Computer Physics Communications 265 (2021) 108004

Contents lists available at ScienceDirect

Computer Physics Communications

www.elsevier.com/locate/cpc

Optimized cycle basis in volume integral formulations for large scale eddy-current problems ☆

Dimitri Voltolina [a,*], Riccardo Torchio [b], Paolo Bettini [a,b], Ruben Specogna [c], Piergiorgio Alotto [b]

[a] Centro Ricerche Fusione (CRF), University of Padova, 35127 Padova, Italy
[b] Department of Industrial Engineering, University of Padova, 35131 Padova, Italy
[c] Polytechnic Department of Engineering and Architecture, University of Udine, 33100 Udine, Italy

ARTICLE INFO

Article history:
Received 15 June 2020
Received in revised form 3 February 2021
Accepted 19 April 2021
Available online 26 April 2021

Keywords:
Volume integral formulations
Eddy-current problems
Cohomology
Multiply connected domains
Low-rank approximation
Thermonuclear fusion devices

ABSTRACT

We present a Volume Integral formulation for the solution of large scale eddy-current problems coupled with low-rank approximation techniques. Two alternative approaches are introduced to map the problem unknowns into a subset of grid elements forming a base of global or mixed (global and local) cycles, respectively, and guarantee the well-posedness of the problem both in simply and multiply connected domains. The paper shows that the adoption of mixed cycles is computationally more efficient than global ones. In particular, integral formulations based on global cycles cannot be safely coupled with low-rank approximation techniques, which, however, are crucial to increase the size of the largest solvable problem, like the ones involving conducting structures in magnetic confinement fusion devices. The aim of this paper is to demonstrate how such bottleneck can be overcome by considering local and global cycles differently, on the basis of the cohomology theory. An improved, efficient, and robust algorithm for computing a base of global cycles is described in detail. In particular, the presented algorithm is able to almost minimize the cohomology basis length, i.e. the number of mesh edges forming such a basis, in order to allow an efficient solution of large scale problems. Furthermore, a novel and general method to handle global and local cycles together, in the context of low-rank approximated matrices, is shown to be efficient for the solution of large scale eddy-current problems in multiply connected domains. Along the manuscript, pseudo-codes are given, which clarify the proposed methods and help to implement them by Volume Integral Equation practitioners.

1. Introduction

Integral formulations have shown to be suitable alternatives to usual Finite Element Methods (FEM) for the solution of eddy-current problems involving complex 3D conducting structures, especially when embedded in large non-conducting regions [1–3]. An example is represented by magnetic confinement fusion (MCF) devices, where large metallic structures can include a lot of small and tailored components (e.g. the ports for diagnostics, heating and current drive and vacuum systems, the insulations gaps, the divertor plates, etc.). The study of this kind of devices requires both high solution accuracy and high computational efficiency (in terms of solution time and memory) of the formulation, in order to solve large scale problems on standard workstations [4].

In this regard, we compare two Volume Integral approaches, which rely on a discrete reformulation of Maxwell's equations over a pair of interlocked grids [5,6] and start from a common discrete equation referred as Electric Field Integral Equation (EFIE). Despite both approaches give the same numerical results, the different techniques adopted to impose the solenoidality condition of the current density field lead to a different mapping of the problem unknowns on a basis of grid cycles. The former, called **L**-approach in this paper, strictly relies on a circuit interpretation of the eddy-current problem, as typical of Partial Element Equivalent Circuit (PEEC) methods [7–10]. The problem is solved by the direct application of the loop current method of circuit theory, without the introduction of any vector potential, hence the well-posedness is guaranteed both in simply and multiply connected domains. The latter, called **C**-approach, solves the eddy-current problem by the common definition of the electric vector potential [11]. The well-posedness of the problem in multiply connected domains is addressed with cohomology theory [12], resulting into a topological pre-processing, required by the formulation, to identify a basis of

☆ The review of this paper was arranged by Prof. N.S. Scott.

* Corresponding author.
E-mail address: dimitri.voltolina@studenti.unipd.it (D. Voltolina).

https://doi.org/10.1016/j.cpc.2021.108004
0010-4655/

the first cohomology group generators of the conductor's boundary [13].

The aim of the paper is to investigate which approach performs better in the solution of large scale problems, with a particular focus on MCF devices. The coupling of volume integral formulations with low-rank approximation techniques has already demonstrated to be an efficient computational method to increase the size of the largest solvable problem on standard workstations [4,14,15]. However, the integral approach has to be carefully formulated in order to obtain an efficient, but at the same time accurate, approximation of the dense matrices arising from the chosen approach. As regards the **C**-approach, starting from the algorithm described in [16], a new robust algorithm is proposed for the computation of the first group cohomology basis of the conductor's boundary, which ideally matches with the adopted low-rank approximation method. The obtained *optimal* cohomology basis allows for reducing the computational costs of system assembling and solution, without losing the physical properties introduced by the cohomology generators. Moreover, we also propose a clustering technique which allows for efficiently applying low-rank compression techniques in the **C**-approach.

The algorithm is implemented on the basis of graph theory and of the duality between the interlocked grids. Hence, without losing any formality, it can be easily followed by those unfamiliar with homology and cohomology theories [12]. The algorithm is robust and efficient, since it requires algebraic operations with integer numbers only, avoiding any loss of numerical precision arising from the use of floating point arithmetic. Moreover, in contrast with previous approaches such as [17], it computes an *optimal* basis, in the sense that, first, it almost minimizes the basis length, i.e. the number of mesh edges which form such a basis, allowing a significant speed-up in the system assembling, which otherwise could represent a bottleneck in the problem solution. Moreover, the computed basis provides a full rank system to be solved, with a time consumption of a few seconds for the basis computation in meshes of tens of thousands of elements, which is essentially negligible if compared to the time required by the system assembly and solution.

On the contrary, the adoption of a global basis in the **L**-approach does not allow an efficient and correct coupling with low-rank approximation techniques, thus limiting the applicability of the **L**-approach to large scale problems.

The paper is organized as follows. In Section 2, the equations of the eddy-current problem are derived for the two presented approaches. In Section 3 and Section 4, the two approaches are explained in detail. Section 5 rigorously describes the implemented algorithm for computing an optimal basis of the first cohomology group, while Section 6 deals with the coupling with low-rank approximation techniques. Numerical results are presented in Section 7 as regards the numerical accuracy and the computational performances of the approaches, and in Section 8 regarding the computational costs of the cohomology basis computation. Finally, in Section 9, conclusions are drawn.

2. Volume Integral formulation

Let $\Omega \subset \mathbb{R}^3$ be a conducting domain with electric resistivity $\rho(\mathbf{r})$, with $\mathbf{r} \in \Omega$. Assuming a magneto quasi-static (MQS) approximation of the electromagnetic fields, the following set of equations holds for every point $\mathbf{r} \in \Omega$:

$$\nabla \times \mathbf{e} = -i\omega \mathbf{b} \tag{1}$$

$$\nabla \times \mathbf{h} = \mathbf{j} \tag{2}$$

$$\nabla \cdot \mathbf{b} = 0 \tag{3}$$

$$\nabla \cdot \mathbf{j} = 0 \tag{4}$$

$$\mathbf{e} = \rho \mathbf{j}, \tag{5}$$

where ω is the angular frequency, $\mathbf{j}$ is the eddy-current density, $\mathbf{h}$ and $\mathbf{b}$ are the magnetic field and the magnetic flux density, respectively. Taking into account the solenoidality condition of the magnetic flux density (3), Faraday's law (1) is rewritten as

$$\nabla \times \mathbf{e} + i\omega \nabla \times \mathbf{a} = -i\omega \nabla \times \mathbf{a}_0, \tag{6}$$

where $\mathbf{a}$ and $\mathbf{a}_0$ are the magnetic vector potential due to the eddy-currents and the external currents, respectively. The previous equation can be rewritten, introducing the electric scalar potential ϕ, as follows:

$$\mathbf{e} + i\omega \mathbf{a} + \nabla \phi = -i\omega \mathbf{a}_0. \tag{7}$$

We recall that $\mathbf{a}$ and ϕ can be expressed as functions of the current density $\mathbf{j}$ by the following integral expressions [18]:

$$\mathbf{a}(\mathbf{r}) = \mu_0 \int_\Omega g(\mathbf{r}, \mathbf{r}') \mathbf{j}(\mathbf{r}') \, d\mathbf{r}' \tag{8}$$

$$\phi(\mathbf{r}) = -\frac{1}{i\omega\varepsilon_0} \int_\Omega g(\mathbf{r}, \mathbf{r}') \nabla' \cdot \mathbf{j}(\mathbf{r}') \, d\mathbf{r}' - \frac{1}{i\omega\varepsilon_0} \int_{\partial\Omega} g(\mathbf{r}, \mathbf{r}') \mathbf{j}(\mathbf{r}') \cdot \mathbf{n}(\mathbf{r}') \, d\mathbf{r}', \tag{9}$$

where $\mathbf{r}' \in \Omega$ is the source point, $g = (4\pi |\mathbf{r} - \mathbf{r}'|)^{-1}$ is the free-space Green's function (i.e. the fundamental solution of Maxwell's equations in vacuum [19,9]), $\mathbf{n}$ is the unit vector normal to the domain boundary in $\mathbf{r}'$, and $\nabla' \cdot$ indicates that the divergence operator is applied on the source point $\mathbf{r}'$. By using (8) and (9) in (7), the integral equation, usually referred as *Electric Field Integral Equation* (EFIE), is obtained, which relates the total electric field and the scattered electric field (i.e. the term $j\omega\mathbf{a} + \nabla\phi$) to the eddy-current field $\mathbf{j}$. Moreover, (4) implies that no charge accumulations can appear in the conducting domain Ω, thus the MQS-EFIE (7) becomes

$$\mathbf{e} + i\omega \mathbf{a} = -i\omega \mathbf{a}_0. \tag{10}$$

The above equation is the common starting point for the solution of eddy-current problems by the two integral formulations presented in this work.

In both approaches, the discretization of the electromagnetic fields is carried out over a pair of interlocked grids, through the natural association of physical variables to the grid elements [5,6]. However, different approaches can be followed to apply condition (4), resulting into different algebraic systems of discrete equations.

Let $\mathcal{K}$ be a volumetric grid of the domain Ω, consisting of N_n nodes, N_e edges, N_f faces and N_v volumes. Let us define with n_i, e_i, f_i, v_i the generic i-th node, edge, face and volume of $\mathcal{K}$, respectively. The sparse matrices **G**, **C** and **D** store the topological incidences between the pairs edges-nodes, faces-edges and volumes-faces of $\mathcal{K}$, respectively, providing a discrete form of the gradient, curl, and divergence operators in $\mathcal{K}$ [5]. A *dual grid* $\tilde{\mathcal{K}} = D(\mathcal{K})$, where D is the duality operator [5], is constructed by taking the barycentric subdivision of $\mathcal{K}$, which results into $\tilde{N}_n = N_v$ dual nodes, $\tilde{N}_e = N_f$ dual edges and $\tilde{N}_f = N_e$ dual faces. Similarly to primal grid elements, $\tilde{n}_j$, $\tilde{e}_j$, $\tilde{f}_j$ correspond to the j-th node, edge, and face of $\tilde{\mathcal{K}}$, respectively.

The following relations hold between incidence matrices of the interlocked grids:

$$\tilde{\mathbf{G}} = -\mathbf{D}^T \tag{11}$$

$$\tilde{\mathbf{C}} = \mathbf{C}^T. \tag{12}$$

According to Tonti's scheme for Maxwell's equations [5], the current density field $\mathbf{j}$ is expanded by means of face basis functions:

$$\mathbf{j}(\mathbf{r}) = \sum_{j=1}^{N_f} \mathbf{w}_j(\mathbf{r}) I_j, \tag{13}$$

where $\mathbf{w}_j$ is the vector basis function associated to f_j and I_j is the current across such face. When the grid consists of standard elements, like tetrahedra or hexahedra, one may use standard Whitney face basis functions [20], like in [1]. Alternatively, the geometric basis functions proposed in [21] are suitable for general polyhedral elements. By using (13), (5) and (8) inside (10), one gets

$$\sum_{j=1}^{N_f} I_j \left[\rho(\mathbf{r}) \mathbf{w}_j(\mathbf{r}) + i\omega\mu_0 \int_\Omega g(\mathbf{r}, \mathbf{r}') \mathbf{w}_j(\mathbf{r}') d\mathbf{r}' \right] = -i\omega \mathbf{a}_0(\mathbf{r}). \tag{14}$$

A Galerkin testing of the previous equation yields the following system of linear equations, that enforces the discrete form of the MQS-EFIE on the dual edges of $\tilde{K}$:

$$\begin{cases} (\mathbf{R} + i\omega\mathbf{M})\mathbf{I} = \mathbf{Z}\mathbf{I} = -i\omega\tilde{\mathbf{A}}_0 & \text{(a)} \\ \mathbf{D}\mathbf{I} = -\tilde{\mathbf{G}}^T\mathbf{I} = \mathbf{0} & \text{(b)} \\ I_j = 0, \quad \forall f_j \in \partial\mathcal{K}, & \text{(c)} \end{cases} \tag{15}$$

where the vector $\mathbf{I}$ stores the unknown currents I_i, $i \in \{1, \ldots, N_f\}$, and the generic i-th entry of $\tilde{\mathbf{A}}_0$ is given by

$$\tilde{A}_{0i} = \int_\Omega \mathbf{a}_0 \cdot \mathbf{w}_i d\Omega. \tag{16}$$

Eq. (15b) expresses the discrete counterpart of the solenoidality condition (4), while (15c) represents the restriction of such condition to the domain boundary $\partial\Omega$, given that $\mathbf{j}(\mathbf{r}) \cdot \mathbf{n}(\mathbf{r}) = 0, \forall \mathbf{r} \in \partial\Omega$. Furthermore, the *resistance matrix* $\mathbf{R}$ and the *inductance matrix* $\mathbf{M}$ are introduced. The former expresses the discrete counterpart of the constitutive relation (5), while the latter takes into account the mutual magnetic couplings. A generic (i, j) entry of the local matrices is computed as follows [22,13]:

$$R_{ij} = \int_{\Omega_i} \rho(\mathbf{r}) \mathbf{w}_i(\mathbf{r}) \cdot \mathbf{w}_j(\mathbf{r}) d\mathbf{r} \tag{17}$$

$$M_{ij} = \mu_0 \int_{\Omega_i} \int_{\Omega_j} \mathbf{w}_i(\mathbf{r}) \cdot \mathbf{w}_j(\mathbf{r}') g(\mathbf{r}, \mathbf{r}') d\mathbf{r}' d\mathbf{r}, \tag{18}$$

where $\mathbf{r} \in \Omega_i$, $\mathbf{r}' \in \Omega_j$ and $\Omega_i \subset \Omega$ is the support of $\mathbf{w}_i$, i.e.:

$$\Omega_i = \bigcup_{k=1, v_k \ni f_i}^{N_v} v_k. \tag{19}$$

Different approaches can be developed in order to solve (15a) together with conditions (15b) and (15c). According to the chosen approach, the unknown currents $\mathbf{I}$ are mapped from the primal faces of $\mathcal{K}$ to a different set of topological elements, consisting of closed loops (i.e. cycles) of edges belonging to the primal grid $\mathcal{K}$ or the dual grid $\tilde{\mathcal{K}}$. In this regard, a different topological preprocessing is required to guarantee the uniqueness of the solution, in particular when the domain Ω is not simply-connected.

In the following sections, we develop an eddy-current formulation that directly solves (15a) in terms of loop currents and a second one which relies on the electric vector potential. For both approaches a circuit interpretation exists, given that electro-motive forces are associated to dual edges. Indeed, as commonly done in PEEC formulations [7,8,23], (15a) stores the electro-motive forces balance on each dual edge of $\tilde{K}$, where $\mathbf{R}$ and $\mathbf{M}$ take into account the resistive and inductive voltage drops respectively, while capacitive effects are neglected because of the MQS approximation.

For the sake of simplicity, in introducing the two approaches, we assume that the computational domain Ω consists of a single component (conductive region), the extension to multi-component domain problems (e.g. two disjoint conducting spheres) being straightforward.

3. Loop current approach

The loop current approach, from now on **L**-approach, consists in the solution of the electric circuit representing the eddy-current problem on $\mathcal{K}$ by means of the *loop current method* of circuit theory [24]. In this method, one constructs and solves a system of equations in which *all* the unknowns are *loop currents*, which may be defined as the currents that flow in a cycle basis of the dual edge-dual node graph of $\tilde{\mathcal{K}}$, i.e. a maximal set of independent cycles. Kirchhoff's voltage laws (KVL) are enforced on such a set of independent loops.

Let's consider the connected dual graph $\tilde{\mathcal{G}} \subset \tilde{\mathcal{K}}$ corresponding to the $\tilde{N}_e$ edges and the $\tilde{N}_n$ nodes of $\tilde{\mathcal{K}}$. The boundary condition (15c) is simply enforced by removing the dual graph edges associated to the boundary faces of $\mathcal{K}$ (hence, from now on, $\tilde{N}_e$ will correspond to the number of faces $f_j \in \mathcal{K} \setminus \partial\mathcal{K}$). We remind that the rank of matrix $\tilde{\mathbf{G}}$ is $\tilde{N}_n - 1$,[1] hence to derive a set of independent equations one dual node has to be "grounded", i.e. the corresponding column in $\tilde{\mathbf{G}}$ is removed. Given an arbitrary tree-cotree decomposition of $\tilde{\mathcal{G}}$, let $\tilde{\mathbf{G}}_{reo}$ be the reordered incidence matrix of $\tilde{\mathcal{G}}$, obtained by a row-swapping of $\tilde{\mathbf{G}}$, such that

$$\tilde{\mathbf{G}}_{reo} = \begin{bmatrix} \tilde{\mathbf{G}}_t \\ \tilde{\mathbf{G}}_c \end{bmatrix}, \tag{20}$$

where $\tilde{\mathbf{G}}_t$ is the $(\tilde{N}_n - 1) \times (\tilde{N}_n - 1)$ submatrix of $\tilde{\mathbf{G}}$, having as rows only the ones corresponding to a tree of $\tilde{\mathcal{G}}$, while $\tilde{\mathbf{G}}_c$ is the $(\tilde{N}_e - \tilde{N}_n + 1) \times (\tilde{N}_n - 1)$ submatrix of the corresponding cotree. Since matrix $\tilde{\mathbf{G}}_t$ is non-singular, it can be symbolically inverted obtaining the new incidence matrix [25]

$$\tilde{\mathbf{G}}' = \tilde{\mathbf{G}}_{reo} \tilde{\mathbf{G}}_t^{-1} = \begin{bmatrix} \mathbb{1} \\ \tilde{\mathbf{F}} \end{bmatrix}, \tag{21}$$

where $\tilde{\mathbf{F}}$ is a $(\tilde{N}_e - \tilde{N}_n + 1) \times (\tilde{N}_n - 1)$ matrix, named *fundamental cutset matrix*, which stores the incidences between the cotree edges and a set of independent cutset of $\tilde{\mathcal{G}}$. Using (21) and by reordering the vector $\mathbf{I}$ in the same manner as $\tilde{\mathbf{G}}_{reo}$, (15b) holds also for $\tilde{\mathbf{G}}'$ [25]:

$$\tilde{\mathbf{G}}'^T \begin{bmatrix} \mathbf{I}_t \\ \mathbf{I}_c \end{bmatrix} = \tilde{\mathbf{G}}_t^{-T} \tilde{\mathbf{G}}_{reo}^T \begin{bmatrix} \mathbf{I}_t \\ \mathbf{I}_c \end{bmatrix} = \mathbf{0}. \tag{22}$$

[1] If the domain is formed by N_c disjoint subdomains, i.e. $\Omega = \bigcup_{k=1}^{N_c} \Omega_k$, rank of $\tilde{\mathbf{G}}$ would be $\tilde{N}_n - N_c$.

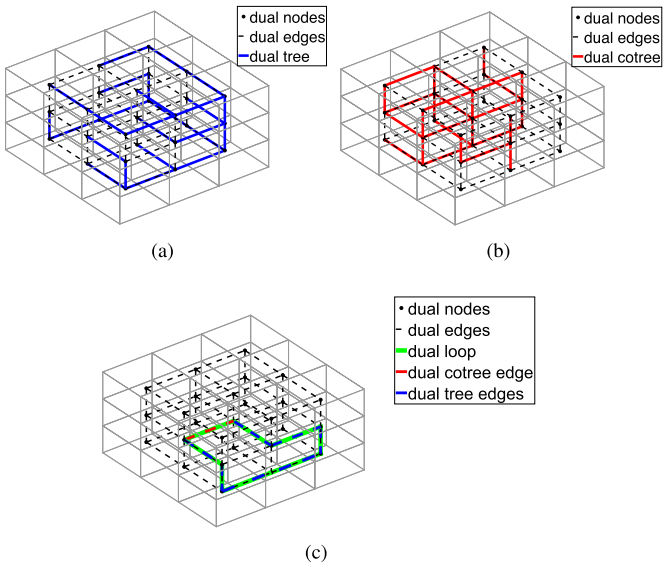


Fig. 1. Example of dual graph tree-cotree decomposition. (a) and (b) show the tree and cotree edges of $\tilde{\mathcal{G}}$, once removed the edges $\tilde{e}_j = D(f_j) : f_j \in \partial\mathcal{K}$, respectively. In (c) one of the independent cycle is shown (green), built up by the relative cotree edge (in red) and other tree edges (in blue). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Then, it is straightforward to prove that the unknown currents associated to the tree edges of $\tilde{\mathcal{G}}$ are linearly dependent from the ones belonging to the cotree edges with

$$\begin{bmatrix} \mathbf{I}_t \\ \mathbf{I}_c \end{bmatrix} = \begin{bmatrix} -\tilde{\mathbf{F}}^T \\ \mathbb{1} \end{bmatrix} \mathbf{I}_c = \tilde{\mathbf{L}}\mathbf{I}_c. \tag{23}$$

In the end, by pre-multiplying (15a) by $\tilde{\mathbf{L}}^T$—which corresponds to the enforcement of the KVL on the dual cycle basis—and by using (23), a new algebraic form of the MQS-EFIE is obtained

$$\left(\tilde{\mathbf{L}}^T \mathbf{Z} \tilde{\mathbf{L}}\right) \mathbf{I}_c = -i\omega \tilde{\mathbf{L}}^T \tilde{\mathbf{A}}_0. \tag{24}$$

Matrix $\tilde{\mathbf{L}}$ maps the current vector $\mathbf{I}$ into a new vector of linearly independent unknowns, which are the loop currents I_{c_i}, $i \in \{1, \ldots, \tilde{N}_e - \tilde{N}_n + 1\}$, associated to the cotree edges of $\tilde{\mathcal{G}}$. Each cotree edge closes an independent cycle $\tilde{z}_i$ formed by itself and the edges of the tree (like the one in Fig. 1c). This is in analogy with the loop current method of circuit theory, where the Kirchhoff's voltage equations are written for a set of independent loops of the circuit graph.

A topological preprocessing is required for the computation of $\tilde{\mathbf{F}}$. Such operation can be achieved by actually inverting $\tilde{\mathbf{G}}_t$ as in (21) or, more efficiently, by computing the reduced row echelon form (RREF) of $\tilde{\mathbf{G}}_{reo}$, since $(\tilde{\mathbf{G}}')^T = RREF(\tilde{\mathbf{G}}_{reo}^T)$. Nevertheless, the resulting matrices in both approaches are affected by matrix fill-in (i.e. loops consisting of a big number of edges are generated), thus the number of non-zeros of $\tilde{\mathbf{F}}$ may increase if compared with the sparse matrix $\tilde{\mathbf{G}}_t$. A technique used to reduce the fill-in of sparse matrices is nested dissection which, in this work, has been implemented through the METIS library [26]. The *ndmetis* program [27] performs a reordering of the $\tilde{\mathcal{G}}$ dual edges such that, once $\tilde{\mathbf{F}}$ is computed, the length (i.e. the number of edges) of each cycle is reduced (Fig. 2).

The described formulation can be easily implemented, as shown in Algorithm 1, without caring about the topological properties of the domain. Indeed, since this approach does not rely on a vector potential, (24) is well-posed and admits a unique solution both in simply-connected and multiply connected domains.

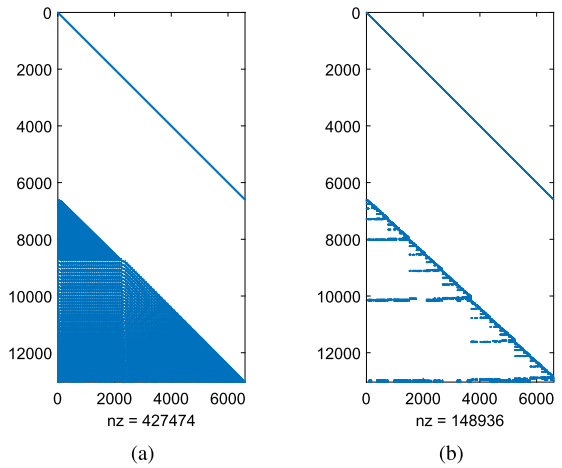


Fig. 2. Non-zero entries (nz) in a 13034×6599 $\tilde{\mathbf{G}}'$ matrix computed without any $\tilde{\mathcal{G}}$ reordering (a) or using *ndmetis* (b).

Algorithm 1 **L**-approach.

Input: $\mathcal{K}$ (primal grid of Ω), ω;
Output: Eddy-current vector in Ω;
1: Compute topological matrices $\mathbf{G}, \mathbf{C}, \mathbf{D}$;
2: Swap $\tilde{\mathbf{G}}$ rows using nested-dissection algorithm (e.g. using METIS library);
3: Construct a tree T of $\tilde{\mathcal{G}}$, and the corresponding cotree C, such that every $\tilde{e}_j = D(f_j)$ with $f_j \in \partial\mathcal{K}$ will be in T;
4: Reorder $\tilde{\mathbf{G}}_{reo}$ as in (20);
5: Get $\tilde{\mathbf{F}}$ from $RREF(\tilde{\mathbf{G}}_{reo}^T)$;
6: Compute matrices $\mathbf{R}$ and $\mathbf{M}$;
7: Assemble and solve system (24);
8: Get all the currents as $\mathbf{I} = \tilde{\mathbf{L}}\mathbf{I}_c$;
9: Reconstruct eddy-current vector $\mathbf{j}$ (13);
10: **Return:** $\mathbf{j}(\mathbf{r}), \mathbf{r} \in \Omega$;

4. Vector potential approach

The vector potential approach, from now on **C**-approach, relies on the use of the electric vector potential [1,11,13].

Indeed, since $\mathbf{j}$ is a solenoidal field, it can be expressed as the curl of a vector potential in a simply-connected domain. Therefore the vector of currents $\mathbf{I}$ is written as

$$\mathbf{I} = \mathbf{C}\mathbf{T}, \tag{25}$$

where the degrees of freedom (DoFs) where the vector $\mathbf{T}$ stores the integral of the electric vector potential on primal grid edges.

More in general, if Ω is a multiply connected domain, $\mathbf{I}$ is redefined as

$$\mathbf{I} = \mathbf{C}\mathbf{T} + \mathbf{W}\mathbf{i}, \tag{26}$$

where $\mathbf{i}$ is the vector of independent currents [13,17,28], and the columns of $\mathbf{W}$ store the representatives of generators of the *second relative cohomology group* $H^2(\mathcal{K}, \partial\mathcal{K})$ [12,29] (see Fig. 3a). The second relative cohomology group, by its very definition, spans solenoidal fields tangent to $\partial\mathcal{K}$ that are not curl of some vector field. As an example, $\mathbf{W}$ for a solid toric conductor is formed by a single column whose entries, if interpreted as electric current DoFs, form a unit current that flows on a dual 1-cycle[2] around the torus (see Fig. 3a). More in general, $\mathbf{W}$ has a number of columns equal to $\beta_1(\mathcal{K})$, where $\beta_1(\mathcal{K})$ is the number of 1-dimensional holes

[2] By definition, a p-cycle is a linear combination of p-cells of $\mathcal{K}$ (where $p = 0, 1, 2, 3$ denotes a node, edge, face or volume), called p-chain, which boundary is zero. It is straightforward to extend such definition to dual p-cycles on $\tilde{\mathcal{K}}$.

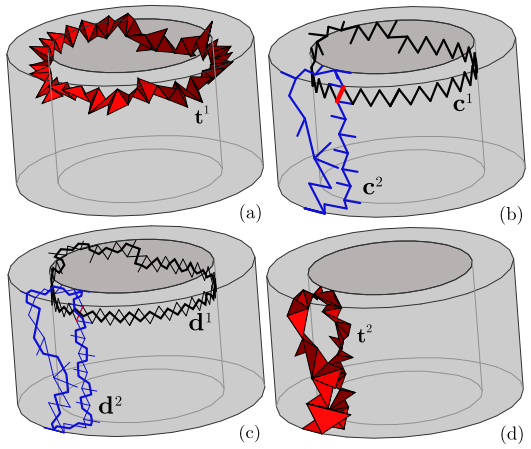


Fig. 3. Example of cohomology generators for a solid torus $\mathcal{K}$. (a) The *support* [16] of a representative $\mathbf{t}^1$ of the $H^2(\mathcal{K},\partial\mathcal{K})$ generator. It can be thought as a *thinned* unit current that flows around the torus. In this example, $\mathbf{t}^1$ is obtained as $\mathbf{t}^1=\mathbf{C}\mathbf{c}^1$, where $\mathbf{c}^1$ is represented in Fig. 3b. (b) The 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 $H^2(\mathcal{K},\partial\mathcal{K})$.

in $\mathcal{K}$ (1st Betti number).[3] Indeed, a basis of $H^2(\mathcal{K},\partial\mathcal{K})$ contains $\beta_1(\mathcal{K})$ independent 2-cycles, i.e. $rank(\mathbf{W})=\beta_1(\mathcal{K})$. The independent currents $\mathbf{i}$ are additional unknowns of the eddy-current problem in the case of multiply connected domain.

The formulation requires an algorithm for the automatic computation of $\mathbf{W}$. For efficiency, it is preferable to construct $\mathbf{W}$ by working on $\partial\mathcal{K}$ only, because there are less geometric elements to process in $\partial\mathcal{K}$ than in the whole $\mathcal{K}$. In this regard, $\mathbf{W}$ is usually computed as

$$\mathbf{W}=\mathbf{C}\mathbf{H}, \tag{27}$$

where the columns of $\mathbf{H}$ store some of the representatives of generators of the *first cohomology group* of the boundary $H^1(\partial\mathcal{K})$ [12,13], and consequently, the currents in (26) become

$$\mathbf{I}=\mathbf{C}\,(\mathbf{T}+\mathbf{H}\mathbf{i})\,. \tag{28}$$

Then, by using (28) in (15a) and pre-multiplying by $[\mathbf{C}\ \mathbf{C}\mathbf{H}]^T$, the MQS-EFIE is re-written as

$$\begin{bmatrix}\mathbf{K} & \mathbf{K}\mathbf{H}\\ \mathbf{H}^T\mathbf{K} & \mathbf{H}^T\mathbf{K}\mathbf{H}\end{bmatrix}\begin{bmatrix}\mathbf{T}\\ \mathbf{i}\end{bmatrix}=-i\omega\begin{bmatrix}\mathbf{C}^T\tilde{\mathbf{A}}_0\\ \mathbf{H}^T\mathbf{C}^T\tilde{\mathbf{A}}_0\end{bmatrix}, \tag{29}$$

where

$$\mathbf{K}=\mathbf{C}^T\,(\mathbf{R}+i\omega\mathbf{M})\,\mathbf{C}. \tag{30}$$

The first and the second rows in (29) enforce the discrete Faraday's law on local and global cycles, respectively.

Concerning the boundary condition (15c), we set to zero the entries of the vector $\mathbf{T}$ related to edges on $\partial\mathcal{K}$. To reduce the unknowns and to impose a tree-cotree gauge [11], we also set to zero the entries of the vector $\mathbf{T}$ on a suitable tree of $\mathcal{K}\setminus\partial\mathcal{K}$. Once $\mathbf{H}$ is constructed (which is the topic of Section 5), the above described approach can be easily implemented following Algorithm 2.

As previously stated, $\mathbf{H}$ is constructed by processing the geometric elements belonging to $\partial\mathcal{K}$ only. Nevertheless, there is also a downside to this approach. The major difficulty here is that the $H^1(\partial\mathcal{K})$ cohomology group produces twice the number of generators of a $H^2(\mathcal{K},\partial\mathcal{K})$ basis, since a basis of $H^1(\partial\mathcal{K})$ is formed by $\beta_1(\partial\mathcal{K})=2\beta_1(\mathcal{K})$ 1-cycles, where $\beta_1(\partial\mathcal{K})$ is the number of 1-dimensional holes in $\partial\mathcal{K}$.[4] For example, when dealing with the solid torus, two possible representatives of boundary generators of H^1 are depicted in Fig. 3b. 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 $\partial\mathcal{K}$. It is always possible to find a basis of $H^1(\partial\mathcal{K})$ generators such that half generators have the dual cycle in $\partial\mathcal{K}$ (i.e. a cycle formed by *dual edges* [12] in $\partial\mathcal{K}$) that is the boundary of a 2-chain, i.e. a surface, which lies in $\mathbb{R}^3\backslash\mathcal{K}$.

Such kind of dual cycles, like $\mathbf{d}^1$ in Fig. 3c, are said to be *homologically trivial* in the insulator $\mathbb{R}^3\backslash\mathcal{K}$ [12]. They produce a $H^2(\mathcal{K},\partial\mathcal{K})$ basis when pre-multiplied by $\mathbf{C}$, see Fig. 3a and [16]. We will refer to these cycles as *relevant* generators of the first cohomology group $H^1(\partial\mathcal{K})$. The remaining dual cycles, like $\mathbf{d}^2$ in Fig. 3c, are the boundary of a 2-chain which lies in $\mathcal{K}$ (homologically trivial in the conductor mesh). They must be discarded to obtain a full rank system (29), since they do not produce a representative of a $H^2(\mathcal{K},\partial\mathcal{K})$ basis, like the one in Fig. 3d.

A first approach, proposed in [17], suggests to still use all representatives of the $H^1(\partial\mathcal{K})$ basis to produce $\mathbf{W}$ by pre-multiplying the representatives $\mathbf{H}$ by $\mathbf{C}$. The obtained $\mathbf{W}$ is called a *lazy cohomology basis* and the obtained system is singular, but generally iterative and direct solver manage to solve such systems.

On the contrary, if one wants to obtain a full rank system, there are essentially two techniques. One may try to add to the basis $\mathbf{W}$ one lazy cohomology generator at a time, but only if the generator is independent with respect to the generators already added previously in the basis. This is the approach followed in [30]. However, checking independency is costly, as it requires to solve one linear system with integer coefficients for each generator. Moreover, since real coefficients are used to this aim in practice, see [30], there is also a loss of robustness, since—because of the finite precision of floating point representation of real numbers—a threshold has to be added for checking independency. Thus, the computational effort of this approach is huge, especially if the number of generators is large.

Another, much more efficient, technique to find the required change of cohomology basis to obtain $\mathbf{W}$ has been introduced in [16,28], and is the approach followed in this work. In the next section, we will discuss in detail how to efficiently compute a lazy cohomology basis and how to retrieve, from such a basis, only the first group relevant generators which produce a basis of $H^2(\mathcal{K},\partial\mathcal{K})$.

Algorithm 2 **C**-approach.

Input: $\mathcal{K}$ (primal grid of Ω), $\mathbf{H}$, ω;
Output: Eddy-current vector in Ω;
1: Compute topological matrices $\mathbf{G},\mathbf{C},\mathbf{D}$;
2: Construct a tree T of $\mathcal{G}$, and the corresponding cotree C, such that every $e_i\in\partial\mathcal{K}$ will be in T;
3: Compute matrices $\mathbf{R}$ and $\mathbf{M}$;
4: **if** $rank(\mathbf{H})=0$ **then**
5: Assemble and solve system (29) for cotree edges only;
6: Get current fluxes as $\mathbf{I}=\mathbf{C}\mathbf{T}$, where $T_i=0$ if $e_i\in T$, $i\in\{1,...,N_e\}$;
7: **else**
8: Assemble and solve system (29) for cotree edges and the basis $H^1(\partial\mathcal{K})$;
9: Get currents as $\mathbf{I}=\mathbf{C}(\mathbf{T}+\mathbf{H}\mathbf{i})$, where $T_i=0$ if $e_i\in T$, $i\in\{1,...,N_e\}$;
10: **end if**
11: Reconstruct eddy-current vector $\mathbf{j}$ (13);
12: **Return:** $\mathbf{j}(\mathbf{r})$, $\mathbf{r}\in\Omega$;

[3] A 1-dimensional hole in $\mathcal{K}$ is represented by any set of points $\mathbf{r}\in\mathbb{R}^3\setminus\mathcal{K}$, which can be surrounded by a 1-cycle of $\mathcal{K}$.

[4] A 1-dimensional hole in $\partial\mathcal{K}$ is represented by any set of points $\mathbf{r}\in\mathbb{R}^3\setminus\partial\mathcal{K}$ which can be surrounded by a 1-cycle of $\partial\mathcal{K}$.

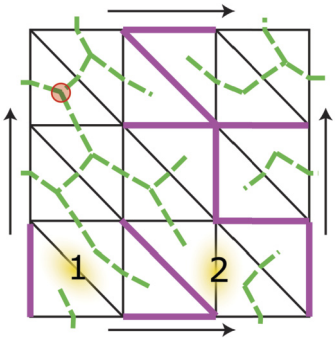


Fig. 4. Construction of a dual BFS tree $\tilde{T}^b$ (in dashed green lines) and a maximal spanning primal tree T^b (in solid thick magenta lines) on the boundary of a torus. The basepoint, from which $\tilde{T}^b$ is rooted, is indicated by the red circle. A 2D representation of the torus is employed: the horizontal arrows indicate the toroidal direction, while the vertical arrows indicate the poloidal one; edges on the left and the right sides are identical, as well as the edges on the top and bottom sides. The edges highlighted in yellow populate the set $\mathcal{R}$, i.e. they do not belong to any of the above trees.

5. Computation of an *optimal* $H^1(\partial\mathcal{K})$ cohomology basis

The computation of a $H^1(\partial\mathcal{K})$ basis has been discussed in several works and a general and fast method, described in [28], has been developed for the computation of the *lazy* first cohomology group generators. In this section, the algorithm described in [28] is reviewed and new steps are introduced in order to compute only the *relevant* generators of the first cohomology group H^1 of the boundary $\partial\mathcal{K}$. Indeed, the presented procedure allows for computing a matrix $\mathbf{H}$ which, once pre-multiplied by $\mathbf{C}$, gives only $\beta_1(\mathcal{K})$ 2-cycles which are representatives of a $H^2(\mathcal{K}, \partial\mathcal{K})$ basis. On the contrary to the use of lazy generators as representatives of $\mathbf{W}$, by using such a basis the obtained system (29) is full rank, with non-negligible computational advantages in the system solution. The presented algorithm is robust and is not affected by numerical precision, since all the arithmetic operations are performed with integer coefficients. Moreover, as will be demonstrated in the last section of the paper, thanks to the easy parallelization of the procedure, the required computation time is very small. A pseudo-code of the complete algorithm, described in the following paragraphs, is reported in Algorithm 3.

5.1. Lazy cohomology generators algorithm

Let $\mathcal{K}^b$ be the boundary $\partial\mathcal{K}$ of the primal grid $\mathcal{K}$, composed by N_f^b faces, N_e^b edges and N_n^b nodes. A barycentric subdivision of $\mathcal{K}^b$ yields to the dual grid $\tilde{\mathcal{K}}^b = D(\mathcal{K}^b)$, consisting of $\tilde{N}_f^b = N_n^b$ faces, $\tilde{N}_e^b = N_e^b$ edges and $\tilde{N}_n^b = N_f^b$ nodes.[5] $\mathcal{G}^b \subset \mathcal{K}^b$ denotes the primal graph of N_e^b edges and N_n^b nodes, while $\tilde{\mathcal{G}}^b \subset \tilde{\mathcal{K}}^b$ is the corresponding interlocked dual graph of $\tilde{N}_e^b$ edges and $\tilde{N}_n^b$ nodes.

We construct a spanning dual tree $\tilde{T}^b$ on $\tilde{\mathcal{G}}^b$, using a Breadth-First-Search (BFS) algorithm or, equivalently, by Dijkstra algorithm [31] weighting all the dual edges with a unitary value (line 7 of Algorithm 3). Since the tree is rooted from an arbitrary dual node $\tilde{n}^* = D(f^*)$ with $f^* \in \mathcal{K}^b$, called *basepoint* (see Fig. 4), let define the distance $dist(\tilde{n}_i, \tilde{T}^b)$ of a node $\tilde{n}_i \in \tilde{T}^b$ as the number of consecutive dual edges which connect $\tilde{n}_i$ with the root node $\tilde{n}^*$. Then, a maximal spanning tree T^b is constructed on $\mathcal{G}^b \setminus D(\tilde{T}^b)$ (e.g. using Kruskal algorithm, as at line 8 of Algorithm 3), i.e. neglecting all the primal edges corresponding to the dual edges of $\tilde{T}^b$, and weighing any remaining edge as:

[5] From now on the duality operator will be applied to the boundary $\mathcal{K}^b$ only.

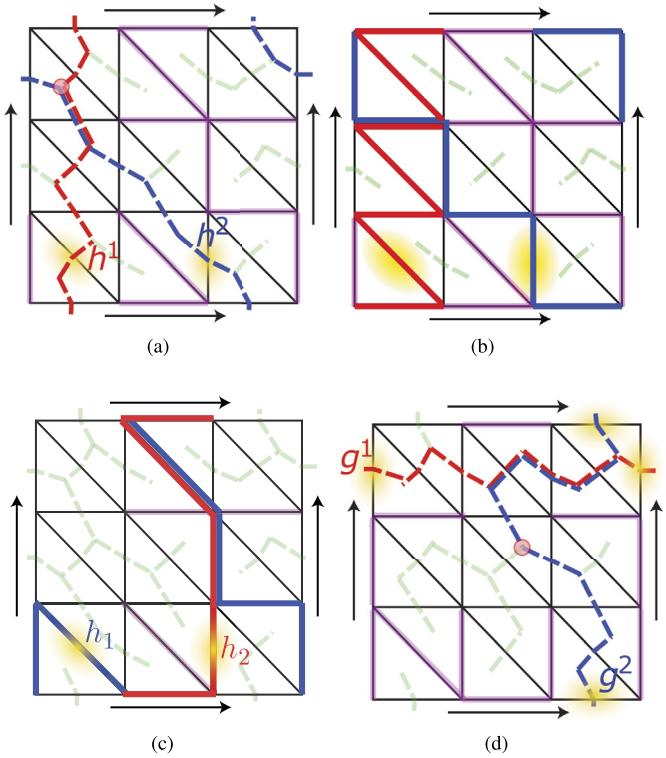


Fig. 5. Cohomology $H^1(\partial\mathcal{K})$ and homology $H_1(\partial\mathcal{K})$ generators of the torus computed with the basepoint of Fig. 4. (a) The cohomology generators can be interpreted on the dual complex as cycles. These cycles are made by dual edges that are dual to primal edges in the support of each cohomology generator. The corresponding primal edges, which incidences are actually stored in $\mathbf{H}$, are shown in (b). The representatives of the homology basis are reported in (c). (d) shows the dual cycles of the cohomology generators obtained with a different basepoint.

$$p_i = dist(\tilde{n}_{i,1}, \tilde{T}^b) + dist(\tilde{n}_{i,2}, \tilde{T}^b), \quad \forall e_i \in \mathcal{G}^b : D(e_i) \notin \tilde{T}^b, \quad (31)$$

where $\tilde{n}_{i,1}$ and $\tilde{n}_{i,2}$ are the dual nodes incident with the dual edge $\tilde{e}_i = D(e_i)$. Once a pair of interlocked trees respecting the properties above has been constructed, the set $\mathcal{R}$ of the primal edges which do not belong to anyone of them will be populated by $2\beta_1(\mathcal{K})$ elements (see Fig. 4), i.e.:

$$\mathcal{R} = \left\{ e_i \in \mathcal{K}^b : e_i \notin T^b \wedge D(e_i) \notin \tilde{T}^b \right\}, \text{ with } card(\mathcal{R}) = 2\beta_1(\mathcal{K}). \quad (32)$$

As reported in lines 10-15 of Algorithm 3, for each edge $e_h \in \mathcal{R}$, $h \in \{1, ..., 2\beta_1(\mathcal{K})\}$, we construct the unique dual 1-cycle $D(h^h) \in \tilde{T}^b \cup D(e_h)$, see Fig. 5a. The set of such cycles represents the shortest basis of the first cohomology group H^1 of $\mathcal{K}^b$ with the basepoint $\tilde{n}^*$. For each cohomology generator h^h, its incidences with the edge set of $\mathcal{K}$ are stored in the columns of $\mathbf{H}$. Fig. 5b shows (thick solid lines) the representatives of $H^1(\partial\mathcal{K})$ on the corresponding primal grid edges, whose incidences are actually stored in $\mathbf{H}$.

Similarly, the shortest basis of the first homology group H_1 of $\mathcal{K}^b$, with the basepoint $\tilde{n}^*$, can be formed by closing, for each edge $e_h \in \mathcal{R}$, $h \in \{1, ..., 2\beta_1(\mathcal{K})\}$, the unique 1-cycle $h_h \in T^b \cup e_h$, see Fig. 5c.

Nevertheless, the assembling time of matrix blocks in (29) involving $\mathbf{H}$ can represent a bottleneck for the presented formulation when applied to large scale problems, as will discussed in the next section concerning the low-rank approximation of system (29). For this reason, we are interested in reducing as much as possible the cohomology generators length (i.e. the number of edges of each cycle). Consequently, a *minimal length lazy cohomology basis* H^1 of $\mathcal{K}^b$,

Table 1
Left. *Signature* of h^1 and h^2 cohomology generators of Fig. 5a with respect to the homology basis represented by h_1, h_2 generators in Fig. 5c. Right: *Signature* of cohomology generators g^1 and g^2 of Fig. 5d with respect to the homology basis represented by h_1, h_2 generators.

	h_1	h_2
h^1	1	0
h^2	0	1

	h_1	h_2
g^1	1	1
g^2	1	0

indicated with L, is computed by iteratively replicating the procedure described above, but changing from one iteration to the other the starting basepoint $\tilde{n}^*$, as explicated by the for-cycle from line 6 to line 25 of Algorithm 3. For every basepoint, once the relative basis $H^1(\partial\mathcal{K})$ has been computed, we store only a triplet of values for every generator $h^h \in H^1(\partial\mathcal{K})$, $h \in \{1, ..., 2\beta_1(\mathcal{K})\}$, which are classified in the structure M in lines 16-23 of Algorithm 3:

- the *length* of h^h;
- its *coordinates*, i.e. the index of the basepoint used for constructing $\tilde{T}^b$ and the local index h of the generator in $H^1(\partial\mathcal{K})$;
- the *intersection*, i.e. the dot product $\langle h^h, h_k \rangle$ of h^h with any cycle h_k belonging to a fixed homology basis H_1 of $\mathcal{K}^b$, $h \in \{1, .., 2\beta_1(\mathcal{K})\}$, $k \in \{1, .., 2\beta_1(\mathcal{K})\}$.

It is worth noting that, storing only a triplet M of values for each basepoint avoids to store all the possible cohomology generators $H^1(\partial\mathcal{K})$ before selecting only the ones with minimal length, thus reducing the computer memory requirements of the procedure. Furthermore, since each cohomology basis starts from a different basepoint, this iterative procedure can be easily parallelized on different CPU threads, gaining a significant speed-up with respect to a serial computation.

Then, as shown in the Algorithm 3 from line 26 to line 37, all the generators are sorted in ascending length order and starting from the shortest, each generator is added to the minimal length cohomology generator basis L only if it is linearly independent from the others already stored in L.

The independence of the generators can be checked by the assembly of a $2\beta_1(\mathcal{K}) \times 2\beta_1(\mathcal{K})$ matrix $\mathbf{N}$. Its rows store the dot product of the chosen cohomology generator with a fixed homology basis. The collection of all these products for one cohomology generator h is called *signature* of h. The homology basis can be computed, for example, with the first selected basepoint (see lines 12-14 of Algorithm 3). As an example, the *signature* of h^1 and h^2 cohomology generators shown in Fig. 5a, with respect to the homology basis represented by h_1, h_2 generators in Fig. 5c, is shown in Table 1-left. Since the two bases are computed with the same basepoint, the matrix collecting all the *signatures* is the identity matrix. If the cohomology generators g^1 and g^2 are obtained from a different basepoint, as shown in Fig. 5d, one gets the *signatures* in Table 1-right. As can be guess from Fig. 5a and Fig. 5d, cohomology generators h^1 and g^2 are equivalent (i.e. in the same cohomology class) because their *signatures* are the same.

Thus, a generator is added to L only if its addition increases the rank of $\mathbf{N}$ (lines 31-33 of Algorithm 3). The computation of the minimal length (lazy) cohomology basis $H^1(\partial\mathcal{K})$ can be stopped as soon as $rank(\mathbf{N}) = 2\beta_1(\mathcal{K})$, as shown in lines 34-36 of Algorithm 3.

Finally, the incidences $\mathbf{H}$ of the minimal generators can be recomputed using the coordinates previously stored, as shown in lines 38-43 of Algorithm 3. It is worth noting that, in this case, the generators independency check is robust and really fast, because of the small size of the integer coefficient matrix $\mathbf{N}$, yet it does not require to fix any floating point precision.

In the case of very fine meshes characterized by high values of $\beta_1(\mathcal{K})$, finding the minimal length basis $H^1(\partial\mathcal{K})$ can be a time consuming operation. Nevertheless, as will be shown in the numerical results section, limiting the number of basepoints can significantly speed up the basis computation, with a negligible difference in the algebraic operations with $\mathbf{H}$, without affecting the solution accuracy of (29). A set of, e.g., $5\beta_1(\mathcal{K})$ basepoints, can be chosen as randomly distributed faces of $\partial\mathcal{K}$ or as evenly spaced faces of $\partial\mathcal{K}$. Both approaches yield to an *optimal* cohomology basis $H^1(\partial\mathcal{K})$, very similar to the minimal one, which otherwise should be computed considering all the possible basepoints in $\partial\mathcal{K}$. In the last section of the paper, we will show some numerical results concerning the sensitivity of the basis length on the number of basepoints.

Algorithm 3 Computing relevant minimal length cohomology generators $H^1(\partial\mathcal{K})$.

Input: $\mathcal{G}^b$, $\tilde{\mathcal{G}}^b$, a set $\mathcal{B}$ of N_b chosen basepoints $\tilde{n}^*$;
Output: $\mathbf{H}$;
1: $L = \emptyset$ - lazy minimal length basis of H^1 in $\partial\mathcal{K}$;
2: $R = \emptyset$ - relevant minimal length basis of H^1 in $\partial\mathcal{K}$;
3: $\mathbf{H} = \emptyset$;
4: $i = 1$ - global index of cohomology generators;
5: $b = 1$ - basepoint index;
6: **for** every basepoint $\tilde{n}^*_b \in \mathcal{B}$ **do**
7: construct a spanning dual tree $\tilde{T}^b$ on $\tilde{\mathcal{G}}^b$, rooted in $\tilde{n}^*_b$, with BFS or Dijkstra algorithm;
8: construct a maximal spanning primal tree T^b on $\mathcal{G}^b \setminus D(\tilde{T}^b)$, with Kruskal algorithm;
9: Compute $\mathcal{R}$ as in (32);
10: **for** every edge $e_j \in \mathcal{R}$, $j \in \{1, ..., 2\beta_1(\mathcal{K})\}$ **do**
11: construct cohomology vector $h^j \in H^1(\partial\mathcal{K})$ basis;
12: **if** $b = 1$ **then**
13: construct homology vector $h_j \in H_1(\partial\mathcal{K})$ basis;
14: **end if**
15: **end for**
16: **for** $j = 1$ to $2\beta_1(\mathcal{K})$ **do**
17: $M[i].dist =$ length of $D(h^j)$;
18: $M[i].coord = (b, j)$;
19: **for** $k = 1$ to $2\beta_1(\mathcal{K})$ **do**
20: $M[i].intersect[k] = \langle h^j, h_k \rangle$;
21: **end for**
22: $i = i + 1$;
23: **end for**
24: $b = b + 1$;
25: **end for**
26: Sort M elements in ascending order according to the *dist* parameter;
27: $G = \emptyset$ - vector of triplets as M, corresponding to the objective minimal length basis;
28: **for** $i = 1$ to $2\beta_1(\mathcal{K})N_b$ **do**
29: construct $\mathbf{N}$ matrix storing $G[k].intersect$, with $k \in \{1, ..., card(G)\}$;
30: construct $\mathbf{N}'$ matrix storing $G[k].intersect \cup M[i].intersect$, with $k \in \{1, ..., card(G)\}$;
31: **if** $rank(\mathbf{N}') > rank(\mathbf{N})$ **then**
32: $G = G \cup M[i]$;
33: **end if**
34: **if** $card(G) = 2\beta_1(\mathcal{K})$ **then**
35: break;
36: **end if**
37: **end for**
38: **for** $k = 1$ to $2\beta_1(\mathcal{K})$ **do**
39: $(b, j) = G[k].coord$;
40: $h^k =$ recompute the j-th generator rooted in the b-th basepoint;
41: $L = L \cup h^k$;
42: Store h^k in $\mathbf{H}(:, k)$;
43: **end for**
44: $\mathbf{H}_{lazy} = \mathbf{H}$
45: **for** $i = 1$ to $2\beta_1(\mathcal{K})$ **do**
46: **for** $j = 1$ to $2\beta_1(\mathcal{K})$ **do**
47: construct pushed generator $\hat{h}^j$;
48: $\mathbf{F}(i, j) = linking\_number(\, D(h^i), D(\hat{h}^j)\,)$
49: **end for**
50: **end for**
51: $\mathbf{H} = \mathbf{H}_{lazy}\, null(\mathbf{F}) - R$ is the set of columns of $\mathbf{H}$;
52: **return** $\mathbf{H}$;

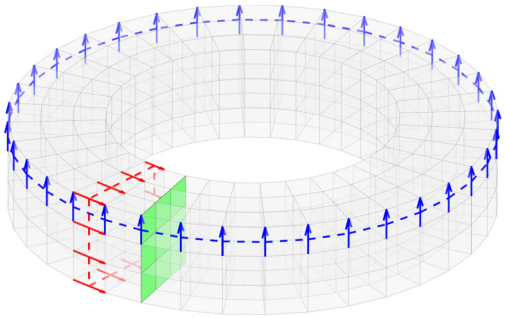


Fig. 6. First group cohomology generators h^1 (blue arrows) and h^2 (red arrows) of the boundary of a torus ($\beta_1(\mathcal{K}) = 1$). The dual cycles $D(h^1)$ and $D(h^2)$ are represented with dashed lines.

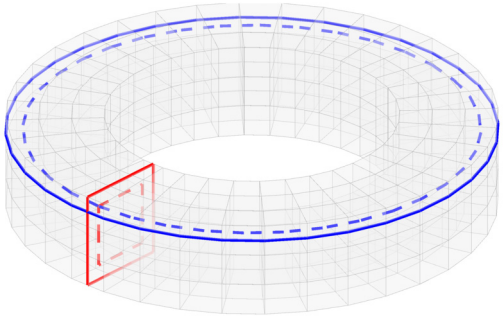


Fig. 7. Dual cycles $D(h)$ (solid lines) of the cohomology generators of the torus in Fig. 6 and dual cycles $D(\hat{h})$ (dashed lines) of the corresponding "pushed" generators from $\partial\mathcal{K}$ to $\mathcal{K}$. Once pushed, $\hat{h}^1$ (dashed blue) still links h^2 (solid red), thus $\mathbf{F}(2,1) \neq 0$. On the contrary, the pushing of $\hat{h}^2$ (dashed red) cancels out the linking with h^1 (solid blue), thus $\mathbf{F}(1,2) = 0$. $\mathbf{F}(1,1) = \mathbf{F}(2,2) = 0$ by definition of linking number. Thus, for a solid torus only the toroidal generator, like the blue solid one or equivalently $\mathbf{d}^1$ in Fig. 3c, is the boundary of a surface which lies in the insulator $\mathbb{R}^3 \setminus \mathcal{K}$ and it is stored in $\mathbf{H}$. On the contrary, the red generator, or $\mathbf{d}^2$ in Fig. 3c, it is the boundary of a surface which entirely belong to $\mathcal{K}$, thus it is not a representative of $\mathbf{H}$.

5.2. From lazy to relevant cohomology generators

We now introduce a further step to Algorithm 3. Although the lazy cohomology generators algorithm is able to compute a minimal length cohomology basis $H^1(\partial\mathcal{K})$, using such a basis system (29) is not full rank. We remark that only half of the representatives $\mathbf{H}$ computed by the lazy cohomology generators algorithm, once pre multiplied by $\mathbf{C}$ (see (27)), give the correct representatives $\mathbf{W}$ of the second relative cohomology basis $H^2(\mathcal{K}, \partial\mathcal{K})$. Only a subset $R \subset L$, composed by $\beta_1(\mathcal{K})$ elements, contains the *relevant* first cohomology group generators and this justifies the name *lazy* first cohomology group usually used for L. Our approach for retrieving the relevant only first group cohomology generators relies in the computation of the *linking numbers* (lines 44-51 of Algorithm 3) between each couple of 1-cycles in $\tilde{\mathcal{K}}$ that are dual of the generators in L (Fig. 6). Fig. 6 clarifies the necessity of a cohomology basis in the case of a torus. Indeed, (25) applied to the green faces would give a zero net current, since the entries of $\mathbf{T}$ are fixed to be zero for all the edges belonging to $\partial\mathcal{K}$ [11]. Its flux is restored by the independent current associated to h^1 (blue arrows), while h^2 (red arrows) represents a not-relevant generator, since it does not contribute to the flux of such faces (no incidences between h^2 edges and such kind of faces). The theoretical background of the approach is given in [32].

For the sake of clarity, we denote with $\mathbf{H}_{lazy}$ the incidence matrix $\mathbf{H}$ computed using the presented algorithm so far, i.e. up to line 43 of Algorithm 3. Let be $\mathbf{F}$ the $2\beta_1(\mathcal{K}) \times 2\beta_1(\mathcal{K})$ matrix whose (i, j) entry is defined as

$$F_{ij} = linking\_number(D(h^i), D(\hat{h}^j)), \tag{33}$$

where h^i, $i \in \{1, ..., 2\beta_1(\mathcal{K})\}$ is the i-th generator in L and $\hat{h}^j$ is the j-th generator "pushed" from $\partial\mathcal{K}$ inside $\mathcal{K}$. The "pushing" of generators is easily achievable, thanks to the duality between primal and dual grids, pre-multiplying the generators matrix $\mathbf{H}_{lazy}$ by the incidence matrix $\mathbf{C}$, such that

$$\hat{\mathbf{H}} = \mathbf{C}\mathbf{H}_{lazy} = \tilde{\mathbf{C}}^T\mathbf{H}_{lazy} = \left[\hat{h}^1 \cdots \hat{h}^j \cdots \hat{h}^{2\beta_1(\mathcal{K})}\right], \tag{34}$$

where $\hat{\mathbf{H}}$ stores the incidences of the pushed cohomology generators with the face set of $\mathcal{K}$. It can be easily seen that, while $D(h^j) \in \tilde{\mathcal{K}}^b$ holds, on the contrary $D(\hat{h}^j)$ is a 1-cycle entirely belonging to $\tilde{\mathcal{K}} \setminus \partial\tilde{\mathcal{K}}$. However, only half of the representatives of $\hat{\mathbf{H}}$ are also representatives of $\mathbf{W}$. In particular, the dual generators $D(\hat{h}^j)$ pushed from those dual generators h^j that are boundary of a surface in the insulator $\mathbb{R}^3 \setminus \mathcal{K}$ ($\mathbf{d}^1$ in Fig. 3), still link at least one dual cycle $D(h^i)$, $i \in \{1, ..., 2\beta_1(\mathcal{K})\}$, as for example $\hat{h}^1$ (dashed blue cycle) in Fig. 7.

On the contrary, every remaining dual cycle (i.e. those that are boundary of a surface in the conductor mesh $\mathcal{K}$, like $\mathbf{d}^2$ in Fig. 3), once pushed into $\tilde{\mathcal{K}}$, does not link any dual cycle $D(h^i)$, $i \in \{1, ..., 2\beta_1(\mathcal{K})\}$, see for example $\hat{h}^2$ (dashed red cycle) in Fig. 7, and must be discarded. On the basis of above comments, the retrieving of relevant generators belonging to R is performed by computing the integer null space of matrix $\mathbf{F}$. Thus, $\mathbf{H}$ is computed by a linear combination of the columns of $\mathbf{H}_{lazy}$, such that:

$$\mathbf{H} = \mathbf{H}_{lazy}\, null(\mathbf{F}), \tag{35}$$

where now $rank(\mathbf{H}) = \beta_1(\mathcal{K})$.

An accurate computation of the linking number between two cycles requires the use of interval arithmetic [33]. For generators retrieving purposes, we simply need to evaluate if one pushed generator $\hat{h}^j$ still links at least one of the not-pushed generators h^i. Consequently, the linking number computation can be implemented in floating point arithmetic, then rounding the result of (33) to an integer coefficient. In this regard, a shorter length lazy cohomology basis helps in reducing the time required for the linking number evaluation, hence justifies the computation of a minimal (or close to minimal) length basis. On the other side, the remaining operations, i.e. the computation of the null space and the generators retrieving by using (35), are performed with integer coefficients only, which again reinforces the robustness of the algorithm, that is a crucial point when large scale devices are considered.

5.3. C-approach vs L-approach

Clearly, the **C**-approach may be interpreted as an optimized implementation of the **L**-approach. It has been already emphasized that the **L**-approach corresponds to loop analysis of circuit theory. The **C**-approach considers differently local versus global cycles, where global cycles arise only when the topology of the conductor is not trivial, to obtain something similar to the *mesh analysis* of network theory [24].

The **C**-approach performs better than the **L**-approach in various respects. First, as regards the topological pre-processing required by the two formulations, the RREF required by **L**-approach has a cubical complexity, i.e. $O(N^3)$, where N is the number of mesh edges. Thus, the computation time for the pre-processing required by the **L**-approach rapidly increases with problem dimensions. On the other side, all the operations involved to compute a relevant $H^1(\partial\mathcal{K})$ basis scale with the number of 1-dimensional holes in $\mathcal{K}$, which are small compared to the mesh edges. Moreover, the most apparent advantage of having a short cycle basis is that low-rank

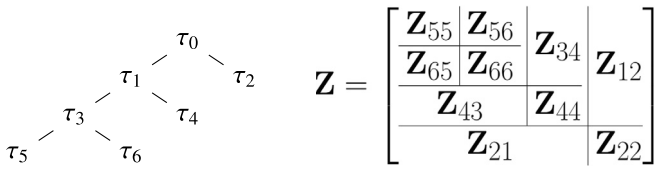


Fig. 8. Example of CT (with 3 levels) and corresponding matrix partitioning.

approximation techniques perform much better when the cycle basis is localized. This yields a limited applicability of the **L**-approach to large scale problems, like the ones involving thermonuclear fusion devices. This aspect is in detail addressed in the next section.

Let us observe that also the **L**-approach actually uses (co)homology generators, given that the computation of the cycle basis on the dual graph $\tilde{\mathcal{G}}$ used in Section 3 is exactly the computation of the generators of the $H_1(\mathcal{C}, \mathbb{Z})$ homology group [34, p. 506], where $\mathcal{C}$ is the 1-dimensional simplicial complex representing the dual graph $\tilde{\mathcal{G}}$.

6. Low-rank approximation of dense matrices

As a common drawback of integral methods, both the presented approaches end up with dense algebraic systems whose storage and assembling cost grow with N^2, where N is the number of DoFs. Moreover, the solution of (24) or (29) by a direct solver, e.g. using an LU factorization of the system matrix, has a $\mathcal{O}(N^3)$ complexity. Nevertheless, coupling integral formulations with low-rank approximation techniques based on hierarchical-matrix representations is an efficient solution to increase the size of the largest solvable problem on standard workstations [4,14], since, the solution complexity of performing an LU factorization is reduced approximately to $Nlog(N)$ [35]. In this section we briefly review the low-rank approximation of dense matrices based on $\mathcal{H}$-matrices. The state of the art of hierarchical-matrices also covers other data-sparse formats like HODLR matrices, HSS matrices and $\mathcal{H}^2$ matrices, which provide similar results in terms of accuracy and compression. For a detailed discussion on such formats the reader is referred to [36]. Efficient libraries for implementing low-rank approximations of dense matrices can be found in HLIBPro [37], *STRUMPACK Dense Package* [38], *hm-toolbox* [39] and HODLRlib [40].

A low-rank approximation of the dense matrix **Z** in (15a) is feasible since the integral kernel of a generic entry Z_{ij} has a strong geometrical dependence because of the Green's function, see (18). Indeed, the larger is the geometric distance between the integration domains Ω_i and Ω_j, the smoother the integrand function. Thus, as algebraic consequence, if the index set $\{1, ..., N\}$ corresponds to suitably ordered mesh elements, then **Z** exhibits rank-deficient properties in its off-diagonal blocks, which can be exploited by a hierarchical partitioning of the matrix. We would like to stress the requirement of "suitably ordered mesh elements", since this strongly affects the accuracy and the efficiency of the approximation. The index set $\{1, ..., N\}$ of the mesh elements corresponding to the system DoFs, i.e. the faces of $\mathcal{K}$ in (15a), should be reordered so that close indices in $\{1, ..., N\}$ correspond to near faces in the 3D space. Such reordering is performed by clustering the mesh elements, e.g. using a geometric bisection algorithm, and identifying each face in the 3D space with a DoF *coordinate* $\hat{\mathbf{r}}$, that corresponds to the face barycentre. Then, according to a user-controlled binary cluster tree (CT), the **Z** matrix is partitioned such that the entries belonging to a matrix sub-block $\mathbf{Z}(\tau_i, \tau_j)$ correspond to mesh elements belonging to a couple of CT nodes τ_i and τ_j (Fig. 8), for which the following geometrical mean quantities are defined:

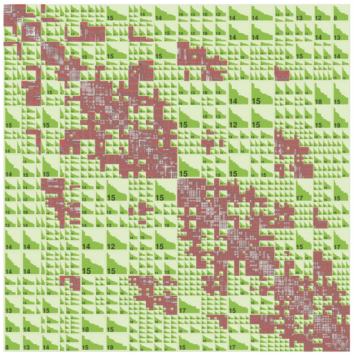


Fig. 9. Example of $\mathcal{H}$-matrix partitioned in admissible low-rank blocks (green) and inadmissible high-rank blocks (red). The number printed inside each block refers to the total count of columns evaluated by ACA for such sub-matrix.

$$diam(\tau_i) := max\left\{\|\hat{\mathbf{r}}_k - \hat{\mathbf{r}}_h\|_2 : \hat{\mathbf{r}}_k, \hat{\mathbf{r}}_h \in \tau_i\right\} \tag{36}$$

$$dist(\tau_i, \tau_j) := min\left\{\|\hat{\mathbf{r}}_k - \hat{\mathbf{r}}_h\|_2 : \hat{\mathbf{r}}_k \in \tau_i, \hat{\mathbf{r}}_h \in \tau_j\right\}. \tag{37}$$

At each CT level, every couple of nodes is tested against the following admissibility criterion:

$$min\{diam(\tau_i), diam(\tau_j)\} \leq \eta\, dist(\tau_i, \tau_j), \tag{38}$$

where η is the admissibility parameter ($0 < \eta < 2$). If (38) is true, the corresponding sub-matrix $\mathbf{Z}(\tau_i, \tau_j) \in \mathbb{C}^{m\times n}$ is approximated (i.e. compressed) using an SVD-like factorization, which stores a reduced number of sub-matrix columns such that:

$$\frac{\|\mathbf{Z}(\tau_i, \tau_j) - \mathbf{U}_{k'}\mathbf{V}_{k'}^*\|_F}{\|\mathbf{Z}(\tau_i, \tau_j)\|_F} < \varepsilon \quad \text{with } k' \leq k, \tag{39}$$

where ε is the approximation tolerance, k' and k are the approximate and exact rank of $\mathbf{Z}(\tau_i, \tau_j)$, respectively, and $\mathbf{U} \in \mathbb{C}^{m\times k'}$, $\mathbf{V} \in \mathbb{C}^{n\times k'}$ are its low-rank factors. On the contrary, non-admissible blocks are factorized and stored as dense sub-matrices without any kind of approximation. Because of its low-rank properties, matrix **Z** will be composed mostly by admissible blocks (Fig. 9), which should be directly approximated in order to avoid a high memory consumption during the factorization. As example, the factorization can be performed using the Adaptive Cross Approximation (ACA) algorithm [41]: the columns of a generic admissible sub-block $\mathbf{Z}(\tau_i, \tau_j)$ are adaptively computed until condition (39) is reached, avoiding to entirely store $\mathbf{Z}(\tau_i, \tau_j)$ before its decomposition in factors **U** and **V**. User must provide an element-access routine able to compute any generic entry of **Z** required by ACA. It is worth noting that the time required by ACA to compress each matrix sub-block is highly dependent by the programming efficiency of such routine which, usually, cannot be optimized as in the case of assembling the whole matrix **Z** at once. Indeed, if ACA needed all matrix entries, the required time will be greater than the one of an optimized "routine". Nevertheless, thanks to low-rank properties of **Z**, the reduced number of matrix evaluations performed by ACA gives a reasonable speed-up in assembling the compressed version of **Z** [14].

In the **C**-formulation, since the unknown vector **T** is associated to the edges of $\mathcal{K}$, the edge barycentres are used as DoFs coordinates. It is worth noting that each entry of the system matrix (29) arises from the double integral of a function different from the integrand in Z_{ij}. For example, the kernel of entries belonging to the first matrix block **K** involves the curl of the face basis functions which integration support Ω_i is defined as

$$\Omega_i = \bigcup_{k=1}^{N_v} v_k : v_k \ni e_i, \quad i \in \{1, ..., N_e\}. \tag{40}$$

Yet, the discrete curl $\mathbf{C}$ is a local operator, which maps the problem unknowns to local (i.e. short) cycles. The distance between the DoF coordinates involved in K_{ij} is not far from the mean distance between their supports Ω_i and Ω_j, thus the smoothness of the kernel is preserved and a low-rank approximation of $\mathbf{K}$ is still feasible. On the contrary, the remaining matrix blocks (i.e. $\mathbf{KH}$, $\mathbf{H}^T\mathbf{K}$ and $\mathbf{H}^T\mathbf{KH}$) cannot be safely approximated since they involve the mutual coupling between primal grid edges and cohomology generators or the mutual coupling between generators themselves. Matrix $\mathbf{H}$ is not a local operator and the supports of the independent currents are all the volumes incident with the generator cycle. Hence, the kernel smoothness may be lost along these global cycles, leading to an inefficient and inaccurate low-rank approximation.

Luckily, for almost all applications of interest the total number of entries in such blocks is really small compared to those in $\mathbf{K}$, hence they can be stored as dense sub-matrices without any approximation and with a negligible effect on memory allocation. Nevertheless, the time required for the entire computation of their entries grows quadratically with the generators length. For example, the assembling time for the (i, j) entry of $\mathbf{H}^T\mathbf{KH}$ has a $\mathcal{O}(N_i N_j)$ complexity, where N_i and N_j are the length of the i-th and j-th cohomology generators in $\mathbf{H}$, respectively. Hence, the computation of a minimal length cohomology basis is crucial to speed-up the assembling of system (29), particularly when considering fine meshes with relatively high values of $\beta_1(\mathcal{K})$.

Different methods can be adopted to assemble the system block matrix (29) and to avoid the compression of blocks $\mathbf{KH}$, $\mathbf{H}^T\mathbf{K}$ and $\mathbf{H}^T\mathbf{KH}$. However, since such implementation could be highly dependent on the used library, in the last paragraph of this section we will present a general method to handle this unusual $\mathcal{H}$-matrix structure.

In the $\mathbf{L}$-formulation, the DoF coordinates are defined as the barycentres of those faces in $\mathcal{K}$ which correspond to the dual cotree edges of $\tilde{\mathcal{G}}$. $\tilde{\mathbf{L}}$ is a non-local operator which maps the DoFs from the $\mathcal{K}$ faces to a set of independent dual loops in $\tilde{\mathcal{K}}$ and a generic entry (i, j) of $\tilde{\mathbf{L}}^T\mathbf{Z}\tilde{\mathbf{L}}$ involves the double integral on supports Ω_i and Ω_j that are all the volumes incident with the global dual loops $\tilde{z}_i$ and $\tilde{z}_j$ individuated by the dual cotree edges $\tilde{e}_i$ and $\tilde{e}_j$, respectively:

$$\Omega_i = \bigcup_{k=1}^{N_v} v_k : v_k \ni f_h, \quad \forall f_h = D(\tilde{e}_h) : \tilde{e}_h \in \tilde{z}_i. \tag{41}$$

Thus, *all* DoFs are associated to global (i.e. long) cycles which support may cover a great part of the domain Ω and this cancels the low-rank properties of $\tilde{\mathbf{L}}^T\mathbf{Z}\tilde{\mathbf{L}}$ off-diagonal blocks. To sum up, the dense system matrix arising from the $\mathbf{L}$-formulation cannot be safely approximated unless sufficiently minimal length dual loops are provided, such that they do not spread far from the corresponding dual cotree edge.

6.1. Assembling of $\mathcal{H}$-block-matrices with partitioned cluster tree

In the previous section it has been pointed out that blocks in (29) involving the matrix $\mathbf{H}$ of cohomology generators cannot be safely compressed in a low-rank format. The resulting block $\mathcal{H}$-matrix, i.e. $\mathcal{H}(\mathbf{Z})$, must then admit a hierarchical structure where sub-blocks $\mathbf{KH}$, $\mathbf{H}^T\mathbf{K}$ and $\mathbf{H}^T\mathbf{KH}$ are treated as high-rank matrices. The approach we adopted relies on a modified construction of the CT. Normally, in a cluster tree the index set $\{1, ..., N\}$ is recursively halved and the resulting sub-sets are arranged on nested levels of the tree (Fig. 10a). Differently, in our approach we force a first division of the index set by separating the indices related to the mesh edges from the ones associated to the cohomology generators, each one is then included into a separate index set with unitary cardinality (Fig. 10b):

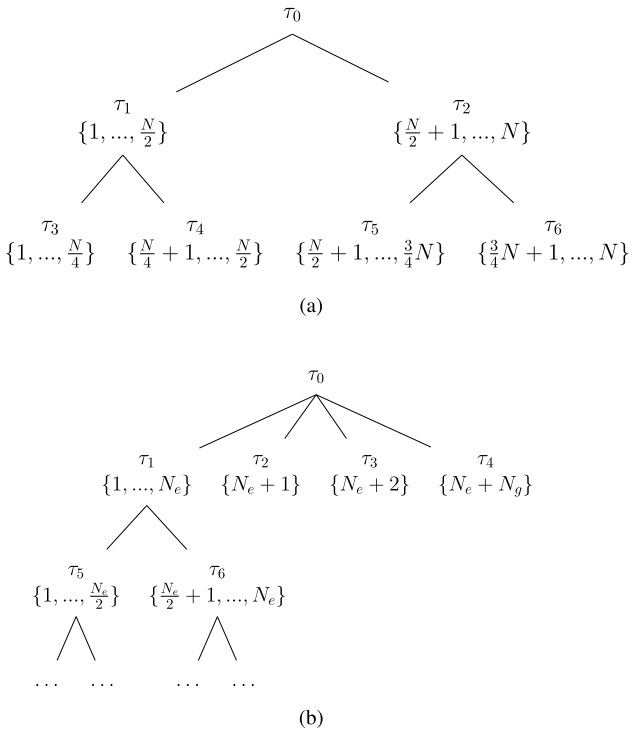


Fig. 10. Comparison of a standard binary CT and a modified CT partitioned according to (42). In this example $N_g = 3$.

$$\{1, ..., N\} = \{1, ..., N_e\} \cup \{N_e + 1\} \cup \{N_e + 2\} \cup ... \cup \left\{N_e + N_g\right\}, \tag{42}$$

with $N = N_e + N_g$. Furthermore, a fictitious set of coordinates, equal to a unique arbitrary numerical value, is associated to the N_g unknowns related to cohomology generators. This allows the low-rank approximation algorithm to test the CT nodes against the admissibility criterion. However, independently on its admissibility, each sub-block $\mathbf{Z}(\tau_i, \tau_j)$ with $i = 1$ and $1 < j \leq 1 + N_g$ is a 1-D vector of rank $k' = k$ (see Fig. 11), hence it will be stored by an entire computation of its entries. The same statement holds for blocks $\mathbf{Z}(\tau_i, \tau_j)$ with $1 < i \leq 1 + N_g$ and $j = 1$, as well as when $1 < i \leq 1 + N_g$ and $1 < j \leq 1 + N_g$ being in the last case $\mathbf{Z}(\tau_i, \tau_j)$ a trivial block with only one element. The implemented approach does not affect the low-rank approximation algorithm, since it relies on the CT construction only. Hence, it can be applied independently on the used library, provided that the user must be able to customize the tree construction.

A particular case consists of 3D objects characterized by small holes, if compared to the characteristic dimensions of the object: let consider, as example, a metallic supporting plate which is drilled in different positions to allow its fastening using some screws. In such cases, if a minimal length cohomology basis is computed, the generators are cycles whose dimensions are not so different from the ones of the local cycles corresponding to the DoFs in $\mathbf{T}$. Hence, cohomology generators can be treated as local cycles too and, associating to each one a DoF coordinate corresponding to the generator barycentre (i.e. the mean point between the barycentre of edges forming such generator), the overall system (29) is compatible with a low-rank approximation.

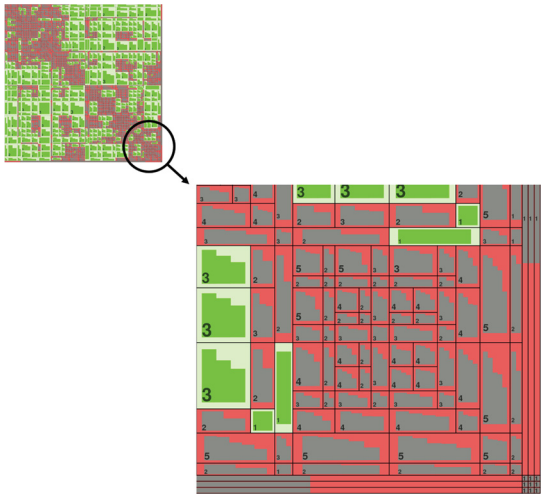


Fig. 11. $\mathcal{H}$-block-matrix partitioned using the modified CT. The vector partitioning of blocks involving cohomology generators (in this example $N_g = 3$) can be easily recognized in the zoomed picture.

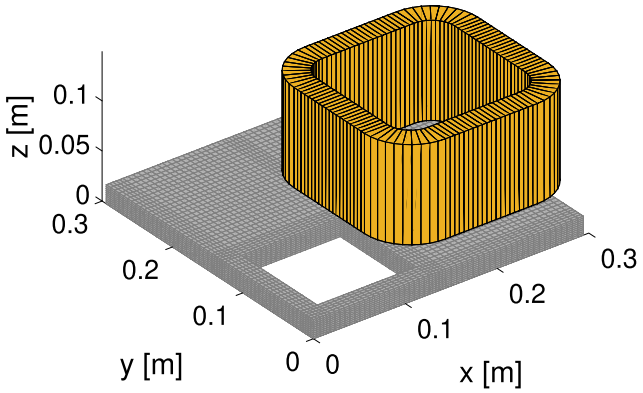


Fig. 12. TEAM Workshop Problem 7: asymmetrical conductor with a hole and excitation coil.

7. Numerical examples

In this section we test the accuracy and the computational performances of the two discussed formulations on the basis of two numerical examples. First, the TEAM Workshop Problem 7 [42] is chosen as benchmark case for both formulations. In the second example we consider a more complex device consisting of a typical vacuum vessel of a MCF device. While in the former case no low-rank approximation has been implemented, in the latter, because of the large size of the problem, we tested the compression performances comparing the approximate solution of the problem with the exact one (i.e. solving the dense uncompressed system). The simulations have been run on a Linux machine equipped with 16-core/32-thread processor (Intel® Xeon® Gold 6130 @2.10 GHz), for a total of 256 cores and 3 TB of RAM [43]. Both approaches have been implemented in the FORTRAN language. If (24) or (29) are assembled without low-rank approximation, the matrices assembling is performed in parallel via OpenMP, then the resulting system is directly solved using the LAPACK *zgesv* subroutine, based on LU decomposition. On the contrary, if compression is enabled, the low-rank approximation is implemented by the HLIBPro library which relies on a parallel computation of the entries required by the ACA algorithm and uses an $\mathcal{H}$–LU preconditioned GMRES solver. We remark that, as stated in the previous section, a low-rank approximation is feasible only for system (29), i.e. for the **C**-formulation.

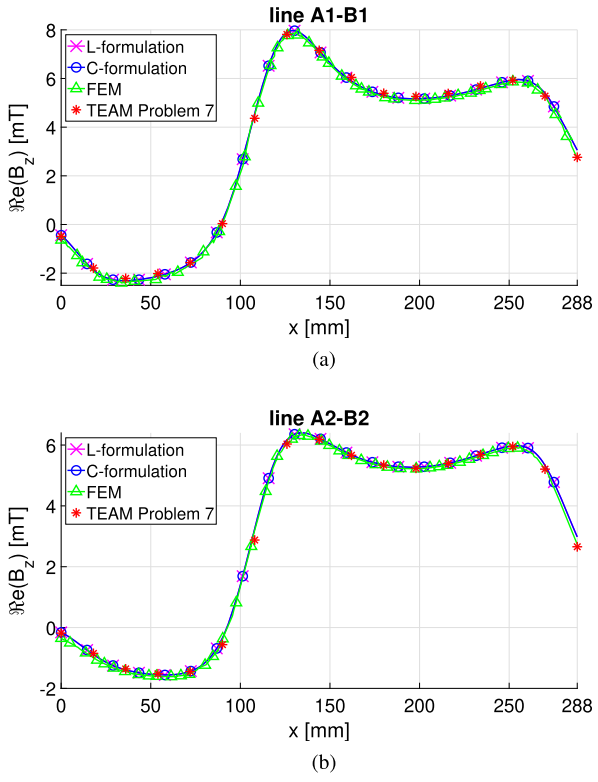


Fig. 13. Comparison of the **B** z–component (real part) computed at 50 Hz with the two integral formulations, the FEM software and results of TEAM Workshop Problem 7, along line A1-B1 (a) and line A2-B2 (b).

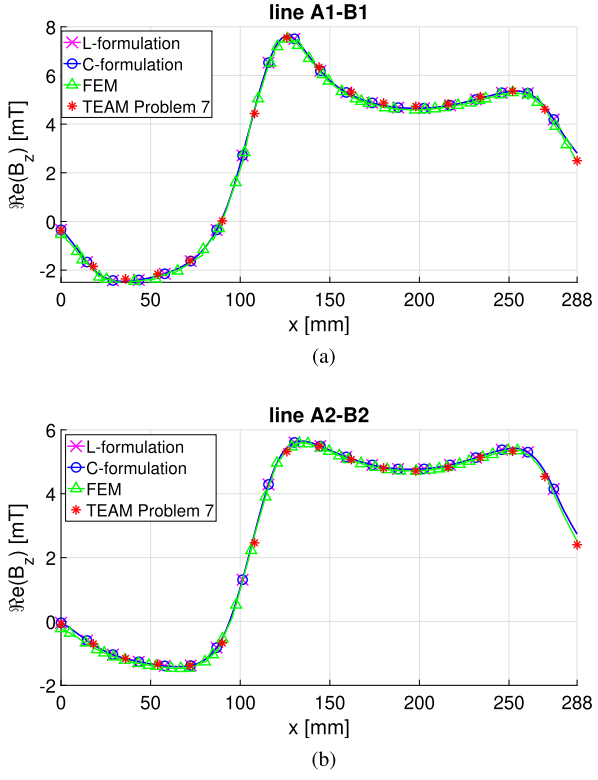
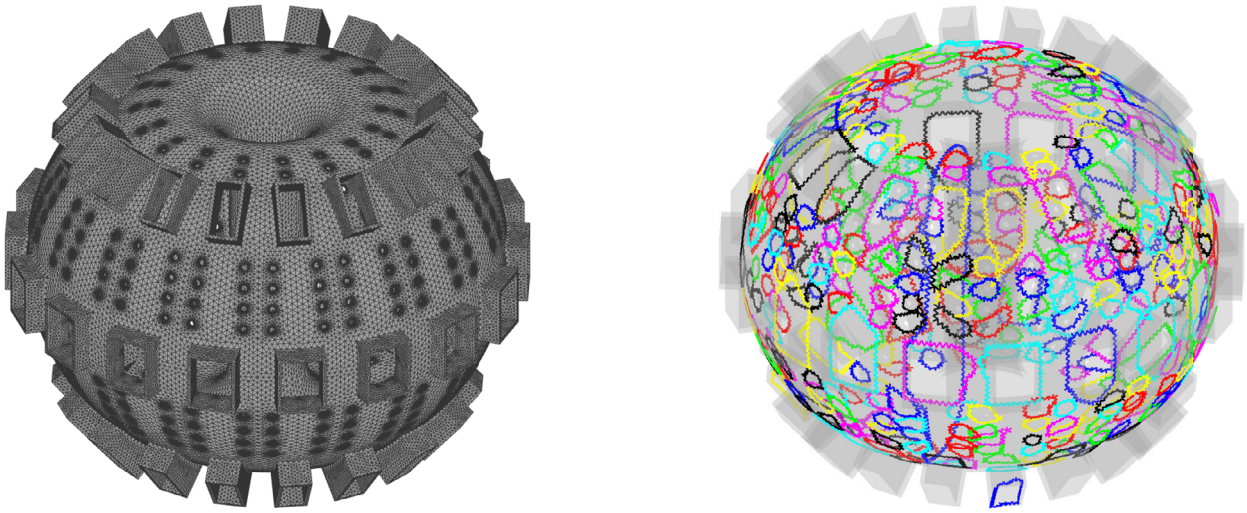


Fig. 14. Comparison of the **B** z–component (real part) computed at 200 Hz with the two integral formulations, the FEM software and results of TEAM Workshop Problem 7, along line A1-B1 (a) and line A2-B2 (b).



(a) (b)

Fig. 15. Geometry (a) and first cohomology group generators $H^1(\partial\mathcal{K})$ (b) of the vacuum vessel of Example B.

7.1. Example A

The TEAM Workshop Problem 7 consists of an asymmetrical conductor ($\rho = 0.028$ μΩ m) with a hole, excited by a coil with 2742 AT, at 50 Hz and 200 Hz frequencies (Fig. 12). To solve the problem, the conductor is meshed with 26250 hexahedra, leading to a number of unknowns $N = 48376$ in both approaches. The exciting field is addressed by analytical formulas. No low-rank approximation of matrices is applied in this example. The magnetic flux density is computed along two reference lines located at $y = 72$ mm, $z = 34$ mm (line A1-B1) and at $y = 144$ mm, $z = 34$ mm (line A2-B2), whose points vary from $x = 0$ mm to $x = 288$ mm. In this example, the results obtained by the two integral formulations are compared with the experimental data of the TEAM Workshop Problem 7. For the sake of comparison, we introduce a third set of values computed using a commercial FEM software.

Fig. 13 and Fig. 14 show the results of the simulations at 50 Hz and 200 Hz, respectively. The two integral formulations lead to numerically identical results since they both rely on the same set of discrete equations (15a). The computed values are in good agreement with the FEM simulation, as well as with the trend of experimental points of the TEAM Workshop Problem 7.

7.2. Example B

In the second example presented in this work, we model a typical vacuum vessel used in thermonuclear fusion machines, in which a vertical field of 1 T (50 Hz) simulates the magnetizing field of the machine magnet system. The device, having an electrical resistivity of 7.2E-07 Ωm, is characterized by a complex geometry having 415 ports, that are used for diagnostic and pumping purposes. First, we consider a mesh of 252670 tetrahedra, leading to a system of 163160 DoFs. The geometry of the device and the relative cohomology generators $H^1(\partial\mathcal{K})$, computed using the Algorithm 3, are shown in Fig. 15.

The problem is solved by applying a low-rank approximation of the dense matrices of (29) and without compression, i.e. maintaining the full rank structure of such matrices (dense case). In Table 2, the computational costs for solving the described problem are summarized, comparing the assembling, solution, and total times, as well as the total number of matrix entry evaluations and the Peak Memory Usage (PMU) reached during such steps. In this case, handling the dense matrices with a low-rank approximation allows almost halving the computational times and reducing the PMU to about the 8% of the one used in the dense case. The PMU ratio is confirmed by the ratio between the total number of entry evaluations effectively performed by ACA and the total matrix entries.

Table 2
Comparison of the computational costs between dense and approximated cases with $N_v = 252670$.

	HLIBPro	Dense	Ratio [%]
Assembling time [s]	7632	18669	40.9
n. of entry eval.	1.92E09	163160^2	7.2
Solution time [s]	3805	7826	48.6
Total time [s]	11437	26495	43.2
PMU [GB]	34.8	426	8.1

Then, the accuracy of the approximation is checked by evaluating the magnetic flux density on a toroidal surface close to the vessel internal wall (first wall). This kind of computation is typical in fusion applications, in order to estimate the "error-fields" arising from the eddy-currents induced in the machine conducting structures. Such eddy-currents have a greater amplitude around the machine ports and gaps, and produce unavoidable magnetic fields which break the toroidal symmetry of the machine [44]. Considering a toroidal coordinate system (r, θ, ϕ), the described surface is obtained by revolving a set of 100 points on the poloidal plane ($\phi = 0$ rad) for 150 positions between 0 and 2π radians in the toroidal direction ϕ, leading to a total of $N_p = 15000$ evaluation points. The radial components of the magnetic flux density are compared, on the toroidal surface described above, by defining the relative error in the evaluation point $\mathbf{r}_i(\theta, \phi)$, with $i \in \{1, ..., N_p\}$, and the overall mean square error ϵ_2 as

$$\epsilon_i = log_{10}\left|\frac{B_i^* - B_i}{B_i}\right| \tag{43}$$

$$\epsilon_2 = \sqrt{\frac{\sum_{i=1}^{N_p}(B_i^* - B_i)^2}{\sum_{i=1}^{N_p}(B_i)^2}}, \tag{44}$$

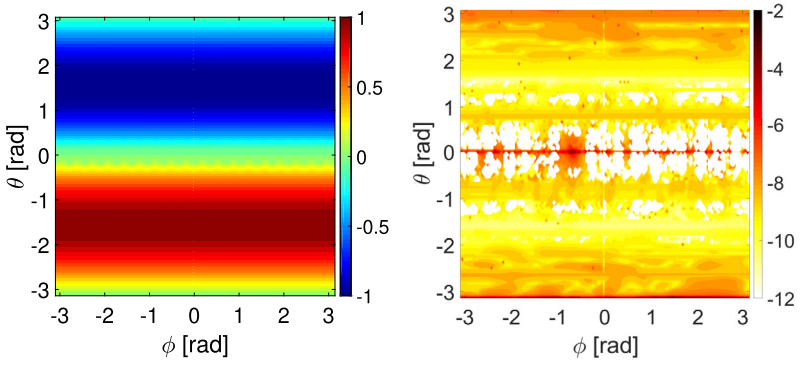


Fig. 16. Real part of the radial component $B_r(\theta, \phi)$ (values in T) on the first wall toroidal surface (a) and relative error ϵ_i (b). $\epsilon_2 = 3.5\text{E-}08$.

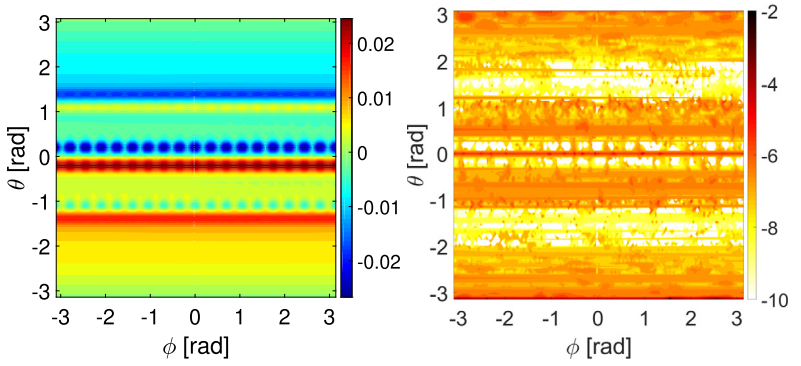


Fig. 17. Imaginary part of the radial component $B_r(\theta, \phi)$ (values in T) on the first wall toroidal surface (a) and relative error ϵ_i (b). $\epsilon_2 = 1.6\text{E-}07$.

Table 3
Comparison of the computational costs between dense and approximated cases with $N_v = 621844$.

	HLIBPro	Dense	Ratio [%]
Assembling time [s]	26456	–	–
n. of entry eval.	6.13 E09	423858^2	3.4
Solution time [s]	12616	–	–
Total time [s]	39072	–	–
PMU [GB]	110	2874	3.9

where B_i and B_i^* represent a generic component of the magnetic flux density in $\mathbf{r}_i$ evaluated in the dense and approximate cases, respectively. The error in the approximate case is negligible compared to the dense case, as demonstrated by the error maps shown in Fig. 16 and Fig. 17.

Finally, we improve the mesh resolution reaching an object discretization of 621884 tetrahedra, with 423858 DoFs. Because of the size of the problem, a dense solution is not considered since the PMU would be very close to the used workstation memory limit. The results are compared with the previous case by evaluating the magnetic flux density on the vacuum vessel first wall. The mean square error ϵ_2 is lower than 1% both for the real and the imaginary part of the magnetic flux density, which demonstrates the accuracy of the Volume Integral formulation even with a coarser discretization.

Moreover, the computational costs shown in Table 3 are extremely promising: a high number of DoFs can be managed with reasonable computation times and memory usage, which would be unfeasible by solving the dense problem on a standard workstation. This confirms the advantages, already observed in [14], of coupling Volume Integral methods with low-rank approximation techniques and proves the applicability of such methods also for very large scale problems.

Table 4
Plate with 4 holes: $H^1(\partial\mathcal{K})$ basis length and computation times according to the number of basepoints.

N_{BP}	H^1 length	T_{BP} [s]	T_{H^1} [s]	T_{TOT} [s]
1	588	6,20E-06	2,12E-01	2,12E-01
2	329	5,60E-04	5,14E-01	5,15E-01
5	224	1,19E-03	5,18E-01	5,19E-01
10	189	2,96E-03	5,07E-01	5,10E-01
15	178	6,02E-03	5,06E-01	5,12E-01
20	163	1,04E-02	6,02E-01	6,13E-01
30	160	2,23E-02	6,06E-01	6,29E-01
40	153	3,96E-02	6,56E-01	6,95E-01
50	150	6,23E-02	8,27E-01	8,89E-01
60	148	8,84E-02	8,27E-01	9,15E-01

8. $H^1(\partial\mathcal{K})$ basis length sensitivity to basepoints number

The last section of this work aims at defining a rule of thumb in choosing the basepoints number in order to compute an optimal first group cohomology generator basis $H^1(\partial\mathcal{K})$. The objective of this quantification is to reach a cohomology basis close the absolute minimal basis, but guaranteeing a reasonable computational cost for its evaluation.

As a first example, we consider the simple geometry of a thick plate with four holes ($\beta_1(\mathcal{K}) = 4$) and we compute the minimal length basis achievable starting from a number of basepoints in the range from 1 to 60. The plate is discretized by means of a tetrahedral mesh consisting of 36518 volumes and 8900 boundary faces. Results are summarized in Table 4, where N_{BP} indicates the number of chosen basepoints, T_{TOT} is the total time needed for the

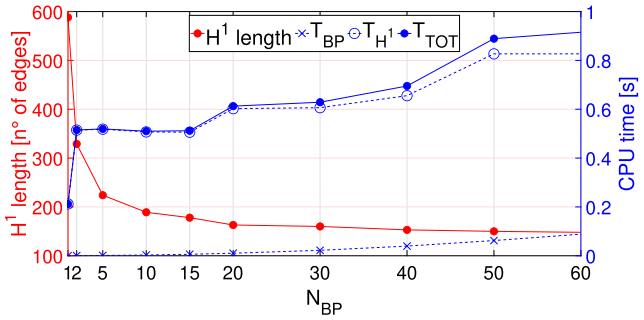


Fig. 18. Plate with 4 holes: computational cost of finding the $H^1(\partial\mathcal{K})$ minimal length basis achievable from N_{BP} selected basepoints.

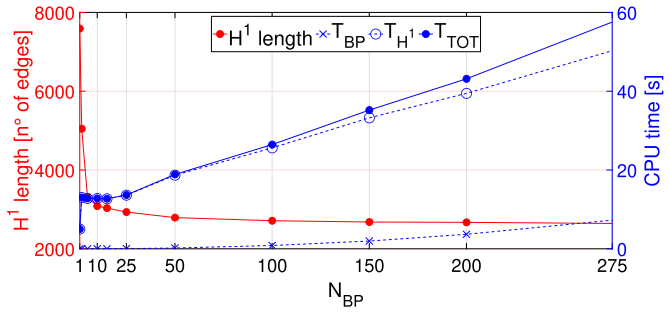


Fig. 19. Disk with 55 holes: computational cost of finding the $H^1(\partial\mathcal{K})$ minimal length basis achievable from N_{BP} selected basepoints.

Table 5
Disk with 55 holes: $H^1(\partial\mathcal{K})$ basis length and computation times according to the number of basepoints.

N_{BP}	H^1 length	T_{BP} [s]	T_{H^1} [s]	T_{TOT} [s]
1	7591	5,96E-06	5,01	5,01
2	5048	1,65E-03	13,04	1,30E+01
5	3313	3,14E-03	12,79	1,28E+01
10	3084	9,61E-03	12,78	1,28E+01
15	3033	1,85E-02	12,7	1,27E+01
25	2933	5,20E-02	13,61	1,37E+01
50	2792	2,00E-01	18,68	1,89E+01
100	2713	8,40E-01	25,62	2,65E+01
150	2680	1,97	33,24	3,52E+01
200	2672	3,69	39,46	4,32E+01
275	2643	7,27	50,29	5,76E+01

basis computation, which requires a first step for the basepoints collocation (T_{BP}) and a second step for sorting the generators according to their length (expressed in number of edges) and selecting the shortest ones (T_{H^1}). A *Maxmin Selection* algorithm [16] is used to obtain a set of N_{BP} basepoints evenly distributed on the domain boundary.

As shown in Fig. 18, the larger the number of basepoints, the shorter the $H^1(\partial\mathcal{K})$ basis length. Indeed, a higher N_{BP} allows a wider distribution of the basepoints on the object boundary and then a higher probability to collocate the basepoints close to the domain holes. On the contrary, a single basepoint makes the basis construction strongly dependent on its position, preventing to find a short basis, as shown in Fig. 20a where cohomology generators start from a single basepoint located on the top corner of the plate. Nevertheless, the $H^1(\partial\mathcal{K})$ basis length tends to saturate by increasing the basepoints number and a value $N_{BP} = 5\beta_1(\mathcal{K})$ can be a good compromise.

Then, we consider a second example consisting of a typical disk brake geometry (65419 tetrahedra and 27158 boundary faces) used for bikes or motorbikes, which is characterized by a total of 55 holes ($\beta_1(\mathcal{K}) = 55$) (Table 5, Fig. 21). The computation of a non-optimal $H^1(\partial\mathcal{K})$ basis (i.e. $N_{BP} = 1$) requires only few seconds, since it does not require to iterate on different basepoints to find a minimal length basis. The comparison of Fig. 19 with Fig. 18 shows that the initial slope of the $H^1(\partial\mathcal{K})$ curve tends to increase with the number of holes, and the achieved basis length saturates sooner by increasing N_{BP}. Therefore, with relatively high values of $\beta_1(\mathcal{K})$, the choice of $N_{BP} = 5\beta_1(\mathcal{K})$ may not guarantee the computational efficiency of the algorithm because of the longer time required for sorting the basis. We suggest to fix a maximum value of N_{BP} (e.g. equal to 50), such that a general choice in the number of basepoints could be $N_{BP} = min(5\beta_1(\mathcal{K}), 50)$.

9. Conclusions

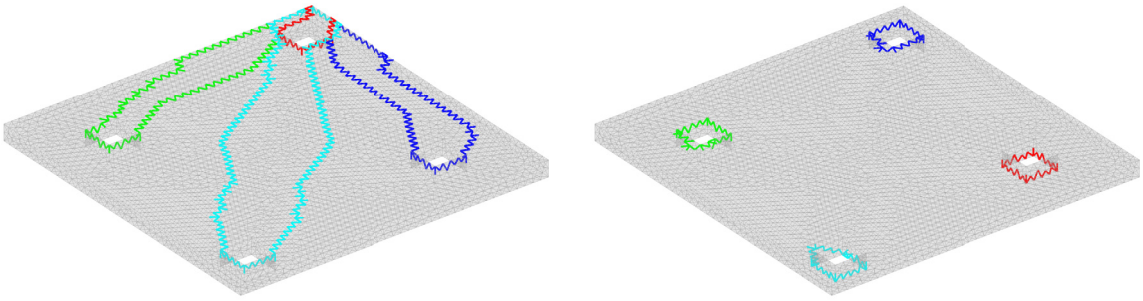
We have presented a Volume Integral (VI) formulation and compared two approaches which both rely on a cycle basis which ensures the solenoidality condition of the current density in simply and multiply connected domains. While the **L**-approach relies on global cycles only, the **C**-approach is based on local cycles and combines global cycles only when the domain is not simply connected. Consequently, the **C**-approach shows better performances both in the pre-processing and in the solution of large scale problems.

An efficient, general, and robust algorithm for computing a basis of the first cohomology group $H^1(\partial\mathcal{K})$ has been described in detail. Important modifications have been finally carried out in order to improve the original lazy cohomology generators algorithm and, for the first time, a complete procedure which starts from a lazy cohomology basis and ends with a minimal length (optimized) basis of relevant generators applied to a Volume Integral formulation (i.e. the **C**-approach) has been detailed. In this regard, the optimization of the basis length positively affects the computational time spent both in the generators retrievial (from lazy to relevant generators) and in the system assembly. A rule of thumb in the choice of the number of basepoints has been given, in order to guarantee that the time required for the basis computation becomes negligible if compared to the one spent in system assembly and solution. Moreover, we have shown how cohomology and low-rank approximations are compatible in solving large scale problems in multiply connected domains by suitably handling the hierarchical matrix cluster tree. The presented approach is independent on the used compression library, and has a negligible impact on the computational costs if combined with a minimal length cohomology basis.

On the contrary, the non applicability of low-rank approximation techniques represents an important bottleneck for the **L**-approach, that unavoidably limits the size of the largest problem solvable on standard workstations. In particular, in MCF applications, the problem size grows rapidly with the complexity of machine geometry, characterized, for example, by a large number of pumping ports, insulation gaps, very narrow regions and unconventionally shaped elements. Despite integral formulations having shown to be more efficient than conventional FEM approaches for studying this kind of devices [1], only optimized integral methods (like the **C**-approach, in this work) perform well in these applications [4].

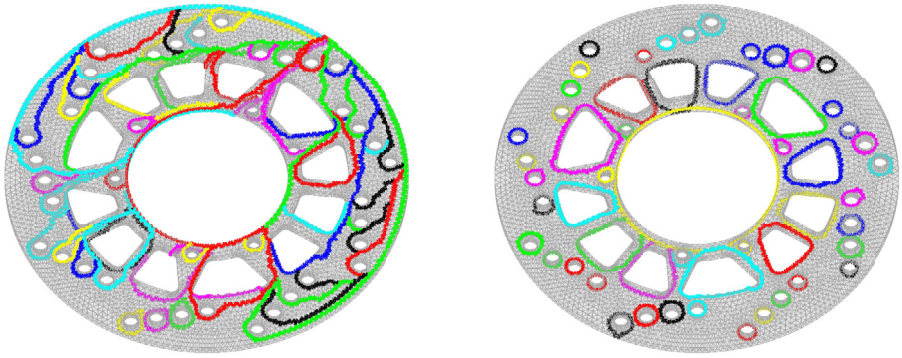
It is worth remarking that the proposed algorithm for the construction of a minimal length cohomology basis is also suitable for high-frequency problems in multiply connected domains where, usually, loop-star and loop-tree decomposition techniques are adopted [45,46].

Finally, it can be extended straightforwardly to the case of nonlinear materials, with interesting potential applications in MCF de-



(a) (b)

Fig. 20. Plate with 4 holes: $H^1(\partial\mathcal{K})$ basis computed with a single basepoint (a) and minimal length $H^1(\partial\mathcal{K})$ basis obtained with $N_{BP} = 5\beta_1(\mathcal{K}) = 20$ (b).



(a) (b)

Fig. 21. Disk with 55 holes: $H^1(\partial\mathcal{K})$ basis computed with a single basepoint (a) and minimal length $H^1(\partial\mathcal{K})$ basis obtained with $N_{BP} = 5\beta_1 = 275$ (b).

vices (e.g. modelling of ferritic steel tiles (FSTs) installed to reduce the toroidal magnetic field ripple).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 and 2019-2020 under grant agreement No. 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission. The authors would like to thank the University of Padova Strategic Research Infrastructure, Grant 2017, for allowing the access to the efficient computational infrastructure "CAPRI: Calcolo ad Alte Prestazioni per la Ricerca e l'Innovazione". Author Riccardo Torchio was financially supported by the Università degli Studi di Padova, Italy, BIRD 2019 program [no. BIRD 195949/19].

References

[1] R. Albanese, G. Rubinacci, IEE Proc. A 135 (1988) 457–462, https://doi.org/10.1049/ip-a-1.1988.0072.
[2] G. Meunier, O. Chadebec, J.-M. Guichon, IEEE Trans. Magn. 51 (2015), https://doi.org/10.1109/TMAG.2014.2365500.
[3] T.-T. Nguyen, G. Meunier, J.-M. Guichon, O. Chadebec, T.-S. Nguyen, IEEE Trans. Magn. 50 (2014) 549–552, https://doi.org/10.1109/TMAG.2013.2282957.
[4] P. Alotto, P. Bettini, R. Specogna, IEEE Trans. Magn. 52 (2016), https://doi.org/10.1109/TMAG.2015.2488699.
[5] E. Tonti, The Mathematical Structure of Classical and Relativistic Physics: A General Classification Diagram, Modeling and Simulation in Science, Engineering and Technology, Springer New York, 2013.
[6] P. Alotto, F. Freschi, M. Repetto, C. Rosso, The Cell Method for Electrical Engineering and Multiphysics Problems: An Introduction, Lecture Notes in Electrical Engineering, Springer Berlin Heidelberg, 2013.
[7] R. Torchio, IEEE Trans. Antennas Propag. 67 (2019) 7452–7465, https://doi.org/10.1109/TAP.2019.2927789.
[8] A. Ruehli, IEEE Trans. Microw. Theory Tech. 22 (1974) 216–221, https://doi.org/10.1109/TMTT.1974.1128204.
[9] T. Bauernfeind, P. Baumgartner, O. Bíró, A. Hackl, C. Magele, W. Renhart, R. Torchio, IEEE Trans. Magn. 54 (2018) 1–4, https://doi.org/10.1109/TMAG.2017.2771366.
[10] R. Torchio, F. Moro, G. Meunier, J.-M. Guichon, O. Chadebec, IEEE Trans. Magn. 55 (2019) 1–4, https://doi.org/10.1109/TMAG.2018.2889435.
[11] R. Albanese, G. Rubinacci, Adv. Imaging Electron Phys. 102 (1997) 1–86, https://doi.org/10.1016/S1076-5670(08)70121-6.
[12] J. Munkres, Elements of Algebraic Topology, CRC Press, 2018.
[13] P. Bettini, M. Passarotto, R. Specogna, IEEE Trans. Magn. 53 (2017), https://doi.org/10.1109/TMAG.2017.2663112.
[14] D. Voltolina, P. Bettini, P. Alotto, F. Moro, R. Torchio, IEEE Trans. Magn. 55 (2019), https://doi.org/10.1109/TMAG.2019.2894618.
[15] V. Cirimele, R. Torchio, A. Virgillito, F. Freschi, P. Alotto, Energies 12 (2019), https://doi.org/10.3390/en12142677.
[16] P. Dlotko, R. Specogna, IEEE Trans. Magn. 50 (2014) 577–580, https://doi.org/10.1109/TMAG.2013.2281076.
[17] P. Bettini, R. Specogna, J. Comput. Phys. 273 (2014) 100–117, https://doi.org/10.1016/j.jcp.2014.04.060.
[18] J. Stratton, Electromagnetic Theory, IEEE Press Series on Electromagnetic Wave Theory, Wiley, 2007.
[19] P. Yla-Oijala, J. Markkanen, S. Jarvenpaa, S.P. Kiminki, Prog. Electromagn. Res. 149 (2014) 15–44, https://doi.org/10.2528/PIER14070105.
[20] A. Bossavit, I. Mayergoyz, Computational Electromagnetism: Variational Formulations, Complementarity, Edge Elements, Electromagnetism, Elsevier Science, 1998.
[21] L. Codecasa, R. Specogna, F. Trevisan, J. Comput. Phys. 229 (2010) 7401–7410, https://doi.org/10.1016/j.jcp.2010.06.023.
[22] M. Kamon, J. White, M. Tsuk, IEEE Trans. Microw. Theory Tech. 42 (1994) 1750–1758, https://doi.org/10.1109/22.310584.

[23] J. Siau, G. Meunier, O. Chadebec, J.-M. Guichon, R. Perrin-Bit, IEEE Trans. Electromagn. Compat. 58 (2016) 1587–1594, https://doi.org/10.1109/TEMC.2016.2559801.
[24] N. Balabanian, T. Bickart, Electrical Network Theory, R.E. Krieger Publishing Company, 1983.
[25] I. Munteanu, Int. Compumag Soc. Newsl. 9 (2002) 10–14.
[26] METIS (ver. 5.1.0), http://glaros.dtc.umn.edu/gkhome/metis/metis/download. (Accessed 30 January 2018).
[27] D. La Salle, G. Karypis, Efficient Nested Dissection for Multicore Architectures, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9233, 2015, pp. 467–478.
[28] P. Dłotko, R. Specogna, Comput. Phys. Commun. 184 (2013) 2257–2266, https://doi.org/10.1016/j.cpc.2013.05.006.
[29] P. Giblin, Graphs, Surfaces and Homology, Cambridge University Press, 2010.
[30] G. Rubinacci, A. Tamburrino, IEEE Trans. Magn. 46 (2010) 2791–2794, https://doi.org/10.1109/TMAG.2010.2043722.
[31] T. Cormen, C. Leiserson, R. Rivest, C. Stein, Graph Algorithms, MIT Press, 2003.
[32] R. Hiptmair, J. Ostrowski, SIAM J. Comput. 31 (2002) 1405–1423, https://doi.org/10.1137/S0097539701386526.
[33] Z. Arai, Nonlinear Theory Appl., IEICE 4 (2013) 104–110, https://doi.org/10.1587/nolta.4.104.
[34] P. Bamberg, S. Sternberg, A Course in Mathematics for Students of Physics: Volume 2, Cambridge University Press, 1988.
[35] W. Hackbusch, B. Khoromskij, J. Comput. Appl. Math. 125 (2000) 479–501, https://doi.org/10.1016/S0377-0427(00)00486-6, Numerical Analysis 2000. Vol. VI: Ordinary Differential Equations and Integral Equations.
[36] J. Ballani, D. Kressner, Matrices with Hierarchical Low-Rank Structures, Springer International Publishing, Cham, 2016, pp. 161–209.
[37] HLIBpro (ver. 2.6), http://hlibpro.com. (Accessed 11 January 2017).
[38] STRUMPACK (ver 1.1.1), http://portal.nersc.gov/project/sparse/strumpack. (Accessed 11 January 2017).
[39] D. Kressner, S. Massei, L. Robol, hm-toolbox: Matlab software for hodlr and hss matrices, https://github.com/numpi/hm-toolbox, arXiv preprint, arXiv:1909.07909.
[40] HODLRlib (ver. 3.1415), http://hodlrlib.readthedocs.io/en/latest/. (Accessed 2 January 2018).
[41] M. Bebendorf, Approximation of Discrete Integral Operators, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 99–192.
[42] K. Fujiwara, T. Nakata, Compel 9 (1990) 137–154, https://doi.org/10.1108/eb010071.
[43] CAPRI: Calcolo ad alte prestazioni per la ricerca e l'innovazione, http://dei.unipd.it, University of Padova Strategic Infrastructure Grant 2017.
[44] P. Bettini, C. Finotti, L. Grando, G. Marchiori, R. Specogna, Fusion Eng. Des. 123 (2017) 518–521, https://doi.org/10.1016/j.fusengdes.2017.06.001.
[45] C. Forestiere, G. Miano, G. Rubinacci, A. Tamburrino, L. Udpa, S. Ventre, IEEE Trans. Antennas Propag. 65 (2017) 1224–1235, https://doi.org/10.1109/TAP.2016.2647585.
[46] F.P. Andriulli, IEEE Trans. Antennas Propag. 60 (2012) 2347–2356, https://doi.org/10.1109/TAP.2012.2189723.