Cyclic Symmetry in Volume Integral Formulations for Eddy Currents: Cohomology Computation and Gauging

Mauro Passarotto and Ruben Specogna

EMCLab, Polytechnic Department of Engineering and Architecture, Università di Udine, 33100 Udine, Italy

This contribution addresses the solution of eddy-current problems by means of a volume integral formulation based on the electric vector potential on a computational domain that exhibits a cyclic symmetry. Even if grids discretizing the domain are typically composed of tetrahedral or hexahedral elements, the proposed approach also works for general polyhedral meshes, such as those ones obtained by *subgridding*. In this article, an algorithm to compute a set of suitable cohomology generators needed when the conductors are not simply connected is introduced first. Besides being purely combinatorial, with linear-time worst case complexity and suitable with polyhedral meshes, it reuses a code that computes generators for triangular surface meshes, with obvious advantages concerning the implementation effort. Second, the formulation and the algorithm for cohomology computation are tweaked to be able to solve eddy-current problems with cyclic symmetry reserving specific attention to the construction of suitable tree–cotree decomposition for the problem gauging.

Index Terms— **Cohomology, cyclic symmetry, eddy currents, gauging, partial element equivalent circuit (PEEC), subgridding, volume integral formulation.**

I. Introduction

INTEGRAL and partial element equivalent circuit (PEEC) formulations have the advantage of limiting the discretized domain of a given eddy-current problem with non-magnetic materials to the conducting parts only. A popular integral formulation to solve eddy-current problems by using the electric vector potential on tetrahedral or hexahedral meshes has been introduced in [1]. In [2], a similar formulation has been proposed that, however, uses *cohomology theory* [3] to rigorously treat non-simply connected conductors and it is suitable for general polyhedral meshes. Polyhedral meshes have the potential of seamlessly dealing with hybrid meshes made of tetrahedra, hexahedra, and all other possible shapes. Polyhedral elements also enable the *subgridding* for adaptive mesh refinement without recurring to hanging nodes or refinement of adjacent elements.

In this article, we solve eddy-current problems with the formulation of [2] on polyhedral meshes built, for example, with *subgridding*. We first propose a novel algorithm to compute the *cohomology generators* [3] required when the conductors are not simply connected. The main advantage of this new algorithm, besides being very fast, is that it reuses the code for triangular meshes previously developed, thus minimizing the new implementation effort required when dealing with hexahedral or general polyhedral meshes.

The second contribution of this article, as a natural continuation of [4], is a technique to exploit cyclic symmetry [5] applied to the volume integral formulation [2] in the presence of a polyhedral grid and non-simply connected conductors. Moreover, by extending the idea of [4] to a volume integral formulation, the new non-trivial issue of a proper tree–cotree decomposition for the problem *gauging* [1] arises. For this reason, in this article, it is shown how to build a cyclic symmetric and consistent tree on the symmetry cell only, without spoiling the problem symmetry. In conclusion, this approach can be used in addition to the well-known matrix compression techniques, as the adaptive cross approximation [6] to reduce storage requirement for the system matrix, which is, in case of integral formulations, a fully populated matrix.

The rest of this article is organized as follows. In Section II, the volume integral formulation [2] is recalled. Section III presents the novel algorithm for cohomology computation when the mesh is composed of general polyhedral elements. Section IV describes how to exploit the cyclic symmetry and how to gauge the problem appropriately with suitable tree–cotree decomposition. Finally, in Section V, the numerical results are presented.

II. Volume Integral Formulation

The polyhedral mesh $\mathcal{K}$ covering the conductive regions $\boldsymbol{\Omega}_c$ is composed by N_v polyhedral volume elements, N_f polygonal faces, N_e edges, and N_n nodes. The incidences of these oriented geometrical elements are stored in the usual element-face $\mathbf{D}$ and face-edge $\mathbf{C}$ incidence matrices. The vector of currents flowing through the mesh faces is

$$\mathbf{I} = \mathbf{CT} + \mathbf{Wi} \tag{1}$$

where the degrees of freedom (DoFs) array $\mathbf{T}$ stores the integral of the electric vector potential on mesh edges, and $\mathbf{i}$ the array of independent currents [2] and the columns of $\mathbf{W}$ store the representatives of generators of the second relative cohomology group $H^2(\mathcal{K}, \partial\mathcal{K})$ [3], see Fig. 1(a). The second relative cohomology group allows, by its very definition, to span solenoidal fields tangent to $\partial\mathcal{K}$ that are not curl of something. Thus, the term $\mathbf{Wi}$ of (2), relative to cohomology, together with $\mathbf{CT}$, allows to represent all possible solenoidal currents $\mathbf{I}$. As an example, the $\mathbf{W}$ for a solid torus is formed by

Manuscript received August 18, 2019; revised October 17, 2019; accepted November 5, 2019. Date of current version January 20, 2020. Corresponding author: M. Passarotto (e-mail: passarotto.mauro@spes.uniud.it).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMAG.2019.2953395

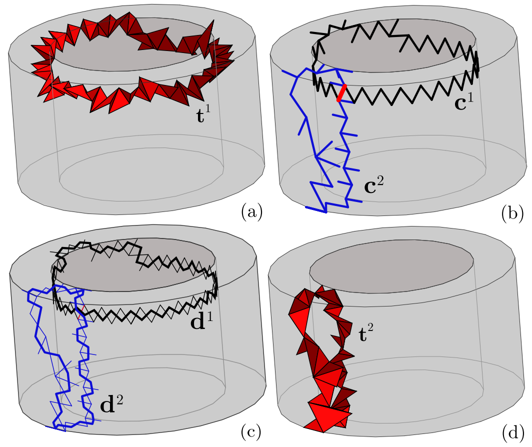

Fig. 1. Example of cohomology generators for a solid torus $\mathcal{K}$. (a) *Support* [7] of a representative $\mathbf{t}^1$ of the $H^2(\mathcal{K}, \partial\mathcal{K})$ generator. It can be thought as a *thinned* unit current that flows around the torus. In this example, $\mathbf{t}^1$ is obtained as $\mathbf{t}^1 = \mathbf{C}\mathbf{c}^1$, where $\mathbf{c}^1$ is represented in (b). (b) Support of two representatives of $H_1(\partial\mathcal{K})$ generators. The thick red edge belongs to both supports of $\mathbf{c}^1$ and $\mathbf{c}^2$. They correspond to the poloidal and toroidal currents that flow on $\partial\mathcal{K}$. (c) $\mathbf{d}^i$, with $i \in \{1, 2\}$, is the cycle made of dual edges which are dual to $\mathbf{c}^i$ in $\partial\mathcal{K}$. $\mathbf{d}^1$ is homologically trivial in $\mathbb{R}^3\backslash\mathcal{K}$, whereas $\mathbf{d}^2$ is trivial in $\mathcal{K}$. (d) $\mathbf{t}^2 = \mathbf{C}\mathbf{c}^2$ is trivial in $H^2(\mathcal{K}, \partial\mathcal{K})$.

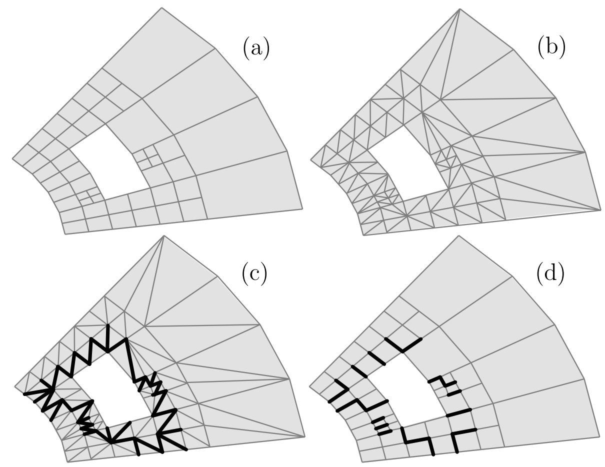

Fig. 2. Zoom on a *part* of the polygonal mesh of $\partial\mathcal{K}$, where the cohomology computation has to be performed. The four steps of the novel algorithm for cohomology computation for polyhedral meshes are shown.

a single column whose entries, interpreted as electric current DoFs, form a unit *thinned* current that flows around the torus, see Fig. 1(a). The independent currents $\mathbf{i}$ are not known, and thus, they form additional unknowns of the problem.

The formulation requires the automatic computation of $\mathbf{W}$ through a fast and general algorithm. For efficiency, it is preferable to construct $\mathbf{W}$ by working on $\partial\mathcal{K}$ only, simply because there are fewer geometric elements to process in $\partial\mathcal{K}$ than in the whole $\mathcal{K}$. Moreover, the algorithms are intrinsically simpler when working on manifold combinatorial surfaces, such as $\partial\mathcal{K}$. This is the reason why matrix $\mathbf{H}$, whose columns store some of the representatives of generators of the first cohomology group $H^1(\partial\mathcal{K})$ [3], is usually employed. Yet, the major difficulty here is that the $H^1(\partial\mathcal{K})$ cohomology group produces twice the number of generators of an $H^2(\mathcal{K}, \partial\mathcal{K})$ basis. For example, when dealing with a solid torus, as shown in Fig. 1(b), the two boundary generators correspond to the poloidal and toroidal currents that flow in $\partial\mathcal{K}$. It is always possible to find a basis of $H^1(\partial\mathcal{K})$ generators such that half of the generators have the dual cycle (i.e., a cycle formed by *dual edges* [3] in $\partial\mathcal{K}$) in $\partial\mathcal{K}$ that is *homologically trivial* in the insulator $\mathbb{R}^3\backslash\mathcal{K}$ [i.e., the cycle is the boundary of a two-chain on the *dual complex* [3] whose support lies inside $\mathbb{R}^3\backslash\mathcal{K}$, such as the dual cycle $\mathbf{d}^1$ in Fig. 1(c)] and the ones whose dual cycle in $\partial\mathcal{K}$ are homologically trivial in the conductor mesh $\mathcal{K}$ [such as the dual cycle $\mathbf{d}^2$ in Fig. 1(c)]. Only the formers produce an $H^2(\mathcal{K}, \partial\mathcal{K})$ basis when pre-multiplied by $\mathbf{C}$, see Fig. 1(a) and [7]. The others, such as the one in Fig. 1(d), must be discarded to obtain a full-rank system. A technique to find the required change of cohomology basis to obtain the matrix $\mathbf{H}$ has been described in [7]. Consequently, $\mathbf{W}$ is computed as $\mathbf{W} = \mathbf{CH}$ and the current is thus represented with

$$\mathbf{I} = \mathbf{C}\,(\mathbf{T} + \mathbf{H}\mathbf{i})\,. \tag{2}$$

Let us define the matrix

$$\mathbf{K} = \mathbf{C}^T\,(\mathbf{R} + i\omega\mathbf{M})\,\mathbf{C} \tag{3}$$

and $\mathbf{b}_s = -i\omega\mathbf{C}^T\tilde{\mathbf{A}}_s$, where $\mathbf{R}$ and $\mathbf{M}$ are, respectively, the resistance and inductance constitutive matrices whose computation is detailed in [2], whereas $\tilde{\mathbf{A}}_s$ is the integral of the magnetic vector potential along the dual-grid edges due to a source of the magnetic field. Let us also define $\tilde{\mathbf{U}}$ as the array of electromotive forces along dual edges and $\tilde{\Phi}$ the magnetic flux through dual faces. The magnetic flux is written as $\tilde{\Phi} = \mathbf{C}^T(\tilde{\mathbf{A}}+\tilde{\mathbf{A}}_s)$, where $\tilde{\mathbf{A}}$ is the circulation of the unknown magnetic vector potential on mesh edges.

By enforcing the discrete Faraday's law locally as $\mathbf{C}^T\tilde{\mathbf{U}} + i\omega\tilde{\Phi} = \mathbf{0}$ and globally as $\mathbf{H}^T(\mathbf{C}^T\tilde{\mathbf{U}} + i\omega\tilde{\Phi}) = \mathbf{0}$, and by considering the discrete constitutive relations $\tilde{\mathbf{U}} = \mathbf{RI}$ and $\tilde{\mathbf{A}} = \mathbf{MI}$, the complete set of equations reads as

$$\begin{bmatrix} \mathbf{K} & \mathbf{KH} \\ \mathbf{H}^T\mathbf{K} & \mathbf{H}^T\mathbf{KH} \end{bmatrix} \begin{bmatrix} \mathbf{T} \\ \mathbf{i} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_s \\ \mathbf{H}^T\mathbf{b}_s \end{bmatrix}. \tag{4}$$

Concerning boundary conditions, we set to zero the entries of the array $\mathbf{T}$ relative to edges on $\partial\mathcal{K}$. To reduce the unknowns, we apply a tree–cotree gauge [1] by setting to zero the entries of the array $\mathbf{T}$ on a suitable tree inside $\mathcal{K}$.

III. New Algorithm for Cohomology Computation

There are various types of software that compute cohomology for triangulated surfaces, see, for example, [8]–[10] or [11]. On the contrary, as far as we know, there are no off-the-shelf implementations for more general meshes.

Motivated by minimizing the implementation effort, we propose here a way to reuse any implementation for computing generators on triangulated surfaces. The idea is summarized in the following four steps, see Fig. 2.

1) First, the boundary of the conductor mesh is considered, see a part of such a combinatorial surface in Fig. 2(a).
2) Second, the boundary mesh is partitioned into triangles obtaining a triangulated surface, see Fig. 2(b). Let P be a polygon represented by an ordered list of vertices $v_0, v_1, v_2, \ldots, v_n$, see an example in Fig. 3(a). A chord

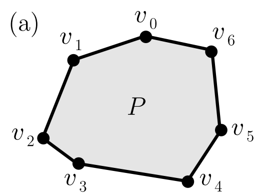

Fig. 3. (a) Polygon P. (b) Partition of P into triangles without adding new nodes.

in P is a line segment that connects two non-adjacent vertices in P. A triangulation of P into $n-2$ triangles is obtained by drawing $n-3$ chords, see Fig. 3(b). Then, the triangles are $v_0-v_1-v_2, v_0-v_2-v_3, v_0-v_3-v_4, \ldots, v_0-v_{n-1}-v_n$. We remark that since we need a topological partitioning (i.e., the coordinates of vertices are irrelevant), the same algorithm is used in the case of concave polygons.

3) Third, any already available software is used to get cohomology generators for the obtained triangulated surface, such as [8]–[10] or [11]. The support of a possible representative of a cohomology generator $H_1(\partial\mathcal{K})$ is shown in Fig. 2(c).
4) Finally, the fourth step just loads the mesh edges with non-zero coefficients in the representative of the cohomology generator of the triangulated surface. Each edge is described as a pair of nodes that form its boundary. If the edge, identified with the pair of nodes, is present in the polygonal mesh, we assemble the coefficient to the matrix $\mathbf{H}$; otherwise, the coefficient is discarded.

It is clear that if step 3) of the previous algorithm is performed in linear time, as suggested in [7], the worst case complexity of the whole algorithm is linear with the number of elements.

Let us show that each array $\mathbf{c}$ of coefficients obtained with this algorithm is a representative of a cohomology generator for the polyhedral mesh.

We have to show first that the one-cochain $\mathbf{c}$ is a one-cocycle (i.e., a curl-free discrete field). Formally, $\mathbf{c}$ is such that $\mathbf{C}_b\mathbf{c} = \mathbf{0}$, where $\mathbf{C}_b$ is the face-edge incidence matrix restricted to $\partial\mathcal{K}$. This is equivalent to show that, for each polygon P of $\partial\mathcal{K}$, the sum of the coefficients in ∂P is zero. This is indeed the case since P is a sum of triangles (thus, a linear combination) and each triangle has zero circulation in its boundary, since the coefficients computed by a software for cohomology computation on the triangulation fulfill $\mathbf{C}_\triangle\mathbf{c}_\triangle = \mathbf{0}$, where $\mathbf{C}_\triangle$ and $\mathbf{c}_\triangle$ are the face-edge incidence matrix and the one-cocycle relative to the triangulation, see Fig. 4(a) and (b).

Finally, the dual cycle in $\partial\mathcal{K}$ with respect to $\mathbf{c}$ is clearly homologous to the dual cycle of $\mathbf{c}_\triangle$, since they differ by a boundary, see Fig. 4(c) and (d).

IV. Exploiting Cyclic Symmetry

When the conductor has a *cyclic symmetry* [5], it is wise to solve the problem on $\mathcal{K}$ by solving a family of problems on the *symmetry cell* $\mathcal{S}$, see Fig. 5(a) and (b). This is performed by using the *discrete Fourier transform* [6]. Once that the matrix becomes block circulant, the standard technique recalled in [6]

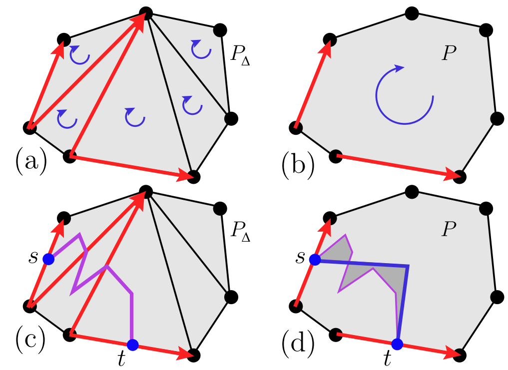

Fig. 4. Generic polygon P and its triangulation $P_\triangle$. (a) Thick edges (in red) have a coefficient different than zero in a one-cochain $\mathbf{c}_\triangle$ obtained by software for cohomology computation. They are one-cocycles, and therefore, the circulation of the coefficients on the boundary of each triangle is zero. (b) P is a sum of triangles, and therefore, the circulation of the coefficients in the boundary of P is again zero. (c) Part inside $P_\triangle$ of the dual cycle which is dual to $\mathbf{c}_\triangle$. (d) Part inside P of the dual cycle which is dual to $\mathbf{c}$. Both parts of the dual cycles start in point s and end in point t. They are homologous in each polygon P since they differ by a boundary (the dark gray area).

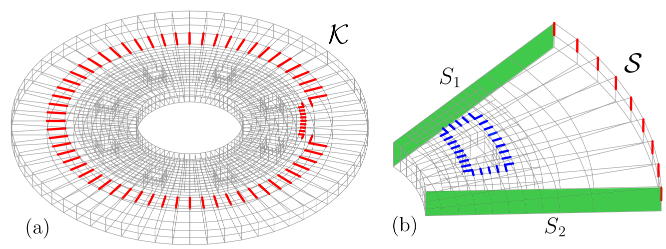

Fig. 5. Support of a cohomology generator for (a) complete geometry $\mathcal{K}$—a disk with nine holes—and (b) supports of the two generators for the symmetry cell $\mathcal{S}$.

can be used to find the solution by solving one or more problems in the symmetry cell $\mathcal{S}$ only.

The main point is that the matrix of the considered volume integral formulation is *not* block circulant in general, and therefore the exploitation of cyclic symmetry is still an open issue. Indeed, the part due to the non-local Faraday's laws originating from cohomology theory spoils the symmetry of the problem. This has been recognized recently by Rubinacci *et al.* [12], where they propose eliminating the non-block-circulant blocks by using a costly algorithm.

One of the purposes of this article is to show that it is possible to write the system directly in a block-circulant form instead, thus avoiding the unnecessary complications of [12]. It is easy to realize that to obtain a block-circulant matrix, the representatives of the cohomology generators have to *share the same cyclic symmetry* as the geometry. Fig. 5(a) represents the support of a representative of a cohomology generator in $\mathcal{K}$, which does not share the symmetry of the domain. To build representatives with the appropriate geometric symmetry, we use an idea similar to what was proposed in [4] for a boundary integral formulation. Let us define a novel complex $\mathcal{S}'$ as the *quotient space* [3] of $\mathcal{S}$ formed by identifying the first symmetry boundary S_1 of $\mathcal{S}$ with the other symmetry boundary S_2, see Fig. 5(b). This means simply that we *glue* S_1 with S_2, i.e., the labels of all geometric elements on S_2 are

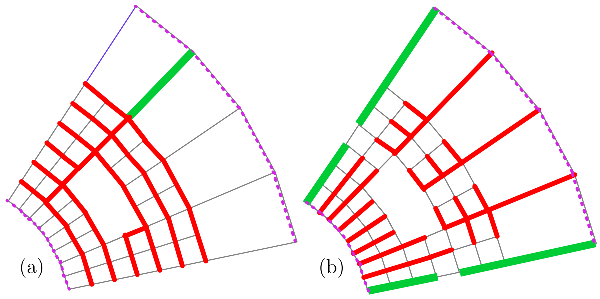

Fig. 6. Top view of the 3-D $\mathcal{S}$ of Fig. 5(b). (a) Standard tree on $\mathcal{S}$ is not a tree on the full domain $\mathcal{K}$. (b) Proposed tree on $\mathcal{S}$ is symmetric and extends as a tree on the full domain $\mathcal{K}$.

changed with the ones of the corresponding elements on S_1. Then, one may use any algorithm for cohomology computation on $\mathcal{S}'$. The representatives of the cohomology generators on $\mathcal{S}$ are obtained from the ones on $\mathcal{S}'$ by copying the coefficients on edges of S_1 to the edges on S_2. Fig. 5(b) represents the support of the generators on $\mathcal{S}$ obtained by using the proposed technique. We remark that there is no assumption on the topology of S_1 and S_2.

Another contribution of this article is to show that also the spanning tree on $\mathcal{K}$ for the problem gauging should be carefully constructed to share the same symmetry of the problem, again in order to get a block-circulant matrix. Gauging in the considered context is useful to reduce the number of unknowns and to get a full-rank linear system.

To get a spanning tree that shares the same symmetries of the whole domain and for efficiency, it is desirable to build the spanning tree on $\mathcal{S}$ only. The spanning tree on the whole $\mathcal{K}$ is then obtained after all symmetry cells are glued together to obtain $\mathcal{K}$. For example, if one applies the classical recipe of [1], one may produce a tree in the boundary $\partial\mathcal{K} \cap \mathcal{S}$ first, see the violet dotted edges in Fig. 6(a). We recall that the tree has to be built first on the boundary to be able to apply the boundary conditions, see [1] for more details. Then, the tree is made in the interior (in red), and finally, one needs to add one edge to each connected component of $\partial\mathcal{K} \cap \mathcal{S}$ (the thick green edge). As can be guessed from Fig. 6(a), this is a spanning tree on $\mathcal{S}$, but it is not suitable for our purposes since it produces cycles when the symmetry cells are glued together to obtain $\mathcal{K}$.

The solution we propose to solve this problem is to find the tree on $\mathcal{S}'$. First, a tree is produced on $\partial\mathcal{S}'$; see the violet dotted edges in Fig. 6(b). Then, the tree is extended in the interior of $\mathcal{S}'$. Considering Fig. 6(b), the thick green edges are the parts of the tree on S_1 and S_2, which are the same since the edges on S_1 and S_2 are topologically identified as the same edges. This identification allows, similarly to what is performed for the generators, to produce a tree for $\mathcal{S}$ starting from the tree on $\mathcal{S}'$. It is clear that this tree cannot exhibit a cycle when extended on $\mathcal{K}$ since such a tree would form a cycle also on $\mathcal{S}'$.

V. Numerical Results

In Fig. 7, we show the real current density $Re\{\mathbf{J}\}$ flowing in the structure when a symmetric source of magnetic field

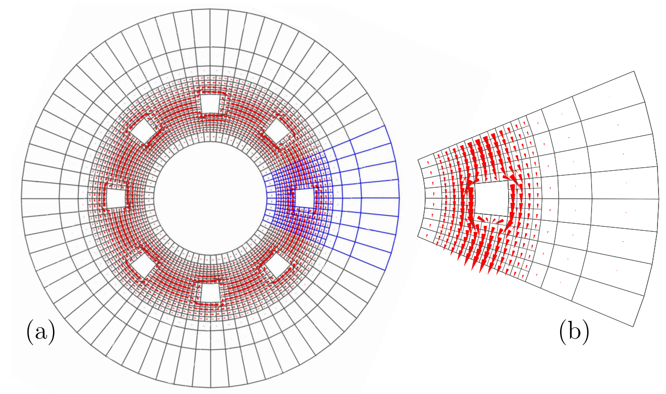

Fig. 7. (a) Real part of the current density $Re\{\mathbf{J}\}$ on the conductor $\mathcal{K}$ and (b) $Re\{\mathbf{J}\}$ in the symmetry cell $\mathcal{S}$.

is positioned above the conducting domain Ω_c. The coil is powered with a current $I_c = 100$ At at a frequency $f = 200$ Hz. The current computed using the complete structure $\mathcal{K}$ is the same as the one computed using the symmetric cell $\mathcal{S}$ up to the linear solver tolerance.

Acknowledgment

This work was supported by Higher Education and Development Op. 1 of the Friuli-Venezia Giulia region.

References

[1] R. Albanese and G. Rubinacci, “Integral formulation for 3D eddy-current computation using edge elements,” *Proc. Inst. Elect. Eng. A Phys. Sci., Meas. Instrum., Manage. Educ., Rev.*, vol. 135, no. 7, pp. 457–462, Sep. 1988.

[2] P. Bettini, M. Passarotto, and R. Specogna, “A volume integral formulation for solving eddy current problems on polyhedral meshes,” *IEEE Trans. Magn.*, vol. 53, no. 6, Jun. 2017, Art. no. 7204904.

[3] J. R. Munkres, *Elements of Algebraic Topology*. New York, NY, USA: Addison-Wesley, 1984.

[4] B. Kapidani, M. Passarotto, and R. Specogna, “Exploiting cyclic symmetry in stream function-based boundary integral formulations,” *IEEE Trans. Magn.*, vol. 55, no. 6, Jun. 2019, Art. no. 7200504.

[5] A. Bossavit, “Symmetry, groups, and boundary value problems. A progressive introduction to noncommutative harmonic analysis of partial differential equations in domains with geometrical symmetry,” *Comput. Methods Appl. Mech. Eng.*, vol. 56, no. 2, pp. 167–215, 1986.

[6] S. Kurz, O. Rain, and S. Rjasanow, “Application of the adaptive cross approximation technique for the coupled BE-FE solution of symmetric electromagnetic problems,” *Comput. Mech.*, vol. 32, nos. 4–6, pp. 423–429, 2003.

[7] P. Dłotko and R. Specogna, “Physics inspired algorithms for (co)homology computations of three-dimensional combinatorial manifolds with boundary,” *Comput. Phys. Commun.*, vol. 184, no. 10, pp. 2257–2266, Oct. 2013.

[8] P. Bettini and R. Specogna, “A boundary integral method for computing eddy currents in thin conductors of arbitrary topology,” *IEEE Trans. Magn.*, vol. 51, no. 3, Mar. 2015, Art. no. 7203904.

[9] B. Kapidani, P. Dłotko, P. Alotto, P. Bettini, and R. Specogna, “Computation of relative 1-cohomology generators from a 1-homology basis for eddy currents boundary integral formulations,” *IEEE Trans. Magn.*, vol. 52, no. 10, Oct. 2016, Art. no. 7210006.

[10] M. Pellikka, S. Suuriniemi, L. Kettunen, and C. Geuzaine, “Homology and cohomology computation in finite element modeling,” *SIAM J. Sci. Comput.*, vol. 35, no. 5, pp. B1195–B1214, 2013.

[11] T. K. Dey, J. Sun, and Y. Wang, “Approximating cycles in a shortest basis of the first homology group from point data,” *Inverse Problems*, vol. 27, no. 12, 2011, Art. no. 124004.

[12] G. Rubinacci, A. Tamburrino, S. Ventre, and F. Villone, “Application of n-fold rotational symmetries to eddy currents integral model in the time domain,” *IEEE Trans. Magn.*, to be published.