Computer Physics Communications 184 (2013) 2257–2266

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Physics inspired algorithms for (co)homology computations of three-dimensional combinatorial manifolds with boundary

Paweł Dłotko [a,b], Ruben Specogna [c,*]

[a] Department of Mathematics, University of Pennsylvania, 209 South 33rd Street, Philadelphia, PA 19104-6395, USA
[b] Jagiellonian University, Institute of Computer Science, Lojasiewicza 4, 30348 Kraków, Poland
[c] Università di Udine, Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica, Via delle Scienze 206, 33100 Udine, Italy

ARTICLE INFO

Article history:
Received 5 December 2012
Received in revised form
3 May 2013
Accepted 8 May 2013
Available online 18 May 2013

Keywords:
Physics inspired algorithms
Algebraic topology
(co)homology
First De Rham cohomology group generators
Discrete Hodge decomposition
Computational physics
Magneto-quasistatics
Eddy-currents

ABSTRACT

The issue of computing (co)homology generators of a cell complex is gaining a pivotal role in various branches of science. While this issue may be rigorously solved in polynomial time, it is still overly demanding for large scale problems. Drawing inspiration from low-frequency electrodynamics, this paper presents a physics inspired algorithm for first cohomology group computations on three-dimensional complexes. The algorithm is general and exhibits orders of magnitude speed up with respect to competing ones, allowing to handle problems not addressable before. In particular, when generators are employed in the physical modeling of magneto-quasistatic problems, this algorithm solves one of the most long-lasting problems in low-frequency computational electromagnetics. In this case, the effectiveness of the algorithm and its ease of implementation may be even improved by introducing the novel concept of *lazy cohomology generators*.

1. Introduction

The availability of unprecedented computing power and efficient numerical methods produced a dramatic increase in applications of computational (co)homology [1–4] (i.e. the computation of generators of the (co)homology group). (Co)homology, in fact, has been already shown to be essential in unexpected areas of science, ranging from computer aided design (CAD) for feature detection [5], parametrization and mesh generation [6], shape analysis and pattern recognition [7], to sensors networks [8] and robot motion planning [9], medicine and biology [10] and quantum chemistry [11]. Focusing on physics, (co)homology generators have been used for example to detect the chaotic behavior in sampled experimental data [12–14], in quantum information theory [15] and electromagnetism [16–23].

The problem addressed in this paper is the computation of the first cohomology group generators for three-dimensional combinatorial manifolds with boundary, being the computation of the zeroth and second absolute (co)homology groups already satisfactorily solved in the literature, see for example [22].

Integer cohomology generators – unlike the real and complex ones – may be rigorously computed in polynomial time by finding the *Smith normal form* (SNF) [4] of the coboundary matrix. However, this approach is computationally not attractive because the best implementation of the SNF exhibits a hyper-cubical complexity [24]. To briefly survey the techniques already introduced in the literature for 1st cohomology group computations, it is convenient to partition them into three classes. First, *combinatorial methods* [21,25,26] use sparse matrix data structures and various *reductions* of the input cell complex $\mathcal{M}$ to speed up the SNF computation to a point that may be used for practical problems. Second, *homology-based methods* [27,28,17,20] start by computing a homology generator that is "dual" to the cohomology generator that one aims to compute (this duality will be explained in Section 3). The idea behind this technique has been first introduced by Kotiuga [27,28,17] for the computation of *cuts* (i.e. generators of the 2nd cohomology group $H_2(\mathcal{M}, \partial\mathcal{M})$ starting from a 1st homology basis $H_1(\tilde{\mathcal{M}})$ on the dual complex $\tilde{\mathcal{M}}$. In [20], a dual version of the Kotiuga's algorithm – called *Generalized Spanning Tree Technique* (GSTT) – has been presented to compute the 1st cohomology group basis $H^1(\mathcal{M}) \cong H_2(\tilde{\mathcal{M}}, \partial\tilde{\mathcal{M}})$ starting from a 1st homology

* Corresponding author. Tel.: +39 0432 558037; fax: +39 0432 558251.
E-mail addresses: dlotko@sas.upenn.edu (P. Dłotko), ruben.specogna@uniud.it (R. Specogna).

0010-4655/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cpc.2013.05.006

group basis $H_1(\mathcal{M})$. Recently, a third class of *physics inspired methods* to compute cohomology generators has been introduced. As opposed to any other method, this class of algorithms constructs the cohomology generators starting from a set of electric currents. The *Thinned Current Technique* [29] (TCT) – the first physics inspired algorithm – presents some attractive features as the orders of magnitude reduction of the computational time and the simple implementation. Nonetheless, its main limitation is that when the complex $\mathcal{M}$ is not skeletonizable – i.e. it cannot be homotopically retracted to a graph – this algorithm cannot be used and another technique, as [21], has to be employed instead. For a comprehensive survey on other algorithms proposed in the literature (most of them patently incorrect) please refer to [23].

To introduce a novel physics inspired algorithm for the first cohomology group computations, let us focus on the interplay between (co)homology, discrete Hodge decomposition [30,16,18,17,31] and physical modeling that appears when considering problems whose definition of potentials is not straightforward. One of the most studied examples where this happens occurs in low-frequency electrodynamics. Electromagnetic phenomena are governed by Maxwell's laws [32] and material constitutive relations. For slowly time-varying fields, whose change in magnetic field energy is dominant and electromagnetic wave propagation can be ignored, it is typical to brake the symmetry of Maxwell's laws by neglecting the displacement current in the Ampère–Maxwell's equation [32]. Using this magneto-quasistatic (MQS) approximation, the magnetic field is irrotational in the insulators thanks to Ampère's law. The fact that the insulating region is in most cases not simply connected prevents it to be exact. The consequence is that a magnetic scalar potential cannot be introduced naïvely in the insulating regions. Yet, using a scalar potential is tempting since formulations based on it are computationally much more efficient than the ones using the classical magnetic vector potential.

How to define a magnetic scalar potential in non simply connected domains has drawn a considerable effort in the computational electromagnetics community in the last twenty-five years. A connection of this issue with (co)homology theory has been advocated many years ago by Kotiuga [16], that solved this problem by introducing the *cuts*[1] together with an algorithm to compute them [27,28,17]. Despite most scientists keep using other heuristic and sometimes patently incorrect definition of potentials and related algorithms (see references and counter-examples in [23]), it is getting accepted that the first cohomology group generators over integers of the cell complex modeling the insulating region are needed to make the problem well defined [20,23]. Introductory material on this subject comprising an informal introduction to algebraic topology, how to model physical variables as cochains with complex coefficients and how to rephrase Maxwell's laws in algebraic form, can be found in [20,23,29].

The quest for an algorithm for first cohomology group computation that is both general and exhibits a linear average complexity is still open. This is surprising since the research on this issue has been pushed forward by many leading software houses having at least part of the core business in solving MQS problems. The fact that this issue has been considered unsolved for so many years indicates that computing cohomology generators quickly (and correctly) is not straightforward. Also the implementation complexity may affect negatively the technology transfer. Developing a simple and fast algorithm would enable to embed it in the next-generation of electromagnetic Computer-Aided Engineering (CAE) softwares.

This article fills this gap by exploiting a novel physics inspired approach to compute cohomology generators suitable for physical modeling called the *Dłotko–Specogna* (DS) algorithm. Moreover, the novel concept of *lazy cohomology generators* is introduced to speed up and simplify the implementation when cohomology generators are employed, for example, in computational physics.

The paper is structured as follows. Section 2 is a mild introduction to (co)homology theory that may be skipped by readers familiar with this topic. In Section 3 the role of (co)homology theory in low-frequency electrodynamics is recalled. The *Dłotko–Specogna* (DS) algorithm and the novel concept of *lazy cohomology generators* are introduced in Section 4. In Section 5 some numerical experiments are presented to compare the novel algorithm to other state-of-the-art algorithms in terms of efficiency and robustness. Finally, in Section 6, the conclusions are drawn.

2. Mild introduction to algebraic topology

In this section, a mild introduction to algebraic topology is provided. For a rigorous one, please consult [33]. Let us consider a discretization $\mathcal{K}$ of a given three-dimensional space as a cell complex (more precisely, a regular CW-complex [33]). For simplicity, one may think about a simplicial complex. $\mathcal{K}$ is a *combinatorial manifold* if the link of every vertex is a sphere or a disk. The link of a vertex $v \in \mathcal{K}$ consists of all elements $s \in \mathcal{K}$ such that $s \cap v = \emptyset$ and there exist $t \in \mathcal{K}$ such that $v, s \in t$.

Homology and its dual cohomology theory are mathematical tools to describe "holes" of a given space in a rigorous way. As an example, let us consider the two-dimensional simplicial complex $\mathcal{K}$ of the annulus represented in the Fig. 1. The zero-dimensional holes are defined as the connected components of $\mathcal{K}$. In the example, $\mathcal{K}$ is formed by a single connected component. Clearly, there is one (one-dimensional) hole in the annulus. This hole may be surrounded with a one-dimensional oriented cycle, as in Fig. 1 upper left. This kind of cycle represents a generator of the first homology group of the annulus.

The concept of holes may be generalized to higher dimensions. In the example, there are no holes of dimensions two and higher. Yet, two-dimensional holes for three-dimensional complexes are voids as the ones that may be encircled by a ball contained in $\mathcal{K}$ (i.e. cavities of the complex). In this case, the collection of oriented two-dimensional cells on the surface of the ball represents a second homology generator of the complex. The number of i-dimensional holes is often referred to as the ith Betti number $\beta_i(\mathcal{K})$.

In homology theory some cycles are considered equivalent. If two i-dimensional cycles (with appropriate orientation) form the boundary of a set made of $(i+1)$-dimensional elements, then we say that they are in the same homology (equivalence) class, or simply that they are equal. Two cycles in the same homology class are represented in Fig. 1 lower left. If some cycle alone is a boundary, then we say that it is homologically trivial.

More formally, a i-dimensional *chain* (i-chain) c of a complex $\mathcal{K}$ is a formal sum of elements of $\mathcal{K}$ of dimension i with coefficients (in this paper we consider integer or complex coefficients). The group of i-chains of a complex $\mathcal{K}$ are denoted as $C_i(\mathcal{K})$. Every chain is uniquely determined by its coefficients on all i-dimensional elements of $\mathcal{K}$. The chain c is a *cycle* if its boundary ∂c vanishes. c is *homologically nontrivial* if it is not a boundary of a higher dimensional chain. The group of i-dimensional cycles is denoted by $Z_i(\mathcal{K})$. The group of i-dimensional boundaries is denoted by $B_i(\mathcal{K})$. Since every boundary is a cycle, one may define the ith *homology group* as

[1] The use of cuts (in place of the 1st cohomology group generators) is considered nowadays obsolete, since they are required to be embedded submanifolds (i.e. they have to be surfaces that do not self-intersect). Kotiuga proposed a very nice technique to construct them by means of the solution of non-physical Poisson problems. Each cut is then constructed as a level set of the solution of these problems. As the reader can guess, solving a Poisson problem for each generator is extremely time consuming and – at least at the present state – can be hardly justified in practice.

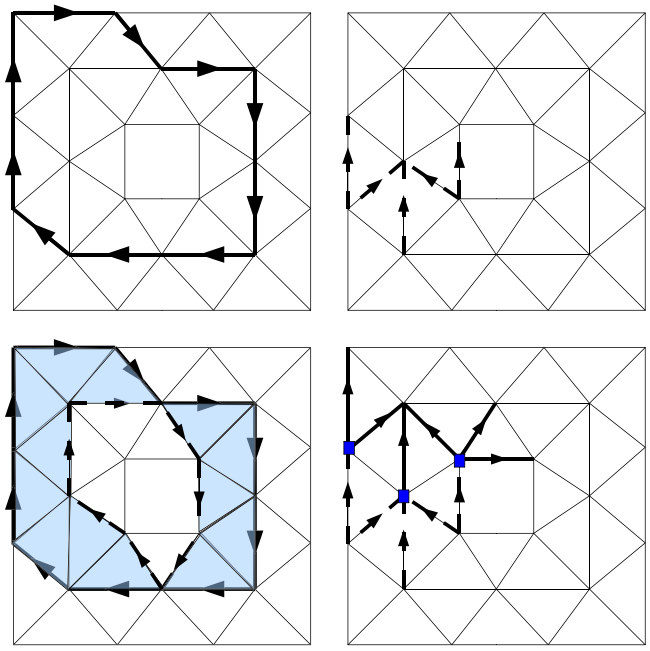

Fig. 1. On the left (upper and lower), representatives of homology generators of the annulus. On the right (upper and lower), representatives of cohomology generators of the annulus.

the quotient $H_i(\mathcal{K}) = Z_i(\mathcal{K})/B_i(\mathcal{K})$. By $H_i(\mathcal{K}, \mathbb{Z})$ and $H_i(\mathcal{K}, \mathbb{C})$ we denote the integer and complex homology groups, respectively.

Homology has a dual theory called cohomology. The elements dual to the i-dimensional chains (i-chains) are called i-dimensional *cochains* (i-cochains) and they are functions from the i-chains to integers/complex numbers. The value of the i-cochain $\mathbf{c}^*$ on a given i-chain c is determined by the value of $\mathbf{c}^*$ on the i-dimensional cells (i-cells) of $\mathcal{K}$. Therefore, in this paper, we compute some i-cochains by providing their values on all i-cells of $\mathcal{K}$. The *cocycles* and *coboundaries* may be also introduced analogously as in the case of homology.

In the case of complexes embedded in a tree-dimensional space, a straightforward duality between homology and cohomology generators exists. Let us fix the $c_1, \ldots, c_n$ representatives of a homology basis. For a cocycle $\mathbf{c}_i^*$ we may define the discrete analogous of integration of $\mathbf{c}_i^*$ on a cycle c by using a dot product $\langle \mathbf{c}_i^*, c\rangle$ between the vectors representing the coefficients of $\mathbf{c}_i^*$ and c. We say that $\mathbf{c}_i^*$ is a cohomology generator dual to c_i if $\langle \mathbf{c}_i^*, c_j\rangle = \delta_{ij}$ (for more details consult [34]). A possible cohomology generator in the considered example may be visualized in Fig. 1 upper right.

As in the case of homology, in cohomology two cocycles $\mathbf{c}^*$ and $\mathbf{c}^{*'}$ are considered equal if there exists a $(i-1)$-cochain $\mathbf{s}$ such that $\mathbf{c}^*$ and $\mathbf{c}^{'}$ are common *co*boundary of elements of $\mathbf{s}$, see Fig. 1 bottom right for an example.

More formally, a *cochain* with integer/complex coefficients is a map from the group of chains to integers/complex numbers. A cochain $\mathbf{c}^*$ is a *cocycle* if its coboundary $\delta\mathbf{c}^*$ vanishes. δ is the coboundary operator [33] defined with the Generalized Stokes Theorem $\langle \delta\mathbf{c}^*, c\rangle = \langle \mathbf{c}^*, \partial c\rangle$. $\mathbf{c}^*$ is *cohomologically nontrivial* if it is not a coboundary. The group of i-dimensional cocycles of the complex $\mathcal{K}$ is denoted by $Z^i(\mathcal{K})$, whereas the group of i-dimensional coboundaries is denoted as $B^i(\mathcal{K})$. The ith *cohomology group* is the quotient $H^i(\mathcal{K}) = Z^i(\mathcal{K})/B^i(\mathcal{K})$. By $H^i(\mathcal{K}, \mathbb{Z})$ and $H^i(\mathcal{K}, \mathbb{C})$ we denote integer and complex cohomology groups, respectively. By a *(co)homology basis* we mean a set of linearly independent (co)homology generators that span the (co)homology group.

One may also consider the so-called *relative* (co)homology group of a complex $\mathcal{K}$ modulo a sub-complex $\mathcal{K}_0 \subset \mathcal{K}$ in which $\mathcal{K}_0$ is forgotten. Formally, to define a relative (co)homology, one needs to take the quotient group $C_*(\mathcal{K})/C_*(\mathcal{K}_0)$ in the case of homology or $C^*(\mathcal{K})/C^*(\mathcal{K}_0)$ in the case of cohomology and compute its (co)homology ($*$ represents any integer number). The relative (co)homology groups are denoted by $H_i(\mathcal{K}, \mathcal{K}_0)$ and $H^i(\mathcal{K}, \mathcal{K}_0)$. The details may be found in [33] or, more informally, in [34,22].

The *dual complex* $\tilde{\mathcal{K}}$ [4] and [20, Section 3] is obtained from $\mathcal{K}$ by using the *barycentric subdivision*. Let us define the dual cell complex $\tilde{\mathcal{K}} = D(\mathcal{K})$ in the following way:

(1) For every polyhedron $t \in \mathcal{K}$, the dual node $\tilde{n} = D(t)$ is defined as the barycenter of t.
(2) For every 2-cell $f \in \mathcal{K}$ that is a common face of polyhedra $t_1, t_2 \in \mathcal{K}$, the dual edge $\tilde{e} = D(f)$ is defined as the sum of a segment of line joining the barycenter of f with $D(t_1)$ and a segment of line joining the barycenter of f with $D(t_2)$.
(3) For every edge $e \in \mathcal{K}$ let $f_1, \ldots, f_n \in \mathcal{K}$ be the 2-cells incidental to e. The dual face $\tilde{f} = D(e)$ is then defined as a (set-theoretical) sum $\bigcup_{i=1}^{n} \mathrm{conv}[B(e), D(f_i)]$, where $B(e)$ denotes the barycenter of the edge e and $\mathrm{conv}[\cdot]$ the convex hull.
(4) For every node $n \in \mathcal{K}$ let $e_1, \ldots, e_n \in \mathcal{K}$ be the edges incidental to n. The dual volume $\tilde{t} = D(n)$ is the volume bounded by $D(e_1), \ldots, D(e_n)$.

3. (Co)homology and low-frequency electrodynamics

Let us cover the topologically trivial computational domain by a conformal polyhedral mesh $\mathcal{K}$ which is a regular CW-complex [33]. As it is going to be explained in Section 4.3, the requirement of dealing with a homologically trivial complex does not limit the generality of the technique described in this paper. Two sub-complexes $\mathcal{K}_c$ and $\mathcal{K}_a$ of $\mathcal{K}$ are introduced that contain mesh elements belonging to the conducting and insulating regions, respectively. Let us define the potentials in $\mathcal{K}_a$ for the harmonic analysis of a MQS problem formulated by using a magnetic scalar potential, as the $\mathbf{T} - \boldsymbol{\Omega}$ formulation [20,23,29]. The algebraic Ampère's law is enforced on every 2-cell with $\delta\mathbf{F} = \mathbf{I}$, where $\mathbf{I} \in C^2(\mathcal{K}, \mathbb{C})$ is the complex-valued electric current 2-cochain and $\mathbf{F} \in C^1(\mathcal{K}, \mathbb{C})$ is the magneto-motive force (m.m.f.) complex-valued 1-cochain. Let us denote by $\mathbf{F}_a$ the restriction of $\mathbf{F}$ to $\mathcal{K}_a$. Since the current is zero for all 2-cells belonging to $\mathcal{K}_a$, $\mathbf{F}_a$ is a 1-cocycle in $\mathcal{K}_a$ ($\mathbf{F}_a \in Z^1(\mathcal{K}_a, \mathbb{C})$). From the definition of the cohomology group basis, the 1-cocycle $\mathbf{F}_a$ may be expressed as a linear combination of a basis of the 1st cohomology group $H^1(\mathcal{K}_a, \mathbb{C})$ plus a 1-coboundary $B^1(\mathcal{K}_a, \mathbb{C})$. The 1-coboundary $B^1(\mathcal{K}_a, \mathbb{C})$ is provided by taking the 0-coboundary of a complex-valued 0-cochain $\boldsymbol{\Omega}_a$ whose coefficients represent the magnetic scalar potential sampled on mesh nodes. Since $\mathcal{K}_a$ is embedded in $\mathbb{R}^3$, the (co)homology groups are torsion free [4] and the basis of $H^1(\mathcal{K}_a, \mathbb{C})$ may be obtained from a basis of $H^1(\mathcal{K}_a, \mathbb{Z})$ where the elements of $\mathbb{Z}$ are treated as elements of $\mathbb{C}$ [33]. Then, the *nonlocal* (i.e. applied not locally on each 2-cell as $\delta\mathbf{F} = \mathbf{I}$, but on an arbitrary 2-chain) algebraic Ampère's law [20,23] $\langle \mathbf{F}_a, c_j\rangle = \langle \mathbf{I}, s_j\rangle$ (see Fig. 2) implicitly holds for any 1-cycle $c_j \in Z_1(\mathcal{K}_a, \mathbb{Z})$, with $c_j = \partial s_j$, by setting

$$\mathbf{F}_a = \delta\boldsymbol{\Omega}_a + \sum_{j=1}^{\beta_1(\mathcal{K}_a)} i_j\, \mathbf{h}^j, \tag{1}$$

where $\langle\cdot,\cdot\rangle$ denotes the dot product, $\{\mathbf{h}^j\}_{j=1}^{\beta_1(\mathcal{K}_a)}$ are the representatives of the 1st cohomology group $H^1(\mathcal{K}_a, \mathbb{Z})$ generators and $\beta_1(\mathcal{K}_a)$ is the 1st Betti number of $\mathcal{K}_a$. Fig. 3a shows an example of cohomology generator for the torus. Instead of representing the edges belonging to the support of $\mathbf{h}^1$, the figure shows the support of the dual 2-chain $D(\mathbf{h}^1)$ on the complex dual to $\mathcal{K}_a$. When it is not confusing, by cohomology generators we refer to both the cohomology classes and their representatives. Physically, the $\{i_j\}_{j=1}^{\beta_1(\mathcal{K}_a)}$

may be interpreted as a set of *independent currents* [20,23] flowing in the branches of the conductors $\mathcal{K}_c$. Fig. 2 shows the independent current flowing in a solid 2-dimensional torus. For a n-fold 2-dimensional solid torus, there are n independent currents.

Let us fix the generators $\{\mathbf{h}^j\}_{j=1}^{\beta_1(\mathcal{K}_a)}$ of $H^1(\mathcal{K}_a, \mathbb{Z})$. There exists a set of cycles $\{c_i\}_{i=1}^{\beta_1(\mathcal{K}_a)}$ being a $H_1(\mathcal{K}_a, \mathbb{Z})$ basis such that $\langle \mathbf{h}^j, c_i \rangle = \delta_{ij}$ holds [33,34]. Since $\mathcal{K}$ is homologically trivial, there exist 2-chains $\{s_i\}_{i=1}^{\beta_1(\mathcal{K}_a)} \in C_2(\mathcal{K}, \mathbb{Z})$ whose boundaries are the $\{c_i\}_{i=1}^{\beta_1(\mathcal{K}_a)}$. Their restrictions $\{\sigma_i\}_{i=1}^{\beta_1(\mathcal{K}_a)}$ to $\mathcal{K}_c$ form a $H_2(\mathcal{K}_c, \mathcal{K}_c \cap \mathcal{K}_a, \mathbb{Z})$ basis [29], see Fig. 2, on the right. The independent currents are exactly the currents linked in nonlocal algebraic Ampère's law [20] by the 1-cycles $\{c_i\}_{i=1}^{\beta_1(\mathcal{K}_a)}$ in such a way that the j-th independent current may be defined with $\langle \mathbf{F}, c_j \rangle = i_j = \langle \mathbf{I}, s_j \rangle = \langle \mathbf{I}, \sigma_j \rangle$. Then, the currents linked by any other cycle in $Z_1(\mathcal{K}_a, \mathbb{Z})$ may be obtained by linear combinations of the independent currents. In the end, the induced currents may be obtained as described in [23] by solving a linear system obtained by enforcing the discrete magnetic Gauss's law, the Faraday's law and discrete counterparts of the constitutive laws together with (1).

We note that, the trace $\mathbf{b}^i$ of the cohomology generator $\mathbf{h}^i$ on $\mathcal{K}_a \cap \mathcal{K}_c$ is a cohomology generator of $H^1(\mathcal{K}_a \cap \mathcal{K}_c, \mathbb{Z})$, see Fig. 3b. By using the Generalized Stokes Theorem, $\langle \mathbf{h}^i, c_j \rangle = \delta_{ij} = \langle \delta \mathbf{h}^i, s_j \rangle$. Since $\mathbf{h}^i$ is a 1-cocycle in $\mathcal{K}_a$, we get

$$\langle \delta \mathbf{h}^i, \sigma_j \rangle = \langle \mathbf{t}^i, \sigma_j \rangle = \delta_{ij}, \tag{2}$$

where the 2-cocycles $\mathbf{t}^i$ are defined as $\mathbf{t}^i = \delta \mathbf{h}^i$. Given that $\mathbf{h}^i$ is zero in the interior of $\mathcal{K}_c$, the support of the 2-cocycle $\mathbf{t}^i$ is a set of 2-cells in the interior of $\mathcal{K}_c$ that have at least one edge in $\mathbf{b}^i$, see Fig. 3c. The current 2-cocycle $\mathbf{I}$ may be expressed in all $\mathcal{K}$ by summing up the contributions $\mathbf{F}_c = \mathbf{T} + \delta \boldsymbol{\Omega}_c$ of the restriction of $\mathbf{F}$ to $\mathcal{K}_c - \mathcal{K}_a$ and (1) as $\mathbf{I} = \delta \mathbf{F} = \delta(\mathbf{T} + \delta \boldsymbol{\Omega} + \sum_{j=1}^{\beta_1(\mathcal{K}_a)} i_j \mathbf{h}^j)$, where $\mathbf{T}$ is the electric vector potential. Considering that $\mathbf{t}^j = \delta \mathbf{h}^j$, we get

$$\mathbf{I} = \delta \mathbf{T} + \sum_{j=1}^{\beta_1(\mathcal{K}_a)} i_j \mathbf{t}^j. \tag{3}$$

We may therefore imagine the independent currents i_j as currents that flow in the support of the 2-cocycles $\mathbf{t}^j$.

The dual of $\mathbf{t}^i$ is a 1-chain $D(\mathbf{t}^i)$ on the complex $D(\mathcal{K}_c)$, see Fig. 4a. Also the dual of $\mathbf{b}^i$ is a 1-chain $D(\mathbf{b}^i)$ on the complex $D(\mathcal{K}_c)$, see Fig. 4b. Both $\{D(\mathbf{t}^i)\}_{i=1}^{\beta_1(\mathcal{K}_a)}$ and $\{D(\mathbf{b}^i)\}_{i=1}^{\beta_1(\mathcal{K}_a)}$ are representatives of the same $H_1(D(\mathcal{K}_c), D(\partial \mathcal{K}))$ basis.

4. Dłotko–Specogna (DS) algorithm and lazy cohomology generators

Usually, cohomology generators are found first [27,28,17,20,21,23] and the current distribution inside $\mathcal{K}_c$ is a result of the MQS simulation. In principle, taking inspiration from physics, one may do it the other way around. Thanks to (2), the j-th 1-cocycle $\mathbf{h}^j$ may be computed by imposing a unity current in σ_j and a zero current in the other sigmas, where the $\{\sigma_i\}_{i=1}^{\beta_1(\mathcal{K}_a)}$ are $H_2(\mathcal{K}_c, \mathcal{K}_c \cap \mathcal{K}_a, \mathbb{Z})$ generators. So, let us suppose that a set of current 2-cocycles $\{\mathbf{t}^1, \ldots, \mathbf{t}^{\beta_1(\mathcal{K}_a)}\}$ is constructed such that $\langle \mathbf{t}^i, \sigma_j \rangle = \delta_{ij}$ for every representative σ_j of the $H_2(\mathcal{K}_c, \mathcal{K}_c \cap \mathcal{K}_a, \mathbb{Z})$ generators. This is clearly equivalent to fix a basis of $H^1(\mathcal{K}_a)$, since it fixes the evaluation of the cohomology generators on the dual homology $H_1(\mathcal{K}_a)$ basis provided by the 1-cycles $\{\partial \sigma_i\}_{i=1}^{\beta_1(\mathcal{K}_a)}$. In fact, thanks to (2), $\langle \mathbf{h}^i, c_j \rangle = \delta_{ij} = \langle \mathbf{h}^i, \partial \sigma_j \rangle = \langle \mathbf{t}^i, \sigma_j \rangle$. Therefore, finding the cohomology basis is just a matter of solving $\beta_1(\mathcal{K}_a)$ linear systems $\delta \mathbf{h}^i = \mathbf{t}^i$ in the complex $\mathcal{K}$ and restricting the output cocycle to $\mathcal{K}_a$ (it is clear, in fact, that the restriction of a cocycle is a cocycle). Surprisingly, these systems may be solved in most cases by back-substitutions only with the *Extended Spanning Tree Technique* (ESTT) [35,29], without using any integer linear solver and even a sparse matrix data structure.

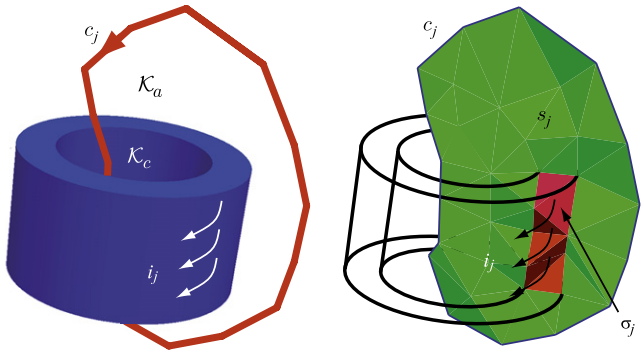

Fig. 2. On the left, the independent current flowing in a torus. On the right, the 1-cycle $c_j \in Z_1(\mathcal{K}_a)$ and a 2-chain s_j such that $c_j = \partial s_j$.

The core of this physics inspired approach to cohomology computation is how to obtain such currents $\{\mathbf{t}^1, \ldots, \mathbf{t}^{\beta_1(\mathcal{K}_a)}\}$. The key observation is that the cohomology classes of the 1-cocycles $\{\mathbf{h}^i\}_{i=1}^{\beta_1(\mathcal{K}_a)}$ do not depend on how the unity currents are distributed inside $\mathcal{K}_c$. Moreover, if each unity current flows in a ring whose sections are "single 2-cells", then the 2-cocycle condition together with the constraint $\langle \mathbf{t}^i, \sigma_j \rangle = \delta_{ij}$ for every representative σ_j of $H_2(\mathcal{K}_c, \mathcal{K}_c \cap \mathcal{K}_a, \mathbb{Z})$ generators may be trivially imposed by assigning a unity current (with the appropriate sign that depends on incidence) to those 2-cells. The resulting 2-cocycles $\{\mathbf{t}^1, \ldots, \mathbf{t}^{\beta_1(\mathcal{K}_a)}\}$ are called *thinned currents* [29].

In TCT algorithm [29], a physically-based method to construct thinned currents is employed. A thinning is applied on $\mathcal{K}_c$ in such a way that the conductors become a "single 3-cell thick". The thinned conductors may be viewed on the dual complex as a graph representing the skeleton of $\mathcal{K}_c$. By computing independent cycles on that graph, by orienting them and by considering the 2-cells that are dual to the dual 1-cells in the graph, the thinned currents are found. This approach presents two problems: on one hand, this approach does not work for conductors that do not homotopically retract to a graph, on the other hand finding the skeleton is time consuming since all elements of $\mathcal{K}_c$ have to be processed.

The DS algorithm, introduced for the first time in the present paper, solves all of these problems, resulting in a general and extremely fast algorithm. The DS algorithm may be summarized as follows:

(1) Find the n combinatorial 2-manifolds that represent the connected components $\mathcal{C}_1, \ldots, \mathcal{C}_n$ of $\mathcal{K}_c \cap \mathcal{K}_a$. This part requires $O(card(\mathcal{K}))$ time.
(2) Compute the 1st cohomology $H^1(\mathcal{C}_i, \mathbb{Z})$ generators $\mathbf{c}^1, \ldots, \mathbf{c}^{2g}$ for each $\mathcal{C}_i$, where g is the genus of $\mathcal{C}_i$. This may be performed in linear time worst-case complexity $O(card(\partial \mathcal{K}_c)\, g)$ with the graph-theoretic algorithm presented in [36] (see Appendix A for a short presentation). An example of $H^1(\mathcal{C}_i)$ generators for a solid 2-torus is presented in Fig. 5a.
(3) For every connected component $\mathcal{C}_i$, find the *thinned currents* $\mathbf{t}^1, \ldots, \mathbf{t}^{2g}$ corresponding to $\mathbf{c}^1, \ldots, \mathbf{c}^{2g}$ in $O(card(\partial \mathcal{K}_c)\, g)$ time with the following algorithm. The value of the cochain $\mathbf{t}$ on a cell E is $\langle \mathbf{t}, E \rangle$, whereas $\kappa(A, B)$ denotes the incidence between cells A and B, see [33]. Initially, set $\langle \mathbf{t}^i, T \rangle = 0$ for all 2-cells $T \in \mathcal{K}_c$.

for each 1-cell E with nonzero coefficient c_E in $\mathbf{c}^i$
  for each 2-cell $T \in \mathcal{K}_c$ with E in the boundary
    $\langle \mathbf{t}^i, T \rangle += c_E \kappa(T, E)$;

For an example, see Fig. 5b–c. In Appendix B, it is proved that the $\mathbf{t}^i$ obtained in this way are cocycles for every i.

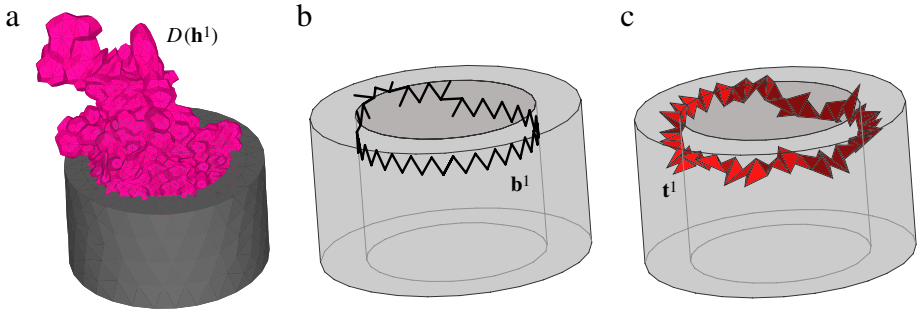

Fig. 3. (a) The 2-chain $D(\mathbf{h}^1)$ on the complex dual to $\mathcal{K}_a$ is the dual w.r.t. the cohomology generator $\mathbf{h}^1$. (b) $\mathbf{b}^1$ is the trace on $\mathcal{K}_c \cap \mathcal{K}_a$ of $\mathbf{h}^1$. (c) The 2-cocycle $\mathbf{t}^1 = \delta\mathbf{h}^1$.

(4) For every connected component $\mathcal{C}_i$, solve in $\mathcal{K}$ the integer systems $\delta\,\mathbf{h}^j = \mathbf{t}^j$, $j \in \{1, \ldots, 2g\}$. This may be performed without solving any system by the application of a *vectorialized* version of the ESTT algorithm [35,29]. Vectorialized means that the ESTT algorithm is applied to all $\mathbf{t}^1, \ldots, \mathbf{t}^{2g}$ thinned currents at the same time. Algorithmically this may be easily achieved by changing a real number to a vector of $2g$ real numbers in the ESTT algorithm, in such a way that the algorithm may be easily parallelized.
(5) For every connected component $\mathcal{C}_i$, store the restrictions of $\mathbf{h}^1, \ldots, \mathbf{h}^{2g}$ to $\mathcal{K}_a$. Computationally, the restrictions are performed by setting the coefficient of every edge in the interior of $\mathcal{K}_c$ to zero in every cochains $\mathbf{h}^1, \ldots, \mathbf{h}^{2g}$. The computational effort required is $O(card(\mathcal{K})\, g)$.

We would like to point out that in our implementation we process all connected components of the boundary in parallel. This detail is omitted in the pseudocode above for the sake of clarity in the presentation.

The ESTT algorithm terminates always since the $\mathbf{t}^i$ are 2-cocycles for every i, as shown in Appendix B. For a proof that the ESTT always terminates given a 2-cocycle as input please consult [35]. If the genus g is bounded by a constant, as it happens always in practical problems, the worst case complexity of the DS algorithm is cubical w.r.t the number of 1-cells of the complex. Yet, the average complexity of the DS algorithm has been linear $O(card(\mathcal{K}))$ in all tested problems, provided BFS or minimal diameter trees are used in the ESTT algorithm [35]. This is because, even if it is very hard to prove it, in practice there is no need to introduce any symbolic variable in the ESTT algorithm to guarantee its termination [35]. So the typical complexity is linear and the algorithm is also purely graph-theoretic, so straightforward to implement.

4.1. Lazy cohomology generators or $H^1(\mathcal{K}_a, \mathbb{Z})$ basis

The 1-cocycles obtained by the DS algorithm are not a basis of $H^1(\mathcal{K}_a, \mathbb{Z})$, since the number of obtained generators is twice the cardinality of its basis. Still, the cocycles obtained from the DS algorithm span $H^1(\mathcal{K}_a, \mathbb{Z})$ but some of them are linear combinations of the others, see Appendix C for the proof. To show this informally, let us consider the $\mathbf{c}^1, \ldots, \mathbf{c}^{2n}$ generators of the $H^1(\mathcal{K}_c \cap \mathcal{K}_a)$ basis obtained by the DS algorithm at point (1). Then, as demonstrated in [37], any of them may be expressed as $\mathbf{c}^i = \sum_{h=1}^n \alpha_h \mathbf{a}^h + \sum_{k=1}^n \beta_k \mathbf{b}^k$, where $\{D(\mathbf{b}^k)\}_{k=1}^n$ is a basis of $H_1(D(\mathcal{K}_a))$ whereas $\{D(\mathbf{a}^k)\}_{k=1}^n$ is a basis of $H_1(D(\mathcal{K}_c), D(\partial\mathcal{K}))$. If one produces the thinned currents $\{\mathbf{t}^i\}_{i=1}^{2n}$ from boundary generators $\{\mathbf{c}^i\}_{i=1}^{2n}$, one would get $\mathbf{t}^i = \sum_{h=1}^n \alpha_h \mathbf{t}_a^h$, where $\{\mathbf{t}_a^h\}_{h=1}^n$ are the thinned currents produced with $\{\mathbf{a}^h\}_{h=1}^n$ boundary generators. This is because $\{D(\mathbf{b}^k)\}_{k=1}^n$ are trivial in $H_1(D(\mathcal{K}_c), D(\mathcal{K}))$ and adding them does not change the homology class of $D(\mathbf{t}^i)$ and therefore also the cohomology class of $\mathbf{t}^i$. It is clear that by running the ESTT algorithm on the thinned currents obtained from $\mathbf{c}^i$, one obtains always 1-cocycles in the class $\mathbf{h}^i = \sum_{h=1}^n \alpha_h \mathbf{h}_a^h$, with $\delta\mathbf{h}_a^h = \mathbf{t}_a^h$. $\{\mathbf{h}_a^i\}_{i=1}^n$ are cohomology $H^1(\mathcal{K}_a)$ generators. Therefore, the DS algorithm automatically ignores the components of boundary cocycles that are trivial in $\mathcal{K}_c$ and the produced 1-cocycles span $H^1(\mathcal{K}_a, \mathbb{Z})$. What outlined a moment ago motivates the introduction of the novel concept of *lazy cohomology generators*.

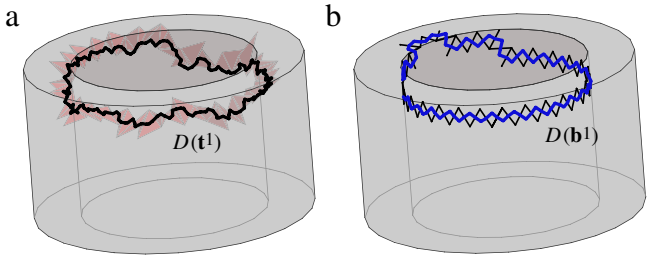

Fig. 4. (a) The 1-chain $D(\mathbf{t}^1)$ on the complex dual to $\mathcal{K}_c$, is the dual of $\mathbf{t}^1$. (b) The 1-chain $D(\mathbf{b}^1)$ on the complex dual to $\mathcal{K}_c \cap \mathcal{K}_a$, is the dual of $\mathbf{b}^1$. $D(\mathbf{t}^1)$ may be thought as obtained by submerging $D(\mathbf{b}^1)$ inside $D(\mathcal{K}_c)$.

Definition 1. A set of 1-cocycles that generate $H^1(\mathcal{K}_a, \mathbb{Z})$ are defined as *lazy cohomology generators*.

To obtain a $H^1(\mathcal{K}_a, \mathbb{Z})$ cohomology basis with the DS algorithm, not all boundary generators $H^1(\mathcal{C}_i, \mathbb{Z})$ have to be used. The required ones, i.e. the ones such that their duals generate $H_1(D(\mathcal{K}_c), D(\partial\mathcal{K}))$, may be found by a change of basis obtained by computing some linking numbers and the SNF of a small matrix, as already well documented in [37] (see Appendix D for details about the idea and its implementation). Nonetheless, due to conceptual and implementation difficulties, one welcomes the possibility to bypass this additional step. This is investigated in what follows.

4.2. Lazy cohomology generators in physical modeling

Apart the DS algorithm, the other fundamental contribution of this paper is to show that avoiding the change of basis of the $H^1(\mathcal{C}_i, \mathbb{Z})$ boundary generators is not only possible but even more convenient in most applications. This is realized by employing directly the lazy cohomology generators in the physical modeling, something that has been never considered in the literature. Lazy cohomology generators are employed in a MQS formulation, for example the $\mathbf{T} - \boldsymbol{\Omega}$ [19,20,23], as if they were a set of standard $H^1(\mathcal{K}_a, \mathbb{Z})$ generators (i.e. the fact that they are lazy generators and not standard ones is completely transparent to the user). Namely, a nonlocal Faraday's equation [20,23] is written on the support of the j-th lazy cohomology generator as $\langle \tilde{\mathbf{U}}, \partial\tilde{h}_j \rangle = -i\,\omega\,\langle \tilde{\boldsymbol{\Phi}}, \tilde{h}_j \rangle$, where $\tilde{\mathbf{U}}$ is the electro-motive force (e.m.f.) 1-cochain on the dual complex, $\tilde{\boldsymbol{\Phi}}$ is the magnetic flux 2-cochain on the dual complex and

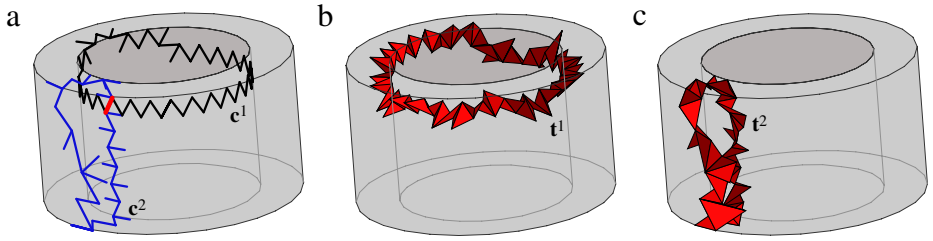

Fig. 5. (a) The two cohomology generators for a solid 2-torus intersecting in the thicker edge. (b–c) The support of the thinned currents $\mathbf{t}^1$ and $\mathbf{t}^2$ corresponding to $\mathbf{c}^1$ and $\mathbf{c}^2$, respectively.

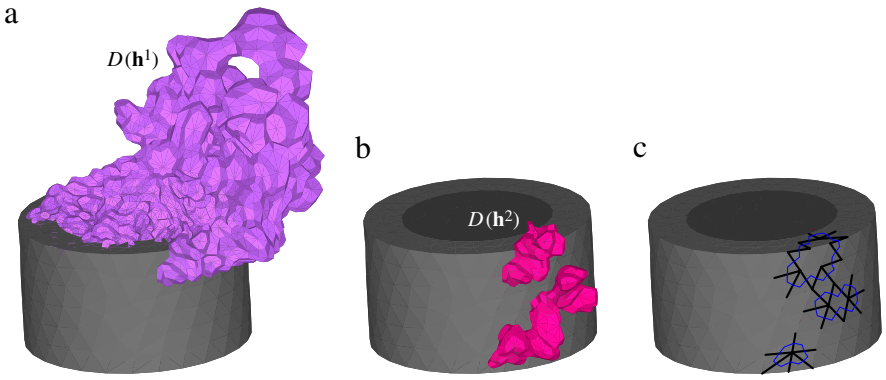

Fig. 6. The two 2-cycles on the dual complex $D(\mathbf{h}^1)$ and $D(\mathbf{h}^2)$ that are the dual to the lazy cohomology generators $\mathbf{h}^1$ and $\mathbf{h}^2$ obtained by the thinned currents $\mathbf{t}^1$ and $\mathbf{t}^2$ in Fig. 5b–c. (a) $\mathbf{h}^1$ is nontrivial in $H^1(\mathcal{K}_a, \mathbb{Z})$, whereas (b) $\mathbf{h}^2$ is trivial in $H^1(\mathcal{K}_a, \mathbb{Z})$. (c) The trace on $\mathcal{K}_c \cap \mathcal{K}_a$ of $\mathbf{h}^2$.

$\tilde{h}_j = D(\mathbf{h}^j)$, D being the *dual map* [4] that maps elements of the original complex to elements of the dual complex.

If one interpolates the lazy cohomology generators with Whitney forms [38,18], one gets a set of lazy generators for the 1st de Rham cohomology group [39,31]. This way, the lazy generators produced by the DS algorithm may be also used as nonlocal basis functions suitable for edge element formulations based on the magnetic scalar potential arising from Finite Elements, as the t-ω or h-ϕ formulations [40,18,41].

What remains to be explained is why one may safely use lazy generators in the physical modeling. Lazy cohomology generators span a $H^1(\mathcal{K}_a, \mathbb{Z})$ basis, but some of the lazy cohomology generators may be a linear combination of generators already considered or even cohomologically trivial. To understand why this should not be deemed as a problem, let us start by noticing that the system of equations to solve before adding nonlocal Faraday's equations is already overdetermined [20,23]. In fact, adding an arbitrary 1-coboundary $\delta\mathbf{W}$ to the electric vector potential $\mathbf{T}$ does not alter the current $\mathbf{I}$, since $\mathbf{I} = \delta\mathbf{T} = \delta(\mathbf{T} + \delta\mathbf{W})$, where $\mathbf{W}$ is an arbitrary 0-cochain.[2] Therefore, the (local) algebraic Faraday's laws [20,23] $\langle\tilde{\mathbf{U}}, \partial\tilde{f}\rangle = -i\,\omega\,\langle\tilde{\boldsymbol{\Phi}}, \tilde{f}\rangle$ enforced on every dual face $\tilde{f}$ in $\mathcal{K}_c$ are linearly dependent. Even though a full-rank system may be obtained by a *tree-cotree gauging* (i.e. by setting the electric vector potential on a tree of 1-cells in the interior of $\mathcal{K}_c$ to zero [42,43] and by eliminating the corresponding local Faraday's equations), it is widely known that with iterative linear solvers (as the ones that have to be used for huge problems) it is much more efficient to use *ungauged* [42,43] formulations. What is important from the modeling point of view is that even if the vector potential is not unique, its curl – i.e. the current density – is [42].

Coming back to the main issue, let us show why the lazy cohomology generators work with some examples. Consider $\mathcal{K}_c$ as a solid 2-torus, see Fig. 5a. Let us assume that $\mathbf{c}^1$ and $\mathbf{c}^2$ in Fig. 5a are the representatives of the $H^1(\mathcal{K}_c \cap \mathcal{K}_a, \mathbb{Z})$ generators. Then, by starting from the corresponding thinned currents $\mathbf{t}^1$ and $\mathbf{t}^2$ in Fig. 5b–c, a lazy generator $\mathbf{h}^1$ cohomologous to the standard $H^1(\mathcal{K}_a, \mathbb{Z})$ generator is obtained (see Fig. 6a) together with a lazy generator $\mathbf{h}^2$ that is trivial in $H^1(\mathcal{K}_a, \mathbb{Z})$ (see Fig. 6b). Since the solution in terms of induced currents does not depend on the representatives of the fixed cohomology basis, one may use any representative of the given basis. In this case, one gets the same solution if the cohomologically trivial lazy generator $\mathbf{h}^2$ is replaced by any trivial generator whose support does not touch $\mathcal{K}_a \cap \mathcal{K}_c$ and $\partial\mathcal{K}$. Then, since the trivial lazy generator may be expressed by a 1-coboundary $\delta\mathbf{W}$ in $\mathcal{K}$, $\mathbf{I} = \delta\mathbf{F} = \delta(\mathbf{T} + \delta\boldsymbol{\Omega} + i_1\mathbf{h}^1 + i_2\mathbf{h}^2) = \delta(\mathbf{T} + \delta\boldsymbol{\Omega} + i_1\mathbf{h}^1 + i_2\delta\mathbf{W}) = \delta(\mathbf{T} + i_1\mathbf{h}^1)$. Therefore, we can conclude that adding some generator trivial in $H^1(\mathcal{K}_a, \mathbb{Z})$ does not alter the solution in term of induced currents for the same reasons why ungauged formulations work. We note also that, given an arbitrary 1-cycle $c \in Z_1(\mathcal{K}_a)$, the dot product of the trivial lazy generator with c is zero. Therefore, trivial generators verify trivially the nonlocal algebraic Ampère's law and the current i_j does not represent the current linked by the dual homology generator. Instead, the value of the i_j current relative to a trivial generator is not unique and it is determined by the solution of the system of equations. This is not surprising, since the i_j current in this case does not have a physical meaning.

Let us now see the other possible case, namely when lazy generators are dependent but nontrivial in $H^1(\mathcal{K}_a, \mathbb{Z})$. To this aim, let us assume for example that $\hat{\mathbf{c}}^1 = \mathbf{c}^1$ and $\hat{\mathbf{c}}^2 = \mathbf{c}^1 + \mathbf{c}^2$ in Fig. 5a are the representatives of $H^1(\mathcal{K}_c \cap \mathcal{K}_a, \mathbb{Z})$ generators. Then, by starting from the corresponding thinned currents, a standard $H^1(\mathcal{K}_a, \mathbb{Z})$ generator $\hat{\mathbf{h}}^1$ is obtained together with another lazy generator $\hat{\mathbf{h}}^2$ in the same class. Then, $\mathbf{I} = \delta\mathbf{F} = \delta(\mathbf{T} + \delta\boldsymbol{\Omega} + \hat{i}_1\hat{\mathbf{h}}^1 + \hat{i}_2\hat{\mathbf{h}}^2) = \delta(\mathbf{T} + (\hat{i}_1 + \hat{i}_2)\mathbf{h}^1)$, where we used the standard generator $\mathbf{h}^1$ in place of the two lazy generators in the same class since again

[2] Defining $\mathbf{F}_c = \mathbf{T}_c + \delta\boldsymbol{\Omega}_c$ in $\mathcal{K}_c$ does not help to obtain a full-rank system, since magnetic Gauss's laws in $\mathcal{K}_c$ are a consequence of the Faraday's laws. One may even avoid to define $\boldsymbol{\Omega}_c$ (and enforce Gauss's laws in $\mathcal{K}_c$) obtaining the h-ϕ formulation [40,18] in place of the t-ω.

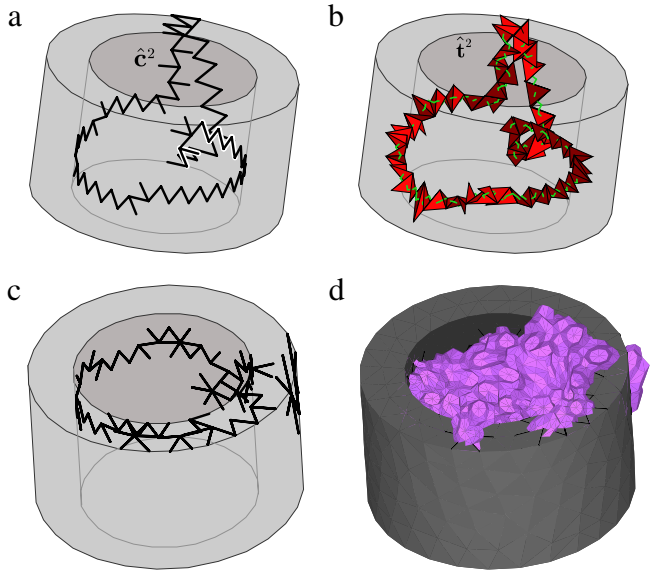

Fig. 7. (a) An example of $\hat{\mathbf{c}}^2$ generator. (b) The thinned current $\hat{\mathbf{t}}^2$ corresponding to $\hat{\mathbf{c}}^2$. (c) The boundary of the lazy cohomology generator $\hat{\mathbf{h}}^2$ corresponding to $\hat{\mathbf{t}}^2$ is not in the same $H^1(\mathcal{K}_a \cap \mathcal{K}_c)$ cohomology class as $\hat{\mathbf{c}}^2$. (d) $D(\hat{\mathbf{h}}^2)$.

we use the property that the solution does not depend on the particular representative chosen of a given cohomology class. Since in the standard case $\mathbf{I} = \delta(\mathbf{T} + i_1\mathbf{h}^1)$ and we implicitly get from Ampère's law $i_1 = \hat{i}_1 + \hat{i}_2$, the two solutions in term of induced currents are again the same.

It is important to remark that the DS algorithm has nothing to do with techniques similar to the ones described in [41] that "grow a surface on the dual complex of $\mathcal{K}_a$" starting from the representatives of the $H^1(\mathcal{K}_c \cap \mathcal{K}_a, \mathbb{Z})$ cohomology generators. Such techniques are patently incorrect, since it is obvious that starting from a mixed surface generator as $\hat{\mathbf{c}}^2$ is impossible to extended it as a 1-cocycle in $\mathcal{K}_a$. In fact, given for example a mixed generator as $\hat{\mathbf{c}}^2$ in Fig. 7a, there exist 1-cycles $c \in Z_1(\mathcal{K}_a \cap \mathcal{K}_c)$ that are homologically trivial in $\mathcal{K}_a$ such that their dot product $\langle \hat{\mathbf{c}}^2, c\rangle$ is nonzero. This yields to an inconsistency, since the fact that the $\mathbf{h}^i$ are 1-cocycles in $\mathcal{K}_a$ implies that $\langle \mathbf{h}^i, c\rangle = 0$ for all $c \in B_1(\mathcal{K}_a)$.

It is therefore important to understand why the DS algorithm does no suffer from this severe limitation. The intuitive explanation is that the trace of a lazy generator on $\mathcal{K}_c \cap \mathcal{K}_a$ obtained by the DS algorithm is not in the same cohomology class as the $H^1(\mathcal{K}_c \cap \mathcal{K}_a, \mathbb{Z})$ generator that produced it in general. In fact, the boundary of $D(\mathbf{h}^1)$ is required to be $D(\mathbf{t}^1)$ (plus, in some cases when $\mathcal{K}_c$ touches $\partial\mathcal{K}$, some part in $D(\partial\mathcal{K})$) and not $D(\mathbf{c}^1)$. Moreover, the 1-cochains $\mathbf{h}^i$ are constructed by using $\mathcal{K} = \mathcal{K}_a \cup \mathcal{K}_c$ and they are restricted to $\mathcal{K}_a$ only in the last stage of the algorithm. For example, the trivial lazy generator $\mathbf{h}^2$ on Fig. 6b exhibits a trace on $\mathcal{K}_c \cap \mathcal{K}_a$ that is trivial in $H^1(\mathcal{K}_c \cap \mathcal{K}_a, \mathbb{Z})$, see Fig. 6c, whereas it has been produced with the $H^1(\mathcal{K}_c \cap \mathcal{K}_a, \mathbb{Z})$ generator $\mathbf{c}^2$ in Fig. 5a. Concerning the other example, the generator $\hat{\mathbf{h}}^2$ (see Fig. 7d), produced starting by $\hat{\mathbf{c}}^2 = \mathbf{c}^1 + \mathbf{c}^2$ in Fig. 7a, has a trace on $\mathcal{K}_c \cap \mathcal{K}_a$ that is cohomologous to $\mathbf{c}^1$, see Fig. 7c. Therefore, as previously indicated, the DS algorithm automatically ignores the components of boundary cocycles that are trivial in $\mathcal{K}_c$.

To conclude, we verified both theoretically and with numerical experiments that the use of linearly dependent cocycles in the physical modeling does not introduce either any inconsistency in the formulation of the boundary value problem or any penalties in the computational time employed by the simulation due, for example, to a hypothetical increase of the condition number of the linear system matrix or to the use of twice as many cohomology generators as needed.

4.3. Extension of the DS algorithm for homologically nontrivial $\mathcal{K}$

The DS algorithm may be used also in applications were $\mathcal{K} = \mathcal{K}_c \cup \mathcal{K}_a$ is topologically nontrivial. This covers also the situations where only the homologically nontrivial complex $\mathcal{K}_a$ is available (in this case we assume that $\mathcal{K}_c$ is void). In the development of the DS algorithm, we assumed (as typically performed in MQS problems) that $\mathcal{K} = \mathcal{K}_c \cup \mathcal{K}_a$ is a topologically trivial cell complex. This assumption is required because the ESTT algorithm [35] works only for homologically trivial complexes. To use the DS algorithm on an arbitrary three-dimensional combinatorial manifold with boundary $\mathcal{K} = \mathcal{K}_c \cup \mathcal{K}_a$ one needs to construct the complement $\mathcal{C}$ of $\mathcal{K}$ with respect to a box or sphere containing $\mathcal{K}$. One may employ an efficient mesh generator as TetGen (http://www.tetgen.org/) to produce the mesh[3] of the complement $\mathcal{C}$ with respect to a box containing $\mathcal{K}$. The complement $\mathcal{C}$ is treated as a conducting region along with $\mathcal{K}_c$ in the DS algorithm. From Theorem 2 in [29], the first cohomology groups of $\mathcal{K}_a \subset \mathcal{K} \cup \mathcal{C}$ may be obtained in the following way:

(1) Take every connected component C of $\mathcal{K}_c \cup \mathcal{C}$.
(2) With DS algorithm compute the lazy generators of the complement of C in $\mathcal{K} \cup \mathcal{C}$.
(3) Restrict the cocycles obtained in (2) to $\mathcal{K}_a$.

This way, the algorithm presented in this paper may be used for an arbitrary three-dimensional combinatorial manifolds with boundary.

5. Numerical experiments

The DS algorithm has been integrated into the research software CDICE [45] implemented in Fortran 90. Three competing algorithms presented in [20,21,29] are considered. Two computers are used to the computations: a six years old Intel Core 2 Duo T7700 2.4 GHz laptop with 4 GB of RAM and 64 GB of RAM and Intel Xenon E7-8830 2.13 GHz processors computer. As the mesh size increases, standard cohomology computations end up in failures due to having exceeded the memory limit of the laptop. When the laptop runs out of memory, the large computer is used. The DS algorithm has been executed on the laptop up to five millions of tetrahedra without encountering any problem, which shows that it is quite economical in terms of memory usage.

Table 1 reports the time required (in seconds) for cohomology computation by the various algorithms for the benchmark problems described in [29] (not depicted here to save space). The timings obtained on the large computer are given in brackets. The Table presents also the time that DS algorithm requires for computing a standard cohomology basis (see Appendix D for implementation details).

The algorithm is going to be exploited to solve MQS problems arising in fusion engineering and design, in engineering and optimization of electromagnetic devices and the analysis of features of magnetic fields generated by current-carrying thick knots, see Fig. 8. We remark that the concept of lazy (co)homology generators is not limited to MQS problems, but they may be profitably used in many other problems arising in computational physics.

6. Conclusions

The DS algorithm introduced in this paper, even though is general and straightforward to implement and parallelize, outperforms all competing state-of-the-art algorithms for first

[3] Producing such a mesh is extremely fast, since the mesh does not need to be refined. In other words, the so called *initial mesh* (that is produced from the boundary mesh $\partial\mathcal{C}$ without adding any interior point [44]) is enough for this application.

Table 1
Time required (in seconds) for cohomology computation with various algorithms.

Benchmark	$\beta_1(\mathcal{K}_a)$	Tetrahedra	$H^1(\mathcal{K}_a, \mathbb{Z})$ [20,34]	$H^1(\mathcal{K}_a, \mathbb{Z})$ [21]	TCT [29]	DS lazy	DS $H^1(\mathcal{K}_a, \mathbb{Z})$
Trefoil knot	1	199,208	24	23	0.6	0.3	1.1
Spiral	1	1,842,070	(424)	(612)	10.1	1.7	4.1
Micro-inductor	1	2,197,192	(59,359)	(>70,000)	24.5	2.4	4.2
Micro-transformer	2	2,582,830	(>70,000)	(>70,000)	32.8	3.6	7.6
Micro-coaxial line	6	4,861,655	(612,828)	(6128)	86.1	10.6	26.8
Toroidal shell	2	2,769,200	(1503)	(>70,000)	(>70,000)	3.4	3.9

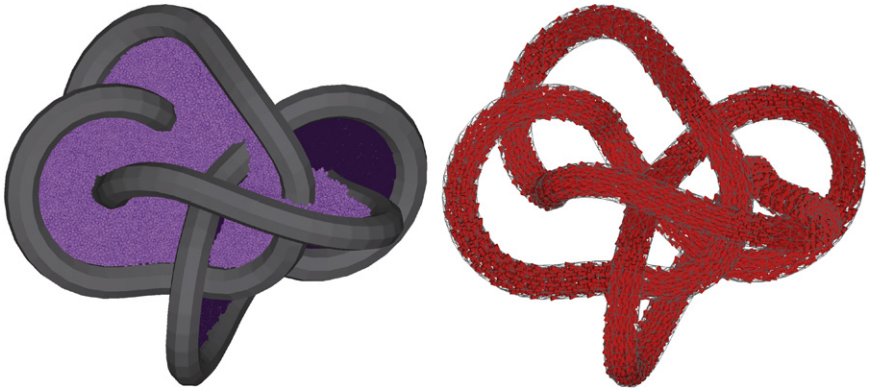

Fig. 8. On the left, a complicated thick knot and the first cohomology group generator of its complement. For clarity, the dual faces in the support of $\tilde{h}_j = D(\mathbf{h}^j)$ are shown. On the right, the current density obtained by the MQS solver.

cohomology group computations. The time required for computing cohomology generators with the DS algorithm is so limited that it allows to remove what has been considered for more than twenty-five years as the main simulation bottleneck for low-frequency electrodynamics problems. Therefore, we expect that the DS algorithm will be embedded in the next-generation electromagnetic CAE softwares and that lazy cohomology generators will be seen as a major step forward in all physical problems requiring cohomology generators.

Acknowledgments

The first author was partially supported by DARPA grant FA9550-12-1-0416. The second author was partially supported by the Italian Ministry of Education, University and Research (MIUR) project PRIN 2009LTRYRE.

Appendix A. Cohomology generators of 2-manifolds

In this section, the algorithm to compute $H^1(\mathcal{K}_c \cap \mathcal{K}_a)$ generators presented in [36] is recalled. For simplicity, we consider 2-manifolds without boundary only (in which case $\mathcal{K}_c \cap \mathcal{K}_a = \partial\mathcal{K}_c$. Therefore, later in this section we write $\partial\mathcal{K}_c$ instead of $\mathcal{K}_c \cap \mathcal{K}_a$). In [36] it is described how to compute $H^1(\mathcal{K}_c \cap \mathcal{K}_a)$ for $\mathcal{K}_c \cap \mathcal{K}_a$ being a combinatorial 2-manifold with nonempty boundary.

By the *primal skeleton* of $\partial\mathcal{K}_c$ we mean the graph consisting of all the vertices and edges in $\partial\mathcal{K}_c$. By the *dual skeleton* of $\partial\mathcal{K}_c$ we mean a graph whose vertices are the 2-cells in $\partial\mathcal{K}_c$ and an edge is put between two vertices iff. the corresponding faces in $\partial\mathcal{K}_c$ share an edge in $\partial\mathcal{K}_c$. We want to point out that edges of both the primal and dual skeletons correspond to edges of $\partial\mathcal{K}_c$.

Let us fix a spanning tree T of the primal skeleton. Let us also fix a spanning tree T' of dual skeleton. We assume, that T and T' do not share edges of $\partial\mathcal{K}_c$. In [36,46] it is shown that the number of edges in $\partial\mathcal{K}_c$ that are neither in T not in T' is the first Betti number of $\partial\mathcal{K}_c$. Moreover, the $H_1(\partial\mathcal{K}_c)$ generators are the cycles closed in T by those edges, whereas the $H^1(\partial\mathcal{K}_c)$ generators are the cycles closed in T' by those edges.

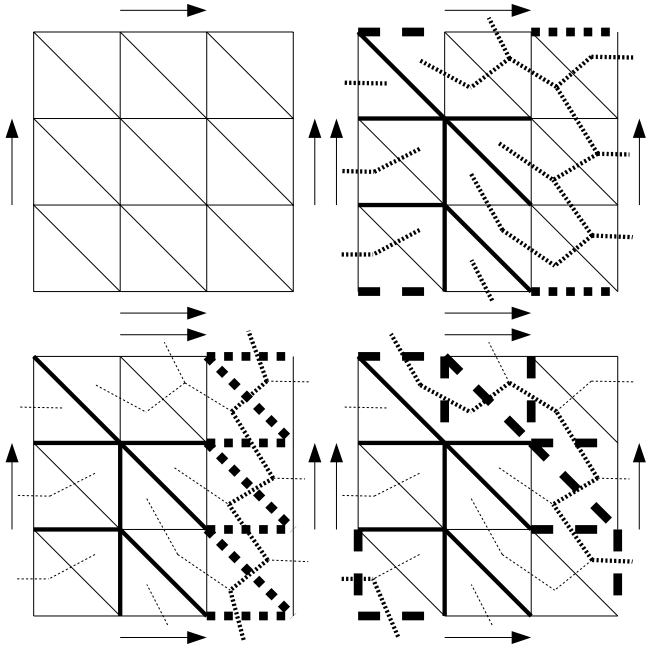

Fig. A.1. Upper left, the standard triangulation of a torus. Opposite sides are identified. Upper right, tree on primal (solid bold), and dual (dotted bold) skeleton. With double bold, the two edges not belonging to one of the trees are depicted. Lower left, a cohomology generator closed by the first edge. Lower right, cohomology generator closed by the second edge.

The idea of the procedure is presented in Fig. A.1.

In order to obtain the coefficients of the cocycle, a simple procedure that orients the cycle is used. Let v and w be the vertices of the edge that close the cycle. With a BFS strategy [47], a distance function on the tree (or dual tree) from v is built as long as w is not reached. Then, a path in the tree from w to v is found by following the decreasing values of the defined function. In this way we obtain a cycle in the graph. The obvious details on how to orient the cycle are left to the reader.

Appendix B. Proof that the thinned currents produced by the DS algorithm are cocycles

In this section, we show that the output $\mathbf{t}$ of the algorithm to obtain thinned currents from $H^1(\mathcal{K}_c \cap \mathcal{K}_a)$ generators (that is part of DS algorithm) is indeed a 2-cocycle. We need it to be a cocycle, since this is a necessary and sufficient condition for the ESTT algorithm termination [35] due to the fact that $\mathcal{K}$ is topologically trivial. Let $\mathbf{c}$ be the 1-cochain from which $\mathbf{t}$ has been constructed in DS algorithm. To show that $\mathbf{t}$ is a cocycle, first we need to remind one of the Massey's equations [33] that is a property of any regular CW-complex. Let T be a 3-dimensional cell having 1-dimensional cell E in boundary. Then there exist exactly two 2-dimensional cells T_1 and T_2 in boundary of T both having E in their boundary. Moreover, the incidence indices satisfy the following equation (Theorem IX.7.2 in [33]):

$$\kappa(K, T_1)\kappa(T_1, E) + \kappa(K, T_2)\kappa(T_2, E) = 0.$$

In order to show that $\mathbf{t}$ is a cocycle, we have to show that $\delta\mathbf{t} = 0$. This is equivalent to showing that for every 3-dimensional cell W, $\langle \delta\mathbf{t}, W\rangle = 0$. From the DS algorithm, we know that only the 2-simplices $T \in \mathcal{K}_c$ that have at least one edge $e \in (\mathcal{K}_c \cap \mathcal{K}_a)$ such that $\langle \mathbf{c}, e\rangle \neq 0$ may be nonzero in $\mathbf{t}$. Therefore, $\langle \delta\mathbf{t}, W\rangle$ may be nonzero only for 3-dimensional cells in $\mathcal{K}_c$ that have at least one edge $E \in (\mathcal{K}_c \cap \mathcal{K}_a)$ such that $\langle \mathbf{c}, E\rangle \neq 0$.

At the beginning, let us consider the case when W has a unique edge in the boundary. The other cases will immediately follow from this one. Let us assume that W has only one edge E in $(\mathcal{K}_c \cap \mathcal{K}_a)$ such that $\langle \mathbf{c}, E\rangle \neq 0$. Then, exactly the two 2-cells T_1 and T_2 from the Massey's equation will be nonzero in the cochain $\mathbf{t}$. From the DS algorithm, we have $\langle \mathbf{t}, T_1\rangle = \langle \mathbf{c}, E\rangle\kappa(T_1, E)$ and $\langle \mathbf{t}, T_2\rangle = \langle \mathbf{c}, E\rangle \kappa(T_2, E)$, respectively. From the Massey's equation, we know that $\kappa(W, T_1)\kappa(T_1, E) + \kappa(W, T_2)\kappa(T_2, E) = 0$. After multiplying this equation by $\langle \mathbf{c}, E\rangle$ we get

$$\kappa(W, T_1)\kappa(T_1, E)\langle \mathbf{c}, E\rangle + \kappa(W, T_2)\kappa(T_2, E)\langle \mathbf{c}, E\rangle = 0.$$

Therefore,

$$\begin{aligned} 0 &= \kappa(W, T_1)(\kappa(T_1, E)\langle \mathbf{c}, E\rangle) + \kappa(W, T_2)(\kappa(T_2, E)\langle \mathbf{c}, E\rangle) \\ &= \kappa(W, T_1)\langle \mathbf{t}, T_1\rangle + \kappa(W, T_2)\langle \mathbf{t}, T_2\rangle = \langle \delta\mathbf{t}, W\rangle \end{aligned}$$

that proves that $\langle \delta\mathbf{t}, W\rangle = 0$.

Now, let us assume that W has the edges $E_1, \ldots, E_l \in (\mathcal{K}_c \cap \mathcal{K}_a)$ such that for each $i \in \{1, \ldots, l\}$, $\langle \mathbf{c}, E_i\rangle \neq 0$. Each E_i contribute in $\langle \delta\mathbf{t}, W\rangle$ with two 2-cells T_1 and T_2 exactly as described in above. Therefore, its total contribution of E_i to $\langle \delta\mathbf{t}, W\rangle$ is zero. Consequently, the total contribution of all edges $E_1, \ldots, E_l$ to $\langle \delta\mathbf{t}, W\rangle$ is again zero. That proves that $\mathbf{t}$ is indeed a 2-cocycle.

Appendix C. Proof that the output of the DS algorithm are lazy cohomology generators

Let us show that the 1-cocycles obtained by the DS algorithm span $H^1(\mathcal{K}_a, \mathbb{Z})$. For this aim, the Mayer–Vietoris sequence for homology is needed [48]:

$$\begin{aligned} H_1(\mathcal{K}, \mathbb{Z}) &\to H_1(\mathcal{K}_c \cap \mathcal{K}_a, \mathbb{Z}) \xrightarrow{i_c^*, i_a^*} H_1(\mathcal{K}_c, \mathbb{Z}) \oplus H_1(\mathcal{K}_a, \mathbb{Z}) \\ &\to H_2(\mathcal{K}, \mathbb{Z}). \end{aligned}$$

Since $\mathcal{K}$ is topologically trivial, $H_i(\mathcal{K}, \mathbb{Z})$ are trivial. Therefore, $(i_c^*, i_a^*) : H_1(\mathcal{K}_c \cap \mathcal{K}_a, \mathbb{Z}) \to H_1(\mathcal{K}_c, \mathbb{Z}) \oplus H_1(\mathcal{K}_a, \mathbb{Z})$ is an isomorphism induced by inclusions $i_c : \mathcal{K}_c \cap \mathcal{K}_a \hookrightarrow \mathcal{K}_c$ and $i_a : \mathcal{K}_c \cap \mathcal{K}_a \hookrightarrow \mathcal{K}_a$. Therefore, having the generators of $H_1(\mathcal{K}_c \cap \mathcal{K}_a, \mathbb{Z})$, one may choose a subset of them that generates either $H_1(\mathcal{K}_c, \mathbb{Z})$ or $H_1(\mathcal{K}_a, \mathbb{Z})$.

Let us now take a basis of $H^1(\mathcal{K}_c \cap \mathcal{K}_a, \mathbb{Z})$. Depending if $\mathcal{K}_c \cap \mathcal{K}_a$ is a combinatorial 2-manifold with boundary or not, we will use Poincare ($H^1(\mathcal{K}_c \cap \mathcal{K}_a, \mathbb{Z}) \simeq H_1(D(\mathcal{K}_c \cap \mathcal{K}_a), \mathbb{Z})$) or Lefschetz ($H^1(\mathcal{K}_c \cap \mathcal{K}_a, \mathbb{Z}) \simeq H_1(D(\mathcal{K}_c \cap \mathcal{K}_a), \partial D(\mathcal{K}_c \cap \mathcal{K}_a), \mathbb{Z})$) duality. For a given 1-cocycle representing $H^1(\mathcal{K}_c \cap \mathcal{K}_a)$ basis, a dual cycle is constructed as in Fig. 4a.

Let us work on a fixed connected component of $\mathcal{K}_c$. For the sake of brevity, by writing $\mathcal{K}_c$ and $\mathcal{K}_c \cap \mathcal{K}_a$ we mean just the fixed connected component of $\mathcal{K}_c$ or $\mathcal{K}_c \cap \mathcal{K}_a$.[4]

For $\mathcal{K}_c \cap \mathcal{K}_a$, let us consider all the cocycles $\mathbf{t}^1, \ldots, \mathbf{t}^{2g}$, where g is the genus of $\mathcal{K}_c \cap \mathcal{K}_a$. Our aim is to show that each $\mathbf{t}^i$ is a thinned current as described in [29]. From the point (3) of the DS algorithm, we have that the 2-cells nonzero in $\mathbf{t}^i$ are the cells that have a face in the corresponding $H^1(\mathcal{K}_a \cap \mathcal{K}_c, \mathbb{Z})$ generator. Let us now shift the cycle dual to the given cohomology generator of $\mathcal{K}_c \cap \mathcal{K}_a$ (see Fig. 4b) to the interior of $\mathcal{K}_c$, see Fig. 4c.

More formally, let us take $c_1, \ldots, c_{2n}$ representing $H_1(D(\mathcal{K}_c \cap \mathcal{K}_a))$ generators on the boundary which are dual to our $H^1(\mathcal{K}_c \cap \mathcal{K}_a)$ generators. Let us shift them in the interior of $\mathcal{K}_c$ by constructing $s(c_1), \ldots, s(c_{2n})$ as in Fig. 4a. It is clear, that c_i and $s(c_i)$ are in the same homology class. Therefore, from the Mayer–Vietoris theorem, we know that we can pick a subset $\{s(c_{l_1}), \ldots, s(c_{l_n})\} \subset \{s(c_1), \ldots, s(c_{2n})\}$ which are a $H_1(\mathcal{K}_c)$ basis.

From the Appendix B, every $\mathbf{t}^i$ is a cocycle. It is clear the $\{s(c_{l_1}), \ldots, s(c_{l_n})\}$ are cycles dual to the cocycles $\{\mathbf{t}^{l_1}, \ldots, \mathbf{t}^{l_n}\}$. Therefore, $\{\mathbf{t}^{l_1}, \ldots, \mathbf{t}^{l_n}\}$ are thinned currents of $\mathcal{K}_c$. Consequently, thanks to Theorem 2 in [29], the output of the ESTT algorithm executed on $\{\mathbf{t}^{l_1}, \ldots, \mathbf{t}^{l_n}\}$ is a $H^1(\mathcal{K}_a)$ basis. Of course, since the ESTT algorithm being part of DS algorithm is run not just on $\{\mathbf{t}^{l_1}, \ldots, \mathbf{t}^{l_n}\}$, but on $\{\mathbf{t}^1, \ldots, \mathbf{t}^{2n}\}$ thinned currents, at the output we get a set of lazy cohomology generators instead of a $H^1(\mathcal{K}_a)$ basis.

Appendix D. Finding cocycles in $\mathcal{K}_c \cap \mathcal{K}_a$ that extend to a cohomology basis of $\mathcal{K}_a$

A technique to partition the homology generators of a combinatorial 2-manifold into two classes – the ones that bound in $\mathcal{K}_c$, and the ones that bound in $\mathcal{K}_a$ – is proposed in [37]. This algorithm is based on the computation of linking numbers between all possible pairs of generators and subsequent SNF computation of a small integer matrix containing the computed linking numbers. Linking numbers are defined for disjoint 1-cycles only, so a pre-processing has been applied in [37] on surface generators to perturb them in such a way no intersections are present between any pair of generators. For this purpose, the *submerged cycles* and *shifted surface cycles* have been found. This is not completely trivial to do algorithmically and cycles have to be "straightened" before [37]. The complexity of the linking number computation is quadratic with the number of edges in $\mathcal{K}_c \cap \mathcal{K}_a$ and a computationally costly interval arithmetic [49] package has to be used to rigorously compute linking numbers without the risk of errors due to the finite precision of real numbers [49,50], especially on coarse meshes.

In this paper, to find the change of basis to find cohomology generators nontrivial in $\mathcal{K}_a$ we use the same idea presented in [37], but finding linking numbers between paths on the dual complex. This approach, contrarily to [37], does not require any extra computations since the surface cycles are obtained by considering the dual 1-cells that are dual to 1-cells in the support of the cohomology generators, while the submerged cycles are defined

[4] In this proof, to keep the notation as simple as possible, we assume that $\mathcal{K}_a$ is connected. In such case, for each connected component of $\mathcal{K}_c$ there is a corresponding connected component of $\mathcal{K}_c \cap \mathcal{K}_a$. In the case when $\mathcal{K}_a$ is not connected, in the following the reader should consider all the connected components of $\mathcal{K}_c \cap \mathcal{K}_a$ incidental to the considered connected component of $\mathcal{K}_c$.

simply as the dual 1-cells that are dual to 2-cells in the support of thinned currents. We want to remark that for this purpose the 2-cells belonging to the support of thinned currents have to be in order. They may be easily ordered in linear time and the technical details are left to the reader.

About the practical implementation, the algorithm presented in [50] is used to compute linking numbers and the interval arithmetic library for Fortran 90 presented in [51] is employed for interval arithmetic computations.

References

[1] H. Poincaré, Analysis situs, J. de l'École Polytechnique 2 (1) (1895) 1–123.
[2] H. Poincaré, Complément à l'analysis situs, Rend. Circ. Mat. Palermo 13 (1899) 285–343.
[3] H. Poincaré, Second complément à l'analysis situs, Proc. Lond. Math. Soc. 32 (1900) 277–308.
[4] J.R. Munkres, Elements of Algebraic Topology, Perseus Books, Cambridge, MA, 1984.
[5] T.H. Dey, K. Li, J. Sun, D. Cohen-Steiner, Computing geometry-aware handle and tunnel loops in 3d models, ACM Trans. Graph. 27 (2008) 1–9.
[6] X. Guo, X. Li, Y. Bao, X. Gu, H. Qin, Meshless thin-shell simulation based on global conformal parameterization, IEEE Trans. Vis. Comput. Gr. 12 (2006) 375–385.
[7] M. Mrozek, M. Zelawski, A. Gryglewski, S. Han, A. Krajniak, Homological methods for extraction and analysis of linear features in multidimensional images, Pattern Recogn. 45 (1) (2012) 285–298.
[8] V. De Silva, R. Ghrist, Homological sensor networks, Not. AMS 54 (2007).
[9] M. Farber, Topological complexity of motion planning, Discr. Comput. Geom. 29 (2) (2003) 211–221.
[10] M. Nicolau, A.J. Levine, G. Carlsson, Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival, Proc. Natl. Acad. Sci. 108 (17) (2011) 7265–7270.
[11] P.G. Mezey, Group theory of electrostatic potentials: a tool for quantum chemical drug design, Int. J. Quant. Chem. 28 (1985) 113–122.
[12] K. Mischaikow, M. Mrozek, J. Reiss, A. Szymczak, Construction of symbolic dynamics from experimental time series, Phys. Rev. Lett. 82 (1999) 1144–1147.
[13] M. Gameiro, K. Mischaikow, W. Kalies, Topological characterization of spatial–temporal chaos, Phys. Rev. E 70 (2004) 035203.
[14] H. Kurtuldu, K. Mischaikow, M.F. Schatz, Extensive scaling from computational homology and Karhunen–Loève decomposition analysis of Rayleigh–Bénard convection experiments, Phys. Rev. Lett. 107 (2011) 034503.
[15] H. Bombin, M.A. Martin-Delgado, Topological quantum distillation, Phys. Rev. Lett. 97 (2006) 180501.
[16] P.R. Kotiuga, Hodge decompositions and computational electromagnetics, Ph.D. Thesis, Department of Electrical Engineering, McGill University, Montréal, 1984.
[17] P.W. Gross, P.R. Kotiuga, Electromagnetic Theory and Computation: A Topological Approach, in: MSRI, vol. 48, Cambridge University Press, 2004.
[18] A. Bossavit, Computational Electromagnetism, Academic Press, 1998.
[19] R. Specogna, S. Suuriniemi, F. Trevisan, Geometric t-ω approach to solve eddy-currents coupled to electric circuits, Int. J. Numer. Meth. Eng. 74 (2008) 101–115.
[20] P. Dłotko, 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.
[21] P. Dłotko, R. Specogna, Efficient cohomology computation for electromagnetic modeling, CMES 60 (2010) 247–278.
[22] R. Specogna, Complementary geometric formulations for electrostatics, Int. J. Numer. Meth. Eng. 86 (2011) 1041–1068.
[23] P. Dłotko, R. Specogna, Cohomology in 3d magneto-quasistatic modeling, Commun. Comput. Phys. 14 (1) (2013) 48–76. arXiv:1111.2374.
[24] A. Storjohann, Near optimal algorithms for computing smith normal form of integer matrices, in: Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation, ISAAC, 1996, pp. 267–274.
[25] M. Pellikka, S. Suuriniemi, L. Kettunen, Homology in electromagnetic boundary value problems, Bound. Value Probl. 2010 (1) (2010) 381953.
[26] M. Pellikka, S. Suuriniemi, L. Kettunen, Powerful heuristics and basis selection bring computational homology to engineers, IEEE Trans. Magn. 47 (5) (2011) 1226–1229.
[27] P.R. Kotiuga, On making cuts for magnetic scalar potentials in multiply connected regions, J. Appl. Phys. 61 (1987) 3916–3918.
[28] P.R. Kotiuga, An algorithm to make cuts for magnetic scalar potentials in tetrahedral meshes based on the finite element method, IEEE Trans. Magn. 25 (1989) 4129–4131.
[29] P. Dłotko, R. Specogna, A novel technique for cohomology computations in engineering practice, Comput. Methods Appl. Mech. Eng. 253 (2013) 530–542.
[30] J. Dodziuk, Finite difference approach to the hodge theory of harmonic forms, Amer. J. Math. 98 (1976) 79–104.
[31] D.N. Arnold, R.S. Falk, R. Winther, Finite element exterior calculus, homological techniques, and applications, Acta Numer. 15 (2006) 1–155.
[32] J.C. Maxwell, A Treatise on Electricity and Magnetism, Clarendon Press, Oxford, 1891.
[33] W.S. Massey, A Basic Course in Algebraic Topology, in: Graduate Texts in Mathematics, vol. 127, Springer-Verlag, 1991.
[34] P. Dłotko, R. Specogna, Critical analysis of the spanning tree techniques, SIAM J. Numer. Anal. 48 (2010) 1601–1624.
[35] P. Dłotko, R. Specogna, Efficient generalized source field computation for h-oriented magnetostatic formulations, Eur. Phys. J. Appl. Phys. 53 (2011) 20801.
[36] P. Dłotko, A fast algorithm to compute cohomology group generators of orientable 2-manifolds, Pattern Recognit. Lett. 33 (2012) 1468–1476.
[37] 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.
[38] A. Bossavit, Whitney forms: a class of finite elements for three-dimensional computations in electromagnetism, IEE Proc. A 135 (8) (1988) 493–500.
[39] R. Bott, L.W. Tu, Differential Forms in Algebraic Topology, Springer-Verlag, 1982.
[40] O. Biro, Edge element formulations of eddy current problems, Comput. Methods Appl. Mech. Eng. 169 (3–4) (1999) 391–405.
[41] Z. Ren, t-ω formulation for eddy-current problems in multiply connected regions, IEEE Trans. Magn. 38 (2) (2002) 557–560.
[42] 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.
[43] Z. Ren, Influence of the rhs on the convergence behaviour of the curl–curl equation, IEEE Trans. Magn. 32 (3) (1996) 655–658.
[44] M. Bern, Compatible tetrahedralizations, in: Proceedings of the Ninth Annual Symposium on Computational Geometry, SCG'93, ACM, New York, NY, USA, 1993, pp. 281–288.
[45] R. Specogna, Cdice research software, 2008–2013. http://www.comphys.com.
[46] J. Erickson, K. Whittlesey, Greedy optimal homotopy and homology generators, in: Proc. 16th Annual ACM–SIAM Symposium Discrete Algorithms, SODA, 2005, pp. 1038–1046.
[47] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, second ed., The MIT Press, 2001.
[48] A. Hatcher, Algebraic Topology, Cambridge University Press, 2001.
[49] B. Hayes, A lucid interval, Amer. Sci. 91 (6) (2003) 484–488.
[50] Z. Arai, A rigorous numerical algorithm for computing the linking number of links, Nonlinear Theory Appl. 4 (1) (2013) 104–110.
[51] R. Baker Kearfott, Algorithm 763: INTERVAL_ARITHMETIC: a fortran 90 module for an interval data type, ACM Trans. Math. Software 22 (4) (1996) 385–392.