However, the solution of the nonlinear problem by the Newton method is extremely time-consuming. This is mainly because a linear system with a dense matrix has to be solved at each iteration of the Newton method. This method is called the virtual-voltage method [3].

The purpose of the present study is to develop numerical techniques for speeding up the virtual-voltage method and to numerically investigate how they affect its performance.

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

*) This article is based on the presentation at the 25th International Toki Conference (ITC25).

2. Governing Equations

We first assume that an HTS film has the same cross section Ω over the thickness and that it is exposed to the time-varying magnetic field $\boldsymbol{B}/\mu_0$. By taking its thickness direction as z-axis and choosing its centroid as the origin, we use the Cartesian coordinate system $\langle O : \boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z \rangle$. Furthermore, the HTS film is assumed to contain m cracks whose cross sections are line segments in the xy plane. Note that the boundary $\partial\Omega$ of Ω is composed of not only the outer boundary C_0 but also crack surfaces $C_1, C_2, \cdots, C_m$. In the following, $\boldsymbol{x}$ and $\boldsymbol{x}'$ denote position vectors of two points in the xy plane, whereas $\boldsymbol{t}$ and $\boldsymbol{n}$ are a unit tangent vector and a unit normal vector on $\partial\Omega$, respectively. In addition, b denotes a film thickness.

In HTS films, the electric field $\boldsymbol{E}$ and the shielding current density $\boldsymbol{j}$ are closely related to each other through the J-E constitutive relation. As the relation, we assume the following power law [2–5]:

$$\boldsymbol{E} = E(j)\,\frac{\boldsymbol{j}}{j}, \quad E(j) = E_{\mathrm{C}}\left(\frac{j}{j_{\mathrm{C}}}\right)^N,$$

where $j \equiv |\boldsymbol{j}|$. Also, j_{C} and E_{C} denote the critical current density and the critical electric field, respectively, and N is a positive constant.

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 equation [2,3]:

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

Here, $\langle\ \rangle$ denotes an average operator over the thickness and the operator $\hat{W}$ is defined by

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

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 $t = 0$, $T \in H(\bar{\Omega})$ and $h_i(\boldsymbol{E}) \equiv \oint_{C_i} \boldsymbol{E} \cdot \boldsymbol{t}\, ds = 0$ $(i = 1, 2, \cdots, m)$. Here, s is an arclength along crack surfaces $C_1, C_2, \cdots, C_m$ and $H(\bar{\Omega})$ is a function space defined by

$$H(\bar{\Omega}) \equiv \left\{ w(\boldsymbol{x}) : w = 0 \text{ on } C_0,\ \frac{\partial w}{\partial s} = 0 \text{ on } \bigcup_{i=1}^{m} C_i \right\}.$$

The boundary condition $T \in H(\bar{\Omega})$ is derived from $\boldsymbol{j} \cdot \boldsymbol{n} = 0$ on $\partial\Omega$, whereas $h_i(\boldsymbol{E}) = 0$ is the integral form of Faraday's law around the crack surface C_i. By solving (1) together with the initial and boundary conditions, we can investigate the time evolution of the shielding current density.

3. Virtual-Voltage Method

Throughout this section, a superscripts (l) denotes a value at time $t = l\Delta t$, where Δt is a time-step size. For example, $T(\boldsymbol{x}, l\Delta t)$ is denoted by $T^{(l)}(\boldsymbol{x})$. Also, an inner product is defined by

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

If the initial-boundary-value problem of (1) is discretized with respect to time, $T^{(l)}(\boldsymbol{x})$ becomes a solution of the following nonlinear boundary-value problem:

$$G(T) \equiv \mu_0 \hat{W} T + \Delta t\, \boldsymbol{e}_z \cdot (\nabla \times \boldsymbol{E}) - u = 0 \text{ in } \Omega, \quad (2)$$

$$h_i(\boldsymbol{E}) = 0 \quad (i = 1, 2, \cdots, m), \quad (3)$$

$$T \in H(\bar{\Omega}), \quad (4)$$

where $u \equiv \mu_0 \hat{W} T^{(l-1)} - \langle (\boldsymbol{B}^{(l)} - \boldsymbol{B}^{(l-1)}) \cdot \boldsymbol{e}_z \rangle$.

Although the above boundary-value problem can be easily solved with the finite element method (FEM), the accuracy of its numerical solution will be degraded with a decrease in the film thickness b. Especially, the numerically calculated value $N_i(\boldsymbol{E})$ of $h_i(\boldsymbol{E})$ does not always become negligibly small. In order to resolve this difficulty, the authors proposed the virtual-voltage method [3]. In the method, virtual voltages $\phi_1, \phi_2, \cdots, \phi_m$ are applied around $C_1, C_2, \cdots, C_m$, respectively, so as to have $N_i(\boldsymbol{E}) = 0$ exactly satisfied. In other words, (3) is replaced with $h_i(\boldsymbol{E}) = \phi_i$ and $N_i(\boldsymbol{E}) = 0$ $(i = 1, 2, \cdots, m)$. The resulting boundary-value problem is solved for $(T, \{\phi_i\}_{i=1}^m)$ by means of the Newton method.

In each iteration of Newton method, the following linear boundary-value problem is solved for $(\delta T, \{\delta\phi_i\}_{i=1}^m)$:

$$\delta G = -G(T), \quad (5)$$

$$\delta h_i - \delta\phi_i = -[h_i(\boldsymbol{E}) - \phi_i] \quad (i = 1, 2, \cdots, m), \quad (6)$$

$$\delta N_i = 0 \quad (i = 1, 2, \cdots, m), \quad (7)$$

$$\delta T \in H(\bar{\Omega}). \quad (8)$$

Here, δT and $\delta\phi_i$ are corrections of T and ϕ_i, respectively, whereas δG, δh_i and δN_i denote Fréchet derivatives of $G(T)$, $h_i(\boldsymbol{E})$ and $N_i(\boldsymbol{E})$, respectively. After a straightforward calculation, we can prove that (5) and (6) are equivalent to the following weak form:

$$\begin{aligned} &\forall w \in H(\bar{\Omega}) : \\ &\mu_0 (w, \hat{W}(T + \delta T))_\Omega + \Delta t\, (1, (\nabla w \times \boldsymbol{e}_z) \cdot (\boldsymbol{E} + \delta\boldsymbol{E}))_\Omega \\ &\quad + \Delta t \sum_{i=1}^{m} w(C_i)\, (\phi_i + \delta\phi_i) = (w, u)_\Omega. \end{aligned} \quad (9)$$

Here, $w(C_i)$ is a unknown constant[1], which $w(\boldsymbol{x})$ takes on C_i, and $\delta\boldsymbol{E}$ is given by

$$\delta\boldsymbol{E} = \frac{2}{b} \left\{ \frac{d}{dj} \left[\frac{E(j)}{j} \right] \frac{\boldsymbol{j} \otimes \boldsymbol{j}}{j} + \frac{E(j)}{j} \boldsymbol{I} \right\} \cdot (\nabla \delta T \times \boldsymbol{e}_z),$$

where $\boldsymbol{I}$ is an identity tensor of the 2nd order.

If (7), (8) and (9) are spatially discretized with the FEM, we get

$$\begin{bmatrix} A(\boldsymbol{T}) & C & F \\ C^T & O & O \\ D^T(\boldsymbol{T}) & O & O \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{T} \\ \boldsymbol{\lambda} \\ \delta\boldsymbol{\phi} \end{bmatrix} = \boldsymbol{G}(\boldsymbol{T}, \boldsymbol{\phi}), \quad (10)$$

where $\delta\boldsymbol{T}$ is a nodal vector corresponding δT and $\delta\boldsymbol{\phi}$ is defined by $\delta\boldsymbol{\phi} \equiv [\delta\phi_1, \delta\phi_2, \cdots, \delta\phi_m]^T$. In addition, $A(\boldsymbol{T})$, C and F are the $n \times n$, $n \times k$ and $n \times m$ matrices, respectively, and $D^T(\boldsymbol{T})\, \delta\boldsymbol{T} = \boldsymbol{0}$ is a discretized form of (7). Furthermore, $\boldsymbol{G}(\boldsymbol{T}, \boldsymbol{\phi}) \in \mathbb{R}^{n+k+m}$ is a vector-valued function of $\boldsymbol{T}$ and $\boldsymbol{\phi}$. Also, n and k denote the total number of nodes and that of nodes only on crack surfaces, respectively, and they are assumed to satisfy $n \gg k > m$. Equation (10) has an asymmetric coefficient matrix and, hence, it has been so far solved with the LU decomposition. In contrast, we propose that (10) be solved by GMRES. Since the operation count required for GMRES is almost proportional to the square of the number of unknowns, the operation count required for the virtual-voltage method is expected to be reduced to $O(n^2)$. For this reason, GMRES is incorporated into the virtual-voltage method as a linear-system solver.

For the purpose of comparing the speed of virtual-voltage method with that of the DAE method, CPU times required for both methods are measured on FUJITSU PRIMEHPC FX100 of LHD Numerical Analysis Server in National Institute of Fusion Science. Dependences of CPU times on the number n of nodes are depicted in Fig. 1. Although CPU times for both methods are in rough proportion to $n^{2.5\pm0.1}$, the virtual-voltage method is about 10 times faster than the DAE method.

4. Acceleration Techniques

4.1 Speedup strategy

As is apparent from Section 3, the numerical solution of (10) becomes a rate-determining step at each time in the virtual-voltage method. On the other hand, GMRES is an iterative linear-system solver and matrix-vector

[1] Since $\partial w/\partial s = 0$ is satisfied on C_i, $w(\boldsymbol{x})$ becomes constant on the crack surface C_i.

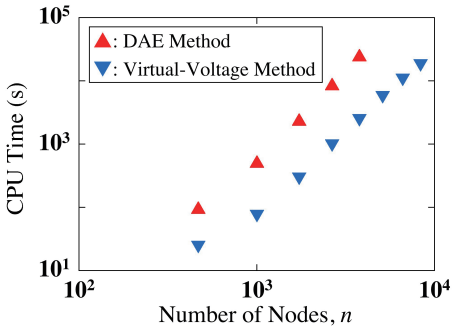


Fig. 1 CPU times as functions of the number of nodes. Here, the initial-boundary-value problem of (1) is solved from $t = 0$ to $t = Aw/(4v)$ by means of either the DAE method or the virtual voltage method. Physical and geometrical parameters are assumed to be the same values as used in Section 4.3.

multiplications are the most time-consuming at each iteration of GMRES. Thus, in order to accelerate the virtual-voltage method, matrix-vector multiplications have to be performed with high speed.

Let W be an FEM matrix, corresponding to the operator $\hat{W}$, in which the boundary condition on C_0 is also taken into account. Apparently, W is a dense matrix. If the contribution of W is subtracted from the coefficient matrix in (10), the remaining matrix becomes sparse. This means that the matrix-vector multiplication $W\boldsymbol{v}$ is the most time-consuming at each iteration of GMRES. Therefore, the virtual-voltage method can be accelerated by using the high-speed matrix-vector multiplication $W\boldsymbol{v}$.

In order to realize the fast matrix-vector multiplication $W\boldsymbol{v}$, the following two transformations are applied to W.

1. The matrix W is divided into a set of submatrices.
2. Only if a $p \times q$ submatrix $W_{(\sigma,\tau)}$ satisfies a specified condition, it is approximated by the ACA decomposition [6, 7]: $W_{(\sigma,\tau)} \cong UV^T$. Here, U and V are $p \times r$ and $q \times r$ matrices, respectively, and r denotes an approximate rank of $W_{(\sigma,\tau)}$.

In fact, $O(pq)$ and $O(r(p+q))$ operations are required for matrix-vector multiplications $W_{(\sigma,\tau)}\boldsymbol{v}_\tau$ and $UV^T\boldsymbol{v}_\tau$, respectively. Hence, if the inequality $r < pq/(p+q)$ is satisfied for all approximated submatrices, the high-speed matrix-vector multiplication $W\boldsymbol{v}$ is realized.

4.2 $\mathcal{H}$-matrix method

In the present study, the above two transformations are simultaneously performed by means of the $\mathcal{H}$-matrix method [6, 7]. In the $\mathcal{H}$-matrix method, a cluster tree is first generated on the basis of the information on node positions. In other words, after a set of all nodes is assumed as a root cluster σ_{R}, a cluster tree is constructed by using the following two steps:

1. A minimum axis-parallel rectangle is formed so as to contain all nodes belonging to the cluster. Such a rectangle is called a bounding box of the cluster.
2. The cluster is divided into two child clusters by means of a line segment linking middle points on two longer sides of the bounding box.

The above two steps are recursively repeated while the number of nodes contained in a cluster is greater than a certain threshold. In the following, each cluster is considered to contain not a set of nodes but a set of node numbers.

Next, W is transformed to an $\mathcal{H}$-matrix by using the cluster tree. Specifically, for a cluster pair of σ and τ on the same level in the tree, we check whether the admissibility condition, $\min[\mathrm{diag}(\sigma), \mathrm{diag}(\tau)] \leq \eta\,\mathrm{dist}(\sigma,\tau)$, is satisfied or not. Here, $\mathrm{diag}(\sigma)$ denotes a diagonal length of the bounding box of σ, whereas $\mathrm{dist}(\sigma,\tau)$ is a distance between bounding boxes of σ and τ. In addition, η is a positive constant. The results of this check are classified into the following three cases:

A For the case where the admissibility condition is satisfied, the submatrix $W_{(\sigma,\tau)}$, which corresponds to σ and τ, is approximated by the ACA decomposition.

B For the case where the admissibility condition is not fulfilled and either σ or τ is a leaf of the cluster tree, $W_{(\sigma,\tau)}$ is not approximated but stored as it is.

C Otherwise, the admissibility condition is checked for four pairs, each of which is composed of child clusters of σ and τ.

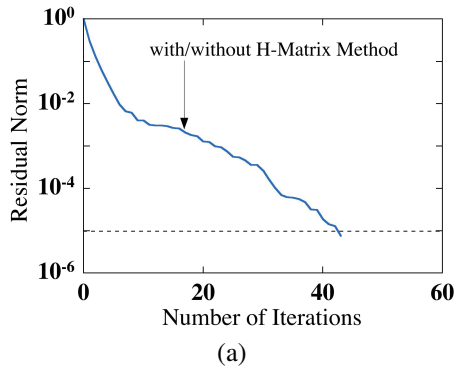
By starting the above check for a pair of two root clusters[2], σ_{R} and σ_{R}, we can obtain an $\mathcal{H}$-matrix for W.

4.3 Speedup effect of $\mathcal{H}$-matrix method

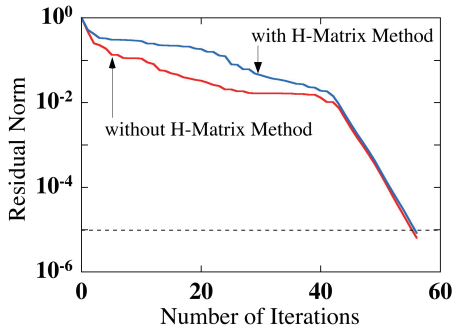
The authors developed a numerical code for analyzing the time evolution of the shielding current density in an HTS film with cracks. For the purpose of further speeding up the code, the $\mathcal{H}$-matrix method is implemented to GMRES.

In order to investigate the effect of the $\mathcal{H}$-matrix method on the speedup of the shielding current analysis, the scanning permanent-magnet method (SPM) [2, 8] is numerically simulated. In the SPM, a cylindrical permanent magnet of radius R and height H is moved along the surface of an HTS film and, simultaneously, an electromagnetic force acting on the film is monitored. During the movement of the magnet, the distance L between the magnet bottom and the film surface is kept constant. In the following, an HTS film is assumed to have a rectangular cross section Ω of width w and length Aw, and cross sections of cracks are assumed to be line segments of length L_{c}. In addition, the longitudinal direction of Ω is taken as x-axis. Also, the symmetry axis of the magnet is denoted by $(x, y) = (x_{\mathrm{A}}, y_{\mathrm{A}})$, and its movement is assumed

[2]The matrix W can be expressed as its submatrix $W_{(\sigma_{\mathrm{R}},\sigma_{\mathrm{R}})}$ corresponding to a pair of two root clusters, σ_{R} and σ_{R}. Hence, the admissibility condition is first checked for a pair of σ_{R} and σ_{R}.



(a)



(b)

Fig. 2 Residual histories of GMRES at $t = Aw/(1200v)$. Here, (a) and (b) are obtained for the 1st and 13th iterations of the Newton method, respectively.

as $x_A = vt - Aw/2$ and $y_A = \text{const}$. Here, v is a scanning speed. Throughout the present study, the physical and geometrical parameters are fixed as follows: $R = 0.8$ mm, $H = 2$ mm, $L = 0.5$ mm, $v = 10$ cm/s, $y_A = 0$ mm, $j_C = 1$ MA/cm^2, $E_C = 1$ mV/m, $N = 20$, $b = 1$ μm, $A = 11$, $w = 12$ mm, $L_c = 4$ mm and $m = 2$.

Let us first investigate how the $\mathcal{H}$-matrix method affects the residual history of GMRES. To this end, residual histories are determined for GMRES with/without the $\mathcal{H}$-matrix method and they are depicted in Figs. 2 (a) and 2 (b). At the initial iteration of the Newton method, the residual history is hardly influenced by the implementation of the $\mathcal{H}$-matrix method (see Fig. 2 (a)). On the other hand, at the almost final iteration of the Newton method, the residual norm is slightly affected by the $\mathcal{H}$-matrix method (see Fig. 2 (b)). However, if the number of iterations required for convergence of GMRES is denoted by M_c, the $\mathcal{H}$-matrix method scarcely changes M_c. In order to investigate these tendencies quantitatively, the sum of M_c is calculated as a function of time and is depicted in Fig. 3. This figure indicates that curves for GMRES with and without the $\mathcal{H}$-matrix method are overlapping with each other. This result shows that, at each time step, the $\mathcal{H}$-matrix method has little effect on the total number of GMRES iterations. Hence, it is the speed of the matrix-vector multiplication $\boldsymbol{Wv}$ that essentially determines the speed of the virtual-voltage method.

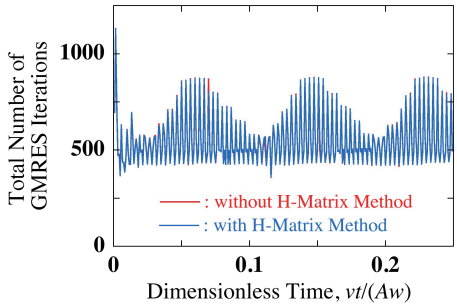


Fig. 3 Total number of GMRES iterations as functions of time.

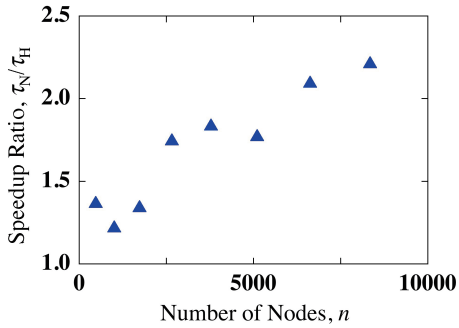


Fig. 4 Dependence of the speedup ratio τ_N/τ_H on the number of nodes.

Finally, we investigate the influence of the $\mathcal{H}$-matrix method on the speedup of the virtual-voltage method. To this end, the speedup ratio τ_N/τ_H is measured as functions of n and is depicted in Fig. 4. Here, τ_N and τ_H denote CPU times required for the execution of the code with and without the $\mathcal{H}$-matrix method, respectively. The speedup ratio τ_N/τ_H increases roughly with increasing number of nodes and it always exceeds unity. In other words, the $\mathcal{H}$-matrix method can accelerate the numerical code for the shielding current analysis and its speedup effect will be enhanced with an increase in the number of nodes. From these results, we can conclude that the $\mathcal{H}$-matrix method is effective for a large-sized shielding current analysis in an HTS film containing cracks.

5. Conclusion

We have proposed numerical techniques for accelerating a linear-system solver that appears in the virtual-voltage method. In addition, we have implemented them to a numerical code for analyzing the shielding current density in an HTS film with cracks. By using the code, the performance of the proposed techniques has been evaluated numerically. Conclusions obtained in the present study are summarized as follow.

1. When GMRES is implemented as a linear-system solver, the virtual-voltage method is about 10 times faster than the DAE method.
2. The implementation of the $\mathcal{H}$-matrix method to GM-

RES enables the virtual-voltage method to show an even higher speed.

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. 26520204, No. 15K05926. A part of this work was also carried out with the support and under the auspices of the NIFS Collaboration Research program (NIFS15KNTS041, NIFS15KNXN297).

[1] Y. Yoshida, M. Uesaka and K. Miya, IEEE. Trans. Magn. **30**, 3503 (1994).
[2] A. Kamitani, T. Takayama, A. Saitoh and H. Nakamura, Plasma Fusion Res. **10**, 3405023 (2015).
[3] A. Kamitani, T. Takayama and S. Ikuno, IEEE Trans. Magn. **49**, 1877 (2013).
[4] R. Brambilla, F. Grilli and L. Martini, IEEE Trans. Appl. Supercond. **22**, 8401006 (2012).
[5] A. Kameni, M. Boubekeur, L. Alloui, F. Bouillault, J. Lambretchs and C. Geuzaine, IEEE Trans. Magn. **50**, 7009204 (2014).
[6] S. Kurz, O. Rain and S. Rjasanow, IEEE Trans. Magn. **38**, 421 (2002).
[7] V. Le-Van, B. Bannwarth, A. Carpentier, O. Chadebec, J.M. Guichon and G. Meunier, IEEE Trans. Magn. **50**, 7010904 (2014).
[8] K. Hattori, A. Saito, Y. Takano, T. Suzuki, H. Yamada, T. Takayama, A. Kamitani and S. Ohshima, Physica C **471**, 1033 (2011).