

Computation of stationary 3D halo currents in fusion devices with accuracy control



Paolo Bettini ^{a,b,*}, Ruben Specogna ^c

^a Università degli Studi di Padova, Dipartimento di Ingegneria Industriale (DII), Via Gradenigo 6/A, 35131 Padova, Italy

^b Consorzio RFX, C.so Stati Uniti 4, 35127 Padova, Italy

^c Università degli Studi di Udine, Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica (DIEGM), Via delle Scienze 206, I-33100 Udine, Italy

ARTICLE INFO

Article history:

Received 22 August 2013

Received in revised form 7 March 2014

Accepted 23 April 2014

Available online 14 May 2014

Keywords:

Magnetic confinement fusion

Halo currents

Neumann boundary value problem

Elliptic partial differential equations

Complementarity

Mixed-hybrid formulation

Lazy cohomology generators

ABSTRACT

This paper addresses the calculation of the resistive distribution of *halo currents* in three-dimensional structures of large magnetic confinement fusion machines. A Neumann electrokinetic problem is solved on a geometry so complicated that *complementarity* is used to monitor the discretization error. An irrotational electric field is obtained by a geometric formulation based on the electric scalar potential, whereas three geometric formulations are compared to obtain a solenoidal current density: a formulation based on the electric vector potential and two geometric formulations inspired from mixed and mixed-hybrid Finite Elements. The electric vector potential formulation is usually considered impractical since an enormous computing power is wasted by the topological pre-processing it requires. To solve this challenging problem, we present novel algorithms based on *lazy cohomology generators* that enable to save orders of magnitude computational time with respect to all other state-of-the-art solutions proposed in literature. Believing that our results are useful in other fields of scientific computing, the proposed algorithm is presented as a detailed pseudocode in such a way that it can be easily implemented.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

A tokamak plasma with a vertically elongated cross section is intrinsically unstable. For elongations up to a critical value, stability can be maintained by surrounding the plasma with close-fitting passive conducting structures, see Fig. 1(a). In this way, the plasma is still unstable, but the vertical motion occurs on the slower resistive time-scale (ms) of these structures, allowing stabilization by means of a feedback control system. Nevertheless, during disruptions,¹ the vertical control can be lost generating what are called Vertical Displacement Events (VDEs). As plasma comes into contact with the vacuum vessel wall² some of the plasma current, the so called *halo current*, flows in a circuit lying partially in the plasma and partially in the vessel wall, see Fig. 1(b). The interaction of the currents injected in the passive structures of the machine with a magnetic field of the order of some Tesla, needed for plasma confinement, causes huge electromagnetic forces.

* Corresponding author at: Università degli Studi di Padova, Dipartimento di Ingegneria Industriale (DII), Via Gradenigo 6/A, 35131 Padova, Italy. Tel.: +39 049 8277545; Fax: +39 049 8277699.

E-mail addresses: paolo.bettini@unipd.it (P. Bettini), ruben.specogna@uniud.it (R. Specogna).

¹ Disruptions are rapid events in which a large fraction of the plasma thermal energy is lost due to the uncontrolled growth of some large-scale plasma instability.

² The vacuum vessel is a hermetically-sealed steel container that surrounds the plasma and houses the fusion reaction.

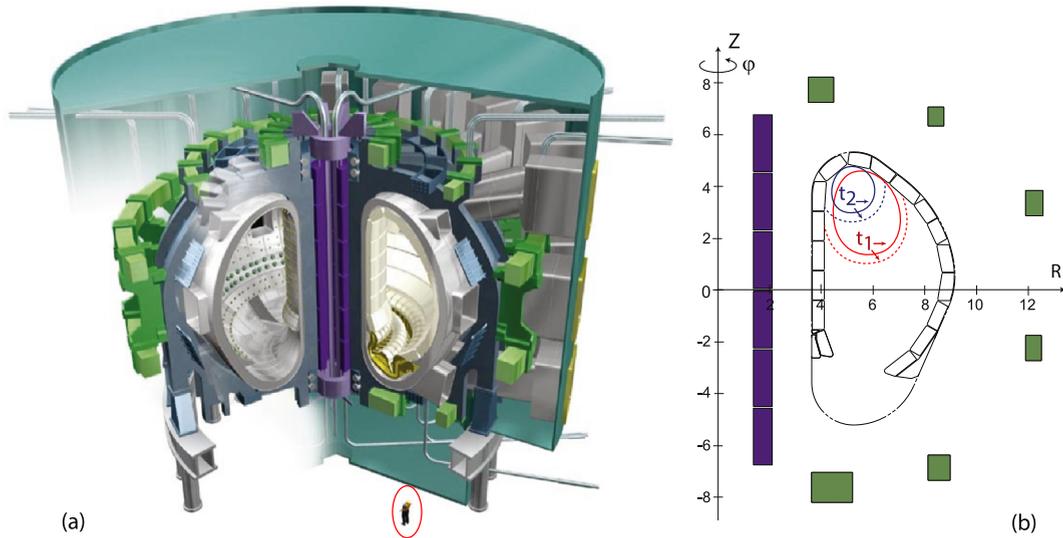


Fig. 1. (a) The ITER Tokamak (nearly 30 metres tall, 23 thousand tons weight) is made up of an estimated one million parts. The man (in the circle) is used as a reference of the machine's scale. Image provided courtesy of Fusion for Energy (F4E). (b) An example of upward symmetric VDE: time slices ($t_1 < t_2$) of plasma last closed flux surface (solid) and halo boundaries (dashed).

A fully self-consistent model that may be used as a predictive tool for these forces is not yet available due to the complexity of the physical magneto-hydrodynamic phenomena and the quite complicated domain of study. Over the years, several 2D numerical codes have been used to model the 2D dynamics and localized forces of VDEs in experimental devices and provide some predictions for fusion reactors under design. On the other hand, halo currents are an intrinsically 3D phenomenon. For this reason, a big effort has been made to simulate asymmetric VDEs and disruptions in Tokamaks also by employing 3D MHD codes (e.g. M3D [1]), although with a resistive axisymmetric wall. Despite halo currents being a typical consequence of fast plasma movements (VDE) towards the wall, it is possible to study, to some extent, a steady state distribution of the halo currents in the 3D structures.

From the engineering point of view, the problem consists of the prediction of electromagnetic forces on the conductive structures surrounding the plasma by solving an electrokinetic Neumann boundary value problem where the boundary data are the currents injected by the plasma to the first wall of the vessel. In [2] two codes have been benchmarked assuming a simplified plasma model and a resistive distribution of halo currents to give the first estimate on halo forces on a typical machine described with a fairly detailed geometric model. The first one is the well-known CARIDDI code [3] based on a gauged electric vector potential formulation, whereas the second one is the CAFE code [4] based on complementary formulations for stationary conduction.

The geometry and topology of the problem is so complicated that tetrahedral mesh generators usually fail during mesh computation. That is why a multiblock structured hexahedral mesh has been used in this paper. The electrokinetic problem may be solved employing the classical electric scalar potential formulation [5]. Formulations based on high order hexahedral elements are not attractive since they suffer from the lack of high order representation of geometry in multiblock mesh generators. Moreover, the complicated problem geometry requires many elements to describe every tiny handle of the geometry in such a way that elements cannot be drastically reduced by using high order elements. In addition, since most physical and geometric parameters are known with a tolerance in the percent range, extreme accuracy is unjustified in practice. On the contrary, the aim is to minimize the setup and the running time of the software tool.

In this paper we use *complementarity* [6–9] to have some estimate on the quality of the solution, exploiting its guaranteed upper and lower bounds for power dissipated [6] and the minimization of the *constitutive error* [7]. Complementarity consists in solving the considered problem two times, employing two complementary formulations that produce respectively an irrotational electric field and a solenoidal current density. Despite the additional computational cost, complementarity has potential advantages. First, the constitutive error provides a robust error estimator for mesh adaptivity [7]. Second, averaging the solutions obtained by both formulations gives a much better approximation that usually enable to obtain a sensible gain from exploiting complementarity [10,11].

The irrotational electric field is usually obtained by means of the standard electric scalar potential formulation \mathbf{V} [5], whereas this paper investigates three alternative formulations (\mathbf{T} , \mathbf{M} , \mathbf{H}) to obtain a solenoidal current density. We remark that these formulations provide exactly the same current density, even in topologically non-trivial domains.

The first formulation, based on an electric vector potential \mathbf{T} , requires a complicated topological pre-processing due to the fact that the domain under study is topologically non-trivial [12]. In particular, for Neumann problems, an $H^2(\mathcal{K} - \partial\mathcal{K})$

cohomology basis³ is needed to render the boundary value problem well defined [2], where \mathcal{K} is the cell complex that encode the incidences of the considered mesh. The representatives of such cohomology generators are called *thick links* [10]. For a rigorous introduction to algebraic topology consult for example [13] or, for non-formal ones, [12,15,10].

The topological pre-processing, if not faced carefully, is unfeasible in practice as both memory requirements and computational timings become prohibitive. This is especially true for applications arising in fusion engineering and design, where the cell complex easily exceed millions elements and the genus of $\partial\mathcal{K}$ a few thousands. That means thousands of generators to retrieve quickly and with reduced support in such a way that they can be stored in memory as sparse vectors. If any of these two requirements is not met, complementarity cannot be exploited in practical problems. And this is exactly the present state of literature. For instance, the code CARIDDI—quite popular in fusion applications—computes directly the kernel of a matrix whose entries are real numbers [2]: this approach is unfeasible for meshes with more than one million elements, whereas it is practical but very time consuming (a few hours) for meshes of some hundreds of thousands hexahedra. Moreover, it is prone to errors due to the finite precision of real numbers. Even state-of-the-art methods to compute cohomology generators as [15,16] would require days to produce the generators for the problems addressed in this paper. This is unacceptable if one thinks that the linear solver requires at least two orders of magnitude less computational effort.

The present paper proposes quite a novel approach to the determination of the required thick links that is able to reduce the computational effort from days to tens of seconds. This enormous reduction of computational effort makes the computation of cohomology generators negligible w.r.t. the time spent solving the linear system of equations. Moreover, defining thick links in the boundary of $\partial\mathcal{K}$ limits also their support in such a way that they occupy a negligible amount of memory. This result is not only due to novel algorithms, but especially to the recent introduction of the original mathematical concept of *lazy cohomology generators* [17,18]. The reasons why they enable this unprecedented speed-up and why are deemed as lazy will be explained after the presentation of the novel algorithm for thick links extraction. We remark that the cohomology generators may be useful also for other applications arising from scientific computing.

The electric vector potential formulation **T** is compared with two original mixed (**M**) and mixed-hybrid (**H**) formulations strongly inspired from Mixed Finite Elements (MFE) and Mixed-Hybrid Finite Elements (MHFE) [19,20]. The geometric formulations differ from the corresponding Finite Elements formulations since they use the dual complex [21] and discrete Hodge operators [22,23] that are symmetric, positive definite, consistent and can be constructed avoiding numerical quadrature, for general polyhedral meshes, by elementary closed form operations involving the geometric elements of the primal and dual complexes.

This paper is structured as follows. In Section 2 the discretization of the numerical domain in primal/dual cell complexes is presented. Section 3 contains a concise introduction to the electric vector potential **T** formulation. Section 4 presents the very idea behind lazy generators and the novel graph-theoretic and general algorithm for lazy cohomology group computations. The algorithm is presented as a collection of detailed pseudocodes in Appendix A, in such a way that it can be easily implemented in any computer code based on Finite Elements, Finite Volumes, integral formulations, compatible or mimetic discretizations and geometric formulations. Section 5 introduces a fast and graph-theoretic method to take into account the source currents without solving any linear system or performing a singular value decomposition. Section 6 describes the mixed **M** and mixed-hybrid **H** geometric formulations. Section 7 is devoted to present the numerical results while in Section 8 the conclusions are drawn.

2. Domain discretization

Let us consider a stationary current conduction problem in a conductive region K , which is a compact connected subset of the three-dimensional Euclidean space. The conductor K is immersed in a perfectly insulating medium and possesses in general topological handles and cavities (meaning that the 1st $\beta_1(K)$ and 2nd $\beta_2(K)$ Betti numbers [13] are nonzero, respectively).

The conductor K is covered by a hexahedral mesh, whose incidences are encoded in the *cell complex* \mathcal{K} represented by the standard incidence matrices **G**, **C** and **D**, [21,9]. **G_b**, **C_b** and **D_b** denote incidence matrices containing incidences between elements on $\partial\mathcal{K}$ only. A dual barycentric complex $\tilde{\mathcal{K}}$ is obtained from \mathcal{K} by using the *barycentric subdivision* [21,15]. Let us define the dual cell complex $\tilde{\mathcal{K}} = D(\mathcal{K})$ in the following way:

- (1) For every hexahedron $v \in \mathcal{K}$, the dual node $\tilde{n} = D(v)$ is defined as the barycenter of v , see Fig. 2(a).
- (2) For every face $f \in \mathcal{K}$ that is a common face of hexahedra $v_1, v_2 \in \mathcal{K}$, the dual edge $\tilde{e} = D(f)$ is defined as the sum of a segment of line joining the barycenter of f with $D(v_1)$ and a segment of line joining the barycenter of f with $D(v_2)$, see Fig. 2(a).
- (3) For every edge $e \in \mathcal{K}$ let $f_1, \dots, f_n \in \mathcal{K}$ be the faces incidental to e . The dual face $\tilde{f} = D(e)$ is then defined as a (set-theoretical) sum $\bigcup_{i=1}^n \text{conv}[B(e), D(f_i)]$, where $B(e)$ denotes the barycenter of the edge e and $\text{conv}[\cdot]$ the convex hull, see Fig. 2(b).
- (4) For every node $n \in \mathcal{K}$ let $e_1, \dots, e_n \in \mathcal{K}$ be the edges incidental to n . The dual volume $\tilde{v} = D(n)$ is the volume bounded by $D(e_1), \dots, D(e_n)$, see Fig. 2(b).

³ We remark that the absolute 2nd cohomology group $H^2(\mathcal{K} - \partial\mathcal{K})$ is torsion-free [13] and any cohomology basis is suitable for this application, so issues related to basis selection [14] are not of interest in this context.

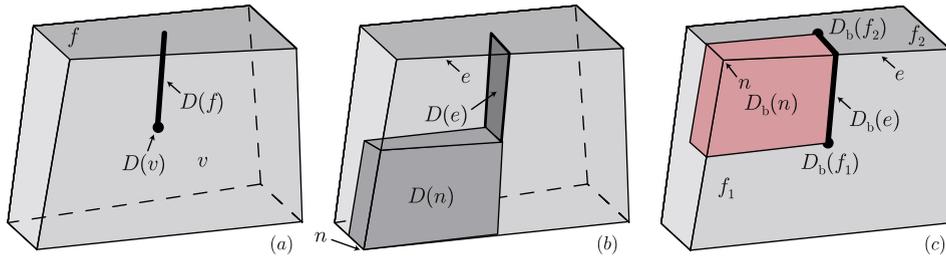


Fig. 2. (a) Example of dual node $D(v)$ and dual edge $D(f)$. (b) Example of dual face $D(e)$ and dual volume $D(n)$. (c) Example of dual node $D_b(f)$, dual edge $D_b(e)$ and dual face $D_b(n)$ on the boundary of the hexahedron.

The matrices $\tilde{\mathbf{G}} = \mathbf{D}^T$, $\tilde{\mathbf{C}} = \mathbf{C}^T$ and $\tilde{\mathbf{D}} = -\mathbf{G}^T$ represent the incidence matrices of $\tilde{\mathcal{K}}$ [21,15].

In case of three-manifolds with boundary, the dual complex $\tilde{\mathcal{K}}$ constructed as previously described is not complete, being undefined on $\partial\mathcal{K}$. In this case, it is customary to complete $\tilde{\mathcal{K}}$ by adding a barycentric dual complex on $\partial\mathcal{K}$ as follows:

- (1) For every face $f \in \partial\mathcal{K}$, the boundary dual node $\tilde{n}_b = D_b(f)$ is defined as the barycenter of f , see Fig. 2(c).
- (2) For every edge $e \in \partial\mathcal{K}$ that is a common edge of faces $f_1, f_2 \in \partial\mathcal{K}$, the boundary dual edge $\tilde{e}_b = D_b(e)$ is defined as the sum of a segment of line joining the barycenter of e with $D_b(f_1)$ and a segment of line joining the barycenter of e with $D_b(f_2)$, see Fig. 2(c).
- (3) For every node $n \in \partial\mathcal{K}$, let $e_1, \dots, e_n \in \partial\mathcal{K}$ be the edges incidental to n . The boundary dual face $\tilde{f}_b = D_b(n)$ is the portion of $\partial\mathcal{K}$ bounded by $D_b(e_1), \dots, D_b(e_n)$, see Fig. 2(c).

As D denotes the standard dual operator, with D_b we refer to its corresponding operator on $\partial\mathcal{K}$. The elements of $\partial\mathcal{K}$, which is a combinatorial two-manifold without boundary, are subject to a Neumann boundary condition, as described in Section 7.

3. Electric vector potential formulation

In the electric scalar potential formulation \mathbf{V} [5,10], Faraday’s law holds by potentials whereas Gauss’s law is imposed by a linear system of equations. In electric vector potential formulations \mathbf{T} , both the association of the physical variables with geometric elements of the cell complexes and the role of the two physical laws are interchanged. That is, in order to formulate the problem by using the vector potential formulation, the following cochains with real coefficients [13,9,10] are introduced:

- Voltage $\tilde{\mathbf{U}} \in C^1(\tilde{\mathcal{K}}, \mathbb{R})$;
- Electric current $\mathbf{I} \in C^2(\mathcal{K}, \mathbb{R})$;
- Electric vector potential $\mathbf{T} \in C^1(\mathcal{K}, \mathbb{R})$.

Differently from [2,4], let us express the current \mathbf{I} as

$$\mathbf{I} = \mathbf{C} \left(\mathbf{T}_s + \mathbf{T} + \sum_{i=1}^N I_g^i \Lambda^i \right), \tag{1}$$

where the 1-chains $\{\Lambda^i\}_{i=1}^N$, $N = \beta_1(\mathcal{K})$, are the *thick links* on $\partial\mathcal{K}^4$ and $\{I_g^i\}_{i=1}^N$ are the corresponding *independent currents*. By construction, the supports of the $\{\Lambda^i\}_{i=1}^N$ are nonzero only on edges of $\partial\mathcal{K}$ and their dual in $\partial\mathcal{K}$, $\{D_b(\Lambda^i)\}_{i=1}^N$, are the representatives of an $H_1(\tilde{\mathcal{K}})$ homology basis. This is shown in Fig. 3, where the support of Λ^1 for a thick torus is visible in Fig. 3(a), the support of its dual on $\partial\mathcal{K} D_b(\Lambda_1)$ in Fig. 3(b), whereas the support of a representative of the $H_1(\tilde{\mathcal{K}})$ generator is represented in Fig. 3(c). How to construct the thick links automatically and efficiently is the aim of Section 4.

We remark that this solution is different w.r.t. the one presented in [2,4] where thick links $\{\Pi^i\}_{i=1}^N$ have been defined as $H^2(\mathcal{K} - \partial\mathcal{K})$ cohomology generators (i.e. their supports are set of faces in the interior of \mathcal{K}). One may easily produce the $\{\Pi^i\}_{i=1}^N$ from $\{\Lambda^i\}_{i=1}^N$ with $\mathbf{C}\Lambda^i = \Pi^i$, for $i = 1, \dots, N$, but the formulation presented in this paper is more efficient and easier to implement. In fact, defining thick links in the boundary of $\partial\mathcal{K}$ limits also their support in such a way that they occupy a negligible amount of memory.

Concerning boundary conditions, \mathbf{T} is imposed to zero on edges belonging to $\partial\mathcal{K}$, whereas \mathbf{T}_s is efficiently determined as explained in Section 5 from the source current \mathbf{I}_s whose coefficients on $\partial\mathcal{K}$ faces are the plasma halo currents imposed through them computed as described in Section 7.

⁴ In [24], which presents an integral formulation for eddy currents, the thick links on the boundary have been called *ribbons*.

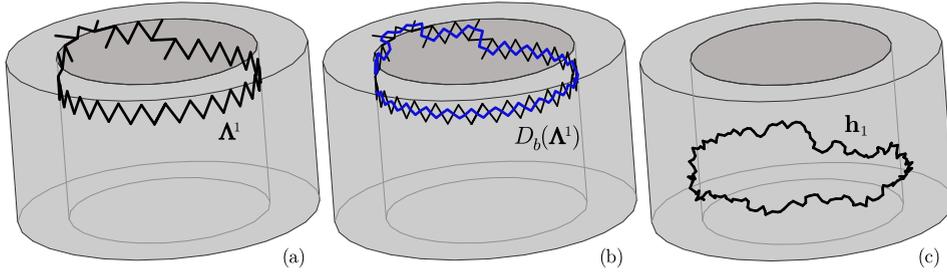


Fig. 3. (a) The support of Λ^1 lies on the boundary of solid two-dimensional torus. (b) $D_b(\Lambda^1)$ is the representative of the $H_1(\tilde{\mathcal{K}})$ generator. (c) The support of the representative \mathbf{h}_1 of the $H_1(\tilde{\mathcal{K}})$ generator. Clearly $D_b(\Lambda^1)$ and \mathbf{h}_1 are homologous since one can construct some 2-chain in $\tilde{\mathcal{K}}$ such that its boundary is the difference between $D_b(\Lambda^1)$ and \mathbf{h}_1 .

By defining the potentials as in (1), the current continuity law

$$\mathbf{DI} = \mathbf{0} \quad (2)$$

is implicitly satisfied, since $\mathbf{DC} = \mathbf{0}$ [21].

The resistance matrix \mathbf{R} is a discrete Hodge operator [25] that relates currents \mathbf{I} to voltages $\tilde{\mathbf{U}}$

$$\tilde{\mathbf{U}} = \mathbf{RI}. \quad (3)$$

\mathbf{R} is the approximate discrete counterpart of the constitutive relation $E = \rho J$ at continuous level, ρ being the resistivity of the material assumed element-wise constant. In this paper, the resistance matrix is constructed by simple closed form expressions avoiding numerical quadrature as described in [22,23]. We note that this technique is valid for general star-shaped polyhedra. Faraday's discrete law,

$$\mathbf{C}^T \tilde{\mathbf{U}} = \mathbf{0} \quad (4)$$

is enforced by a linear system of equations. By substituting (3) and (1) in (4), an algebraic equation is obtained

$$\mathbf{KT} + \sum_{j=1}^N (\mathbf{K}\Lambda^j) I_g^j = -\mathbf{KT}_s, \quad (5)$$

where the stiffness matrix \mathbf{K} is defined as $\mathbf{K} = \mathbf{C}^T \mathbf{RC}$. We note that the stiffness matrix may be computed directly also by using the Finite Element Method [9].

The final algebraic linear system of equations (5) contains as unknowns also the independent currents $\{I_g^j\}_{j=1}^N$. To write as many equations as the unknowns, Faraday's laws have to be written also on every thick link (corresponding to each homology $H_1(\tilde{\mathcal{K}})$ generator) by setting their voltage evaluation to zero

$$\Lambda^{iT} \mathbf{C}^T \tilde{\mathbf{U}} = 0, \quad i \in \{1, \dots, N\}. \quad (6)$$

Since any dual 1-cycle may be obtained as a linear combination of the homology $H_1(\tilde{\mathcal{K}})$ basis plus a boundary, once the voltage on an $H_1(\tilde{\mathcal{K}})$ basis vanishes, it vanishes on any other cycle nontrivial in $H_1(\tilde{\mathcal{K}})$. By using (3) and (1), each one of these equations can be written in terms of unknowns as

$$(\Lambda^{iT} \mathbf{K}) \mathbf{T} + \sum_{j=1}^N (\Lambda^{iT} \mathbf{K}\Lambda^j) I_g^j = -\Lambda^{iT} \mathbf{KT}_s. \quad (7)$$

The sparse and symmetric linear system is solved using a state-of-the-art direct solver (PARDISO, included in the Intel MKL library), with a standard tree-cotree gauge [3].

4. Topological preprocessing for potential definition

The present paper proposes a novel approach for the determination of the required thick links that reduces the computational effort from days to tens of seconds. The novel Algorithm 1 introduced in this paper is an adaptation of the *Łotko–Specogna* (DS) algorithm [17,18] that generates $H^1(\mathcal{K})$ lazy generators to compute an $H^1(\partial\mathcal{K})$ basis instead.

Let \mathbf{G}_b and \mathbf{C}_b be the standard edge-node and face-edge incidence matrices restricted to elements of $\partial\mathcal{K}$. First, an edge spanning tree on $\partial\mathcal{K}$ is found with Algorithm 4 (see Appendix A) by a Breadth First Search (BFS) strategy [26], see Fig. 4(a). Second, a dual edge spanning tree on $D_b(\partial\mathcal{K})$ is computed again by a BFS technique with Algorithm 5 (see Appendix A). This dual tree is constructed by using as graph not all dual edges in $D_b(\partial\mathcal{K})$, but only the edges that are not in the primal

Algorithm 1 Generation of an $H^1(\partial\mathcal{K})$ basis.

Input: $\mathbf{C}_b, \mathbf{G}_b$
Output: representatives $\Lambda^1, \dots, \Lambda^{2g}$ of an $H^1(\partial\mathcal{K})$ basis;
 $\text{primalTree} \leftarrow \text{FindPrimalTree}(\mathbf{G}_b)$;
 $\text{dualTree} \leftarrow \text{FindDualTree}(\mathbf{C}_b, \text{primalTree})$;
 $c \leftarrow 0$;
for each edge E in $\partial\mathcal{K}$ **do**
 if $\text{primalTree}(E) = \text{false}$ and $\text{dualTree}(E) = \text{false}$ **then**
 $c \leftarrow c + 1$;
 $\Lambda^c \leftarrow \text{CocycleRetrieval}(\mathbf{C}_b, \text{dualTree}, E)$;
return $\Lambda^1, \dots, \Lambda^{2g}$;

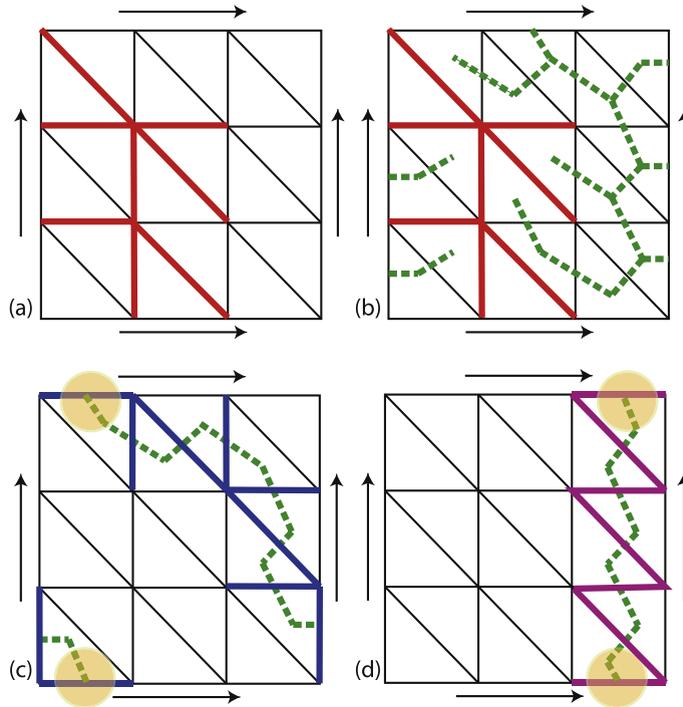


Fig. 4. (a) An edge spanning tree on $\partial\mathcal{K}$, boundary of a solid two-dimensional torus. (Opposite sides are identified.) (b) The dual edge spanning tree on $\partial\mathcal{K}$ is constructed by using as graph the dual edges whose dual are edges not in the primal tree. (c)–(d) The support of the two representatives of the cohomology $H^1(\partial\mathcal{K})$ generators produced by the two free edges indicated with circles in the picture.

tree, see for example Fig. 4(b). Then, each “free” edge in $\partial\mathcal{K}$ that does not belong neither to the primal tree nor to the dual tree produces an $H^1(\partial\mathcal{K})$ generator. In fact, once a free edge E interpreted as a dual edge $D_b(E)$ is added to the dual tree, exactly one cycle is produced on the dual complex $D_b(\partial\mathcal{K})$, see Fig. 4(c)–(d). The dual of these dual cycles, obtained by Algorithm 6 (see Appendix A), are the desired cohomology generators. In particular, to retrieve this cocycle, the two faces F_1, F_2 on $\partial\mathcal{K}$ that have E in their boundaries are found. F_1 and F_2 may be interpreted as dual nodes on the complex $D_b(\partial\mathcal{K})$. Then, a discrete distance field on the dual tree from dual node $D_b(F_1)$ is found by a BFS strategy [26]. The distance field propagation stops when the $D_b(F_2)$ dual node is reached. Finally, the dual cycle is retrieved starting from $D_b(F_2)$ and following back the predecessors until $D_b(F_1)$ is reached.

The algorithm can be heavily parallelized, since all connected components of $\partial\mathcal{K}$ and generators can be processed simultaneously. Secondly, concerning the complexity analysis, the algorithm exhibits an optimal linear worst-case complexity of $O(\text{card}(\partial\mathcal{K})g)$. Since in practical problems the genus g is bounded by a constant (i.e. $O(1)$), the worst-case complexity is linear $O(\text{card}(\partial\mathcal{K}))$ w.r.t. the elements in the boundary of the domain of study.

One may realize that this algorithm does not solve the problem of determining a basis that can be used in the formulation described in the previous section, since the number of obtained generator is twice the number of required ones. For example, for a solid two-dimensional torus (i.e. the genus g of $\partial\mathcal{K}$ is one) we get two generators Λ^1 and Λ^2 in place of one, see Fig. 5(a). Λ^1 is a useful generator since its dual on the boundary $D_b(\Lambda^1)$ is nontrivial in $H_1(\tilde{\mathcal{K}})$, whereas $D_b(\Lambda^2)$ is trivial in $H_1(\tilde{\mathcal{K}})$. In literature, eliminating generators as Λ^2 has always been deemed as necessary [3,27,28,2]. This is not as trivial as it may appear at the first glance because the required and redundant generators may be “mixed”, see for example Fig. 5(b), requiring a considerable effort to untie them. This problem has been elegantly solved and there exist complicated and time consuming techniques based on computing some linking numbers and a Smith Normal Form [13] of an integer

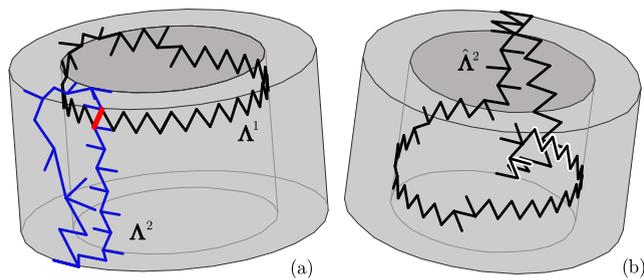


Fig. 5. (a) The two representatives of generators Λ^1 and Λ^2 are a basis of the first cohomology group $H^1(\partial\mathcal{K})$ of the boundary of a solid two-dimensional torus. Λ^1 represents a useful generator whereas eliminating Λ^2 has been always deemed as necessary in literature [3,27,28,2]. (b) The representative of a mixed generator $\hat{\Lambda}^1$ cohomologous to $\Lambda^1 + \Lambda^2$ in $H^1(\partial\mathcal{K})$.

matrix that are able to change the cohomology basis to extract the required generators [27,17,28]. This paper presents an alternative solution to this issue that is described in what follows.

4.1. Lazy cohomology generators

Discriminating the redundant generators is computationally costly and complicated, drastically reducing the efficiency of the whole algorithm and greatly complicating its implementation.

Here, the very idea is to refuse to do it and use all $H^1(\partial\mathcal{K}) \simeq H_1(D_b(\partial\mathcal{K}))$ generators (for example, the ones produced with Algorithm 1) in the formulation. Surprisingly, this key idea has been presented in literature only recently in the different context of eddy current problems, see [17,18]. In this paper we extend this idea to elliptic boundary value problems.

From Mayer–Vietoris exact sequence for homology [13,27], one has that $H_1(D_b(\partial\mathcal{K})) = H_1(\tilde{\mathcal{K}}) \oplus H_1(\tilde{\mathcal{K}}^c)$, where $\tilde{\mathcal{K}}^c$ is the dual complex of a triangulation of the exterior of \mathcal{K} . Since $\beta_1(\tilde{\mathcal{K}}) = \beta_1(\tilde{\mathcal{K}}^c) = g$ [13], g being the genus of $\partial\mathcal{K}$, the representative Λ^i of the i th $H^1(\partial\mathcal{K})$ generator can be expressed as a linear combination $\Lambda^i = \sum_{h=1}^g \alpha_h \mathbf{a}^h + \sum_{k=1}^g \beta_k \mathbf{b}^k$, where $\{D_b(\mathbf{b}^k)\}_{k=1}^g$ is a basis of $H_1(\tilde{\mathcal{K}})$ whereas $\{D_b(\mathbf{a}^k)\}_{k=1}^g$ is a basis of $H_1(\tilde{\mathcal{K}}^c)$. We note that the dual 1-cycles $\{D_b(\mathbf{a}^k)\}_{k=1}^g$ are homologically trivial in $H_1(\tilde{\mathcal{K}})$. This shows that the dual of the $2g$ representatives $\{\Lambda^i\}_{i=1}^{2g}$ of $H^1(\partial\mathcal{K})$ generators span $H_1(\tilde{\mathcal{K}})$ but they are not a basis, some of them being linearly dependent. Such 1-cocycles $\{\Lambda^i\}_{i=1}^{2g}$ are defined as *lazy cohomology generators* [17].

It has been already shown in [17] that lazy generators may be safely used as if they were a standard basis. In fact, if the dual of a lazy generator is trivial in $H_1(\tilde{\mathcal{K}})$, the corresponding nonlocal Faraday’s law (6) does not add any information, since it can be obtained as a linear combination of local Faraday’s laws (4). The other possible case is when lazy generators are dependent but their dual are nontrivial in $H_1(\tilde{\mathcal{K}})$. To consider this case, we first note that one may pick a subset of lazy generators in such a way their dual are an $H_1(\tilde{\mathcal{K}})$ basis [17]. Then, the voltage on the $H_1(\tilde{\mathcal{K}})$ basis is set to zero by the nonlocal Faraday’s law (6). Now, the voltage on every dual 1-cycle vanishes since any dual 1-cycle may be obtained as a linear combination of the homology $H_1(\tilde{\mathcal{K}})$ basis plus a boundary. So no inconsistent constraint can arise if we impose to zero explicitly the voltage on remaining lazy generators. To conclude, the resulting linear system is rank deficient (i.e. some rows of the system are not linearly independent) but consistent. It is easy to catch the analogy of what happens with *ungauged* formulations [29].

The generation of thick links exploiting the idea of lazy generators can be performed with five orders of reduction of computational time, whereas the penalty of using lazy generators in place of standard ones is negligible, since the systems solved with direct solvers or even with iterative solvers converge almost as fast as in the standard case.

In this paper we introduce a further optimization in the algorithm especially suitable when Algorithm 1 is applied to problems that require thousands generators as in fusion engineering problems. In these cases, in fact, the dual cycles retrieval—i.e. *CocycleRetrieval* of Algorithm 1—becomes a bottleneck, since it is still true that g is bounded by a constant, but the constant is relatively large.

4.2. Optimizing the cocycle retrieval

It is intuitive that independently retrieving many thousands of dual cycles at *CocycleRetrieval* of Algorithm 1 is inefficient. In the following, we propose a technique to retrieve simultaneously all generators. The idea is based on the fact that in an acyclic and connected graph (each connected component of $\partial\mathcal{K}$ is considered separately), for each pair of nodes $D_b(F_1)$, $D_b(F_2)$ there exists only one simple path (i.e. a path that crosses any edge at most one time) that connects them.

The whole novel algorithm is illustrated in Algorithm 2. The novel idea is to find just one distance field *dist* on the whole $\partial\mathcal{K}$, by starting from a random face T on each connected component of $\partial\mathcal{K}$.

This can be easily performed by Algorithm 7 (see Appendix A) whose output is an array *dist* containing, for each face in $\partial\mathcal{K}$, an integer representing the topological distance w.r.t. the seed face T of the corresponding connected component of

Algorithm 2 Efficient generation of an $H^1(\partial\mathcal{K})$ basis.

Input: C_b, G_b
Output: representatives $\Lambda^1, \dots, \Lambda^{2g}$ of an $H^1(\partial\mathcal{K})$ basis;
 $primalTree \leftarrow FindPrimalTree(G_b);$
 $dualTree \leftarrow FindDualTree(C_b, primalTree);$
 $[dist, parent] \leftarrow FindBFSDistance(C_b, dualTree);$
 $c \leftarrow 0;$
for each edge E in $\partial\mathcal{K}$ **do**
 if $primalTree(E) = false$ and $dualTree(E) = false$ **then**
 $c \leftarrow c + 1;$
 $\Lambda^c \leftarrow FastCocycleRetrieval(C_b, dist, parent, E);$
return $\Lambda^1, \dots, \Lambda^{2g};$

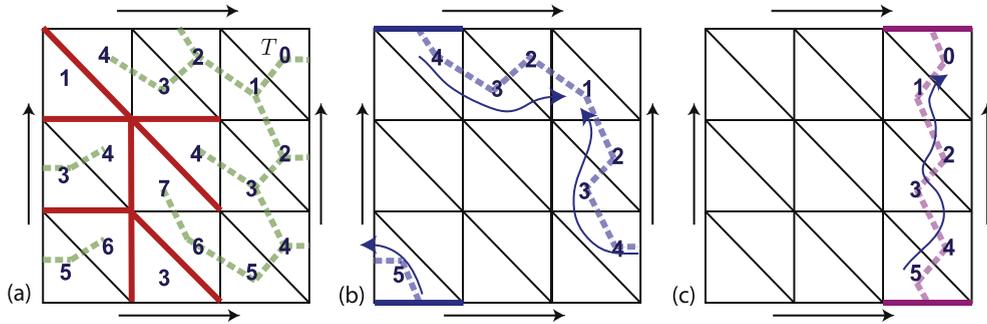


Fig. 6. (a) Distance field $dist$ from T for the boundary of a solid two-dimensional torus. (Opposite sides are identified.) (b)–(c) The support of the two representatives of cohomology $H^1(\partial\mathcal{K})$ generators produced by the two free edges with the distance field $dist$.

$\partial\mathcal{K}$ and the array $parent$ containing the predecessor information. An example of such distance for the boundary of a solid two-dimensional torus is depicted in Fig. 6(a).

Then, each dual cycle may be retrieved by the only distance field available $dist$ by Algorithm 8 (see Appendix A). That is, the dual path between a pair of dual nodes $D_b(F_1), D_b(F_2)$ that are connected by the dual $D_b(E)$ of the free edge E that defines the generic i th $H^1(\partial\mathcal{K})$ generator Λ^i is found by considering the predecessors of $D_b(F_1)$ and $D_b(F_2)$ until the first common predecessor is found. The dual edges traversed during the predecessors visit form the dual cycle and their coefficients of the cocycle Λ^i are easily found, see Fig. 6(b)–(c).

Let us now analyze how the complexity is decreased from $O(card(\partial\mathcal{K})g)$ by using the novel algorithm. The distance field computation on the whole $\partial\mathcal{K}$ takes $O(card(\partial\mathcal{K}))$, whereas the part illustrated in Algorithm 8 considers exactly the edges needed to find the required paths. Therefore, the total worst-case complexity decreases from $O(card(\partial\mathcal{K})g)$ to $O(card(\partial\mathcal{K}) + P)$, where P is the sum of the cardinality of all the dual cycles. In fusion problems, g is a few thousands and each path contains a small subset of $\partial\mathcal{K}$ edges, therefore the gain of using this algorithm may be sensible. The difference in term of computational time is investigated by numerical experiments in Section 7.

5. Efficient computation of sources

In this section we propose an easy graph-theoretic technique to enforce the inhomogeneous Neumann boundary condition on $\partial\mathcal{K}$. In literature usually this is performed by computing the coefficients of the array \mathbf{T}_s in all \mathcal{K} by a quite costly pseudo-inverse, as performed in CARIDDI [3,2].

To have a consistent solution, we first require that $\langle \mathbf{I}_s, \partial\mathcal{K} \rangle = 0$. With this assumption, it is possible to apply a Generalized Spanning Tree Technique (GSTT) [30,2] on the dual complex $\tilde{\mathcal{K}}$ to find the coefficients of \mathbf{I}_s for the faces in the interior of \mathcal{K} .

This is performed by finding a tree with a BFS strategy on the graph obtained by considering all dual nodes in $\tilde{\mathcal{K}}$ and the dual edges dual to primal faces belonging to $\mathcal{K} - \partial\mathcal{K}$. For each connected component of $\partial\mathcal{K}$, one random dual edge, dual to a face in the considered connected component of $\partial\mathcal{K}$, is added to the tree. Then, the coefficients of the faces dual to cotree dual edges are set to zero, whereas the remaining coefficients of \mathbf{I}_s are found iteratively as follows. When all faces of a given element v have been already set but exactly one, the coefficient of the remaining one can be found by imposing \mathbf{I}_s to be a 2-cocycle (i.e. $\mathbf{D}(v, \cdot)\mathbf{I}_s(\cdot) = 0$). This technique, implemented in CAFE code in [2], presents a couple of shortcomings. On one hand, the technique is inefficient, since all elements of \mathcal{K} have to be processed. On the other hand, we need to store in memory one real number for each face of \mathcal{K} .

In this paper we propose a different and efficient solution through the computation of a source electric vector potential \mathbf{T}_s with a graph-theoretic technique. The algorithm is straightforward to implement as it uses most ingredients already computed in the previous section. First, we set \mathbf{T}_s to zero outside $\partial\mathcal{K}$, that enables a reduction of the memory to store its coefficients. Therefore, our aim is to compute the coefficients of \mathbf{T}_s on $\partial\mathcal{K}$ edges. Second, we make a further assumption on the source current \mathbf{I}_s : the evaluation of \mathbf{I}_s vanishes over each connected component of $\partial\mathcal{K}$. This

That is, in order to formulate the problem by using the complementary-dual formulation, the following cochains with real coefficients are introduced:

- Voltage $\tilde{\mathbf{U}} \in C^1(\tilde{\mathcal{K}}, \mathbb{R})$;
- Electric current $\mathbf{I} \in C^2(\mathcal{K}, \mathbb{R})$;
- Electric scalar potential $\tilde{\mathbf{V}} \in C^1(\tilde{\mathcal{K}}, \mathbb{R})$.

The scalar potential $\tilde{\mathbf{V}}$ in the dual nodes is defined through

$$\tilde{\mathbf{U}} = -\tilde{\mathbf{G}}\tilde{\mathbf{V}} = -\mathbf{D}^T\tilde{\mathbf{V}}, \quad (8)$$

what renders the discrete Faraday law $\mathbf{C}^T\tilde{\mathbf{U}} = \mathbf{0}$ implicitly satisfied, since $\mathbf{C}^T\mathbf{D}^T = (\mathbf{DC})^T = \mathbf{0}$ holds [21]. The *resistance matrix* \mathbf{R} , already described in Section 3, relates currents \mathbf{I} to voltages $\tilde{\mathbf{U}}$

$$\tilde{\mathbf{U}} = \mathbf{R}\mathbf{I}. \quad (9)$$

Finally, the current continuity law

$$\mathbf{D}\mathbf{I} = \mathbf{0} \quad (10)$$

has to be enforced by the linear system of equation. By substituting (9) and (8) in (10), an algebraic system with one unknown per element is obtained

$$-\mathbf{DR}^{-1}\mathbf{D}^T\tilde{\mathbf{V}} = \mathbf{0}. \quad (11)$$

This approach, for general hexahedral meshes, is not viable since the \mathbf{R}^{-1} matrix is full.

In an alternative way, following the idea behind Mixed Finite Elements (MFE) formulations [19,20], we introduce the formulation \mathbf{M} writing a system with both \mathbf{I} and $\tilde{\mathbf{V}}$ as unknowns

$$\mathbf{R}\mathbf{I} + \mathbf{D}^T\tilde{\mathbf{V}} = \mathbf{0} \quad (12)$$

$$\mathbf{D}\mathbf{I} = \mathbf{0}. \quad (13)$$

This saddle point problem is solved by a penalty method [31], [19, p. 80], [32]

$$(\mathbf{R} + \gamma\mathbf{D}^T\mathbf{D})\mathbf{I} = \mathbf{0}, \quad (14)$$

where the real parameter $\gamma \gg 1$.

The final system is obtained by enforcing the current \mathbf{I} on faces in $\partial\mathcal{K}$ to \mathbf{I}_s , by standard boundary condition techniques. Eq. (14) sparse, symmetric and positive definite linear system with boundary conditions, that contains as unknowns the currents on faces in $\mathcal{K} \setminus \partial\mathcal{K}$, is solved by using a state-of-the-art direct solver (PARDISO, included in the Intel MKL library).

Another solution (formulation \mathbf{H}) inspired by Mixed-Hybrid Finite Elements (MHFE) formulations [19,33,20], involves a domain decomposition with as many sub-domains as mesh elements (hexahedra in this paper) in such a way that the number of faces is doubled. Lagrange multipliers λ are used to enforce the equality between each pair of currents. Let us denote the doubled cochains as underlined bold characters, i.e. $\underline{\mathbf{I}}$ denotes the currents over the doubled faces.

The mixed system (12) then becomes [19,33]

$$\underline{\mathbf{R}}\mathbf{I} + \underline{\mathbf{D}}^T\tilde{\mathbf{V}} + \mathbf{N}^T\lambda = \mathbf{0} \quad (15)$$

$$\underline{\mathbf{D}}\mathbf{I} = \mathbf{0} \quad (16)$$

$$\underline{\mathbf{N}}\mathbf{I} = \mathbf{I}_s, \quad (17)$$

where, as usual, \mathbf{I}_s is the source current and \mathbf{N} collects the constraints to enforce the equality of currents on each face pair. This matrix has a number of faces as rows and a total number of doubled faces as columns. Its entries for the row f are the incidence numbers $\mathbf{D}(v_1, f)$ and $\mathbf{D}(v_2, f)$, where v_1 and v_2 are the two hexahedra that contain f . These two integers are placed in the appropriate columns corresponding to the labels of f in the doubled list of faces. Of course, for faces f lying in the boundary there is only one of such hexahedron, and the matrix \mathbf{N} is in this case used to enforce the inhomogeneous Neumann boundary condition. The matrix $\underline{\mathbf{R}}$ is block diagonal, each block being the local resistance mass matrix of one hexahedral element. Thus, by inverting $\underline{\mathbf{R}}$, it is possible to easily eliminate the currents unknowns as

$$\underline{\mathbf{D}}\mathbf{R}^{-1}\underline{\mathbf{D}}^T\tilde{\mathbf{V}} + \underline{\mathbf{D}}\mathbf{R}^{-1}\mathbf{N}^T\lambda = \mathbf{0} \quad (18)$$

$$\underline{\mathbf{N}}\mathbf{R}^{-1}\underline{\mathbf{D}}^T\tilde{\mathbf{V}} + \underline{\mathbf{N}}\mathbf{R}^{-1}\mathbf{N}^T\lambda = -\mathbf{I}_s. \quad (19)$$

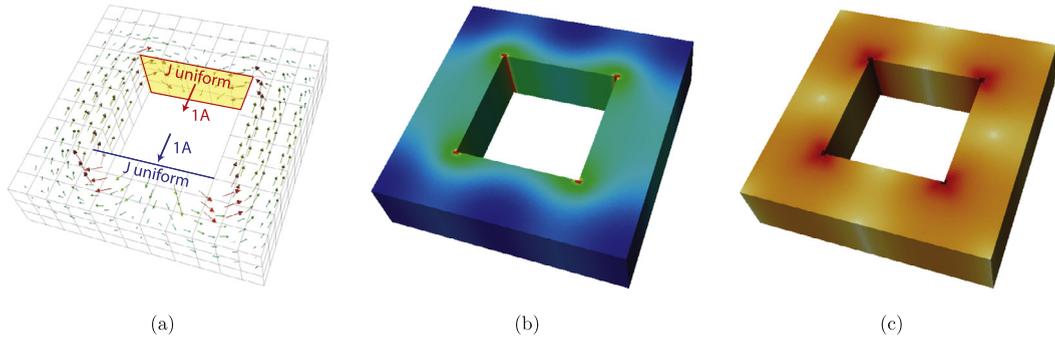


Fig. 8. Geometrically simple but topologically non-trivial 3D benchmark configuration. (a) Current density field lines. Neumann boundary conditions (BCs) are imposed on $\partial\mathcal{K}$. (b) Module of the current density. (c) Distribution of the constitutive error (log scale): the presence of higher values in a specific region gives an idea of the discretization error, providing a criterium to guide a possible adaptive mesh refinement.

Matrix $\mathbf{Q} = \mathbf{DR}^{-1}\mathbf{D}^T$ is diagonal, the value $\mathbf{Q}(v, v)$ being easily computed locally in each element v by $\mathbf{D}_v(\mathbf{R}_v)^{-1}\mathbf{D}_v^T$, where \mathbf{R}_v and \mathbf{D}_v are the local resistance mass matrix and the local volume-face incidence matrix, respectively. Since \mathbf{Q} can be easily inverted, we can also eliminate the potential unknowns obtaining

$$\mathbf{N}[\mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{D}^T\mathbf{Q}^{-1}\mathbf{DR}^{-1}]\mathbf{N}^T\lambda = -\mathbf{I}_s. \quad (20)$$

Finally, to obtain a symmetric and positive definite system, one Lagrange multiplier is eliminated by imposing its coefficient to zero by usual boundary condition techniques, i.e. by not considering it in the assembling process.

The assembling for (20) can be easily performed locally for each mesh element. In fact, for the element v , one first finds the $\mathbf{Q}(v, v) = \mathbf{D}_v(\mathbf{R}_v)^{-1}\mathbf{D}_v^T$ value, then form the local (6×6 for a hexahedron) matrix $\mathbf{J} = \mathbf{R}_v^{-1} - \mathbf{R}_v^{-1}\mathbf{D}_v^T\mathbf{Q}^{-1}\mathbf{D}_v\mathbf{R}_v^{-1}$. The elements of the $\mathbf{J}(i, j)$ matrix have to be assembled in the row corresponding to face i and in the column corresponding to face j by multiplying them by $\mathbf{D}(v, i)\mathbf{D}(v, j)$ to take into account the left and right multiplication by \mathbf{N} and \mathbf{N}^T , respectively. As the Reader can see, both the matrix \mathbf{N} and the labeling for the doubled faces are not explicitly constructed.

7. Numerical results

As a preliminary test, the configuration sketched in Fig. 8 is considered, which is geometrically simple but topologically non-trivial ($\beta_1(K) = 1$); uniform non-homogeneous Neumann boundary conditions (BCs) are imposed on two opposite faces of $\partial\mathcal{K}$ ($I = 1A$), as shown in Fig. 8(a); homogeneous Neumann BCs are imposed on the other faces of $\partial\mathcal{K}$.

In Fig. 9 a study of convergence is presented, in terms of overall power loss in \mathcal{K} : it is clear that the mean of the two solutions (the one provided by formulation \mathbf{V} , the other by formulation \mathbf{T} or, equivalently, by \mathbf{M} or \mathbf{H}) is much more accurate than the solutions provided by each formulation alone, and is relatively close to the reference value even for the coarsest mesh.⁵

Then, a model of a representative fusion device is considered: the discretization of half of the machine is shown in Fig. 10. Note that for this particular geometry (*ITER-like*) it would be possible to take into account just 1/9 of the numerical domain, despite the non-symmetric source term. Nonetheless, since most fusion reactors present no symmetries at all, we consider here a complete model of the machine as a challenging test of the proposed procedure. So far, the entire computational domain has been discretized into a full hexahedral mesh composed by more than 1 million elements, including the main conductive structures surrounding the plasma (vessel, port extensions, blanket modules and divertor). Then, from a uniform refinement of the initial mesh (each element is split into 8 hexahedra, by introducing new nodes corresponding to the barycentres of volumes, faces and edges), the second full hexahedral mesh composed by more than 8 million elements has been produced.⁶ The geometric elements of both initial and refined meshes are summarized in Table 1, together with the corresponding degrees of freedom (DOFs) and non-zero entries of the system matrix for formulation \mathbf{V} , \mathbf{T} (with and w/o gauge), \mathbf{M} and \mathbf{H} .

The boundary conditions (BCs) of the stationary current conduction problem (the normal component of the plasma current to the wall) are provided as a result of a three-dimensional resistive MHD code M3D used to run Tokamak VDE and disruption simulations [34]. M3D code represents the magnetic field in potential form to ensure that the magnetic field is div-free. The simulations rely on three features of the code: a time dependent resistivity (nearly constant along magnetic field lines; outside the last closed flux surface, resistivity is set 10–100 times bigger than in the

⁵ The reference value $P = 2.075$ W is calculated from post-processing of an accurate 2D solution (the numerical domain is discretized in more than 100k second order elements) by means of Comsol®.

⁶ Adaptivity on multiblock structured meshes is not straightforward and usually produces mesh elements of poor quality. Developing a tool for automatic adaptive refinement on multiblock hexahedral meshes is an ongoing work.

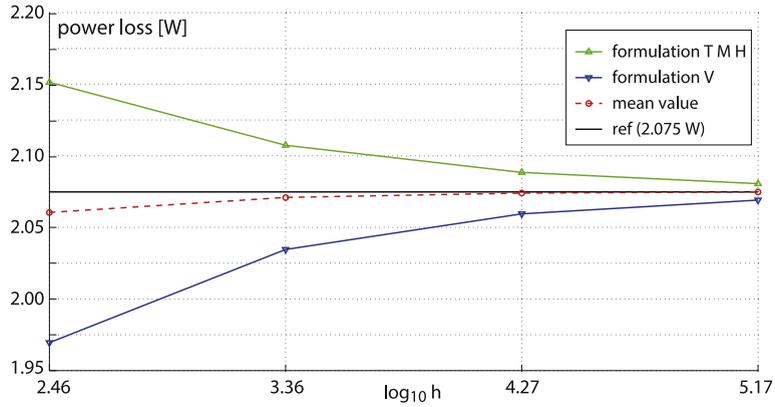


Fig. 9. Study of convergence, in terms of overall power loss in \mathcal{K} as a function of the number of mesh elements (log-scale). The mean of the two solutions (the one provided by formulation **V**, the other by formulation **T** or, equivalently, by **M** or **H**) is much more accurate than the solutions provided by each formulation alone, and is relatively close to the right value even for the coarsest mesh.

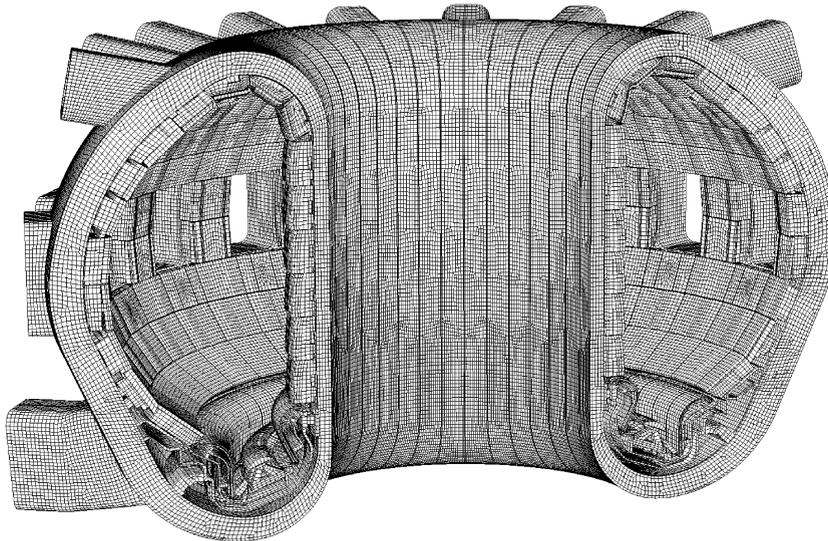


Fig. 10. A relatively complex fusion device: the geometry of half of the machine (an *ITER*-like model including vessel, port extensions, blanket modules and divertor) is discretized in a full hexahedral mesh.

Table 1

Geometric elements (nodes, hexahedra, edges, faces) of initial and refined meshes. Corresponding degrees of freedom (DOFs) and non-zero entries of the system matrix for formulation **V**, **T** (with and w/o gauge), **M** and **H**.

	Initial mesh	Refined mesh
Nodes	1,879,240	11,803,088
Hexahedra	1,088,055	8,704,440
Edges	4,814,281	32,242,940
Faces	4,021,512	29,142,708
DOFs (V)	1,879,239	11,803,087
DOFs (T with gauge)	1,421,947	14,381,081
DOFs (T ungauged)	1,788,065	20,128,560
DOFs (M)	2,506,818	23,083,932
DOFs (H)	4,021,511	29,142,707
Non-zeros (V)	19,088,749	137,149,188
Non-zeros (T with gauge)	12,050,568	139,213,216
Non-zeros (T ungauged)	20,035,111	273,186,474
Non-zeros (M)	11,918,736	123,893,364
Non-zeros (H)	20,342,331	159,709,302

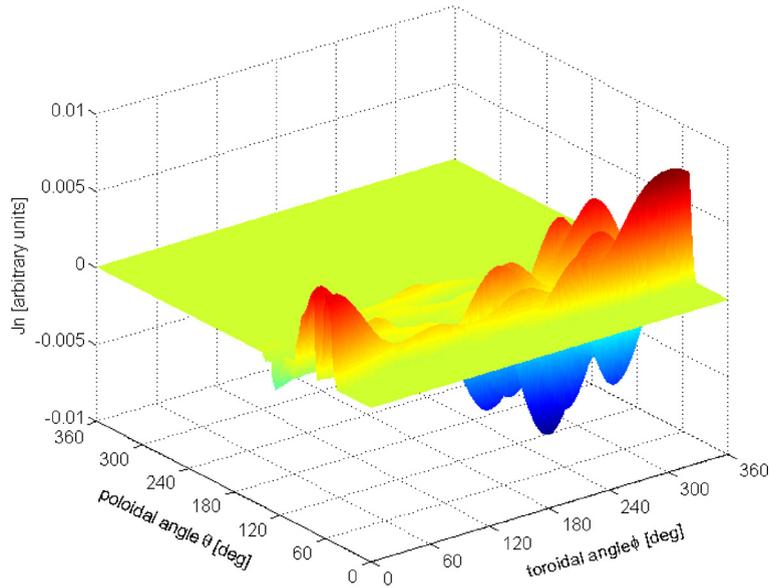


Fig. 11. Typical pattern of the normal component of the current density flowing from plasma to wall and viceversa (post-processing of M3D output).

Table 2

Resume of computational time for different formulations (\mathbf{V} , \mathbf{T} (with and w/o gauge), \mathbf{M} (penalty $\gamma = 10^{10}$), \mathbf{H}) on initial mesh. The computational cost of thick link generation is presented both for Algorithm 1 (adaptation of the *Dlotko-Specogna* (DS) algorithm) and Algorithm 5 (optimization of the cocycle retrieval).

	\mathbf{V}	\mathbf{T}_{gauge}	\mathbf{T}	\mathbf{M}	\mathbf{H}
Thick link generation with Algorithm 5 [s]	–	6.00	6.00	–	–
Thick link generation with Algorithm 1 [s]	–	52.85	52.85	–	–
Source fields \mathbf{T}_s [s]	–	5.95	5.95	–	–
Solution [s]	25.55	16.60	26.00	19.95	32.10

plasma core), an unstructured mesh of the plasma region and resistive wall boundary conditions. The solution inside the resistive wall is matched to the exterior vacuum solution; the exterior problem is solved with Green's function method.

In Fig. 11 a typical pattern of the normal component of the plasma current to the wall is shown. It is calculated by means of M3D at a given time instant during an asymmetric upward VDE, and accounts for two terms, the toroidal derivative of the toroidal plasma current (by definition asymmetric and therefore present only in a 3D plasma model) and the poloidal contribution (related to both symmetric and non-axi-symmetric components of the toroidal magnetic field).

The surface $\partial\mathcal{K}$ is quite complicated, both geometrically and topologically (genus g is 1586). A resume of the computational time spent to generate thick links and to compute source fields \mathbf{T}_s is presented in Tables 2 (initial mesh) and 3 (refined mesh), together with the time spent to solve the linear system for all the formulations with a state-of-the-art direct solver (Intel MKL Pardiso©). Also advanced iterative methods are suitable for the solution of some of the linear systems described in Sections 4 and 6 (e.g. aggregation-based algebraic multigrid solver (AGMG) [35,36] for \mathbf{V} , \mathbf{H} and, surprisingly, also \mathbf{M} ,⁷ or Conjugate Gradient (CG) methods with *ad hoc* preconditioners for \mathbf{V} and ungauged \mathbf{T}) but none of them can overcome the performance of the direct solver.

It must be pointed out that 3172 lazy cohomology generators are computed by Algorithm 5 in 6 s for the initial mesh; a similar amount of time is spent to calculate \mathbf{T}_s , while the solution of the linear system with the formulation \mathbf{T} (with gauge, 1,421,947 DoFs) takes 16.6 s. As far as refined mesh is concerned, lazy cohomology generators and \mathbf{T}_s are calculated in 78.45 s and 28.15 s, respectively. Here, the solution of the linear system takes 276 s. As one can see, thanks to the efficiency of their implementation, the percentage of time spent for computation of cohomology generators and \mathbf{T}_s is of the order of 40% for initial mesh and 30% for refined mesh with respect to the overall time, not including pre-processing and assembling (mesh loading and refinement, cell complex incidence calculation, stiffness matrix assembling); if pre-processing and assembling is taken into account, these percentages become even smaller. Moreover, for a given mesh lazy cohomology generators are computed only once (in the pre-processing phase) and used for any simulation with different BCs. Further-

⁷ It is well known that it is almost impossible for an iterative method to solve the linear system obtained by a penalty method [19, p. 81].

Table 3

Resume of computational time for different formulations (**V**, **T** (with and w/o gauge), **M** (penalty $\gamma = 10^{10}$), **H**) on refined mesh. The computational cost of thick link generation is presented both for **Algorithm 1** (adaptation of the *Dlotko–Specogna* (DS) algorithm) and **Algorithm 5** (optimization of the cocycle retrieval).

	V	T_{gauge}	T	M	H
Thick link generation with Algorithm 5 [s]	–	78.45	78.45	–	–
Thick link generation with Algorithm 1 [s]	–	280.40	280.40	–	–
Source fields T_s [s]	–	28.15	28.15	–	–
Solution [s]	293.50	276.00	510.00	280.50	332.00

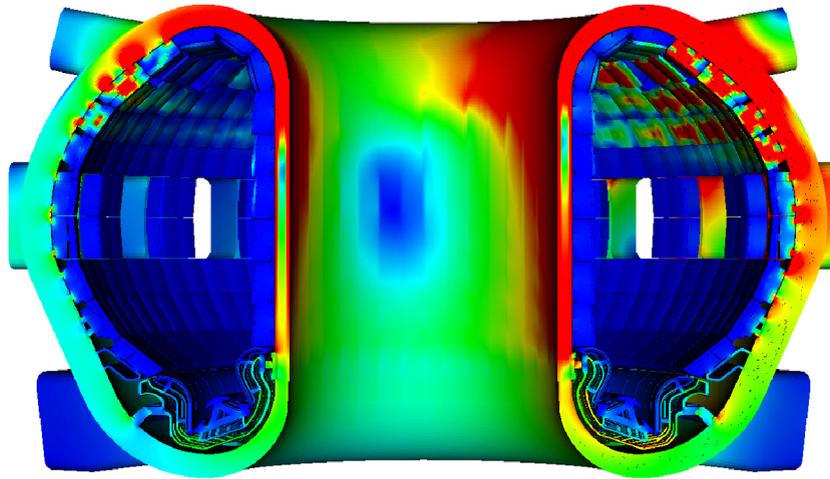


Fig. 12. Module of the current density for a representative case of upward asymmetric VDE. A saturated color map is adopted to highlight the region of stronger plasma-wall interaction.

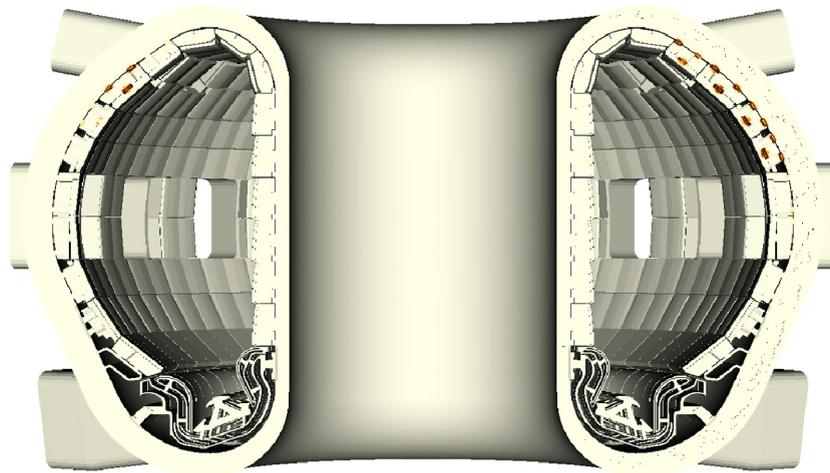


Fig. 13. Distribution of the *constitutive error*. The presence of higher values in a specific region gives an idea of the discretization error, providing a criterium to guide a possible adaptive mesh refinement.

more, since the solution of the linear system is relatively faster for formulation **T**, with respect to **V** (and also w.r.t. **M** and **H**), the overall time for **T** exceeds the one for **V** of no more than 30% for both meshes, providing the conditions for practical use of complementarity on very large scale problems.

In **Fig. 12**, a typical interaction between plasma and FW is shown for an asymmetric upward VDE. The known terms of the problem (currents exchanged between plasma and FW Neumann BCs) are calculated from interpolation of the normal component of the plasma current density at the wall, provided by M3D code (see **Fig. 11**). **Fig. 13** shows the entity of the

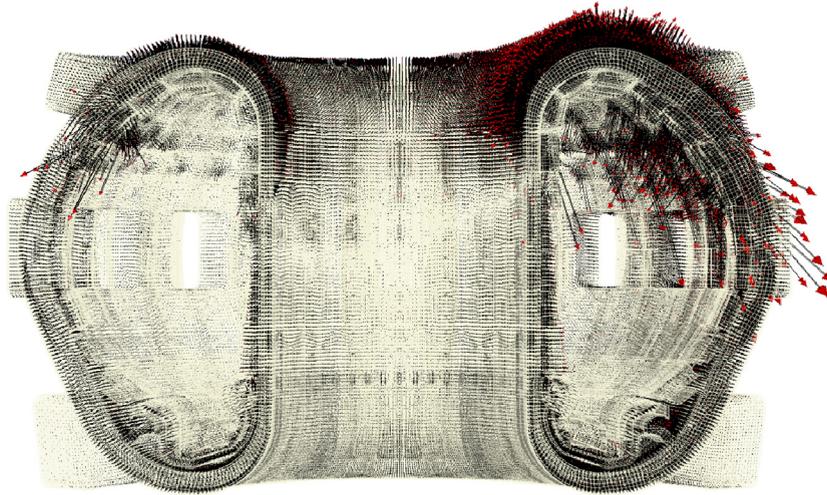


Fig. 14. Detail of force distribution ($\mathbf{f} = \mathbf{J} \times \mathbf{B}$, arbitrary units) calculated for each element of the mesh. A known distribution of magnetic flux density is assumed, which consist only on its toroidal component produced by the so called toroidal field (TF) coils.

Table 4

Cartesian components of the overall electromagnetic force acting on the machine, calculated by complementary formulations (\mathbf{V} , \mathbf{TMH}) on both initial and refined meshes. For each component a relative discrepancy is introduced: e.g. for x component, we get: $\Delta\% = \left| \frac{F_x^V - F_x^T}{F_x^T} \right| \cdot 100$.

	Initial mesh			Refined mesh		
	\mathbf{V}	\mathbf{TMH}	$\Delta\%$	\mathbf{V}	\mathbf{TMH}	$\Delta\%$
F_x [arb.unit]	132,773	131,306	1.1172	132,466	131,978	0.3698
F_y [arb.unit]	-430,813	-425,164	1.3287	-429,533	-427,627	0.4457
F_z [arb.unit]	2,442,859	2,442,910	0.0021	2,442,897	2,442,910	0.0005

constitutive error ε^8 [7]. The detection of high values in a specific region could be profitably used to guide an automatic or manual local refinement of the mesh.

Then, complementarity is used to provide some estimate of electromagnetic force produced by the interaction of currents injected in the conductive structures of the machine with the magnetic field needed for plasma confinement. A known distribution of magnetic flux density is assumed, which consist only on its toroidal component produced by the so called toroidal field (TF) coils. Fig. 14 illustrates the distribution of forces ($\mathbf{f} = \mathbf{J} \times \mathbf{B}$), in arbitrary units, calculated for each element of the mesh. Table 4 summarizes the results for overall electromagnetic force acting on the machine, calculated with complementary formulations (\mathbf{V} , \mathbf{TMH}) on both meshes. For each component a relative discrepancy is also introduced: e.g. for x component, we get: $\Delta\% = \left| \frac{F_x^V - F_x^T}{F_x^T} \right| \cdot 100$. An excellent agreement is achieved, with a maximum relative error of the order of 1%, even for the initial (relatively coarse) mesh.

8. Conclusions

While the electric scalar potential formulation \mathbf{V} is widely used to produce an irrotational electric field for a Poisson-like Neumann boundary value problems, we compared three different formulations to obtain a solenoidal current density. Lazy cohomology generators and fast algorithms to compute them automatically render the simulation time of the vector potential formulation \mathbf{T} comparable with the other two formulations inspired from mixed FE (\mathbf{M}) and mixed-hybrid FE (\mathbf{H}). The formulation \mathbf{M} seems to be the least attractive since it requires a suitable and problem dependent choice of the penalty factor: Too small values produce a current that is not solenoidal, whereas too big values render the system indefinite providing wrong results. As far as \mathbf{T} and \mathbf{H} is concerned, they are almost equivalent (also in terms of computational costs) and the best solution is probably to use both of them to provide a robust cross-check of the results.

Complementarity is used to provide some estimate of electromagnetic force produced by the interaction of halo currents injected in the conductive structures of a representative fusion device. An excellent agreement is achieved, with a maximum relative error of the order of 1%, even for the coarsest mesh.

⁸ $\varepsilon = \int_V \frac{1}{\rho} |\mathbf{E} - \rho \mathbf{j}|^2 dv$ where the electric field \mathbf{E} is interpolated inside each hexahedron v of \mathcal{K} with the \mathbf{V} formulation, whereas the $\rho \mathbf{j}$ is obtained from \mathbf{TMH} .

Acknowledgements

The authors kindly acknowledge R. Paccagnella and H.R. Strauss for providing the output data of Tokamak VDE and disruption simulations carried out by means of M3D code and M. Furno Palumbo for his significant activities on modeling and mesh generation of the large fusion device considered in this investigation.

This work was partially supported by the European Communities under the Contract of Association between EURATOM and ENEA/Consorzio RFX, and by the Italian MIUR under PRIN grants 2010SPS9B3 and 2009LTRYRE. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Appendix A

Algorithm 4 FindPrimalTree.

Input: G_b
Output: *primalTree*;
 initialize vector *primalTree* to *false* for each edge of $\partial\mathcal{K}$;
 initialize vector *cc* to *false* for each node of $\partial\mathcal{K}$;
while true **do**
 node $R \leftarrow 0$;
 for each node R in $\partial\mathcal{K}$ **do**
 if $cc(R) = \text{false}$ **then**
 break;
 if $R = 0$ **then**
 break;
 $cc(R) \leftarrow \text{true}$; initialize queue Q ;
 for each edge E in $\partial\mathcal{K}$ incidental to R found with $G_b(\cdot, R)$ **do**
 enqueue(Q, E);
 while queue Q is not empty **do**
 edge $E \leftarrow \text{dequeue}(Q)$;
 find the nodes N_1 and N_2 boundary of E ;
 if $cc(N_1) = \text{false}$ or $cc(N_2) = \text{false}$ **then**
 if $cc(N_1) = \text{false}$ **then**
 $N \leftarrow N_1$;
 else
 $N \leftarrow N_2$;
 $primalTree(E) \leftarrow \text{true}$;
 $cc(N) \leftarrow \text{true}$;
 for each edge E in $\partial\mathcal{K}$ incidental to N found with $G_b(\cdot, N)$ **do**
 enqueue(Q, E);
return *primalTree*;

Algorithm 5 FindDualTree.

Input: C_b , *primalTree*
Output: *dualTree*;
 initialize vector *dualTree* to *false* for each edge of $\partial\mathcal{K}$;
 initialize vector *cc* to *false* for each face of $\partial\mathcal{K}$;
while true **do**
 face $F \leftarrow 0$;
 for each face F in $\partial\mathcal{K}$ **do**
 if $cc(F) = \text{false}$ **then**
 break;
 if $F = 0$ **then**
 break;
 $cc(F) \leftarrow \text{true}$; initialize queue Q ;
 for each dual edge $D_b(E)$ in $\partial\mathcal{K}$ incidental to dual node $D_b(F)$ found with C_b and such that $primalTree(E) = \text{false}$ **do**
 enqueue(Q, E);
 while queue Q is not empty **do**
 edge $E \leftarrow \text{dequeue}(Q)$;
 find faces F_1 and F_2 such that $D_b(F_1)$ and $D_b(F_2)$ are the boundary of $D_b(E)$;
 if $cc(F_1) = \text{false}$ or $cc(F_2) = \text{false}$ **then**
 if $cc(F_1) = \text{false}$ **then**
 $F \leftarrow F_1$;
 else
 $F \leftarrow F_2$;
 $primalTree(E) \leftarrow \text{true}$;
 $cc(F) \leftarrow \text{true}$;
 for each dual edge $D_b(e)$ in $\partial\mathcal{K}$ incidental to dual node $D_b(F)$ found with C_b and such that $primalTree(e) = \text{false}$ **do**
 enqueue(Q, e);
return *dualTree*;

Algorithm 6 CocycleRetrieval.

Input: $C_b, dualTree, E$
Output: Λ^i ;

initialize vector $dist$ to ∞ for all faces in $\partial\mathcal{K}$;
 find faces F_1, F_2 in $\partial\mathcal{K}$ such that $E \in \partial F_1$ and $E \in \partial F_2$; (Since $\partial\mathcal{K}$ is a combinatorial 2-manifold without boundary, the number of such faces is always exactly two.)
 $dist(F_1) \leftarrow 0$; initialize queue Q ; $enqueue(Q, F_1)$;
while queue Q is not empty **do**
 face $F \leftarrow dequeue(Q)$;
 if $F = F_2$ **then**
 break;
 for all faces $G \in \partial\mathcal{K}$ that share an edge e with F and $dualTree(e) = true$ **do**
 if $dist(G) = \infty$ **then**
 $dist(G) \leftarrow dist(F) + 1$;
 $parent(G) \leftarrow F$;
 $enqueue(Q, G)$;
 $\Lambda^i(E) \leftarrow 1$;
while true **do**
 $G \leftarrow parent(F)$;
 $e \leftarrow$ common edge between triangles F and G ;
 $circulation \leftarrow C_b(F, \cdot) \cdot \Lambda^i(\cdot)$;
 $\Lambda^i(e) \leftarrow -circulation C_b(F, e)$;
 if $dist(G) = 0$ **then**
 break;
 $F \leftarrow G$;
return Λ^i ;

Algorithm 7 FindBFSDistance.

Input: $C_b, dualTree$
Output: $dist, parent$

initialize vector $dist$ to ∞ for all faces in $\partial\mathcal{K}$;
while true **do**
 $T \leftarrow 0$;
 for each face T in $\partial\mathcal{K}$ **do**
 if $dist(T) = \infty$ **then**
 break;
 if $T = 0$ **then**
 break;
 $dist(T) \leftarrow 0$; initialize queue Q ; $enqueue(Q, T)$;
 while queue Q is not empty **do**
 face $F \leftarrow dequeue(Q)$;
 for all faces $G \in \partial\mathcal{K}$ that share an edge E with F and $dualTree(E) = true$ **do**
 if $dist(G) = \infty$ **then**
 $dist(G) \leftarrow dist(F) + 1$;
 $parent(G) \leftarrow F$;
 $enqueue(Q, G)$;
return $dist, parent$;

Algorithm 8 FastCocycleRetrieval.

Input: $E, dist, parent, C_b$
Output: Λ^i

find faces F_1, F_2 in $\partial\mathcal{K}$ such that $E \in \partial F_1$ and $E \in \partial F_2$;
 $n_1 \leftarrow F_1$; $n_2 \leftarrow F_2$; $\Lambda^i(E) \leftarrow 1$;
while true **do**
 if $dist(n_1) < dist(n_2)$ **then**
 $big \leftarrow n_2$; $small \leftarrow n_1$;
 else
 $big \leftarrow n_1$; $small \leftarrow n_2$;
 $circulation \leftarrow C_b(big, \cdot) \cdot \Lambda^i(\cdot)$;
 find edge e shared by faces big and $parent(big)$;
 $\Lambda^i(e) \leftarrow -circulation C_b(big, e)$;
 if $parent(big) = small$ **then**
 break;
 $n_1 \leftarrow small$; $n_2 \leftarrow parent(big)$;
return Λ^i ;

References

- [1] W. Park, E.V. Belova, G.Y. Fu, X.Z. Tang, H.R. Strauss, L.E. Sugiyama, Plasma simulation studies using multilevel physics models, *Phys. Plasmas* 6 (1999) 1796.
- [2] R. Albanese, P. Bettini, N. Marconato, M. Furno Palumbo, S. Peruzzo, G. Rubinacci, R. Specogna, S. Ventre, F. Villone, Numerical modeling of 3d halo current path in iter structures, *Fusion Eng. Des.* 88 (2013) 529–532.
- [3] R. Albanese, G. Rubinacci, F. Villone, An integral computational model for crack simulation and detection via eddy currents, *J. Comput. Phys.* 152 (2) (1999) 736–755.

- [4] P. Bettini, R. Specogna, Lazy cohomology generators enable the use of complementarity for computing halo current resistive distribution in fusion reactors, *IEEE Trans. Magn.* 50 (2014) 7012004.
- [5] G. Strang, G.J. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall, Englewood Cliffs, NJ, 1973.
- [6] J.L. Synge, *The Hypercircle in Mathematical Physics*, Cambridge University Press, New York, 1957.
- [7] J.A.H. Rikabi, C.F. Bryant, E.M. Freeman, An error-based approach to complementary formulations of static field solutions, *Int. J. Numer. Methods Eng.* 26 (9) (1988) 1963–1987.
- [8] J.A.H. Rikabi, Complementary solutions of direct current flow problems, *IEEE Trans. Magn.* 29 (1) (1993) 98–107.
- [9] A. Bossavit, *Computational Electromagnetism*, Academic Press, 1998.
- [10] R. Specogna, Complementary geometric formulations for electrostatics, *Int. J. Numer. Methods Eng.* 86 (2011) 1041–1068.
- [11] R. Specogna, Extraction of VLSI multiconductor transmission line parameters by complementarity, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* 22 (1) (2014) 146–154, <http://dx.doi.org/10.1109/TVLSI.2012.2232320>.
- [12] P.W. Gross, P.R. Kotiuga, *Electromagnetic Theory and Computation: A Topological Approach*, Cambridge University Press, 2004.
- [13] J.R. Munkres, *Elements of Algebraic Topology*, Perseus Books, Cambridge, MA, 1984.
- [14] M. Pellikka, S. Suuriniemi, L. Kettunen, Powerful heuristics and basis selection bring computational homology to engineers, *IEEE Trans. Magn.* 47 (2011) 1226–1229.
- [15] P. Dlotko, R. Specogna, F. Trevisan, Automatic generation of cuts on large-sized meshes for t - ω geometric eddy-current formulation, *Comput. Methods Appl. Mech. Eng.* 198 (2009) 3765–3781.
- [16] P. Dlotko, R. Specogna, Efficient cohomology computation for electromagnetic modeling, *Comput. Model. Eng. Sci.* 60 (2010) 247–278.
- [17] P. Dlotko, R. Specogna, Physics inspired algorithms for (co)homology computations of three-dimensional combinatorial manifolds with boundary, *Comput. Phys. Commun.* 184 (2013) 2257–2266, [arXiv:1212.1360](https://arxiv.org/abs/1212.1360).
- [18] P. Dlotko, R. Specogna, Lazy cohomology generators: a breakthrough in (co)homology computations for CEM, *IEEE Trans. Magn.* 50 (2014) 7014204.
- [19] F. Brezzi, M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer Verlag, Heidelberg, Germany, 1991.
- [20] D. Boffi, F. Brezzi, M. Fortin, *Mixed Finite Element Methods and Applications*, Springer Ser. Comput. Math., vol. 44, 2013, Heidelberg, Germany.
- [21] E. Tonti, On the formal structure of physical theories, *Monogr. Ital. Natl. Res. Council.* (1975).
- [22] L. Codecasa, R. Specogna, F. Trevisan, A new set of basis functions for the discrete geometric approach, *J. Comput. Phys.* 229 (2010) 7401–7410.
- [23] L. Codecasa, R. Specogna, F. Trevisan, Base functions and discrete constitutive relations for staggered polyhedral grids, *Comput. Methods Appl. Mech. Eng.* 198 (2009) 1117–1123.
- [24] L. Kettunen, K. Forsman, A. Bossavit, Formulation of the eddy current problem in multiply connected regions in terms of h , *Int. J. Numer. Methods Eng.* 41 (5) (1998) 935–954.
- [25] T. Tarhasaari, L. Kettunen, Alain Bossavit, Some realizations of a discrete hodge operator: a reinterpretation of finite element techniques [for em field analysis], *IEEE Trans. Magn.* 35 (3) (1999) 1494–1497.
- [26] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to Algorithms*, 2nd edition, The MIT Press, 2001.
- [27] R. Hiptmair, J. Ostrowski, Generators of $h_1(\gamma_h, \mathbb{Z})$ for triangulated surfaces: construction and classification, *SIAM J. Comput.* 31 (5) (2002) 1405–1423.
- [28] A. Alonso Rodriguez, E. Bertolazzi, R. Ghiloni, A. Valli, Construction of a finite element basis of the first de Rham cohomology group and numerical solution of 3d magnetostatic problems, *SIAM J. Numer. Anal.* 51 (4) (2013) 2380–2402.
- [29] O. Biro, S. Pack, K. Richter, On the use of the magnetic vector potential in the nodal and edge finite element analysis of 3d magnetostatic problems, *IEEE Trans. Magn.* 32 (3) (1996) 651–654.
- [30] P. Dlotko, R. Specogna, Critical analysis of the spanning tree techniques, *SIAM J. Numer. Anal.* 48 (2010) 1601–1624.
- [31] R. Courant, Variational methods for the solution of problems of equilibrium and vibration, *Bull. Am. Math. Soc.* 69 (1943) 1–23.
- [32] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, *Acta Numer.* 14 (5) (2005) 1–137.
- [33] A. Bossavit, Mixed-hybrid methods in magnetostatics: complementarity in one stroke, *IEEE Trans. Magn.* 39 (3) (2003) 1099–1102.
- [34] R. Paccagnella, H.R. Strauss, J. Breslau, 3D MHD VDE and disruptions simulations of tokamaks plasmas including some ITER scenarios, *Nucl. Fusion* 49 (2009) 035003.
- [35] Y. Notay, AGMG software and documentation, see <http://homepages.ulb.ac.be/~ynotay/agmg>, 2012.
- [36] Y. Notay, Aggregation-based algebraic multigrid for convection–diffusion equations, *SIAM J. Sci. Comput.* 34 (2012) 2288–2316.