Comput. Methods Appl. Mech. Engrg. 198 (2009) 3765–3781

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

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

Automatic generation of cuts on large-sized meshes for the T–Ω geometric eddy-current formulation

Paweł Dłotko [a,1], Ruben Specogna [b,*], Francesco Trevisan [b]

[a] Jagiellonian University, Institute of Computer Science, Lojasiewicza 4, 30348 Kraków, Poland
[b] Università di Udine, Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica, Via delle Scienze 208, 33100 Udine, Italy

ARTICLE INFO

Article history:
Received 19 February 2009
Received in revised form 1 July 2009
Accepted 3 August 2009
Available online 11 August 2009

Keywords:
Eddy-currents
Scalar potential in multiply-connected regions
Automatic cut generation
Belted tree
Computational homology
Homology generators

ABSTRACT

In this paper a Geometric T–Ω formulation to solve eddy-current problems on a tetrahedral mesh is presented. When non-simply-connected conducting regions are considered, the formulation requires the so-called *thick cuts*, while, in the literature, more attention is usually given to the so-called *thin cuts*. While the automatic construction of thin cuts has been theoretically solved many years ago, no implementation of an algorithm to compute the thick cuts which can be used in practice exists so far.

In this paper, we propose how to fill this gap by introducing an algorithm to automatically compute the thick cuts on real-sized meshes, based on a *belted tree* and a *tree–cotree decomposition*. The belted tree is constructed by means of a homology computation by exploiting efficient reduction methods. A number of benchmarks are presented to demonstrate the generality and the robustness of the algorithm. A rigorous definition of thick cuts, which necessarily has to rely on cohomology, is presented in addition.

1. Introduction

In the recent years the use of algebraic topology gained a considerable importance in the research on computational electromagnetism. In particular, the fundamental works of Branin [1], Tonti [3,4], Weiland [5], Bossavit, Kettunen [6,7] and Kotiuga [8] gave rise to a way to solve Maxwell's equations alternative with respect to the classical Finite Element Method (FEM). This class of numerical methods, that can be called "Discrete Geometric Approach" (DGA), allows the construction of an algebraic system of equations by combining both the physical laws of electromagnetism, formulated exactly in a purely topological way and the constitutive relations, approximated in a geometric way on a specified grid.

The DGA have been already applied as a numerical method to solve various classes of physical problems. In [40] two complementary Discrete Geometric Formulations to solve eddy-current problems are described. A domain of interest D of the eddy-current problem is introduced as a compact subset of the three-dimensional Euclidean space. In D, a set of passive conductive regions D_c is defined. The complement of D_c in D represents the non-conducting region D_a (air region). To take advantage of the irrotationality of the magnetic field in the non-conducting region, the T–Ω formulation, based on the circulation of an electric vector potential T in D_c and a magnetic scalar potential Ω in the whole domain D, is considered in this paper. This formulation uses a magnetic scalar potential in D_a in order to reduce the computational cost with respect to the other complementary formulation $A-\chi$, based on the circulation of a magnetic vector potential A in the whole domain D and an electric scalar potential in the conducting region D_c [40].

The DGA, thanks to the use of physical balance laws, can be elegantly formalized with algebraic topology [1,3]. There exists a large mathematical literature about algebraic topology, which unfortunately is usually far away with respect to the theoretical background available to physicists and especially to engineers. For this reason, in the next section the relevant concepts of algebraic topology are briefly recalled in an informal way. For a more formal introduction on algebraic topology refer to [13–15].

It is well known that the T–Ω formulation requires the so-called *thick cuts* [23,41], while, usually, in the literature the *thin cuts* [19] are introduced instead. Even though the automatic computation of thin cuts has been known for many years [19,18], we are not aware of any implementation that can be used with real-sized finite element meshes. Moreover, as will be illustrated in the paper, extract a thick cut from a general thin cut is not trivial at all.

The aim of the paper is to use the methods from computational topology to design a general and reliable algorithm for the

[*] Corresponding author. Tel.: +39 0432 558285; fax: +39 0432 558251.
E-mail addresses: dlotko@ii.uj.edu.pl (P. Dłotko), ruben.specogna@uniud.it (R. Specogna), trevisan@uniud.it (F. Trevisan).
[1] Partially supported by Polish MNSzW, Grant MNiSzW No. N201 037 31/3151.

0045-7825/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2009.08.007

automatic computation of thick cuts. The thick cuts produced by the algorithm presented in this paper can also be adopted to solve an eddy-current problem using the classical Finite Element Method (FEM) with a *h*-oriented formulation based on edge elements.

The paper is structured as follows: in Section 2 an informal introduction to homology and cohomology theories is presented. In Section 3 the barycentric complex is introduced and its properties are highlighted. The *T*–*Ω* formulation is presented in Section 4. In Section 5 the *loop fields* and the *thick cuts* are defined by using the cohomology theory. In Section 6 the non-local Faraday's equations, arising when a non-simply-connected conductive region is considered, are described. Section 7 contains a critical survey of the algorithms used to automatically generate cuts. The algorithm proposed in this paper to automatically compute the thick cuts is presented in Section 8. Finally, in Section 9 the numerical results on some relevant benchmark problems are shown and in Section 10 the conclusions are drawn.

2. Informal introduction to homology and cohomology theories

2.1. Abstract simplicial complex

The domain of interest *D* is described by a finite element mesh of tetrahedra. It is assumed that the generated mesh reflects the topology of the domain of interest, i.e. the mesh is adequately refined in such a way that all topological features of the domain of interest are captured.

A tetrahedral mesh is composed by (geometrical) *simplices*. In mesh generation literature [11] usually a simplex of dimension *k*, or *k*-simplex, is defined as the convex hull[2] of the $k+1$ affinely independent[3] nodes $\{x_0,\ldots,x_k\}$. The concept of simplices is used to separate the topological information contained in a mesh of an object of interest (like its topological invariants)[4] from its geometric shape. In fact the topology of the mesh can be represented only by the *incidence information*[5] of the simplices, while the geometry is described with the list of the node's coordinates. This separation is quite important, in fact the algorithms presented in this paper will deal only with the incidence information. The geometry will be used only for the field computation in the discrete counterpart of the physical constitutive relations. Considering this aim, it is useful to formalize the incidence without the geometry defining a combinatorial object called *abstract simplicial complex*.

Definition 1. A collection $\mathscr{K}$ of finite, non-empty sets is called an *abstract simplicial complex* if for any set $S\in\mathscr{K}$ every non-empty subset of *S* belongs to $\mathscr{K}$. Any element $S\in\mathscr{K}$ is called *abstract simplex*.

From this point on the word abstract will be omitted from the definitions, since only abstract simplices and abstract simplicial complexes will be considered.

The dimension of a simplex *S* in a simplicial complex $\mathscr{K}$ is an integer, *p*, which is equal to the cardinality of *S* minus one.

Definition 2. For any given *p*-simplex *S*, any non-empty subset of *S* is called a *sub-simplex* of *S*.

For a geometric simplex, which is a subset of $\mathbb{R}^n$ spanned by affinely independent nodes $\{x_0,\ldots,x_n\}$, the corresponding abstract representation consists of the nodes themselves $\{x_0,\ldots,x_n\}$; let us denote 0-simplices for nodes, 1-simplices for edges, 2-simplices for faces, and 3-simplices for tetrahedra.

The analogy between the abstract and geometric simplex entails a corresponding analogy between abstract and geometric simplicial complexes, as presented for example in [15]. The definition of a geometric simplicial complex corresponds to the definition of a *conformal mesh* given in mesh generation literature:

Definition 3. A mesh is said *conformal* if and only if:

(1) Every sub-simplex of a geometric simplex is contained in the mesh.
(2) The intersection of any two geometric simplices is either a common sub-simplex, or it is empty.

Example 1. The reader can easily verify that the mesh on the left in Fig. 1 is a conformal mesh (and in this case its abstract representation is the abstract simplicial complex), while the mesh on the right is a non-conformal mesh (and the abstract representation is not the abstract simplicial complex).

In our implementation a conformal mesh of the domain of interest *D* is produced using the software Netgen [10] and the *abstract simplicial complex* is constructed, using the mesh file, by the algorithm described in Appendix A. The simplices in the abstract simplicial complex are the sets consisting of numbers given to the corresponding geometric nodes by Netgen.

2.2. Oriented simplices and chains

Now, let us consider all orderings of the $k+1$ nodes of a given *k*-simplex *S*. The ordering is defined as the arbitrary bijection from $\{0,\ldots,k\}$ into the set *S*. Orderings can be classified by the equivalence relation[6] introduced as follows. Two orderings S_1, S_2 of a simplex *S* are said to be equivalent, if they differ by an even permutation.[7] If those orderings differ by an odd permutation, they are not equivalent. So all those orderings can be divided into two equivalence classes. Such a class of ordering is called an *(inner) orientation* and the set of all oriented *n*-simplices of a complex $\mathscr{K}$ will be denoted by $S_n(\mathscr{K})$.

Definition 4. An *oriented simplex* σ is a simplex *S* together with a choice of orientation. An *n*-simplex $S=\{x_0,\ldots,x_n\}$ endowed with orientation will be denoted as $\sigma^n=[x_0,\ldots,x_n]$, where $[\cdot]$ stands for an ordered list of nodes.

Example 2. If one considers the 2-simplex (triangle) in Fig. 1 with indices of nodes 1, 2 and 3. As a model decision, one can fix $[1,2,3]$ as positive orientation, see Fig. 2a. As a consequence, the 2-simplex $[2,3,1]$ in Fig. 2b represents the same oriented 2-simplex since it is in the same equivalence class, while for example $[2,1,3]$ in Fig. 2c is negatively oriented, since it is not in the orientation class of simplex $[1,2,3]$ which was chosen as a representative of positively oriented class.

[2] The convex hull of a set of points *S* is the minimal convex set containing these points. For further details see for example [11].

[3] In our case it is required that any three points that constitute a 2-simplex (triangle) must not lie in the same line and that the four points that constitute a 3-simplex (tetrahedron) must not lie in the same plane.

[4] A topological invariant is a topological property that does not depend on continuous deformations of the object, like for example the number of holes.

[5] Intuitively the incidence contains information on how the simplices are glued together in the mesh.

[6] An equivalence relation is a binary relation between two elements of a set which groups them together as being "equivalent" in some way. An equivalence relation $\sim$ is reflexive, symmetric, and transitive. The equivalence class of the element *a*, under $\sim$, denoted $[a]$, is the subset of *S* whose elements *b* are such that $a\sim b$.

[7] A permutation is the bijective map $\Pi:\{0,\ldots,k\}\to\{0,\ldots,k\}$. Even (odd) permutations can be represented as a sequence of an even (odd) number of swaps of neighboring elements.

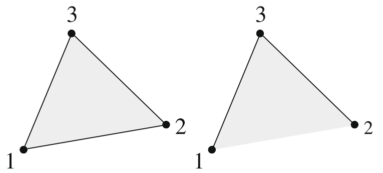

Fig. 1. The abstract simplicial complex corresponding to the geometric 2-simplex (triangle) on the left is: $\mathscr{K} = \{\{1\},\{2\},\{3\},\{1,2\},\{1,3\},\{2,3\},\{1,2,3\}\}$. Consider the abstract representation $\mathscr{K}_1 = \mathscr{K} \setminus \{1,2\}$ of the complex on the right. When one consider the simplex $\{1,2,3\} \in \mathscr{K}_1$ and the sub-simplex $\{1,2\} \subset \{1,2,3\}$, one has that $\{1,2\} \notin \mathscr{K}_1$. Thus one can conclude from the definition that $\mathscr{K}_1$ is not an abstract simplicial complex.

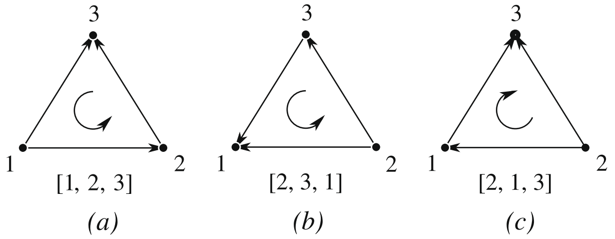

Fig. 2. Positive and negative oriented 2-simplices.

The order of nodes obtained by increasing ordering of the list of node indices is assumed as representative for positive orientation of each simplex. All further computations are independent from the chosen orientation. The orientation of each simplex is fixed with the positive orientation previously defined. This means that, for example, in the computations [2,1] will not appear, being represented by – [1,2].

Definition 5. A *q-chain* c_q with integer coefficients is a formal combination of oriented q-simplices with integer coefficients. The set of all q-chains in a complex $\mathscr{K}$ will be denoted as $C_q(\mathscr{K})$. All the oriented q-simplices form a basis of $C_q(\mathscr{K})$. The elements of $C_q(\mathscr{K})$ form a group with addition.[8]

Example 3. Consider the oriented abstract simplicial complex $\mathscr{K}$ corresponding to the simplices in Fig. 3a. An example of 1-chain, depicted in Fig. 3b, is $c_1 = -\sigma_1^1 - \sigma_4^1 - \sigma_5^1 + \sigma_8^1 + \sigma_9^1$. Since a base of $C_1(\mathscr{K})$ (all 1-simplices with positive orientation) was fixed as in Definition 5, the chain c_1 can be represented by the vector $\{-1,0,0,-1,-1,0,0,1,1,0\}^T$. The coefficient corresponding to the 1-simplex with index 1 is -1 because, for example, one wants all the edges iso-oriented with the path r. An example of 2-chain $c_2 = \sigma_4^2 + \sigma_5^2 = \{0,0,0,1,1\}^T$ is depicted in Fig. 3c.

2.3. Cochains and chain–cochain duality

Definition 6. The group of the *elementary q-cochains* is equal to the group of all maps going from one oriented q-simplex of the complex $\mathscr{K}$ to integers with addition.

Definition 7. A *q-cochain* c^q with integer coefficients is a formal combination of elementary q-cochains. It will be denoted by $C^q(\mathscr{K}) = \{\phi : S_q(\mathscr{K}) \to \mathbb{Z}\}$.

Note that a one-to-one correspondence between a chain and a cochain exists. This chain–cochain natural duality yields

$C_p(\mathscr{K}) \cong$[9] $C^p(\mathscr{K}).$ (1)

2.4. Boundary (coboundary) operator and homology (cohomology)

Definition 8. For $k \geq 1$, the *boundary operator* $\partial_k : C_k(\mathscr{K}) \to C_{k-1}(\mathscr{K})$ is defined in the following way:

(1) For an oriented k-simplex $\sigma^k = [x_0, x_1, \ldots, x_k] \in C_k(\mathscr{K})$ yields $\partial_k(\sigma^k) := \sum_{i=0}^{k}(-1)^i [x_0, x_1, \ldots, \hat{x}_i, \ldots, x_k] \in C_{k-1}(\mathscr{K})$, where $[x_0, x_1, \ldots, \hat{x}_i, \ldots, x_k]$ denotes the simplex $[x_0, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k]$.
(2) For a linear combination of oriented simplices, the linear extension of this operator is applied: $\partial_k(\sum_i a_i \sigma_i^k) = \sum_i a_i \partial_k(\sigma_i^k)$, where σ_i^k is the ith simplex in the sum.

The boundary operator is defined as above in order to comply with the equality $\partial_{k-1} \circ \partial_k = 0$ [15].

Example 4a. Consider the oriented abstract simplicial complex $\mathscr{K}$ corresponding to the simplices in Fig. 4a. The aim now is to find the boundary of the 2-chain $c_2 = \sigma_1^2 + \sigma_2^2$ in Fig. 4b. $\partial_2 c_2 = \partial_2 \sigma_1^2 + \partial_2 \sigma_2^2 =$[10] $\partial_2[1,2,3] + \partial_2[1,3,4] = [2,3] - [1,3] + [1,2] + [3,4] - [1,4] + [1,3] = [2,3] + [1,2] + [3,4] - [1,4] = \sigma_2^1 + \sigma_1^1 + \sigma_4^1 - \sigma_5^1$ that can be represented by the 1-chain $c_1 = \partial_2 c_2 = \{1,1,0,1,-1\}$ in Fig. 4c. If one applies again the boundary operator on c_1, he obtains: $c_0 = \partial_1 c_1 = \partial_1[2,3] + \partial_1[1,2] + \partial_1[3,4] - \partial_1[1,4] = [3] - [2] + [2] - [1] + [4] - [3] - [4] + [1] = 0$.

Since the boundary operator is a linear map between $C_k(\mathscr{K})$ and $C_{k-1}(\mathscr{K})$, once a basis for $C_{k-1}(\mathscr{K})$ and $C_k(\mathscr{K})$ is fixed, it can be represented as the matrix $\mathbf{M}_{\partial_k}$.

Example 4b. In fact, in the Example of Fig. 4 one has:

$$\mathbf{M}_{\partial_1} = \begin{bmatrix} -1 & 0 & -1 & 0 & -1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \quad \mathbf{M}_{\partial_2} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ -1 & 1 \\ 0 & 1 \\ 0 & -1 \end{bmatrix},$$

and it is easy to verify that $\mathbf{M}_{\partial_1}\mathbf{M}_{\partial_2} = 0$.

In the cohomology theory, the so-called *coboundary operator* $\delta^{k-1} : C^{k-1}(\mathscr{K}) \to C^k(\mathscr{K})$ is defined. For our purpose it is enough to say that both boundary and coboundary can be expressed by a matrix. If $\mathbf{M}_{\partial_k}$ is a matrix of boundary operator ∂_k, for the matrix of coboundary operator $\delta^{k-1}, \mathbf{M}_{\delta^{k-1}} = \mathbf{M}_{\partial_k}^T$ holds.

Since in general $\partial_{k-1} \circ \partial_k = 0 (\mathbf{M}_{\partial_{k-1}}\mathbf{M}_{\partial_k} = 0)$, this implies that $(\mathbf{M}_{\partial_{k-1}}\mathbf{M}_{\partial_k})^T = \mathbf{M}_{\partial_k}^T \mathbf{M}_{\partial_{k-1}}^T = \mathbf{M}_{\delta^{k-1}} \circ \mathbf{M}_{\delta^{k-2}} = 0$ holds.

Usually, in computational electromagnetism, the matrices representing the coboundary operator are expressed in terms of *incidence matrices*:

- $\mathbf{G} = \mathbf{M}_{\partial_1}^T$, between the edges and the nodes;
- $\mathbf{C} = \mathbf{M}_{\partial_2}^T$, between the faces and the edges;
- $\mathbf{D} = \mathbf{M}_{\partial_3}^T$, between the tetrahedra and the faces.

[8] A group is an algebraic structure consisting of the set S and binary operation + defined in S. The operation needs to be associative, and the identity element $e \in S$ of the operation + has to exists in S. Moreover each element $x \in S$ has to posses an inverse with respect to the operation +. In this paper only abelian groups will be used, for which the operation is also commutative. The integers with addition are a standard example of an abelian group.

[9] Informally, two groups are isomorphic ($\cong$) if they are identical up to the name of elements and name of operation.

[10] For sake of brevity, the 0-simplex σ_i^0 will be denoted as i.

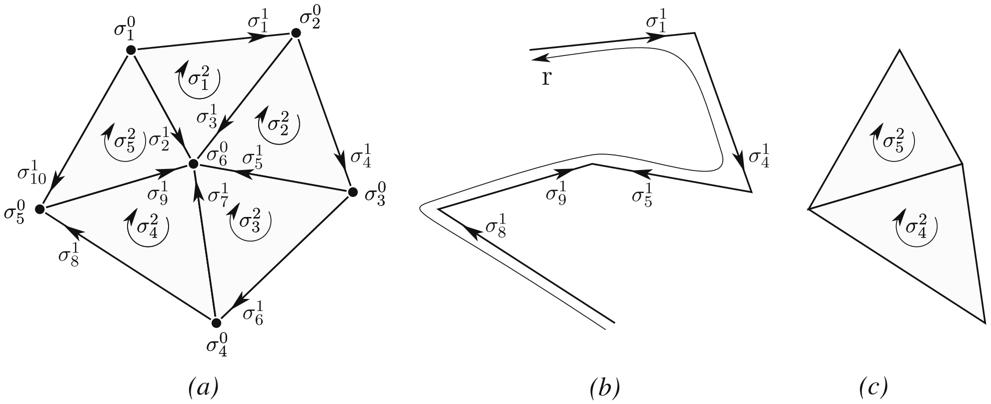

Fig. 3. An example of 1-chain and 2-chain.

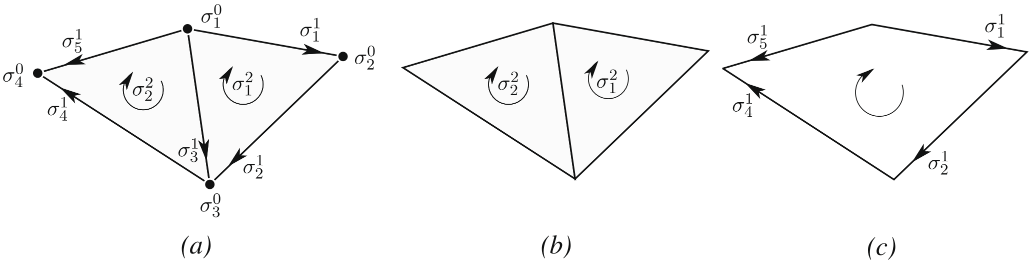

Fig. 4. A complex that is used to exemplify the boundary operator.

The boundary operator gives rise to a classification of chains. From $\partial_{k-1} \circ \partial_k = 0$ it is straightforward to verify that the image of the ∂_k is a subgroup of the kernel of ∂_{k-1}. The $\mathrm{im}(\partial_{k+1})$ is called the *boundary group* of a simplicial complex $\mathscr{K}$ and is denoted by $B_k(\mathscr{K})$. The $\ker(\partial_k)$ is called the *cycles group* of a complex $\mathscr{K}$ and is denoted by $Z_k(\mathscr{K})$. Elements of $Z_k(\mathscr{K})$ are called *cycles* and elements of $B_k(\mathscr{K})$ are called *boundaries* in the simplicial complex $\mathscr{K}$.

Analogous classification can be given for the cochains: $Z^k(\mathscr{K}) = \ker(\delta^k) = \ker(\mathbf{M}^T_{\partial_{k+1}})$ is the group of k-cocycles and $B^k(\mathscr{K}) = \mathrm{im}(\delta^{k-1}) = \mathrm{im}(\mathbf{M}^T_{\partial_k})$ is the group of k-coboundaries.

The *homology group* consists of all the elements from $Z_k(\mathscr{K})$ which are not elements of $B_k(\mathscr{K})$, formally

Definition 9. The *homology group* is the quotient group
$H_k(\mathscr{K}) = Z_k(\mathscr{K})/B_k(\mathscr{K})$ for $k \in \mathbb{N}$.

Given a cycle z in the kth homology group, adding any k-boundary b to it does not make $z + b$ a boundary. For homology theory z and $z + b$ are the same, what motivates the following equivalence relation:

Definition 10. Two k-cycles z_1 and z_2 are *homologous* (or homologically equivalent) if their difference is a boundary

$$z_1 \sim z_2 \iff z_1 - z_2 \in B_k(\mathscr{K}).$$

The homology is an equivalence relation that divides the cycles into equivalence classes, which are called *homology classes*. There exists a set of homology classes such that any other homology class can be written in unique way as a combination of these classes with integer coefficients. The basic set of homology classes will be referred to as *homology generators*. From homology generators one can generate the whole homology group, the set of all homology generators is called a *homology basis*. Only one representatives z from each class is needed, being possible to obtain all other cycles in the class by adding a boundary chain to z. When it is not confusing, for brevity, by homology generators, we will refer to both the equivalence classes of the presented relation and the cycles that represents the equivalence class.

Example 5. Consider $\mathscr{K}$ as the oriented simplicial complex obtained by the complement of the torus (whose triangulated boundary is visible in the Fig. 5a) with respect to a bigger box that contains it (only outlined in the same figure). The boundary of $\mathscr{K}$ is the union of the boundary of the torus and the boundary of the external box. In Fig. 5a four cycles $p_i \in Z_1(\mathscr{K}),\ i = 1, \ldots, 4$ are drawn with a continuous black line; p_1 and p_2 can be boundaries of a surface in $\mathscr{K}$, while p_3 and p_4 cannot be boundaries of any surface entirely contained in $\mathscr{K}$. Thus p_1 and p_2 are trivial (zero) elements of the 1st homology group $H_1(\mathscr{K})$, while p_3 and p_4 are non-zero elements of $H_1(\mathscr{K})$. These type of cycles are also called *non-bounding cycles*. Moreover, since p_3 and p_4 differ by a boundary,[11] they are in the same homology (equivalence) class. In the case presented in the Fig. 5, either p_3 or p_4 can be considered as the $H_1(\mathscr{K})$ homology basis.

Remark. Even though the same conclusions in the Example 5a can be reached considering *homotopy theory*,[12] in general the two theories produce different results [28–30]. This issue will be discussed in Section 7.

[11] In fact the difference of the two cycles can be the boundary of a surface contained in $\mathscr{K}$. In the Example such a surface is the generalized cylinder depicted in Fig. 5b having p_3 and $-p_4$ as boundary.

[12] In homotopy theory a non-trivial cycle is a cycle that cannot be deformed into a point by continuous deformations. Such a cycle is referred to as non-contractible, while the cycles that can be deformed to the point are called contractible. For example p_3 and p_4 cannot be reduced into a point by continuous deformations in $\mathscr{K}$, while p_1 and p_2 are homotopically trivial.

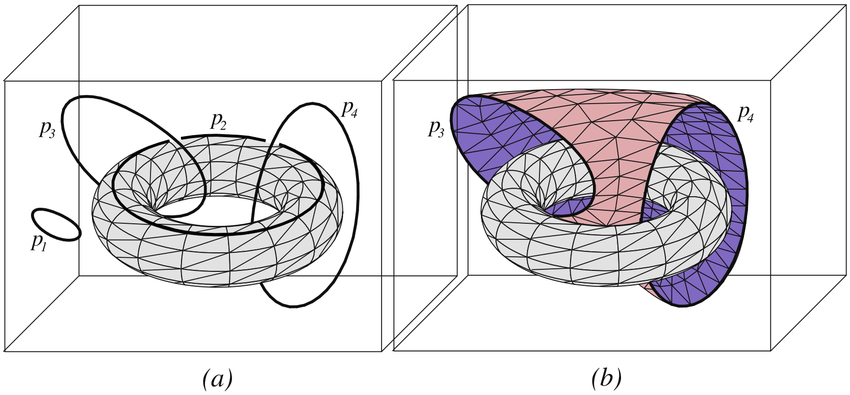

Fig. 5. (a) p_3 and p_4 are two possible absolute 1st homology generators for the torus complement. (b) p_3 and p_4 are the boundary of the depicted cylindrical surface.

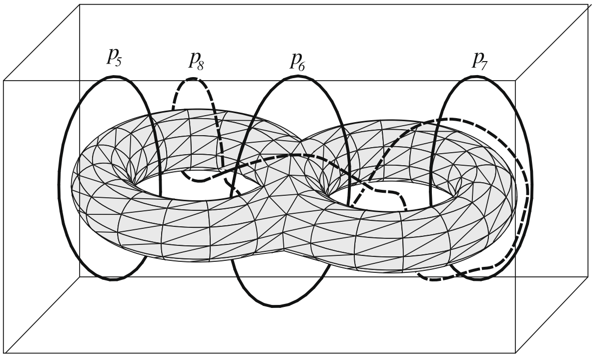

Fig. 6. Some homologically non-trivial cycles for the double torus complement.

Example 6. Consider now $\mathcal{K}$ as the oriented simplicial complex obtained by the complement of the double torus with respect to a bigger box that contains it, Fig. 6. As an example of homology generators one may take p_5 and p_7. In this case $p_6 \in H_1(\mathcal{K})$ can be obtained by linear combination of p_5 and p_7 (and a cycle which is a boundary). Also $p_8 \in H_1(\mathcal{K})$ is a combination of p_5, p_7. The independent cycles p_5 and p_6 or p_5 and p_7 are different examples of basis for $H_1(\mathcal{K})$.

The analog quotient for the cocycles and coboundaries can be defined.

Definition 11. The *cohomology group* is the quotient group $H^k(\mathcal{K}) = Z^k(\mathcal{K})/B^k(\mathcal{K})$, thus contains cocycles that are not coboundaries.

It is demonstrated, for example in [15,17,19], that in case of the complexes $\mathcal{K}$, whose geometric realizations are compact subsets of tree-dimensional Euclidean space, the following isomorphisms exist:

$$H_k(\mathcal{K}) \cong H^k(\mathcal{K}). \tag{2}$$

2.5. Computational homology

The idea of the homology computation is now briefly presented. Based on the abstract simplicial complex, constructed as described in Appendix A, each matrix $\mathbf{M}_{\partial_1}, \mathbf{M}_{\partial_2}, \mathbf{M}_{\partial_3}$ representing the boundary operators in a fixed basis is assumed as given. Then the basis of $Z_k(\mathcal{K}) = \ker(\mathbf{M}_{\partial_k})$ and $B_k(\mathcal{K}) = \mathrm{im}(\mathbf{M}_{\partial_{k+1}})$ are computed. This is achieved by transforming the boundary operator matrices to the so-called *column echelon form* [12] from which one can retrieve the basis of $Z_k(\mathcal{K})$ and $B_k(\mathcal{K})$. The most complicated and time consuming part is to obtain the quotient of those groups $Z_k(\mathcal{K})/B_k(\mathcal{K})$. These computations are implemented in many packages, like [49,50]. During the homology computations *reduction algorithms* can be profitably used in order to decrease the abstract simplicial complex cardinality without changing its topology [12,47,48] before *Smith normal form* [12] computations. In most of the practical cases the reductions are so efficient, that there is no need to use the algebraic Smith normal form diagonalization to obtain the homology generators.

3. Barycentric cell complex

A complex that is more general than the simplicial complex is now introduced. Instead of the notion of simplex, it is based on the notion of cell, that is a general polyhedron homeomorphic[13] to a simplex. For our purposes it is not necessary to define a cell complex in a general way. Only a particular type of cell complex, called the *barycentric complex* $\mathcal{B}$, is defined by means of the *barycentric subdivision* [3] of $\mathcal{K}$. It is fundamental to note that, while the definition of the barycentric complex $\mathcal{B}$ relies on geometry, its incidences can be recovered from the incidences of $\mathcal{K}$ without the geometric construction of the barycentric complex.

The geometric elements of $\mathcal{K}$ are denoted by n for nodes, e for edges, f for triangular faces and v for tetrahedra; whereas the geometric elements of the barycentric complex $\mathcal{B}$ are denoted by $n_{\mathcal{B}}, e_{\mathcal{B}}, f_{\mathcal{B}}$ and $v_{\mathcal{B}}$, respectively.

The nodes of the barycentric complex are defined as the centers of mass of the tetrahedra of $\mathcal{K}$.

Consider two tetrahedra sharing a common triangular face f (2-simplex), Fig. 7a. The edges of the barycentric complex $e_{\mathcal{B}}$ are formed as in Fig. 7a by the union of two segments that join the two centers of mass of the two contiguous tetrahedra with the center of mass of the triangle. The 2-simplex f and the edge of the barycentric complex $e_{\mathcal{B}}$, considered as the union of the two segments, are in one-to-one correspondence. If the triangle is on the boundary of the domain, $e_{\mathcal{B}}$ is composed by just a single segment.

Consider the cluster of tetrahedra constructed by a given edge e. A face of the barycentric complex can be constructed as the union of quadrangles, as in Fig. 7b, one for each tetrahedron T in the cluster. Each quadrangle has one corner in the middle point of the edge, one in the center of mass of T, the other two in the center of mass of the two triangles in the boundary of T that have the edge e in their boundaries. So a one-to-one correspondence between an edge e (1-simplex) and a face of the barycentric complex $f_{\mathcal{B}}$ considered as the union of the quadrangles is evident.

[13] From a topological viewpoint they are equivalent.

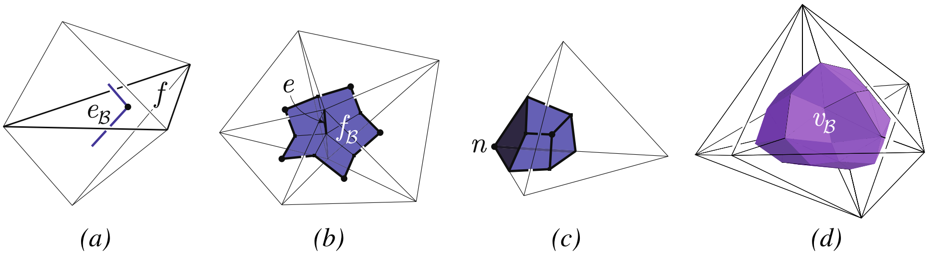

Fig. 7. (a) A face f of the simplicial complex and its one-to-one edge of the barycentric complex $e_\mathscr{B}$; (b) an edge e of the simplicial complex and its one-to-one face of the barycentric complex $f_\mathscr{B}$; (c–d) One-to-one correspondence between a node n of the simplicial complex and the volume of the barycentric complex $v_\mathscr{B}$.

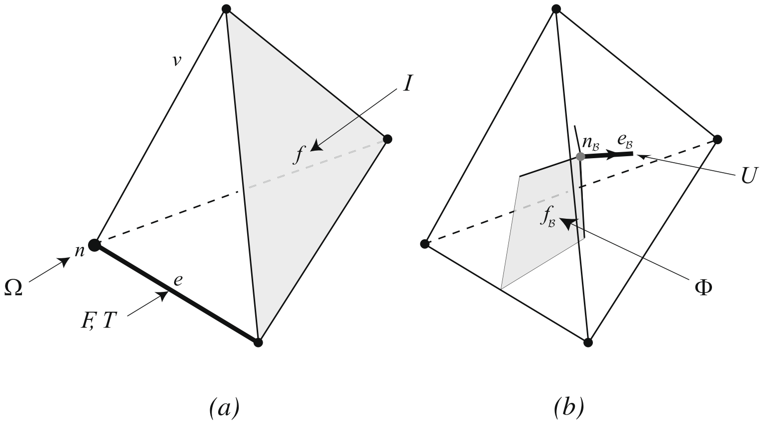

Fig. 8. Association of the integral variables and the geometric entities of $\mathscr{K}$ (a) and $\mathscr{B}$ (b).

Finally consider the cluster of tetrahedra that shares a common node n (0-simplex), Fig. 7d. For every tetrahedron in the cluster an hexahedron is formed, as in Fig. 7c. Considering again all the tetrahedra in the cluster and adding all the respective hexahedra, a solid having as boundary the faces highlighted in Fig. 7d is obtained. Again, a one-to-one correspondence between a node n and a volume of the barycentric complex $v_\mathscr{B}$ is evident.

3.1. $\mathscr{K} - \mathscr{B}$ duality

As noticed in the last paragraphs, another kind of duality with respect to the previously defined chain–cochain duality arise. In fact, every p-simplex of $\mathscr{K}$ is in one-to-one correspondence with one (3-p)-cell of $\mathscr{B}$.[14] This duality can be formalized as

$$C_p(\mathscr{K}) \cong C_{3-p}(\mathscr{B}). \tag{3}$$

The (1) and (3) yield the following isomorphism:

$$C^p(\mathscr{K}) \cong C_{3-p}(\mathscr{B}), \tag{4}$$

which allows to construct the incidence matrices of $\mathscr{B}$ starting from the incidence matrices of $\mathscr{K}$.

Example 7. $\mathbf{G}$ is the incidence matrix between e and n of $\mathscr{K}$. From the isomorphism (4), $\mathbf{G}$ can be reinterpreted on $\mathscr{B}$ as the 3-boundary matrix $\widetilde{\mathbf{D}}^T$. Thus $\widetilde{\mathbf{D}} = \mathbf{G}^T$ is the incidence matrix between $v_\mathscr{B}$ and $f_\mathscr{B}$ of $\mathscr{B}$.[15]

4. Maxwell's balance equations and the *T–Ω* geometric formulation

4.1. Degrees of freedom

The integrals of the electromagnetic field quantities with respect to the oriented geometric elements of the pair of complexes $\mathscr{K}, \mathscr{B}$, are referred to Degrees of Freedom[16] (DoFs) and the arrays they form will be denoted in boldface type. Each entry of a DoFs array is indexed over the corresponding geometric element and the row of array $\mathbf{x}$ relative to the geometric element k will be denoted by $(\mathbf{x})_k$. According to the Tonti's classification of variables [2–4], there is a unique association between every physical variable and the corresponding oriented[17] geometric element, see Fig. 8. In order to formulate an eddy-current problem using the T–Ω formulation, the following DoFs arrays are introduced:

- $\mathbf{\Phi}$ the array of magnetic induction fluxes associated with $f_\mathscr{B} \in D$;
- $\mathbf{F}$ the array of magnetomotive forces (m.m.f.s) associated with $e \in D$;
- $\mathbf{I}$ the array of electric currents associated with $f \in D_c$;
- $\mathbf{U}$ the array of e.m.f.s on edges $e_\mathscr{B} \in D$.

4.2. The *T–Ω* geometric formulation

The algebraic equations governing the discrete eddy-currents problem in the framework of the Discrete Geometric Approach

[14] This is the reason why $\mathscr{B}$ is usually called *dual complex*.

[15] Usually $-\mathbf{G}^T$ is considered as the incidence matrices between $v_\mathscr{B}$ and $f_\mathscr{B}$ of $\mathscr{B}$. The minus sign comes from an assumption that a node is oriented as a sink, whereas the boundary of a volume is oriented by outward normal.

[16] These are just elementary cochains with values in $\mathbb{R}$.

[17] Tonti, in his classification of physical variables, distinguish between inner or outer orientation. For further details see [2–4].

[40] are now recalled. The linearity of the media is assumed, together with a permeability μ_0 in D and a resistivity ρ in D_c. The following DoFs arrays are also introduced:

- $\mathbf{T}$ the array of the circulations of the electric vector potential T along $e \in D_c$.
- $\mathbf{\Omega}$ the array of magnetic scalar potential Ω associated to the nodes $n \in D$.

Using the incidence matrices, Maxwell's laws can be written *exactly* as balance equations between Dofs arrays as

$$\mathbf{G}^T\mathbf{\Phi} = 0 \text{ (a)}, \quad \mathbf{C}^T\mathbf{U} = -i\omega\mathbf{\Phi} \text{ (b)}, \tag{5}$$

where (5a) is the Gauss' magnetic law at discrete level and (5b) is the Faraday's Law. The array $\mathbf{T}$ is defined such that the Ampere's balance law

$$(\mathbf{CF})_f = (\mathbf{I})_f \quad \forall f \in D_c, \tag{6}$$

holds

$$\begin{aligned} (\mathbf{F})_e &= (\mathbf{G\Omega})_e + (\mathbf{T})_e \quad \forall e \in D_c - \partial D_c, \\ (\mathbf{F})_e &= (\mathbf{G\Omega})_e \quad \forall e \in D_a. \end{aligned} \tag{7}$$

In fact one has $\mathbf{C}_c\mathbf{T} = \mathbf{I}$, where $\mathbf{C}_c$ is the sub-matrix of $\mathbf{C}$ relative to the faces and edges in D_c. This implies that the continuity law $\mathbf{D}_c\mathbf{I} = 0$ is identically satisfied,[18] where $\mathbf{D}_c$ is the sub-matrix of $\mathbf{D}$ relative to the volumes and faces in D_c. The interface conditions that avoid the current flow outside the region D_c are already taken into account by considering $T = 0$, $\forall e \in \partial D_c$. In this way, in fact, $(\mathbf{I})_f = 0$, $\forall f \in \partial D_c$ holds.

The discrete counterpart of the constitutive laws can be written as

$$\mathbf{\Phi} = \boldsymbol{\mu}\mathbf{F} \quad \text{in } D\text{(a)}, \quad \mathbf{U}_c = \boldsymbol{\rho}\mathbf{I} \quad \text{in } D_c\text{(b)}, \tag{8}$$

where $\mathbf{U}_c$ is the sub-array of $\mathbf{U}$ relative to edges $e_{\mathscr{B}} \in D_c$. The square matrix $\boldsymbol{\rho}$ ($\dim(\boldsymbol{\rho}) = N_{f_c}, N_{f_c}$ being the number of faces of $\mathscr{K}$ in D_c) is the resistivity matrix such that (8b) holds exactly at least for an element-wise *uniform* current density field J and electric field E in each tetrahedron and it is the approximate discrete counterpart of the constitutive relation $E = \rho J$ at continuous level, ρ being the resistivity assumed element-wise a constant. The square matrix $\boldsymbol{\mu}$ ($\dim(\boldsymbol{\mu}) = N_e, N_e$ being the number of edges in D) is the permeance matrix such that (8a) holds exactly at least for an element-wise *uniform* magnetic field H and induction field B in each tetrahedron and it is the approximate discrete counterpart of the constitutive relation $B = \mu H$ at continuous level, μ being the permeability assumed element-wise a constant.

Classical ways to construct the constitutive matrices $\boldsymbol{\rho}$ and $\boldsymbol{\mu}$ are the Discrete Hodge technique based on Whitney's maps, described for example in [44], or the so-called Galerkin Hodge [45] that produces the same matrix as the Finite Element Method (FEM) with first order edge element basis functions. In this paper, another original solution that use the edge and face vector basis functions defined in [46] is used. These basis functions assure that symmetry, positive-definiteness and consistency[19] properties are satisfied for both the constitutive matrices $\boldsymbol{\rho}$ and $\boldsymbol{\mu}$.

The sources of the problem are enforced in a sub-region D_s of D_a, where the m.m.f. is obtained by

$$(\mathbf{F})_e = (\mathbf{G\Omega})_e + (\mathbf{T}_s)_e \quad \forall e \in D_s, \tag{9}$$

where $\mathbf{T}_s$ is prescribed value of the electric vector potential calculated in such a way that $\mathbf{C}_s\mathbf{T}_s = \mathbf{I}_s$, where $\mathbf{C}_s$ is the sub-matrix of $\mathbf{C}$ relative to the faces and edges in D_s and $\mathbf{I}_s$ is the array containing the impressed currents on $f \in D_s$. To compute the array $\mathbf{T}_s$ form $\mathbf{I}_s$, the iterative technique described in [43] can be used.

By substituting (8a), (8b) and (7) in (5a), the algebraic equations corresponding to the nodes in D are obtained. By substituting (8a), (8b) and (7) in (5b) the algebraic equations corresponding to edges in D_c are derived. The final algebraic system, having $\mathbf{T}$ and $\mathbf{\Omega}$ as unknowns DoFs arrays, can be written as

$$\begin{aligned} &(\mathbf{G}^T\boldsymbol{\mu}\mathbf{G\Omega})_n = \mathbf{0} \quad \forall n \in D_a - D_s, \\ &(\mathbf{G}^T\boldsymbol{\mu}\mathbf{G\Omega})_n = -(\mathbf{G}_s^T\boldsymbol{\mu}_s\mathbf{T}_s)_n \quad \forall n \in D_s, \\ &(\mathbf{G}^T\boldsymbol{\mu}\mathbf{G\Omega})_n + (\mathbf{G}_c^T\boldsymbol{\mu}_c\mathbf{T})_n = \mathbf{0} \quad \forall n \in D_c, \\ &(\mathbf{C}_c^T\boldsymbol{\rho}\mathbf{C}_c\mathbf{T})_e + i\omega(\boldsymbol{\mu}_c(\mathbf{T} + \mathbf{G}_c\mathbf{\Omega}_c))_e = \mathbf{0} \quad \forall e \in D_c. \end{aligned} \tag{10}$$

where the subscripts c and s represent the corresponding sub-array or sub-matrix relative to geometric elements, respectively, in D_c and D_s. The system (10) is singular and, to solve it, a conjugate gradient method without gauge condition is used.

5. From cohomology to thick cuts

In Section 4 the Ampère's balance law was introduced in (6) as a relation imposed between the current associated with a face $f \in D_c$ and the m.m.f.s associated with its boundary edges: $(\mathbf{CF})_f = (\mathbf{I})_f$, $\forall f \in D_c$.

Ampère's balance law can be written also on an arbitrary 1-cycle $c_1 \in Z_1(\mathscr{K}_{D_a})$, where $\mathscr{K}_{D_a}$ is the sub-complex of $\mathscr{K}$ restricted to the non-conducting domain D_a, see Fig. 9. Chosen an orientation for c_1, Ampère's balance law enforce the m.m.f. F_{c_1} evaluated on c_1 to match with the *linked current* I_{c_1}. A 2-chain $S \in \mathscr{K}$ is defined such that has c_1 as boundary, $\partial S = c_1$. Moreover is required that the set of faces, denoted as $|S|$, having non-zero coefficients in the 2-chain S constitute a (orientable) surface. Then the linked current can be obtained by the sum with incidence of the currents crossing S

$$I_{c_1} = \sum_{f_i \in (|S| \cap D_c)} s_{f_i}(\mathbf{I})_{f_i}, \tag{11}$$

where s_{f_i} is the incidence between the orientation of the surface S, inherited from the orientation of its boundary c_1 through the right-hand rule, and the orientation of the face f_i. I_{c_1} does not depend on the choice of S: the current verifies the continuity law $\mathbf{D}_c\mathbf{I} = 0$, thus the flux through all (orientable) surfaces having the same boundary is the same.

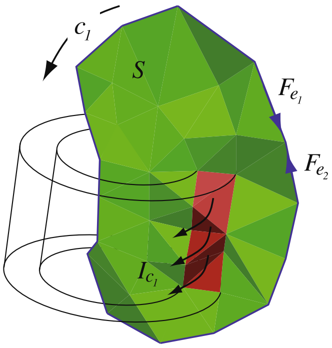

Fig. 9. Consider a conductive region D_c consisting of a circular coil, outlined in the figure. The non-conducting region D_a is assumed to be the complement of D_c with respect to a bigger box, not represented in Figure, which surrounds all D_c. The 1-cycle $c_1 \in Z_1(\mathscr{K}_{D_a})$ (that is non-bounding in D_a) and the 2-chain $S \in C_2(\mathscr{K})$ are shown in the picture. Orienting c_1 as in the Figure, the incidence relative for example to the edges e_1 and e_2 will be, respectively, $s_{e_1} = -1$ and $s_{e_2} = 1$.

[18] Since the coboundary of a coboundary **DC** is identically zero.

[19] A precise definition of the notion of consistency for constitutive matrices is given in [7].

F_{c_1} is obtained as the sum of the contributions due to all m.m.f.s $(\mathbf{F})_e, e \in c_1$, multiplied by the incidence s_{e_i} between the orientation of the edge e_i and the orientation chosen for the cycle c_1. Thanks to the Eq. (7) relative to edges $e \in D_a$, one has

$$F_{c_1} = \sum_{e_i \in c_1} s_{e_i} (\mathbf{F})_{e_i} = \sum_{e_i \in c_1} s_{e_i} (\mathbf{G\Omega})_{e_i}. \tag{12}$$

Since c_1 is a cycle, it is easy to verify that in the sum each node will be counted two times with opposite signs, so F_{c_1} will be always zero for every cycle c_1. This contradicts the Ampère's balance law, since the linked current I_{c_1} is in general non-zero.

Let us consider a cycle $c_2 \in \partial D_c$ that encircle a conductor's section. One can deduce the same inconsistency from the Ampère's balance law applied on c_2, in fact the sum of the T over the cycle c_2 will be always zero. It again follows that this would force to zero the net current on every conductor's section.

This result is not surprising since it is well known that, when homologically non-trivial conductive regions are present, the m.m.f. F_c evaluated along some 1-cycle $c \in Z_1(\mathscr{K}_{D_a})$ cannot be described completely using only a magnetic scalar potential. In particular, the m.m.f.s that cannot be represented by a scalar potential are by Definition 11 the one that are non-zero in the 1st cohomology group $H^1(\mathscr{K}_{D_a})$.[20] These considerations and the fact that Ampère's law has to be verified implicitly in D_a, being not imposed explicitly in (10), motivate the introduction of the so-called *loop fields*.

Definition 12. A *loop field* is a cohomology generator belonging to a basis of $H^1(\mathscr{K}_{D_a})$. Each loop field generator is required to verify Ampère's balance law for every 1-cycle $c \in Z_1(\mathscr{K}_{D_a})$. The loop field will be denoted as the array $\mathbf{T}_0^i$, where each integer $(\mathbf{T}_0^i)_e$, with $e \in D_a$, is the value of the cocycle relative to the 1-simplex e.

The set of loop fields can be constructed in such a way that they correspond naturally to a fixed basis of homology $H_1(\mathscr{K}_{D_a})$ generators.[21] In this case for each loop field $\mathbf{T}_0^i$ the sum with incidence of the $(\mathbf{T}_0^i)_e$ will be the unit current only when evaluated on exactly one $H_1(\mathscr{K}_{D_a})$ homology generator; for the remaining $H_1(\mathscr{K}_{D_a})$ generators this sum will be zero.

Remark. The sums of a loop field evaluated on two cycles, c_1, c_2, which differ by a boundary are equal; The sum around a trivial cycle is zero. This statements will be formally demonstrated in Section 8.

The loop field definition, however, does not imply, that the sum is the unit current only on the cycles homologous to the corresponding $H_1(\mathscr{K}_{D_a})$ generator. Let us take for instance a double torus complement, see Fig. 6. Suppose that the generators of $H_1(\mathscr{K}_{D_a})$ are p_5 and p_7. Once one take a loop field which enforce 1 sum on p_5 and 0 on p_7, it will also enforce 1 on $p_5 + p_7$ which is not homologous to p_5.

We also note that the loop fields are constructed in such a way that every non-bounding cycle crosses at least one edge of at least one loop field.

Using the loop fields, the m.m.f.s associated with an edge $e \in D_a$ can be expressed by

$$(\mathbf{F})_e = (\mathbf{G\Omega})_e + \sum_{j=1}^{\beta_1} i_j (\mathbf{T}_0^j)_e, \tag{13}$$

where i_j is the linked current (which can be known or can be an additional unknown of the problem) and β_1 is the number of cohomology generators.

At this point one would like to retain the speed of the scalar potential formulation limiting the support of the unknown circulations of the electric vector potential T, since, as will be described in Section 6, a bigger support increases the fill-in of the matrix representing the linear system of equations. The thick cuts can informally be presented as an attempt to reduce the support of the electric vector potential.

A *thick cut* can be defined as a "compact" *i*th loop field, such that the dual faces $f_{\mathscr{B}}$ – dual to the set of edges $\mathscr{T}_i$ having a non-zero value in $\mathbf{T}_0^i$ – form on the dual complex an orientable[22] surface Σ_i.

For example, in Fig. 10a it is depicted a polyhedron representing the boundary ∂D_c of a torus. The set of edges $\mathscr{T}_1$ that constitute a possible thick cut are represented by thick edges. One can easily see in Fig. 10b that the union of all the faces $f_{\mathscr{B}}$ dual to edges $e \in \mathscr{T}_1$ form a surface in the barycentric complex.

We note that using thick cuts does not necessarily guarantee that the support is minimized with respect to a general loop field. The surface in fact can fold a big number of times resulting in a large number of thick cut edges. Nevertheless, thick cuts have also a pedagogical advantage, yielding visible the surface used for linked flux evaluation.

6. From thick cuts to non-local Faraday's equations

The domain of definition of the circulations T of the electric vector potential, the edges $e \in D_c$, is extended to the thick cuts edges $e_t \in \{\mathscr{T}_i\}_{i=1}^{\beta_1}$. Since additional unknown DoFs $(\mathbf{T})_{e_t}$, with $e_t \in \{\mathscr{T}_i\}_{i=1}^{\beta_1}$, are introduced with respect to (10), to close the new linear system of equations, additional equations are needed.

A T–Ω formulation suitable for non-simply-connected conductive regions was presented in [41]. The additional equations introduced in [41] have to enforce the sum of the circulations of the electric vector potential along all homologous paths in $\mathscr{K}_{D_a}$ to have the same value. After adding these equations, the linear system of equations becomes non-symmetric.

We note that these equations – and their corresponding new unknowns $(\mathbf{T})_{e_t}, \forall e_t \in \{\mathscr{T}_i\}_{i=1}^{\beta_1}$ – are not explicitly needed, since these constraints are implicitly taken into account when the loop fields are constructed. Thus, in this paper, a new formulation is proposed that uses a reduced number of unknowns and yields a symmetric linear system of equations. Since Ampère's balance law holds in all D, the unknowns that have to be added are just the linked currents i_j introduced in (13), with $j = 1, \ldots, \beta_1$. In fact, once the linked currents i_j are known, the total m.m.f. associated to an edge $e \in D_a$ can be calculated in a post-processing stage using (13). Thus a corresponding number β_1 of new equations have then to be added into the previously introduced linear system of equations (10), suitable only with simply-connected conducting domains: these are the *non-local* Faraday's equations that are described in the following.

6.1. Non-local Faraday's equations

For sake of clarity, consider that only one thick cut is present and it is not self-intersecting, like the one presented in Fig. 10a.

The local Faraday's law (5b) has to be written also for the face $f_{\mathscr{B}}$ in a one-to-one with edge $e \in \mathscr{T}_1$, as already done in the last section for every face $f_{\mathscr{B}}$ in D_c:

$$(\mathbf{C}^T\mathbf{U})_{f_{\mathscr{B}}} + i\omega(\mathbf{\Phi})_{f_{\mathscr{B}}} = \mathbf{0} \quad \forall e \in \mathscr{T}_1.^{23} \tag{14}$$

[20] Due to [19], there is no gain of information in the transition from integer to real coefficients when computing (co)homology.

[21] This reflects the duality $H_1(\mathscr{K}_{D_a}) \simeq H^1(\mathscr{K}_{D_a})$.

[22] It can happen that such a surface is self-intersecting.

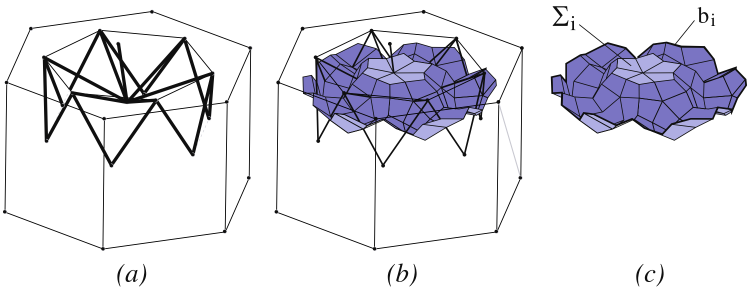

Fig. 10. (a) A polyhedron representing the boundary ∂D_c of a torus and a set of thick cut edges $e \in \mathscr{T}_1$; (b) the cut surface is obtained as the dual faces in the barycentric complex with respect to the thick cut edges. (c) The surface Σ_i and its boundary b_i, composed by edges $e_{\mathscr{B}} \in \mathscr{B}$, are shown.

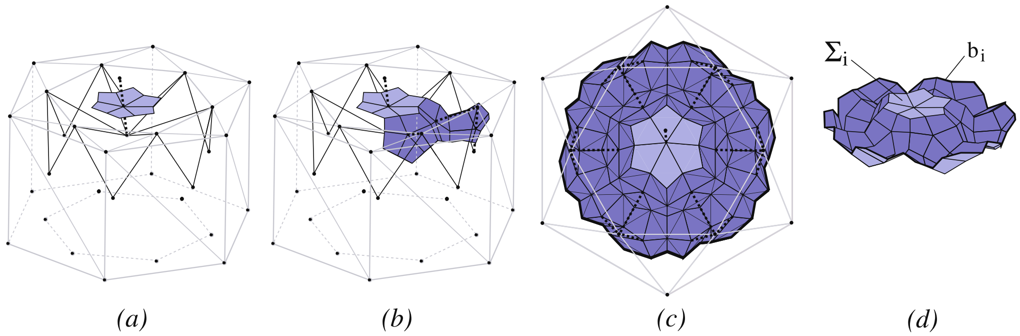

Fig. 11. (a) A starting edge and its one-to-one face $f_{\mathscr{B}}$; (b) another surface where the Faraday's law can be written; (c) from an top view, it is possible to see clearly that the cut surface enters the conductor and its boundary b_i in entirely contained in it; (d) the boundary of Σ_i is the line b_i composed by edges $e_{\mathscr{B}} \in \mathscr{B}$.

[23]These equations cannot be written "locally" for the edges $e \in \mathscr{T}_1$, because in the non-conductive domain $\mathbf{U}$ cannot be expressed in terms of the DoFs $\mathbf{T}$ and $\mathbf{\Omega}$.

Consider the balance Faraday's law (5b) enforced on a face $f_{\mathscr{B}}$ (dual to an edge $e \in \mathscr{T}_1$, see Fig. 11a) and its boundary. One can, as in Fig. 11b, join other neighboring faces to $f_{\mathscr{B}}$ (dual to edges $e \in \mathscr{T}_1$) and write again the Faraday's equation involving the new surface and its boundary. When all faces $f_{\mathscr{B}}$ dual to edges $e \in \mathscr{T}_1$ are joined, as one can see in Fig. 11c, the boundary of the resulting surface is entirely contained inside the conductor, thus $\mathbf{U}$ and $\mathbf{T}$ thanks to (7) and (8b) can be related as $\mathbf{U}_c = \boldsymbol{\rho}\mathbf{C}_c\mathbf{T}$.

The process corresponds to the sum of the local equations and gives rise to the so-called "non-local" Faraday's equation[24]

$$\sum_{e_{\mathscr{B}} \in b_i} s^i_{e_{\mathscr{B}}} (\mathbf{U})_{e_{\mathscr{B}}} = -i\omega \sum_{f_{\mathscr{B}} \in \Sigma_i} s^i_{f_{\mathscr{B}}} (\mathbf{\Phi})_{f_{\mathscr{B}}}, \quad i = 1, \ldots, \beta_1, \tag{15}$$

where $s^i_{f_{\mathscr{B}}}$ is the incidence between a randomly chosen orientation of the cut surface Σ_i and the orientation of the face $f_{\mathscr{B}}$ and $s^i_{e_{\mathscr{B}}}$ is the incidence between the orientation of b_i, inherited through the right-hand rule from the orientation of Σ_i, and the orientation of the edge $e_{\mathscr{B}}$.

Now, the construction of a non-local equation associated to a general loop field is addressed. Let us suppose that non-local equation relative to the ith loop field is considered. Eq. (7), valid only for a simply-connected conducting region D_c, is modified by adding the loop field contribution

$$\begin{aligned} (\mathbf{F})_e &= (\mathbf{G\Omega})_e + (\mathbf{T})_e \quad \forall e \in D_c - \partial D_c, \\ (\mathbf{F})_e &= (\mathbf{G\Omega})_e + \sum_{j=1}^{\beta_1} i_j (\mathbf{T}_0^j)_e, \quad \forall e \in D_a. \end{aligned} \tag{16}$$

The non-local equation is formed as the sum of the local equations belonging to edges $e \in \mathscr{T}_i$ using the coefficients $(\mathbf{T}_0^i)_e$

$$\sum_{e \in \mathscr{T}_i \cap D_c} (\mathbf{T}_0^i)_e (\mathbf{C}_c^T \boldsymbol{\rho} \mathbf{C}_c \mathbf{F}_c)_e + i\omega \sum_{e \in \mathscr{T}_i} (\mathbf{T}_0^i)_e (\boldsymbol{\mu}\mathbf{F})_{f_{\mathscr{B}}} = 0, \quad i = 1, \ldots, \beta_1, \tag{17}$$

where $\mathbf{F}_c$ is the sub-array of $\mathbf{F}$ associated with edges $e \in D_c$. For each edge $e, (\mathbf{F}_c)_e$ and $(\mathbf{F})_e$ can be expressed in terms of the unknown DoFs Ω, T and i by (16).

Defining $\mathbf{C}_a$ as the sub-matrix of $\mathbf{C}$ relative at the entities belonging to $D_a, (\mathbf{C}_a\mathbf{T}_0^j)_f$ is always zero over every face f and for every index j, thus $(\mathbf{CF})_f = (\mathbf{C}_a\mathbf{T}_0^j)_f = 0 = (\mathbf{I})_f, \ \forall f \in \partial D_c$, holds.

7. Automatic generation of cuts

Now that the definition and the requirements for the thick cuts are given, an algorithm for computing them will be provided. In general it is very difficult to define the thick cut edges "by hand". For non-trivial geometries with knots or many links even imagine such cuts can easily become a challenge. This is the reason why efficient, general and reliable algorithm to find cuts is essential. The algorithm should run on a mesh formed by at least hundreds of thousands of tetrahedra in a reasonable computational time.

Several algorithms to automatically generate cuts for the magnetic scalar potential have been proposed in the literature. Most of them, however, compute the so-called *thin cut* surfaces (see Appendix B for definition and examples), while a thick cut is needed for the eddy-current computations with geometric formulations.

7.1. Homotopical approaches

A lot of papers have been published dealing with automatic *homotopy-based* cut construction. These papers have their roots in homotopy theory. For an introduction on homotopy theory, homotopy groups and generators consult for example [17,14]. In

[23] This equation has to be written also for the faces $f_{\mathscr{B}}$ dual to the thick cut edges on ∂D_c, since before only the boundary condition $(\mathbf{T})_e = 0$ was imposed.

[24] This name is due to the fact that the balance equation is not enforced on the neighborhood of a geometric entity, but involves all geometric entities belonging to the cut surface Σ_i and its boundary b_i.

these papers the authors aim to construct the homotopy generators (instead of the homology generators) of the given mesh. Unfortunately the homotopy groups are in general intractable. It is a well known fact, that there do not exist any algorithm which is able to check if the simplicial complex is simply-connected for all possible inputs. There are some theoretical algorithms to compute homotopy groups, but no implementation exists since the enormous memory and time complexity required by these algorithms[25] (see for example [36]). Moreover those theoretical algorithms work under assumption that the input simplicial complex is simply-connected, so they are completely useless for our purpose in all practical cases. All those facts bring us the conclusion that it is hardly possible for a homotopy-based algorithm to work in general case, even with simple non-knotted geometries. In fact, the demonstration that homotopy-based algorithms return always a correct set of cuts is not provided in any paper.

7.1.1. Simple reduction

Many authors attempt to compute second homotopy group generators by using some reduction algorithms [31]. Those methods have their roots on the *simple reductions* used in the computational homology [12]. They essentially reduce the region in a continuous manner, such that the relevant homotopy groups remain invariant. Nevertheless, the *simple reductions* can end up in a sets of simplices that are non-reducible but contractible (for example the Bing's house [32]), being not good as a valid cuts.

The method described in [27,33,35,34] uses a modification of the simple reduction method. In this method the maximal connected and simply-connected set, $\mathscr{C}$, is grown in the non-conducting region of the complex $\mathscr{K}$. From the volume in the non-conducting region that does not belongs to the set $\mathscr{C}$, a set of tetrahedra $\mathscr{D}$ is retrieved; $\mathscr{D}$ is considered as the union of the cut surfaces.

The set $\mathscr{D}$ is constructed in a way that the complement of $\mathscr{D}$ with respect to D_a is always simply-connected. This is an unnecessary additional requirement for a thin cut, that generates when one decides to employ the homotopy theory instead of using the homology theory. In fact, there are examples of thin cuts that do not necessarily leave the complement simply-connected,see for example [28–30].

Furthermore, the biggest practical limitation seams to be the case of a problem for which more then one cut is required. Then the thin cuts have to be generated by separating each of them from set $\mathscr{D}$: such a separation procedure is not described in detail in the literature and seems to be highly non-trivial.

7.2. Approaches based on computational homology

In this Section a review of the approaches based on computational homology that has been developed so far is presented.

7.2.1. Computations of second relative homology group generators

All methods that compute a second relative homology group (see Appendix B) on $\mathscr{K}$, $H_2(\mathscr{K}_{D_a}, \partial\mathscr{K}_{D_a})$, [26] are not useful in practice for the computation of thick cuts. The computations of $H_2(\mathscr{K}_{D_a}, \partial\mathscr{K}_{D_a})$ was the path that was originally followed by the authors. This algorithm, however, did not work in practice since very often the thin cuts obtained were self-intersecting.

A thick cut could be constructed by "growing" a layer of edges from one side of an orientable and non-self-intersecting thin cut. If the thin cuts provided are self-intersecting, an algorithm to extract the thick cut from such a thin cut seems to be highly non-trivial in general case, see Example 8.

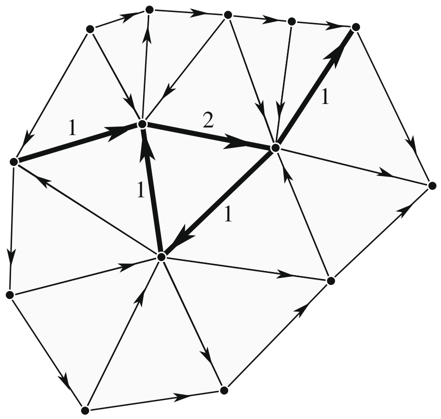

Fig. 12. A portion of a two-dimensional mesh is shown. The edges belonging to a self-intersecting thin cut are highlighted. The integers in the picture denote the coefficients of the non-zero edges in a relative homology generator.

Example 8. When a thin cut is self-intersecting, see Fig. 12 for a two-dimensional example, the coefficient of the chain associated with a face (or an edge in the two-dimensional example) can be an integer greater then 1. As the reader can verify, grow a thick cut on one side of the thin cut in this case is highly non-trivial.

7.2.2. Kotiuga's algorithm

In the classical approach by Kotiuga[20,21,25,24] the following algorithm is used:

(1) Compute the $H^1(\mathscr{B}_{D_a})$ basis on dual complex.
(2) With Finite Element Method (FEM) solve a variational problem for each generator of $H^1(\mathscr{B}_{D_a})$, based on the results of previous step.
(3) The cuts are retrieved by computing the level set of regular value of functions obtained in the step 2.

The presented algorithm, although is known to be the most general, suffers from a serious drawback. The computations of $H^1(\mathscr{B}_{D_a})$ are very time consuming. Kotiuga uses a reduction algorithm based on the spanning tree technique [24] to decrease the size of the complex.

The efficiency of the reduction results strongly dependent on the choice of the tree, but in [24] the construction of an efficient tree is not addressed. A "minimal diameter tree" is proposed in [22]. Even if using this kind of tree, the size of the complex after reduction is big and hardly computable using pure Smith normal form computations. This can be clearly seen from Fig. 6.6 in [24], in which the remaining simplices are 4008 faces and 2888 edges considering a mesh of only 48,463 tetrahedra.

Using a random generated maximal spanning tree, the sizes of the resulting matrices, which need to be diagonalized to Smith normal form, are still really big for real-sized examples. As an example, the sizes of the matrices obtained after the reduction in the benchmarks presented in Section 9, are shown in Table 1.

Computing directly cohomology generators $H^1(\mathscr{K}_{D_a})$ in an efficient way is not possible at present due to the lack of reduction procedures and software designed for cohomology computations.

In our case, the $H_1(\mathscr{K}_{D_a})$ computations are used instead. The algorithm [47], combined with a simple reduction algorithm described in [12] and the recent and powerful *coreduction* algorithm [48], in most of the cases ensures that no Smith diagonalization has to be done at all. What is left after the reductions are, in most practical cases, pure homology generators. This is why in this paper the homology generators are computed and used to generate the loop

[25] Moreover these algorithms require the homology groups to be computed first.

Table 1
Sizes of matrices after reduction based on spanning tree.

Benchmark	# Tetrahedra in $\mathscr{K}$	# Remaining edges × # remaining faces
Coil above a plate	160,576	1668 × 4131
Trefoil knot	199,208	6191 × 14,931
Chain	88,134	2955 × 7117
Borromean rings	254,718	10,964 × 26,428
Inductor	306,085	11,484 × 27,708
Two turn coil	183,051	7225 × 17,391
Folded torus	214,980	7230 × 17,284

fields and the thick cuts. The details of the algorithm are presented in the following section (see Figs. 13–19).

8. Automatic computation of thick cuts based on belted tree

The main steps of the algorithm proposed in this paper are presented below:

1. (1) Find a nice[26] basis for $H_1(\mathscr{K}_{D_a})$ (see Section 8.1).
2. (2) Starting from the homology generators, find a belted tree (see Section 8.2 for its definition and construction).
3. (3) From the belted tree iteratively construct a loop field $\mathbf{T}_0^i$ for each ith generator of $H_1(\mathscr{K}_{D_a})$ (see Section 8.3).
4. (4) Solve a non-physical Poisson problem for each ith loop field (see Section 8.4).
5. (5) Extract the thick cut edges starting from the solution of the non-physical Poisson problems (see Section 8.5).

8.1. Computation of $H_1(\mathscr{K}_{D_a})$

The generators of the $H_1(\mathscr{K}_{D_a})$ are obtained automatically by using [49] software. To use [49] software, the input mesh has to be presented in the form of an abstract simplicial complex, see Appendix A for the algorithm. Then, suitable reduction methods are performed [12,47,48] on the abstract simplicial complex in order to make the structure as small as possible before running the algebraic computations. Once the homology computations are completed, the homology generators are retrieved. For further details on computing homology groups and homology generators consult [12].

The homology generators obtained by the [49] software are very compact and they had never produced intersecting or self-intersecting $H_1(\mathscr{K}_{D_a})$ generators in the tested benchmarks. Those intersections and self-intersections cannot be excluded from the theoretical point of view, but in practice they are extremely rare when using the [49] software. However, if one obtains such a generator, the intersections and self-intersection usually can be removed by local manipulations, such as adding a boundary to some of the generators. In case when local manipulations cannot solve the problem, the homology computation should be restarted with a different reduction algorithm and/or a permutation of indices of the complex. Of course there is no absolute guarantee to get nice generators, but there is extremely low probability not to get them after perturbing the input.

8.2. Construction of a belted tree

Definition 13. A *belted tree* is a sub-graph of $S_1(\mathscr{K})$ for which the only closed cycles are the $H_1(\mathscr{K}_{D_a})$ generators (called *belts*).

The belted tree was introduced in the different context of integral formulations in [23] and for example in [9], without proposing an algorithm to automatically construct it. As far as we are aware of, the automatic construction of a belted tree was addressed only in [38,39], within the different context of integral formulations.[27] As far as we know, our algorithm is the first attempt of automatic construction of belted tree in the context of general eddy-current formulations.

From the previous step of the algorithm, the set of nice $H_1(\mathscr{K}_{D_a})$ generators is obtained. Next, a set $\{E_i\}_{i=1}^{\beta_1}$ of different edges E_i, one per each generator $i = 1, \ldots, \beta_1$, is chosen. The edges from the set $\{E_i\}_{i=1}^{\beta_1}$ are called *belt fasteners*. Starting from the set of homology generator's edges $H_1(\mathscr{K}_{D_a})$ minus all the belt fasteners $\{E_i\}_{i=1}^{\beta_1}$, a maximal possible tree is built. To construct the tree one may use the simple algorithm, that adds an edge to the tree if and only if it does not close a loop. If there are no more such edges in the whole non-conducting region sub-complex $\mathscr{K}_{D_a}$ of a simplicial complex, the algorithm terminates. Once a tree is constructed, all the belt fastener edges $\{E_i\}_{i=1}^{\beta_1}$ are put back into a tree. The tree together with the belt fasteners form the *belted tree*[28] as defined above.

8.3. Construction of the loop fields based on the belted tree

To solve the non-physical problem, the loop fields $\{\mathbf{T}_0^i\}_{i=1}^{\beta_1}$ need to be constructed. To do so, the following algorithm is used, separately for each belted fastener edge $E \in \{E_i\}_{i=1}^{\beta_1}$. First the algorithm is presented, then the demonstration that the output of the algorithm is indeed a loop field will follow.

An iterative technique based on tree–cotree decomposition [29] is used. This technique is described for example in [37], where the belted tree is constructed "by hand".

The aim is to have zero sum around each trivial cycle. It will be demonstrated, that it is enough to enforce $(\mathbf{C}_a\mathbf{T}_0^i)_f = 0,\ \forall f \in D_a$.

To do so, the following algorithm is used, separately for each belt fastener edge $E \in \{E_i\}_{i=1}^{\beta_1}$:

1. (1) Impose 1 on the edge E. For all other belted tree edges, impose 0.
2. (2) Put all triangles in the simplicial complex $\mathscr{K}_{D_a}$ to the list L.
3. (3) While L is not empty do
   - (a) Take a triangle T from the list L.
   - (b) If triangle T has already all three edges imposed, check if the circulation is zero and remove it from the list L.
   - (c) If triangle T has already two edges imposed, then impose the third one in order to get $(\mathbf{C}_a\mathbf{T}_0^i)_T = 0$. Then remove triangle T from the list L.

The strong point of the presented algorithm, that is demonstrated in the following, is that cohomology generators (loop fields) are obtained.

Let us consider the values imposed, by the above algorithm, on the edges of $\mathscr{K}_{D_a}$ as a cochain C. For any cycle $c \in Z_1(\mathscr{K}_{D_a})$ let us

[26] We refer as nice $H_1(\mathscr{K}_{D_a})$ generators, the generators that do not intersect and do not have self-intersections.

[27] When dealing with integral formulations, only the conductive region is meshed. As a consequence, the technique to deal with non-simply-connected conductive regions necessarily has to be strongly different. It is possible to demonstrate that the boundary of thick cuts lying on the conductive regions are the thick cuts suitable for integral formulations.

[28] Of course belted tree is not a tree, since it contains cycles. The cycles contained in the belted tree are just the computed homology generators.

[29] A similar technique based on a maximal spanning tree instead of a belted tree is used by Kotiuga as a reduction method while computing the first cohomology group generators on the dual complex, see points (1)–(4) in Algorithm 6.2, [24], pp. 178–179.

denote the evaluation of a cochain C on a cycle c by a standard scalar product notation $\langle C, c\rangle$ (for a description and properties of the scalar product see [16]). One can easily see that this evaluation is just the sum with incidence of the discrete field C around cycle c. The *generalized Stokes Theorem* $\langle \delta C, d\rangle = \langle C, \partial d\rangle$ for $C \in C^1(\mathscr{K}_{D_a})$ and $d \in C_2(\mathscr{K}_{D_a})$ (see for example [17]) will be used further in this demonstration. Due to the point 3 of the algorithm, $\langle C, \partial T\rangle = 0$ holds for each oriented 2-simplex T. The set of all oriented 2-simplices form a basis of $C_2(\mathscr{K}_{D_a})$. It follows that $0 = \langle C, \partial K\rangle = \langle \delta C, K\rangle$ for every $K \in C_2(\mathscr{K}_{D_a})$, what proofs that C is a cocycle.

To demonstrate that C is not a coboundary, suppose by contrary that it is a coboundary of a cochain $\sum a_{n_i} n_i \in C^0(\mathscr{K}_{D_a})$, thus $\delta(\sum a_{n_i} n_i) = C$. Let us consider a cycle $B \in Z_1(\mathscr{K}_{D_a})$ which is an *active belt*[30] from point (1) of the algorithm and let us denote by E the belt fastener. Let $\partial E = n - m$ for n, m nodes. From the above algorithm it follows $\langle C, B\rangle = \langle C, E\rangle = 1$. The edge E appears only in the coboundary of nodes n and m. Since $\langle C, E\rangle = 1$ it follows, that the values a_n and a_m of a cochain $\sum a_{n_i} n_i$ satisfy $a_n - a_m = 1$. Let $n, k_1, \ldots, k_l, m$ denotes the nodes of a cycle B in the right order (each two neighbor nodes are joined by the edge which is non-zero in a cycle B). Since the edge in B joining n and k_1 is imposed to zero in a cocycle C, the values of coefficients a_n and a_{k_1} in a cochain $\sum a_{n_i} n_i$ are equal. The same argument works for each edge in B joining nodes k_i and k_{i+1} and the edge joining nodes k_l and m. This provides that the values of a_m and a_n are equal integer numbers. But this contradicts the equality $a_n - a_m = 1$. It follows that the cocycle C is not a coboundary.

The above reasoning implies that $C \in H^1(\mathscr{K}_{D_a})$ is a non-zero cohomology class. One can proof that the set of all loop fields obtained from the algorithm form a cohomology basis by using Universal Coefficients Theorem (see for example [16]). The authors decided to omit this proof, since it is long, theoretical and do not give any new light on the practical aspects of the problem.

To ensure the properties after the definition of a loop field, let us show that (a) the evaluation of C on each homologically trivial 1-cycle is 0, (b) the evaluations of C on the cycles in the same homology classes are equal, and (c) the evaluation of C on active belt is equal 1. (c) Follows easily from point 1 of the algorithm. A cycle $d \in Z_1(\mathscr{K}_{D_a})$ is homologically trivial, if there exist a chain $b \in C_2(\mathscr{K}_{D_a})$ such that $\partial b = d$. This provides $\langle C, d\rangle = \langle C, \partial b\rangle = \langle \delta C, b\rangle = 0$, due to the fact that C is cocycle. This proofs (a). For two cycles $[d_1] = [d_2] \in H_1(\mathscr{K}_{D_a})$ in the same homology class, there exist $b \in C_2(\mathscr{K}_{D_a})$ such that $d_1 = d_2 + \partial b$. It follows that $\langle C, d_1\rangle = \langle C, d_2 + \partial b\rangle = \langle C, d_2\rangle + \langle C, \partial b\rangle = \langle C, d_2\rangle$, since $\langle C, \partial b\rangle = 0$, which proofs (b). This demonstrates that the set of cochains obtained by the above algorithm is a loop field.

When the belted tree is constructed starting from nice $H_1(\mathscr{K}_{D_a})$ generators, the triangle T with non-zero $(\mathbf{C}_a \mathbf{T}_0^i)_T$ value at point 3b of the algorithm should never happen. From the presented reasoning it is clear, that when the above algorithm terminates, it returns a valid loop field. It does not terminate only in case of knotted conductors, but one can overcome this problems as described in Section 8.3.1.

At this stage, the loop fields $\{\mathbf{T}_0^i\}_{i=1}^{\beta_1}$ are found. As described in Section 5, the loop fields are enough to solve eddy-current problems with the T–Ω formulation. If one aims to construct a set of thick cuts, a similar technique to the one proposed by Kotiuga can be used, namely the non-physical Poisson problems described in the following have to be solved.

8.3.1. Knotted conductors

When dealing with knotted conductors, the algorithm described in Section 8.3 does not terminate, being impossible to impose all edges in D_a. Anyway, it is still possible to produce a valid loop field also in this case by using a modified version of the algorithm presented in Section 8.3.

The aim of dealing with knotted conductors is to have a general algorithm which produces a valid loop field no matter if a knotted or not knotted conductor is provided as input. Moreover, some applications of knotted conductors come up naturally in the context of force-free magnetic fields, see for example [51,52].

When the algorithm to find a loop field from a belted tree stops, one has to select a random cotree edge E_r which has not already a value imposed. Adding the selected cotree edge to the edges belonging to the belted tree will create some new cycles.

The edges of $H_1(\mathscr{K}_{D_a})$ basis plus the edges of the cycle closed by a cotree edge which does not contain any belt fastener are considered. Since the belted tree is formed by the cycles which are the basis of $H_1(\mathscr{K}_{D_a})$, it is possible to find a combination of these basis cycles which belongs to the same homology class as the cycle consisting of the new cotree edge E_r and some belted tree edges without belt fasteners.

To find the coefficients for the considered edges, one needs usually to solve a linear system of equations, where the unknowns are the coefficients of triangles (the equations are obtained by computing boundaries of those triangles) as well as coefficients on considered edges in the belted tree. All other edges are supposed to have zero coefficients.

The solution of this system will point out the trivial cycle in a belted tree closed by a cotree edge. From the coefficients obtained by solving this linear system of equations, one may retrieve the value to impose on the random cotree edge E_r. Once the cotree edge is imposed, the algorithm continue cycling from the step 3. It should be noted, however, that the system of equation have to be solved rigorously over the integers, which is costly from computational point of view.

8.4. The non-physical Poisson problem

The variables and their association with geometric entities for the non-physical Poisson problem are

- $\mathbf{U}$ the array containing DoFs associated with $e \in D_a$.
- $\mathbf{V}$ the array containing DoFs associated with $n \in D_a$.
- $\mathbf{I}$ the array containing DoFs associated with $f_{\mathscr{B}} \in D_a$.

The variable associated with edge e is expressed by

$$(\mathbf{U})_e = (\mathbf{T}_0^i)_e + (\mathbf{GV})_e. \tag{18}$$

By substituting Eq. (18) and the constitutive relation $\mathbf{I} = \mathbf{SU}$ in the continuity balance law $\mathbf{G}^T\mathbf{I} = 0$ one obtains

$$\mathbf{G}^T\mathbf{SGV} = -\mathbf{G}^T\mathbf{ST}_0^i. \tag{19}$$

The constitutive matrix $\mathbf{S}$ can be constructed in the same way as matrix $\boldsymbol{\mu}$ in Section 4.2, by swapping the material permeability μ with 1.

If more loop fields are present, one solution of the non-physical problem is needed for each ith loop field. In this case the stiffness matrices of these problems are the same, only the right-hand side of the linear system of equations has to be recomputed.

8.5. Thick cut extraction

Once the non-physical Poisson problem is solved, the DoFs $(\mathbf{U})_e$ associated with each edge $e \in D_a$ are calculated using Eq. (18). Based on this DoFs, a *global* nodal variable $(\mathbf{Z})_n$, $\forall n \in D_a$, has to be computed. To do so the following algorithm is used:

[30] By active belt is denoted the belt in which the belt fastener is imposed to 1 by the algorithm.

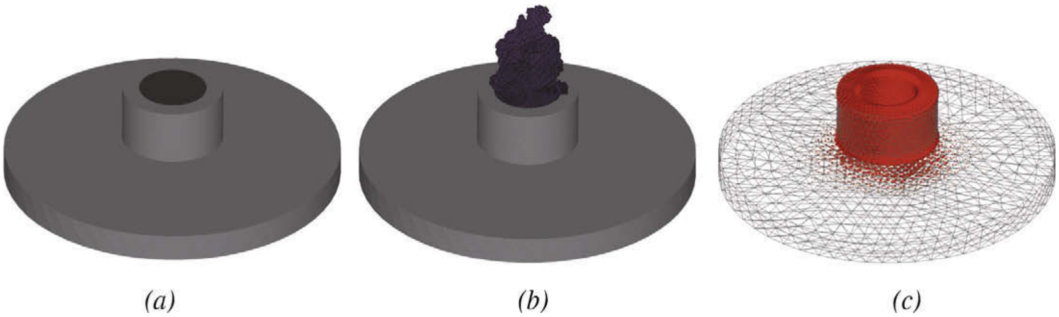

Fig. 13. Coil above a plate benchmark. (a) Geometry; (b) cut surface; (c) current density.

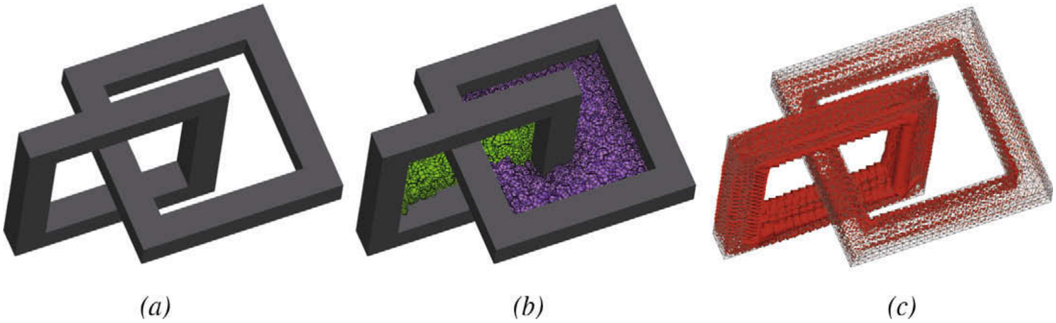

Fig. 14. Chain benchmark. (a) Geometry; (b) cut surface; (c) current density.

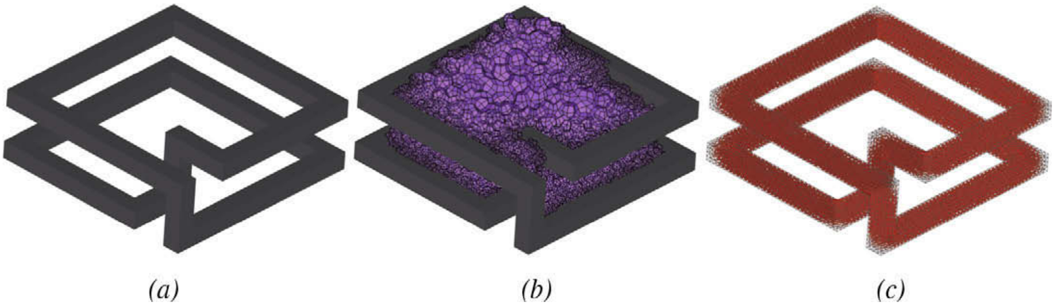

Fig. 15. Two turn coil benchmark. (a) Geometry; (b) cut surface; (c) current density.

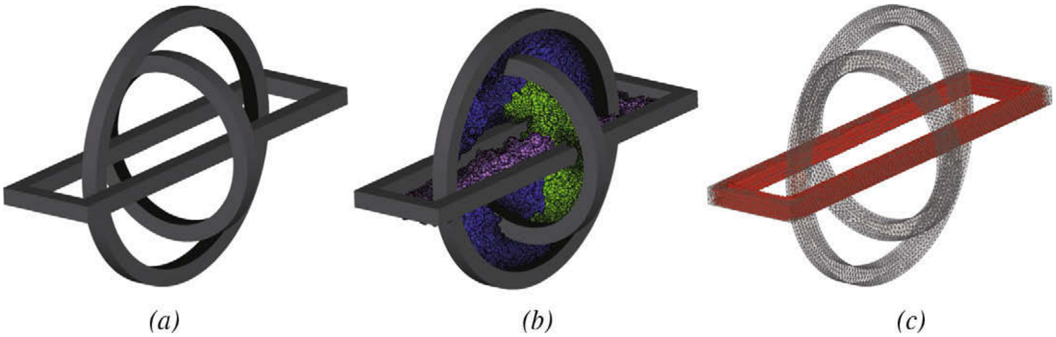

Fig. 16. Borromean rings benchmark. (a) Geometry; (b) cut surface; (c) current density.

```
(1) Take a node $n$ in the $\mathscr{K}_{D_a}$. Impose a 0 value to this node
    $(\mathbf{Z})_n = 0$ and put it into a stack $S$.
(2) While the stack $S$ is non-empty do
    (a) Remove the node $n$ from the top of the stack.
    (b) For each edge $e$ such that $n$ belongs to $e$ do
    (c) If the other node $p$ in the boundary of edge $e$ does not
        have the nodal value $(\mathbf{Z})_p$ already defined, compute it
        with $(\mathbf{Z})_p = (\mathbf{Z})_n + \mathbf{G}(e,p)(\mathbf{U})_e$.
    (d) Put $p$ into the stack $S$.
```

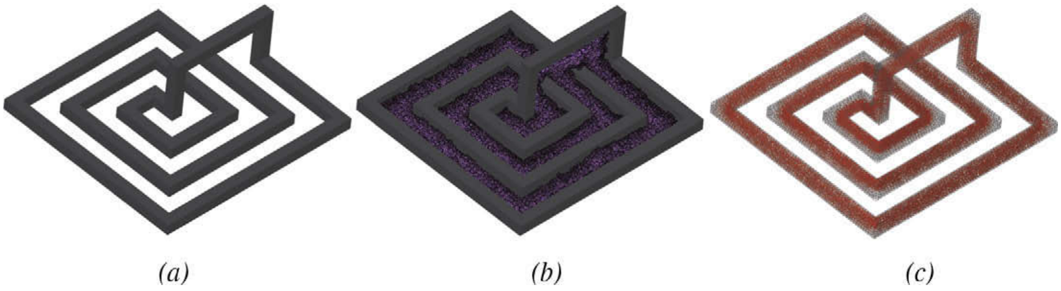

Fig. 17. Inductor benchmark. (a) Geometry; (b) cut surface; (c) current density.

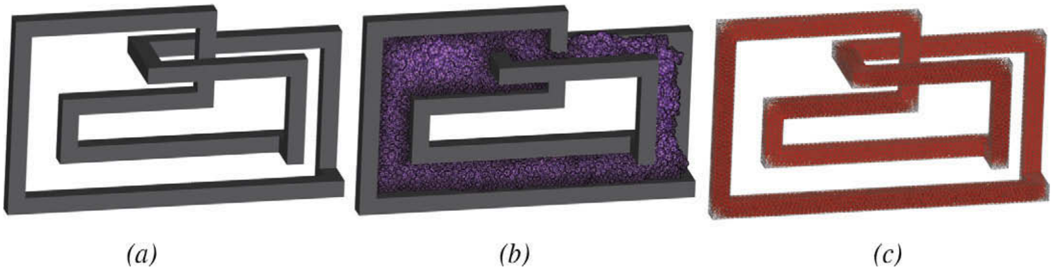

Fig. 18. Folded torus benchmark (inspired by Fig. B in [29]). (a) Geometry; (b) cut surface; (c) current density.

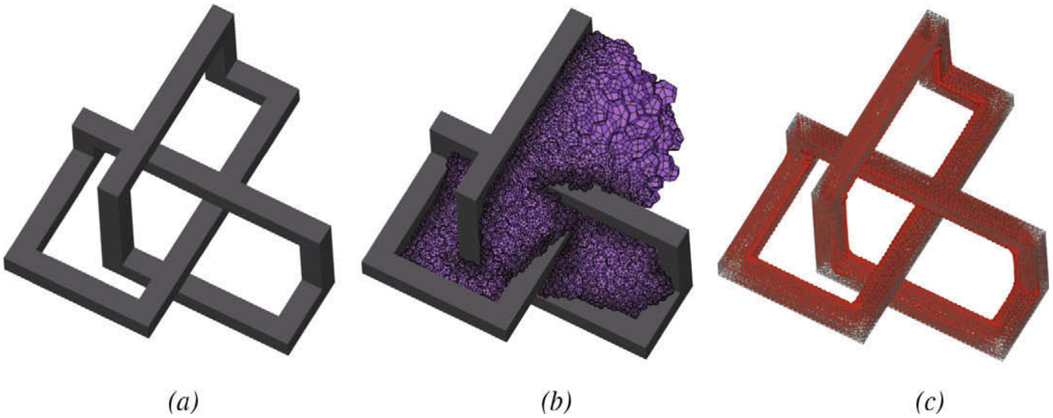

Fig. 19. Trefoil knot benchmark. This result was obtained by means of the modified version of the algorithm described in Section 8.3.1. (a) Geometry; (b) cut surface; (c) current density.

The thick cut edges are the edges in which an integer jump in potential is present.[31]

9. Numerical results

In this section some of the tested benchmark problems are presented. For each benchmark the geometry is shown on the left. On the middle, the resulting thick cut surfaces on $\mathscr{B}$, automatically generated by means of our algorithm, are drawn. On the right, the current density computed by means of the T–Ω geometric formulation using [42] is shown.

10. Conclusions

A T–Ω geometric formulation that yields to a symmetric linear system of equations is presented using tools from algebraic topology, namely homology and cohomology theories. In the paper, this is motivated by the fact that these theories help to gain insight about the so-called thick cuts and the non-local equations. From the pedagogical point of view, the DGA framework results more attractive with respect to classical Finite Element Method (FEM), since the Maxwell's laws are explicitly enforced by means of the coboundary operator.

The algorithms proposed in the literature to automatically compute the cuts are first reviewed. It is shown that most of them, however, construct the thin cuts and not the thick cuts. The algorithm based on the construction of the 2nd relative homology group does not work in practice. In fact the obtained thin cut is often self-intersecting and consequently it is extremely difficult to find an algorithm to produce a thick cut from the obtained thin cut. Homotopy-based methods are not attractive, since homotopy groups are not computable. The algorithm based on the computation of the 1st cohomology generators results to be the most general, while, due to the lack of efficient reduction procedures, it is very time-consuming.

In this paper an original algorithm to directly compute the loop fields and the thick cuts is proposed. It is based on the con-

[31] The same procedure was used by Kotiuga in [24] in a different context of extracting thin cut surface.

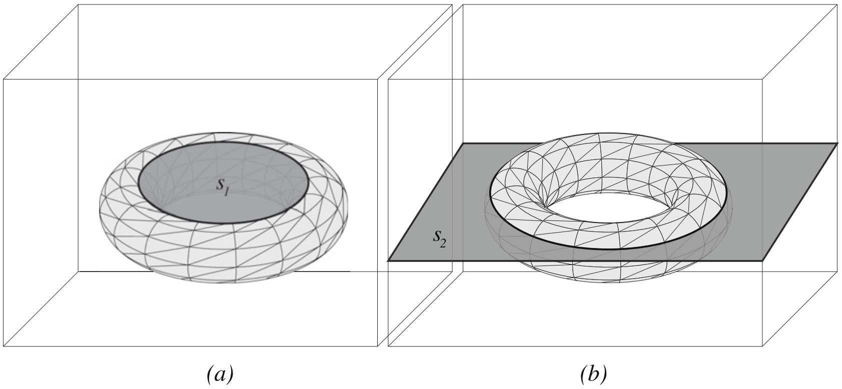

Fig. 20. Two relative homology generators of $H_2(\mathscr{K}_{D_a}, \partial\mathscr{K}_{D_a})$.

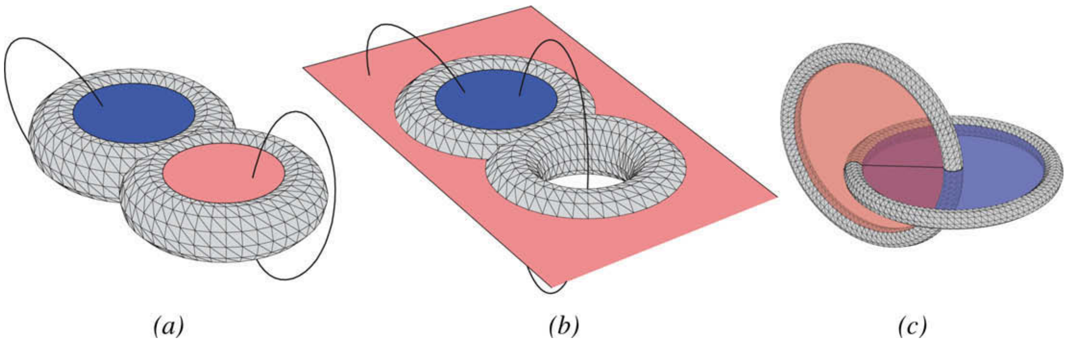

Fig. 21. Examples of cut surfaces.

struction of the so-called belted tree. The belted tree is constructed using a fast computation of the homology generators by means of innovative reduction algorithms. The advantages and limitations of the proposed algorithm are discussed. The presented algorithm was tested on a number of real-sized three-dimensional finite element meshes, showing its utility for practical applications.

Acknowledgements

The authors would like to thank Prof. Robert Kotiuga, Prof. Marian Mrozek and Dr. Saku Suuriniemi for many useful discussions and suggestions.

Appendix A

The algorithm which takes as input the file produced by the Netgen mesh generator (see [10]) and returns the simplicial complex is presented. The first step is to decide on the data structures needed in such an algorithm. A node (zero-dimensional simplex) is represented by an integer number. The number is equal to the number given to this node by the Netgen generator. An edge (one-dimensional simplex) is represented by the pair of nodes which are the endpoints of the given edge. Also triangles (two-dimensional simplices) and tetrahedrons (three-dimensional simplices) are represented by the nodes involved in it. In each presented data structure, all the integers are assumed to be ordered in increasing way. By taking this assumption, the orientation of all simplices in the abstract simplicial complex is fixed.

Consequently, when one restricts the increasing orientation of a simplex to one of its faces, the increasing orientation of the face is obtained. This implies, that in the Definition 8 the orientation of any simplex in the definition of boundary operator will be the same as the fixed orientation of the simplex. It follows, that one does not have to worry about the coefficients coming from the orientation of the simplices, the sign is determined by factor $(-1)^i$ from the equation in Definition 8.

This explanation gives rise to the following formal definition:

```
NODES = list of int;
EDGES = list of [int,int];
TRIANGLES = list of [int,int,int];
TETRAHEDRONS = list of [int,int,int,int];
```

The input set of tetrahedra is obtained from a file generated by the software Netgen. Having defined the basic data structures, the abstract simplicial complex is defined as a structure consisting of all the elements defined above:

```
ABSTRACT SIMPLICIAL COMPLEX
  = {NODES, EDGES, TRIANGLES, TETRAHEDRONS}
```

The algorithm to construct the simplicial complex of the non-conductive region of a mesh is now presented.

```
ComplexGenerator(TETRAHEDRONS tets)
  NODES vert; EDGES ed; TRIANGLES tri; TETRAHEDRONS
tet;
  vert=ed=tri=tet=empty;
  For each tetrahedron T = [v1, v2, v3, v4] from tets do
    for each singleton [a] in T do
      if [a] not exist in vert list then
        Put node [a] into vert list;
    for each doubleton [a,b] in T where a<b do
      if [a,b] not exist in ed then
    Put edge [a,b] into ed list;
for each triple [a,b,c] in T where a<b<c do
  if [a,b,c] not exist in tri then
    Put triangle [a,b,c] into tri list;
Put T into tet list;
```

Appendix B

B.1. Relative homology groups

In Section 2, the definition of the so-called absolute homology group was presented. Also the 2nd *relative homology groups* is useful in computational electromagnetism. Given a sub-complex[32] $\mathscr{A}$ of $\mathscr{K}$, the 2nd relative homology group $H_2(\mathscr{K},\mathscr{A})$ is defined[14] as $H_2(\mathscr{K},\mathscr{A}) = Z_2(\mathscr{K},\mathscr{A})/B_2(\mathscr{K},\mathscr{A})$, where $Z_2(\mathscr{K},\mathscr{A})$ denotes the group of *relative cycles* and $B_2(\mathscr{K},\mathscr{A})$ denotes the group of *relative boundaries*. $z \in C_2(\mathscr{K})$ is a relative cycle modulo $\mathscr{A}(z \in Z_2(\mathscr{K},\mathscr{A}))$ if $\partial z \in C_1(\mathscr{A})$. The $t \in C_2(\mathscr{K})$ is the relative boundary modulo $\mathscr{A}(z \in B_2(\mathscr{K},\mathscr{A}))$ if $t = \partial(x) + y$ for some $x \in C_3(\mathscr{K}), y \in C_2(\mathscr{A})$.

Example 9. Consider the oriented simplicial complex $\mathscr{K}_{D_a}$ of Example 5a. In Fig. 20a and Fig. 20b two different generators s_1 and s_2 of $H_2(\mathscr{K}_{D_a},\partial\mathscr{K}_{D_a})$ (in the same relative homology class) are shown. Such generators are by definition surfaces having all boundaries ∂s_1 and ∂s_2 on $\partial\mathscr{K}_{D_a}$, but that cannot be considered boundaries of any volume entirely contained in $\mathscr{K}_{D_a}$.

B.2. Thin cuts

From the Poincré–Lefschetz duality theorem, one have [15]

$$H^1(\mathscr{K}_{D_a}) \cong H_2(\mathscr{K}_{D_a},\partial\mathscr{K}_{D_a}). \tag{20}$$

This well known duality was used by Kotiuga for the first formal definition of *(thin) cuts* [8]:

Definition 14. A *thin cut* is defined as a 2-chain belonging to an orientable and non-self-intersecting generator of $H_2(\mathscr{K}_{D_a},\partial\mathscr{K}_{D_a})$.

Kotiuga proofed the existence of a set of non-self-intersecting orientable generators of $H_2(\mathscr{K}_{D_a},\partial\mathscr{K}_{D_a})$ [19]. It is easy to show by Example 10c that in general it is impossible to avoid intersections of different cuts.

Example 10. (a) In Fig. 21a and b two valid set of thin cuts for a complex obtained as the complement of a double torus with respect to a surrounding box are shown. Since cuts on the Fig. 21a are homology generators, then by taking a linear combinations of them and adding a boundary one can obtain the cuts in Fig. 21b. This is valid also the other way around. As a different example, considering the complex as the complement of two linked tori with respect to a surrounding box (see Fig. 21c), it is easy to see that in general it is not possible to avoid intersections between different cut surfaces.[33] Moreover in such cases the cut surfaces are non-simply-connected.

References

[1] F.H. Branin Jr., The algebraic-topological basis for network analogies and the vector calculus, in: Proceedings of the Symposium on Generalized Networks, Polytechnic Press, Brooklin, NY, 1966, pp. 453–491.

[2] E. Tonti, On the formal structure of physical theories, Monograph of the Italian National Research Council, 1975.

[3] E. Tonti, Algebraic topology and computational electromagnetism, in: 4th International Workshop on Electric and Magnetic Fields, Marseille (Fr), 12–15 May 1988, pp. 284–294.

[4] E. Tonti, Finite formulation of the electromagnetic field, IEEE Trans. Magn. 38 (2) (2002) 333–336.

[5] T. Weiland, A discretization method for the solution of Maxwell's equations for six-component fields, Electron. Commun. (AEÜ) 31 (3) (1977) 116.

[6] A. Bossavit, How weak is the weak solution in finite elements methods?, IEEE Trans Magn. 34 (5) (1998) 2429–2432.

[7] A. Bossavit, L. Kettunen, Yee-like schemes on staggered cellular grids: a synthesis between FIT and FEM approaches, IEEE Trans. Magn. 36 (4) (2000) 861–867.

[8] P.R. Kotiuga, Hodge decompositions and computational electromagnetics, Ph.D. Thesis, McGill Univ., Montreal, Canada, 1985.

[9] A. Bossavit, Computational Electromagnetism, Academic Press, 1998. ISBN-13 978-0121187101.

[10] J. Schöberl, NETGEN – an advancing front 2D/3D-mesh generator based on abstract rules, Comput. Visual. Sci. 1 (1) (1997) 41–52.

[11] P.L. George, H. Borouchaki, Delaunay Triangulation and Meshing, Editions Hermes, Paris, 1998. ISBN 2-86601-692-0.

[12] T. Kaczynski, K. Mischaikow, M. Mrozek, Computational Homology, Springer-Verlag, New York, 2004.

[13] E. Spanier, Algebraic Topology, McGraw-Hill, New York, 1966.

[14] A. Dold, Lectures on Algebraic Topology, Springer-Verlag, New York, 1980.

[15] J.R. Munkres, Elements of Algebraic Topology, Perseus Books, Cambridge, MA, 1984.

[16] W.S. Massey, Singular Homology Theory, Springer-Verlag, New York, 1980.

[17] A. Hatcher, Algebraic Topology, Cambridge University Press, 2002 (available online).

[18] M.L. Brown, Scalar potentials in multiply connected regions, Int. J. Numer. Meth. Engrg. 20 (1984) 665–680.

[19] P.R. Kotiuga, On making cuts for magnetic scalar potentials in multiply connected regions, J. Appl. Phys. 61 (8) (1987) 3916–3918.

[20] P.R. Kotiuga, Toward an algorithm to make cuts for magnetic scalar potentials in finite element meshes, J. Appl. Phys. 63 (8) (1988) 3357–3359. erratum: 64(8), 4257, 1988.

[21] 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.

[22] A. Murphy, Implementation of a finite element based algorithm to make cuts for magnetic scalar potentials, Masters thesis Dept. of ECS Eng., Boston U., 1991.

[23] L. Kettunen, K. Forsman, A. Bossavit, Formulation of the eddy current problem in multiply connected regions in terms of *h*, Int. J. Numer. Meth. Engrg. 41 (5) (1998) 935–954.

[24] P.W. Gross, P.R. Kotiuga, Electromagnetic Theory and Computation: A Topological Approach, vol. 48, Cambridge University Press, 2004. ISBN 0 521 801605.

[25] P.W. Gross, P.R. Kotiuga, Finite element-based algorithms to make cuts for magnetic scalar potentials: topological constraints and computational complexity, in: F.L. Teixeira, J.A. Kong (Eds.), Geometric Methods for Computational Electromagnetics, Progress in Electromagnetics Research, vol. 32, EMW Publishing, Cambridge MA, 2001, pp. 207–245. ISBN 0966143-6-3.

[26] S. Suuriniemi, Homological computations in electromagnetic modeling, Ph.D. Thesis, Tampere University of Technology, 2004, ISBN 952-15-1237-7.

[27] C.S. Harold, J. Simkin, Cutting multiply connected domains, IEEE Trans. Magn. 21 (6) (1985) 2495–2498.

[28] A. Vourdas, K.J. Binns, Magnetostatics with scalar potentials in multiply connected regions, science, measurement and technology,, IEE Proc. A 136 (2) (1989) 49–54.

[29] A. Vourdas, K.J. Binns, A. Bossavit, Magnetostatics with scalar potentials in multiply connected regions (comments with reply), science, measurement and technology, IEE Proc. A 136 (5) (1989) 260–261.

[30] R. Kotiuga, Magnetostatics with scalar potentials in multiply connected regions, science, measurement and technology, IEE Proc. A 137 (4) (1990) 231–232.

[31] P.J. Leonard, H.C. Lai, R.J. Hill-Cottingham, D. Rodger, Automatic implementation of cuts in multiply connected magnetic scalar region for 3-D eddy current models, IEEE Trans. Magn. 29 (2) (1993) 1368–1371.

[32] R.H. Bing, Some aspects of the topology of 3-manifolds related to the Poincaré Conjecture, in: T.L. Saaty (Ed.), Lectures on Modern Mathematics II, Wiley, 1964, pp. 93–128.

[33] J. Simkin, S.C. Taylor, E.X. Xu, An efficient algorithm for cutting multiply connected regions, IEEE Trans. Magn. 40 (2, Part 2) (2004) 707–709.

[34] Z. Ren, T–Ω formulation for eddy-current problems in multiply connected regions, IEEE Trans. Magn. 38 (2) (2002) 557–560.

[35] P. Dular, Curl-conform source fields in finite element formulations: automatic construction of a reduced form, COMPEL 24 (2) (2005) 364–373.

[36] E.H. Brown Jr., Finite computability of Postnikov complexes, Ann. Math. 65 (1957) 1–20.

[37] F. Henrotte, K. Hameyer, An algorithm to construct the discrete cohomology basis functions required for magnetic scalar potential formulations without cuts, IEEE Trans. Magn. 39 (2003) 1167–1170.

[38] R. Hiptmair, J. Ostrowski, Generators of $H_1(\Gamma_h,\mathbb{Z})$ for triangulated surfaces: construction and classification, SIAM J. Comput. 31 (5) (2002) 1405–1523.

[32] A sub-complex is a subset of a complex that fulfils all the axioms for an abstract simplicial complex.

[33] These results were formally demonstrated by Kotiuga in [20].

[39] F. Rapetti, F. Dubois, A. Bossavit, Discrete vector potentials for nonsimply connected three-dimensional domains, SIAM J. Numer. Anal. 41 (4) (2003) 1505–1527.

[40] F. Trevisan, L. Kettunen, Geometric interpretation of finite dimensional eddy current formulations, Int. J. Numer. Meth. Engrg. 67 (13) (2006) 1888–1908.

[41] R. Specogna, S. Suuriniemi, F. Trevisan, Geometric T–Ω approach to solve eddy-currents coupled to electric circuits, Int. J. Numer. Meth. Engrg. 74 (1) (2008) 101–115.

[42] R. Specogna, F. Trevisan, The Geometric Approach to solve Maxwell's Equations (G.A.M.E.) code, <www.quickgame.org>, copyright 2003–2009.

[43] Y. Le Menach, S. Clenet, F. Piriou, Determination and utilization of the source field in 3D magnetostatic problems, IEEE Trans. Magn. 34 (5) (1998) 2509–2512.

[44] T. Tarhasaari, L. Kettunen, A. Bossavit, Some realizations of a discrete Hodge operator: a reinterpretation of finite element techniques, IEEE Trans. Magn. 35 (1999) 1494–1497.

[45] A. Bossavit, Computational electromagnetism and geometry. (5): the 'Galerkin hodge', J. Jpn. Soc. Appl. Electromagn. Mech. 8 (2) (2000) 203–209.

[46] L. Codecasa, R. Specogna, F. Trevisan, Symmetric positive-definite constitutive matrices for discrete eddy-current problems, IEEE Trans. Magn. 42 (2) (2007) 510–515.

[47] T. Kaczynski, M. Mrozek, M. Ślusarek, Homology computation by reduction of chain complexes, Comput. Math. Appl. 35 (1998) 59–70.

[48] M. Mrozek, B. Batko, Coreduction homology algorithm, Discrete Comput. Geom. (2008), doi:10.1007/s00454-008-9073-y.

[49] Computer Assisted Proofs in Dynamic, capd.wsb-nlu.edu.pl.

[50] Chomp Project, chomp.rutgers.edu.

[51] J.C. Crager, P.R. Kotiuga, Cuts for the magnetic scalar potential in knotted geometries and force-free magnetic fields, IEEE Trans. Magn. 38 (2002) 1309–1312.

[52] P.R. Kotiuga, Topology-based inequalities and inverse problems for near force-free magnetic fields, IEEE Trans. Magn. 40 (2004) 1108–1111.