SIAM J. NUMER. ANAL.
Vol. 48, No. 4, pp. 1601–1624

CRITICAL ANALYSIS OF THE SPANNING TREE TECHNIQUES*

PAWEŁ DŁOTKO† AND RUBEN SPECOGNA‡

Abstract. Two algorithms based upon a tree-cotree decomposition, called in this paper *spanning tree technique* (STT) and *generalized spanning tree technique* (GSTT), have been shown to be useful in computational electromagnetics. The aim of this paper is to give a rigorous description of the GSTT in terms of homology and cohomology theories, together with an analysis of its termination. In particular, the authors aim to show, by concrete counterexamples, that various problems related with both STT and GSTT algorithms exist. The counterexamples clearly demonstrate that the failure of STT and GSTT is not an exceptional event, but something that routinely occurs in practical applications.

Key words. algebraic topology, scalar potential in multiply connected regions, tree-cotree decomposition, belted tree, computational topology, homology theory, cohomology theory, homology and cohomology generators, homology-cohomology duality

AMS subject classifications. 65N30, 78M10, 78M25, 55N99, 55M05, 55N33

DOI. 10.1137/090766334

1. Introduction. The tree-cotree decomposition arises from graph theory and consists in partitioning the edges of a graph into a spanning tree and its complement, referred to as the cotree. The idea of taking advantage of the tree-cotree decomposition is at the root of electric network theory [1], [2]. It has been used, for example, to generate a maximal set of independent Kirchhoff's equations for the network analysis (see, for example, [3]). The connection of electric network analysis to algebraic topology was soon recognized [4], [5], [6], [7], [8], [9], [10]. The tree-cotree decomposition became popular in computational electromagnetics after [11]. It had been widely used as a gauging technique to set well-posed magnetostatic and magneto-quasi-static boundary value problems (BVP). Nowadays, such gauging techniques have lost their importance, since the ungauged formulations were shown to be more effective, improving the condition number of the matrix for the linear system of equations; see, for example, [12].

More recently, two algorithmic techniques based upon the tree-cotree decomposition have been shown to be quite useful in computational electromagnetics. The first one, introduced in [13] (see also [14], [15], [16] and referred to as *spanning tree technique* (STT) in this paper, is commonly employed in order to compute the so-called *generalized source magnetic fields*, needed to enforce the source currents when solving magnetostatic and magneto-quasi-static BVP formulated by using a magnetic scalar potential. In this application, the STT is used to compute a 1-cochain when its coboundary 2-cochain is given as input.

*Received by the editors July 27, 2009; accepted for publication (in revised form) July 16, 2010; published electronically September 7, 2010.
http://www.siam.org/journals/sinum/48-4/76633.html
†Institute of Computer Science, Jagiellonian University, Lojasiewicza 6, 30348 Kraków, Poland (pawel.dlotko@uj.edu.pl). This author's work was partially supported by Polish MNSzW, grant MNiSzW nr N201 037 31/3151.
‡Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica (DIEGM), Università di Udine, Via delle Scienze 208, 33100 Udine, Italy (ruben.specogna@uniud.it).

When solving a magneto-quasi-static BVP by using a scalar potential-based formulation, a basis for the first cohomology group over integers is needed[1] [17], [18], [19]. Hence, over the past twenty years, a considerable effort has been invested by the computational electromagnetics community to develop fast and general algorithms to produce cohomology group generators. The second algorithm analyzed in this paper, referred to as *generalized spanning tree technique* (GSTT), is an attempt to obtain the representatives of the first cohomology group generators when the representatives of the first homology group generators are provided as input. This widely used technique, introduced in [15], is based upon the concept of the so-called *belted tree*, which has been presented in [25] (see also [26], [27], [28], [29]). Nonetheless, in most papers where the GSTT is used, there is no mention on how to automatically and efficiently obtain generators for the first homology group. In [19], homology generators suitable for GSTT are efficiently obtained by effective homology computations based on original reduction techniques [30], [31].

Yet, both STT and GSTT algorithms are considered to be general, and their termination (without returning an error message) is taken for granted in the literature. In this paper, the STT and GSTT algorithms are described in detail in Figures 3.1 and 4.3, respectively, and their termination is analyzed. The main contribution of this paper is to show that both STT and GSTT algorithms exhibit termination problems, which are presented with concrete counterexamples in section 5. The counterexamples clearly show that the failure of both STT and GSTT is not an exceptional event, but something that routinely occurs in practical applications.

The paper is structured as follows. In section 2, to make the paper self-consistent, the relevant concepts of algebraic topology, in particular homology and cohmology theories, are recalled. In sections 2.5 and 2.6, a review about previous results, namely, the absence of torsion and relation between cohomology with integer and real coefficients for simplicial complexes embedded in $\mathbb{R}^3$, are recalled. In section 2.7 some properties of the first cohomology group generators that are extensively used further on in the paper are presented. Section 3 contains a description of the STT. In section 4, the GSTT is introduced in the light of algebraic topology. As far as we are aware, this is the first paper containing a detailed description of the GSTT algorithm together with an analysis of its termination. All of the problems that may occur running the algorithm are easily detected as described in Figure 4.3. Therefore, the algorithm presented in this paper always terminates, and it returns either an error message or a first cohomology group basis of the considered complex. Section 5 contains a selection of counterexamples in which the STT or GSTT fail. Moreover, some conditions are stated for the correct termination of the algorithms, which are left as conjectures. Therefore, the paper intends to give an answer to the open question arisen in [25, p. 238], whether techniques using the belted tree, as the GSTT, are a valid alternative to the direct construction of the first cohomology group basis by means of a cohomology computation. A MATLAB® code which implements the STT and GSTT algorithms is provided to the reader, together with the inputs relative to all counterexamples that are presented in section 5 and to some examples in which the STT or GSTT correctly terminate.

2. Homology and cohomology theory, an introduction. In the considered application, the domain of interest is always a connected subset of $\mathbb{R}^3$, which is de-

[1]The very idea dates back to Maxwell; see "Cyclosis in Surfaces and Regions" [20]. In computational electromagnetic, the idea was formalized and made popular by Kotiuga [21], [22], [23], [50], and Gross and Kotiuga [24], even though his definition, due to the use of finite elements with nodal basis functions, is different with respect to the one proposed in [17], [18], [19].

scribed by a tetrahedral finite element mesh $\mathcal{M}$. The mesh is obtained by using a mesh generator for example, NETGEN, in [32]. Once the mesh is provided, it is easy to derive from it a structure called abstract simplicial complex.

2.1. Abstract simplicial complex. A collection $\mathcal{K}$ of finite and nonempty sets is called an *abstract simplicial complex* if, for every set $S \in \mathcal{K}$, every nonempty subset of S belongs to $\mathcal{K}$. Every set $S \in \mathcal{K}$ is called *abstract simplex*. In this paper, the elements of the abstract simplices are the nodes of the mesh $\mathcal{M}$. A set of nodes form an abstract simplex iff the convex hull of the nodes belonging to the set is an element of a tetrahedron in $\mathcal{M}$. In most of the paper, only abstract simplicial complexes and abstract simplices are considered; for this reason the word *abstract* is omitted when confusion does not arise. Moreover, since our concern is about computer algorithms, only finite complexes are considered. The dimension of a simplex S, referred to as $\dim(S)$, is equal to the cardinality of S minus one. A p-dimensional simplex is called p-simplex. For a given p-simplex S, each nonempty subset of S is called a *subsimplex* of S. A $(p-1)$-dimensional subsimplex of S is referred to as a *face* of S. The geometric realization of a simplex is the convex hull spanned by the elements (nodes) of the considered simplex. The geometric realization of the abstract simplicial complex is the sum of the geometric realizations of all simplices in it, which in our case is the initial mesh $\mathcal{M}$. In this way, the geometric simplex and geometric simplicial complex corresponding to the considered abstract simplicial complex $\mathcal{K}$ are defined.

2.2. Oriented simplices and chains. Let us consider all possible orderings of the elements of a given k-simplex S. We say that two orderings of S are equivalent if they differ by an even permutation. Each equivalence class of this relation is referred to as an orientation of S; see [9], [33], [34]. In this paper, a k-simplex $\{x_0, \dots, x_k\}$ endowed with orientation is denoted by $[x_0, \dots, x_k]$, where $[\bullet]$ stands for an ordered list of nodes. From now on we fix the orientation of all the simplices once and for all. The set of all oriented p-simplices is denoted by $\mathcal{K}_p$. Only oriented simplices are considered further on in the paper; consequently, by the word simplex we always mean an oriented simplex.

A *k-chain* with coefficients in a given group G is a formal combination of k-simplices with coefficients in G. The set of all k-chains in the simplicial complex $\mathcal{K}$ is denoted by $C_k(\mathcal{K}, G)$. The set of all k-simplices in $\mathcal{K}_p$ provides a basis of $C_k(\mathcal{K}, G)$; i.e., every element of $C_k(\mathcal{K}, G)$ can be obtained in a unique way as a combination with coefficients in G of elements belonging to the basis. The elements of $C_k(\mathcal{K}, G)$ form a group with addition called the k-chain group. Let us consider a k-chain $c = \sum_{S \in \mathcal{K}_k} c^S S$, where $c^S \in G$. The *support* $|c|$ of the k-chain c is defined as $|c| = \{S \in \mathcal{K}_k | c^S \neq 0\}$. In this paper we are interested only in the groups $\mathbb{Z}$ and $\mathbb{R}$. Moreover, unless otherwise stated, the group of integers $\mathbb{Z}$ is assumed as the group G. In this case we write simply $C_k(\mathcal{K})$ instead of $C_k(\mathcal{K}, \mathbb{Z})$.

A *k-cochain* with values in the group G is a linear map $c^* : C_k(\mathcal{K}) \to G$. By the value of the cochain we refer to its image considered as an image of the map. All k-cochains of the complex $\mathcal{K}$ form a group with an addition called k-cochain group and denoted by $C^k(\mathcal{K}, G)$. Again, unless otherwise stated, the group G is the group of integers $\mathbb{Z}$, and in this case we write simply $C^k(\mathcal{K})$ instead of $C^k(\mathcal{K}, \mathbb{Z})$.

For each k-simplex $S \in \mathcal{K}_k$, let us define the linear map $S^* : C_k(\mathcal{K}) \to G$ such that $S^*(S) = 1$ and $S^*(K) = 0$ for $K \neq S$. The set $\{S^*\}_{S \in \mathcal{K}_k}$ forms a basis of $C^k(\mathcal{K}, G)$, which is used further on in the paper.

For $c^* \in C^k(\mathcal{K}, G)$ and $e \in C_k(\mathcal{K}, G)$ such that $c^* = \sum_{S \in \mathcal{K}_k} c^S S^*$ and $e = \sum_{S \in \mathcal{K}_k} e^S S$ we denote $\langle c^*, e \rangle = \sum_{S \in \mathcal{K}_k} c^S e^S \in G$.

For $c^* \in C^k(\mathcal{K}, G)$ and $e \in C_k(\mathcal{K}, G)$, the *evaluation* of a cochain c^* on a chain e is, by definition, the number $\langle c^*, e\rangle$.

Due to the bijective correspondence between the basis of $C_k(\mathcal{K}, G)$ and the basis of $C^k(\mathcal{K}, G)$, a one-to-one correspondence between a chain and a cochain exists. This chain-cochain natural duality leads to

$$C_k(\mathcal{K}, G) \cong C^k(\mathcal{K}, G). \tag{2.1}$$

2.3. (Co)boundary operator and (co)homology. For $k \geq 1$, the *boundary operator* $\partial_k : C_k(\mathcal{K}, G) \to C_{k-1}(\mathcal{K}, G)$ is defined in the following way:

1. For a k-simplex $S = [x_0, x_1, \ldots, x_k] \in C_k(\mathcal{K}, G)$ yields $\partial_k S := \sum_{i=0}^{k} (-1)^i [x_0, x_1, \ldots, \hat{x}_i, \ldots, x_k] \in C_{k-1}(\mathcal{K}, G)$, 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 simplices, the linear extension of this operator is applied: $\partial_k(\sum_{S\in\mathcal{K}_k} c^S S) = \sum_{S\in\mathcal{K}_k} c^S \partial_k S$.

It can be verified that $\partial_{k-1} \circ \partial_k = 0$; see, for example, [35].

In cohomology theory, the so-called *coboundary operator* $\delta^{k-1} : C^{k-1}(\mathcal{K}, G) \to C^k(\mathcal{K}, G)$ is defined in the way that for every $c^* \in C^{k-1}(\mathcal{K}, G)$ and $c \in C_k(\mathcal{K}, G)$ the following equation $\langle \delta c^*, c\rangle := \langle c^*, \partial c\rangle$ holds by definition. From the presented definition it is straightforward to see that $\delta^k \circ \delta^{k-1} = 0$.

Since the boundary operator is a linear map between $C_k(\mathcal{K}, G)$ and $C_{k-1}(\mathcal{K}, G)$, it can be represented, using the fixed bases of $C_{k-1}(\mathcal{K}, G)$ and $C_k(\mathcal{K}, G)$, as a matrix called $\mathbf{M}_{\partial_k}$. Suppose that the bases of $C^k(\mathcal{K}, G)$ and $C^{k-1}(\mathcal{K}, G)$ are taken as dual to the fixed bases of $C_k(\mathcal{K}, G)$ and $C_{k-1}(\mathcal{K}, G)$ as described in section 2.2. The coboundary operator δ^{k-1} can be represented, using the considered bases, by the transposed matrix $\mathbf{M}^T_{\partial_k}$. This matrix is also denoted as $\mathbf{M}_{\delta_k}$. The presented matrices representing the boundary and coboundary operators are essential in the computational aspects of the (co)homology theory.

The boundary operator gives rise to a classification of chains. From $\partial_{k-1} \circ \partial_k = 0$, it is straightforward to verify that image $\mathrm{im}(\partial_k)$ is a subgroup of kernel $\ker(\partial_{k-1})$. The image $\mathrm{im}(\partial_k)$ is called a *k-boundary group* of $\mathcal{K}$ and is denoted by $B_k(\mathcal{K}, G)$. The kernel $\ker(\partial_k)$ is called a *k-cycle group* of $\mathcal{K}$ and is denoted by $Z_k(\mathcal{K}, G)$. Elements of $Z_k(\mathcal{K}, G)$ are called *k-cycles* of $\mathcal{K}$, and elements of $B_k(\mathcal{K}, G)$ are called *k-boundaries* of $\mathcal{K}$. An analogous classification can be given for the the following cochains: $Z^k(\mathcal{K}, G) = \ker(\delta^k)$ is the group of k-cocycles of $\mathcal{K}$ and $B^k(\mathcal{K}, G) = \mathrm{im}(\delta^{k-1})$ is the group of k-coboundaries of $\mathcal{K}$. The *homology group* is the quotient group $H_k(\mathcal{K}, G) = Z_k(\mathcal{K}, G)/B_k(\mathcal{K}, G)$ for $k \in \mathbb{N}$. The dimension of the k-homology group is often called as k-Betti number $\beta_k(\mathcal{K}, G) = \dim(H_k(\mathcal{K}, G))$.

The set of equivalence classes of cycles $[h_1], \ldots, [h_n] \in H_k(\mathcal{K}, G)$ is referred to as homology basis if every other class in $H_k(\mathcal{K}, G)$ can be obtained in a unique way as a linear combination of classes $[h_1], \ldots, [h_n]$ with coefficients in G. In the following, by homology generator we refer both to a class $[h_i]$ being an element of the homology basis and to any cycle h_i representing this class.

The *cohomology group* is the quotient group $H^k(\mathcal{K}, G) = Z^k(\mathcal{K}, G)/B^k(\mathcal{K}, G)$ for $k \in \mathbb{N}$. Similarly as for homology, one can define a cohomology basis as a set of equivalence classes of cocycles $[h^1], \ldots, [h^n] \in H^k(\mathcal{K}, G)$ such that every other equivalence class of cocycles can be obtained in a unique way as a linear combination of classes $[h^1], \ldots, [h^n]$ with the coefficients in G. In the following, by cohomology generator we refer both to the equivalence class of a cocycle and to a cocycle representing its equivalence class. The existence of the (co)homology basis for a finite simplicial complex follows from Theorem 2.1 presented in the section 2.4.

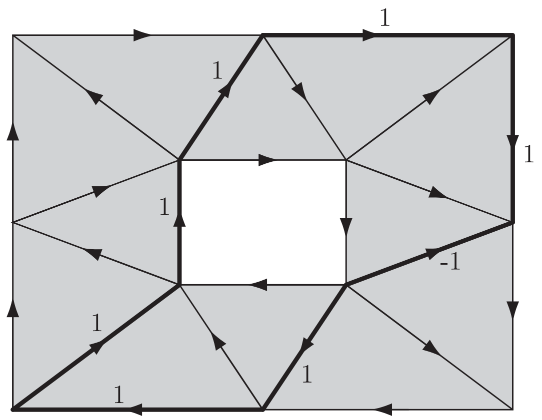

FIG. 2.1. *Support of a generator of the first homology group of an annulus.*

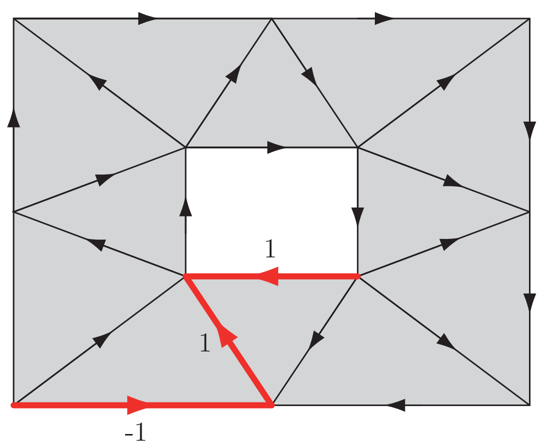

FIG. 2.2. *Support of a generator of the first cohomology group of an annulus.*

When the coefficient group is $\mathbb{Z}$, we simply write $H_k(\mathcal{K})$ and $H^k(\mathcal{K})$ instead of $H_k(\mathcal{K},\mathbb{Z})$ and $H^k(\mathcal{K},\mathbb{Z})$. Analogous simplified notation holds for (co)chains, (co)cycles, and (co)boundaries.

It is possible to define the (co)homology groups for a set $A \subset \mathbb{R}^n$ by using the singular homology theory; see, for example, [35]. The set A can be meshed with the mesh $\mathcal{M}$, and an abstract simplicial complex $\mathcal{K}$ can be produced based on the mesh as described in section 2.1. It is a standard result, in fact, that the singular homology group of A and the homology group of $\mathcal{K}$ are isomorphic.

Let us now present an example of simplices with nonzero coefficients in cycles and cocycles which represent a homology and cohomology basis of an annulus. The gray triangles in Figures 2.1 and 2.2 represent the triangulated region. The 1-chain, which has as support the thick edges and as coefficients the ones in the figure, represents a basis for $H_1(\mathcal{K})$. The 1-cochain, which has as support the thick edges in Figure 2.2 and as coefficients the ones in the same figure, represents a basis for $H^1(\mathcal{K})$.

Let $A_1,\ldots,A_{m+1}$ be abelian groups, and let $\alpha_i : A_i \to A_{i+1}$ be homomorphisms between them. The sequence

$$A_1 \xrightarrow{\alpha_1} A_2 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_{m-1}} A_m \xrightarrow{\alpha_m} A_{m+1}$$

is called exact if $\mathrm{im}(\alpha_i) = \ker(\alpha_{i+1})$ for every $i \in \{1, \ldots, m-1\}$. For $m = 4$ and $A_1 = A_4 = 0$ the exact sequence

$$0 \to A_2 \xrightarrow{\alpha_2} A_3 \xrightarrow{\alpha_3} A_4 \to 0$$

is referred to as short exact sequence. It is straightforward to see that if group A_2 in the short exact sequence is trivial, then $\alpha_3 : A_3 \to A_4$ is an isomorphism.

2.4. Structure of the homology group in case of a simplicial complex embedded in $\mathbb{R}^3$. It can be demonstrated that the homology group of a finite simplicial complex is a finitely generated abelian group; see, for example, [36]. For the definition of a finitely generated group, a cyclic group $\langle g \rangle$ generated by the generator g, and the definition of the direct sum, consult [36]. For finitely generated abelian groups the following classification theorem holds.

Theorem 2.1 (see [36, Theorem 3.61]). *Let G be a finitely generated abelian group. Then G can be decomposed as a direct sum of cyclic groups. More explicitly, there exist generators, $g_1, g_2, \ldots, g_q$ of G and an integer $0 \leq r \leq q$ such that*
- $G = \bigoplus_{i=1}^{q} \langle g_i \rangle$;
- *if $r > 0$, then $g_1, g_2, \ldots, g_r$ are of infinite order;*
- *if $k = q - r > 0$, then $g_{r+1}, g_{r+2}, \ldots, g_{r+k}$ have finite order $b_1, b_2, \ldots, b_k$, where $b_1, b_2, \ldots, b_k \in \mathbb{Z}$.*

The generators $g_{r+1}, g_{r+2}, \ldots, g_{r+k}$ from the Theorem 2.1 span the torsion subgroup of G, and the numbers $b_1, b_2, \ldots, b_k$ are referred to as torsion coefficients. If the homology group does not contain them, then the homology group is said to be torsion free.

A set $X \subset \mathbb{R}^3$ is compact in the standard topology if it is closed and bounded. In our case, all the simplicial complexes are finite, which implies that their geometric realizations are compact sets. A topological space X is said to be contractible if it has the same homotopy type as a point (see [35]). A space X is locally contractible if for every $x \in X$ and every open set U such that $x \in U$, there exists a contractible open set V such that $x \in V \subset U$ (see [37]). It is well known that the geometric realizations of simplicial complexes are locally contractible [37].

The following theorems hold.

Theorem 2.2 (see [35, Corollary 3.45]). *If $X \subset \mathbb{R}^n$ is compact and locally contractible, then $H_i(X, \mathbb{Z})$ is 0 for $i \geq n$ and torsion free for $i = n-1$ and $i = n-2$.*

Theorem 2.3 (see [35, Proposition 2.7]). *If $\mathcal{K}$ is a nonempty and connected simplicial complex, then $H_0(\mathcal{K}) = \mathbb{Z}$.*

Let us use the Theorem 2.2 for $X \subset \mathbb{R}^3$, where X is the geometric realization of the considered simplicial complex $\mathcal{K}$. It implies that $H_2(\mathcal{K}, \mathbb{Z})$ and $H_1(\mathcal{K}, \mathbb{Z})$ are torsion free. Since the geometric realizations X of the considered simplicial complexes $\mathcal{K}$ are connected, due to Theorem 2.3, $H_0(\mathcal{K}, \mathbb{Z}) = \mathbb{Z}$, so it is also torsion free.

2.5. Real and integer (co)homology groups. In this section, the group of (co)chains with real—instead of integer—coefficients are considered. By using such (co)chains it is possible to define the (co)homology groups of a simplicial complex $\mathcal{K}$ over reals, denoted as $H_k(\mathcal{K}, \mathbb{R})$. The cochain values, usually called degrees of freedom (DOFs) in computational physics (see, for example, [38], [25]), have a direct physical interpretation: By using the so-called *de Rham mapping* [39], they are defined as the integrals of the electromagnetic differential forms over the elements of the complex.[2]

[2]For example, the magneto-motive force DOF relative to the 1-simplex e is the integral of the differential 1-form *magnetic field* over the 1-simplex e.

However, unlike in the case of integers, that can be represented in a computer with arbitrary precision, it is not possible to make rigorous homology computations by using real numbers. Fortunately, it is well known in (co)homology theory that the (co)homologies computed over integers are the most universal ones. In fact, it will be demonstrated in the following that, in case of simplicial complexes whose geometric realization can be embedded in $\mathbb{R}^3$, the information brought by integer and real (co)homology is identical. In particular, it will be demonstrated that the integer homology group generators, which can be computed rigorously, are in bijective correspondence to real homology group generators. To demonstrate this fact, let us remind the universal coefficient theorem for homology (For the definition of tensor product and Tor functor consult [35]; their basic properties are cited further on.)

THEOREM 2.4 (see [35, Theorem 3A.4]). *If $\mathcal{K}$ is a simplicial complex, then there are natural short exact sequences*

$$0 \to H_k(\mathcal{K}, \mathbb{Z}) \otimes G \xrightarrow{p} H_k(\mathcal{K}, G) \to Tor(H_{k-1}(\mathcal{K}), G) \to 0$$

for all k and G.

Due to the Theorems 2.2 and 2.3, the homology groups $H_0(\mathcal{K})$, $H_1(\mathcal{K})$, and $H_2(\mathcal{K})$ are torsion free in the case of complexes whose geometric realization can be embedded in $\mathbb{R}^3$.

PROPOSITION 2.5 (see [35, Proposition 3A.5]). *Let A, B be abelian groups. If A or B is a free group, then $Tor(A, B) = 0$.*

From Proposition 2.5 and Theorems 2.2 and 2.3, it follows that $Tor(H_{k-1}(\mathcal{K}), G) = 0$ for $k \in \{1, 2\}$. Theorem 2.4 is used for $k \in \{1, 2\}$. In our case, the coefficient group G used in Theorem 2.4 is the group of real numbers $\mathbb{R}$. From the exactness of the sequence and Proposition 2.5, the following isomorphism holds: $p : H_k(\mathcal{K}, \mathbb{Z}) \otimes \mathbb{R} \to H_k(\mathcal{K}, \mathbb{R})$. Due to the isomorphisms p, each generator of $H_k(\mathcal{K}, \mathbb{Z}) \otimes \mathbb{R}$ corresponds to a unique generator of $H_k(\mathcal{K}, \mathbb{R})$.

Since in the considered case the homology groups $H_k(\mathcal{K}, \mathbb{Z})$ are torsion free, due to the Theorem 2.1 they are isomorphic to the direct sum $\bigoplus_{i=1}^{q} \mathbb{Z}$. From the properties of the tensor product of groups presented in [35, p. 218], taking the tensor product $H_k(\mathcal{K}, \mathbb{Z}) \otimes \mathbb{R}$ is equivalent to multiplying the elements of $H_k(\mathcal{K}, \mathbb{Z})$ basis by real numbers (and treating them as elements of $H_k(\mathcal{K}, \mathbb{R})$).

As a result, there exists a bijective correspondence between the generators of $H_k(\mathcal{K}, \mathbb{Z})$ and $H_k(\mathcal{K}, \mathbb{R})$ for $k \in \{1, 2\}$. From the homology-cohomology duality $H_k(\mathcal{K}, G) \cong H^k(\mathcal{K}, G)$ [34], the same correspondence holds for cohomology group generators.

2.6. Some basic properties of cocyles. In this section, a few basic properties of cochains are recalled. They are extensively used further on in the paper.

LEMMA 2.6. *For a given simplicial complex $\mathcal{K}$, the evaluation of a 1-cocycle $c^* \in Z^1(\mathcal{K})$ on a trivial 1-cycle $c \in B_1(\mathcal{K})$ is zero.*

Proof. Since $c^* \in Z^1(\mathcal{K})$, $\delta c^* = 0$. Since $c \in B_1(\mathcal{K})$, there exists an $e \in C_2(\mathcal{K})$ such that $\partial e = c$. It follows that $\langle c^*, c\rangle = \langle c^*, \partial e\rangle = \langle \delta c^*, e\rangle = \langle 0, e\rangle = 0$. ∎

LEMMA 2.7. *A 1-cochain c^* is a 1-cocycle iff, for every 2-simplex $S \in \mathcal{K}_2$, $\langle c^*, \partial S\rangle = 0$.*

Proof. From Lemma 2.6 it follows that the evaluation of a 1-cocyle c^* on a cycle ∂S is zero for every 2-simplex S. To show the opposite, let us assume, by contrary, that the evaluation $\langle c^*, \partial S\rangle = 0$ for every 2-simplex $S \in \mathcal{K}_2$ and $\delta c^* \neq 0$. This implies that there exists a 2-simplex K which is nonzero in δc^*. It follows that $0 \neq \langle \delta c^*, K\rangle = \langle c^*, \partial K\rangle = 0$. This gives a contradiction. ∎

LEMMA 2.8. *For two 1-cycles c_1 and c_2, which differ by a boundary, and a 1-cocycle c^*, $\langle c^*, c_1\rangle = \langle c^*, c_2\rangle$ holds.*

Proof. Since c_1 and c_2 differ by a boundary, it follows that there exists a 2-cycle $s \in C_2(\mathcal{K})$ such that $c_1 = c_2 + \partial s$. From Lemma 2.6 it follows that $\langle c^*, c_1\rangle = \langle c^*, c_2 + \partial s\rangle = \langle c^*, c_2\rangle + \langle c^*, \partial s\rangle = \langle c^*, c_2\rangle$. ∎

LEMMA 2.9. *Let us fix the cocycle $c^* \in Z^1(\mathcal{K})$ and the cycles $h_i \in Z_1(\mathcal{K}), i \in 1, \ldots, \beta_1(\mathcal{K})$, that represent the $H_1(\mathcal{K})$ basis. Then for any cycle $c \in Z_1(\mathcal{K})$ such that $[c] = \sum \alpha_i[h_i]$ one has $< c^*, c >= \sum \alpha_i \langle c^*, h_i\rangle$.*

Proof. Since $[c] = \sum \alpha_i[h_i]$, there exists $b \in C_2(\mathcal{K})$ such that $c = \sum \alpha_i h_i + \partial b$. From Lemma 2.6 we have $\langle c^*, c\rangle = \langle c^*, \sum_i \alpha_i h_i + \partial b\rangle = \sum_i \langle c^*, \alpha_i h_i\rangle + \langle c^*, \partial b\rangle = \sum_i \alpha_i \langle c^*, h_i\rangle$. ∎

3. The STT.

As already stated in the introduction, the STT is a classical technique used in computational electromagnetics to compute the *generalized source magnetic field*; see, for example, [13], [14], [15], [16]. The generalized source magnetic field is a 1-cochain with real values that are needed to enforce the source term in magnetostatic and magneto-quasi-static[3] BVP. Concerning this application, the simplicial complex $\mathcal{K}$ is always connected and homologically trivial.

DEFINITION 3.1. *The generalized source magnetic field is a 1-cochain $\mathbf{h}^s \in C^1(\mathcal{K}, \mathbb{R})$ that has to verify $\langle \mathbf{h}^s, \partial_2 T\rangle = \langle \mathbf{i}^s, T\rangle$ for each 2-simplex T in $\mathcal{K}_2$. $\mathbf{i}^s \in Z^2(\mathcal{K}, \mathbb{R})$ is a given electric current 2-cocycle. The values of $\mathbf{i}^s$ are real numbers which represent the electric current flowing through each 2-simplex T in $\mathcal{K}_2$.*

The reader should be aware that the cochain $\mathbf{h}^s$ is not unique, as it will be shown in this section.

Let us fix a *spanning tree* $\mathcal{T}$ of $\mathcal{K}_1$. The corresponding cotree $\mathcal{C}$ is obtained as $\mathcal{K}_1 \backslash \mathcal{T}$. Let $\delta^1_{\mathcal{T}}$ and $\delta^1_{\mathcal{C}}$ denote the restriction of the coboundary operator δ^1 to tree and cotree simplices, respectively. Let us denote by $\mathbf{M}_{\delta^1_{\mathcal{C}}}$ and $\mathbf{M}_{\delta^1_{\mathcal{T}}}$ the matrices relative to the restricted operators (i.e., the sizes of all matrices are the same as the size of $\mathbf{M}_{\partial_1}$ and the only nonzero rows correspond to elements in $\mathcal{C}$ and $\mathcal{T}$, respectively).

In the fixed cochain basis, the cochains $\mathbf{h}^s$ and $\mathbf{i}^s$ can be represented by DOF arrays, which are also denoted by $\mathbf{h}^s$ and $\mathbf{i}^s$. The STT is a technique to find $\mathbf{h}^s$ when $\mathbf{i}^s$ is given, without explicitly solving

$$\mathbf{M}_{\delta^1}\mathbf{h}^s = \mathbf{i}^s \tag{3.1}$$

with a linear system of equations solver. The matrix $\mathbf{M}_{\delta^1}$ is obviously not of maximal rank; thus (3.1) has an infinite number of possible solutions. In fact, if two different 1-cochains $\mathbf{h}^s_1$ and $\mathbf{h}^s_2$ that differ by a 0-coboundary of a 0-cochain $\boldsymbol{\omega} \in C^0(\mathcal{K}, \mathbb{R})$ are considered, the following holds:

$$\mathbf{M}_{\delta^1}\mathbf{h}^s_1 = \mathbf{M}_{\delta^1}(\mathbf{h}^s_2 + \mathbf{M}_{\delta^0}\boldsymbol{\omega}) = \mathbf{i}^s \tag{3.2}$$

since $\mathbf{M}_{\delta^1}\mathbf{M}_{\delta^0} = 0$. As a consequence, from (3.1),

$$\mathbf{M}_{\delta^1}\mathbf{h}^s = \mathbf{M}_{\delta^1_{\mathcal{C}}}(\mathbf{h}^s|_{\mathcal{C}}) + \mathbf{M}_{\delta^1_{\mathcal{T}}}(\mathbf{h}^s|_{\mathcal{T}}) = \mathbf{i}^s, \tag{3.3}$$

where $\mathbf{h}^s|_{\mathcal{T}}$ and $\mathbf{h}^s|_{\mathcal{C}}$ denote the restrictions of the cochain $\mathbf{h}^s$ to the tree and cotree 1-simplices, respectively. The value of $\mathbf{h}^s|_{\mathcal{T}}$ relative to 1-simplices in the tree $\mathcal{T}$ can

[3]When solving magneto-quasi-static problems in frequency domain, the generalized source magnetic field is a complex-valued 1-cochain. All the results presented in this paper hold without any modification also in the case of complex-valued 1-cochains.

1. Let $\mathcal{T} \subset \mathcal{K}_1$ be a given spanning tree of $\mathcal{K}_1$ (i.e., all the vertices $\mathcal{K}_0$ are visited by $\mathcal{T}$) and $\mathbf{i}^s \in Z^2(\mathcal{K}, \mathbb{R})$. Let $\mathbf{h}^s$ be the 1-cochain that is about to be constructed.
2. $L := \mathcal{K}_2$;
3. `for` every $E \in \mathcal{K}_1$ set $\langle \mathbf{h}^s, E\rangle :=$ `UNDEFINED`;
4. `for` every $E \in \mathcal{T}$ set $\langle \mathbf{h}^s, E\rangle := 0$;
5. `while(` $L \neq \emptyset$ `)`
   (a) `Lsize` $:= card(L)$;
   (b) `for` every $T \in L$
      i. `if for` every $E \in |\partial T|$, $\langle \mathbf{h}^s, E\rangle \neq$ `UNDEFINED`, `then`
         A. $HsT := \langle \mathbf{h}^s, \partial T\rangle$;
         B. `if` $HsT = \langle \mathbf{i}^s, T\rangle$ `then` remove T from L;
         C. `else return FAILURE`;
      ii. `if` there exist unique $E \in |\partial T|$ such that $\langle \mathbf{h}^s, E\rangle =$ `UNDEFINED`, `then`
         A. set $\langle \mathbf{h}^s, E\rangle$ in a way that equation $\langle \mathbf{h}^s, \partial T\rangle = \langle \mathbf{i}^s, T\rangle$ holds;
         B. remove T from L;
   (c) `if (` `Lsize` $= card(L)$ `) then return INFINITE_LOOP`;
6. `return` $\mathbf{h}^s$;

FIG. 3.1. *The STT algorithm.*

be fixed arbitrarily. In fact, as demonstrated, for example, in [40, p. 106], fixing the values over the spanning tree 1-simplices correspond to eliminating the $\ker(\delta^1)$ (i.e., the system 3.3 has a unique solution when the values of $\mathbf{h}^s$ on $\mathcal{T}$ elements are fixed). Let us fix the values of $\mathbf{h}^s$ relative to 1-simplices of the tree to zero. The rank of the matrix in (3.3) becomes maximal, and a unique solution of

$$\mathbf{M}_{\delta^1}\mathbf{h}^s = \mathbf{M}_{\delta^1_{\mathcal{C}}}(\mathbf{h}^s|_{\mathcal{C}}) = \mathbf{i}^s \tag{3.4}$$

exists. The STT algorithm is now introduced as a technique to solve (3.4) by means of back-substitutions only, without using a linear solver; see the algorithm in Figure 3.1.

The presented STT algorithm starts from setting the value relative to spanning tree edges of the complex to zero. Then all 2-simplices in the complex are loaded into a list L. The `while` loop of the STT algorithm works until there are no more 2-simplices in L. In each iteration, a 2-simplex T that has the value set for 2 or 3 boundary edges is searched. If T has the value already set for all 3 boundary edges, then the evaluation of $\mathbf{t}^i$ on ∂T is checked to be equal to the desired evaluation on ∂T. If it is not, then `FAILURE` is returned by the STT algorithm. If T already has the value set for 2 boundary edges, then the third one is set in order to obtain the desired evaluation. In both cases the 2-simplex T is removed from the list L. In the case when L is nonempty and there is no 2-simplex T that has the value set for either 2 or 3 boundary edges, the algorithm returns `INFINITE_LOOP`.

The question whether this algorithm terminates without returning `FAILURE` or `INFINITE_LOOP` for a given spanning tree is left unaddressed in the literature. However, it is not surprising that not all the linear systems arising in this application can be solved in this way. Several examples of such systems, induced by the corresponding simplicial complexes, are shown in section 5.

In the case when the algorithm returns `INFINITE_LOOP`, since it is known that the topology of the domain is trivial, one can theoretically use the strategy explained in [41] and [42] to solve the problem. The raw idea is as follows:

1. Take an arbitrary cotree edge C such that $\langle \mathbf{h}^{\mathbf{s}}, C\rangle =$ `UNDEFINED`. Together with a part of the tree $\mathcal{T}$, it closes a 1-cycle c;
2. By solving a linear system of equations, find a 2-chain $d = \sum_{F \in \mathcal{K}_2} d^F F$ such that $\partial d = c$;
3. $\langle \mathbf{h}^{\mathbf{s}}, C\rangle := \sum_{F \in \mathcal{K}_2} d^F \langle \mathbf{i}^s, F\rangle$.

In this way, whenever the STT algorithm is about to return `INFINITE_LOOP`, the above procedure can be applied and the STT iterations can continue. However, it should be noted that the presented procedure needs the solution of an underdetermined linear system of equations over integers. Since iterative solvers cannot be used, this yields at least to a cubical complexity with respect to the number of simplices in the complex, while the pure STT algorithm exhibits a linear complexity. Therefore, this solution has exactly the same complexity as solving the original system (3.4) with a linear system solver which provides that, in this case, STT is useless.

4. The GSTT.

The STT can be modified in order to solve a different problem arising when homologically nontrivial complexes are considered. The resulting algorithm, called GSTT, is an attempt to compute the cohomology generators when the representatives of the homology generators are given as input. In the whole section, the set of cycles $\{h_i\}_{i=1}^{\beta_1(\mathcal{K})}$ representing a basis of $H_1(\mathcal{K})$ is fixed.

4.1. Formulation of the problem.

In computational electromagnetics, to solve a magneto-quasi-static BVP using scalar potential-based formulations [19], a family of 1-cochains $\{\mathbf{t}^i\}_{i=1}^{\beta_1(\mathcal{K})}$ having the following properties is needed:

1. For every $i \in \{1, \ldots, \beta_1(\mathcal{K})\}$ and $c \in B_1(\mathcal{K})$, $\langle \mathbf{t}^i, c\rangle = 0$.
2. There exists a set of cycles $\{h_i\}_{i=1}^{\beta_1(\mathcal{K})}$ that represents a basis of $H_1(\mathcal{K})$ such that $\langle \mathbf{t}^i, h_j\rangle = \delta_{ij}$.

Due to the first property, each cochain $\mathbf{t}^i$ is a cocycle. In the following sections, an algorithm to construct such 1-cocycles $\{\mathbf{t}^i\}_{i=1}^{\beta_1(\mathcal{K})}$ for given $h_j, j \in \{1, \ldots, \beta_1(\mathcal{K})\}$ will be presented. Moreover, it will be demonstrated that $\{\mathbf{t}^i\}_{i=1}^{\beta_1(\mathcal{K})}$ are the representatives of a first cohomology group basis.

4.2. Independent constraints on fundamental cycles.

In this section, as an illustration, a naïve approach to find the set of cochains defined in section 4.1 is presented. Similarly to the section 3, let us fix a spanning tree $\mathcal{T}$ and the corresponding cotree $\mathcal{C}$ of $\mathcal{K}_1$.

Since $\mathcal{T}$ is a spanning tree of $\mathcal{K}_1$, for every $E \in \mathcal{C}$ there exist a unique graph-theoretic cycle in $\mathcal{T} \cup E$ denoted by c_E. Based on c_E, by assigning $+1$ and -1 coefficients to the elements of c_E, a cycle (in the sense of homology theory) can be obtained. Moreover, it is straightforward that such a cycle may have the orientation inherited from the orientation of the chosen edge $E \in \mathcal{C}$, in the sense described in the following definition.

Definition 4.1. *The* 1*-cycle* $L_E \in C_1(\mathcal{K})$ *having the coefficient equal* 1 *on the choosen edge* $E \in \mathcal{C}$ *and the coefficient* 1 *or* -1 *in the elements of* $\mathcal{T} \cap c_E$ *and* 0 *elsewhere will be referred to as fundamental cycle.*

Theorem 4.2 (see [9, Theorem 1.20]). *The set of all fundamental cycles* $\{L_E\}_{E\in\mathcal{C}}$ *forms a basis for* $Z_1(\mathcal{K})$.

Considering the fixed basis for the chains, the *fundamental cycle matrix* is now defined. The *fundamental cycle matrix* $\mathbf{B}$ collects the incidence information between each 1-simplex and the fundamental cycles. Let e denote the number of 1-simplices and n the number of 0-simplices in the complex $\mathcal{K}$. The fundamental cycle matrix has a number of rows $e - n + 1$, one for each fundamental cycle, and a number of columns e, one for each 1-simplex. For further details about the fundamental cycle matrix consult [43], [44], [45], [46].

For each homology generator h_i separately, a cochain $\mathbf{b}^i$ is constructed. Let us fix $i \in \{1, \ldots, \beta_1(\mathcal{K})\}$, and let us define the cochain $\mathbf{b}^i$ by fixing to 0 its values for all the 1-simplices belonging to the tree $\mathcal{T}$. For each cotree 1-simplex E that closes the

1. By solving a linear system of equations over integers, find the representation of L_E in the fixed $H_1(\mathcal{K})$ homology basis. Namely, find the integers α_k, $k \in \{1, \dots, \beta_1(\mathcal{K})\}$, such that $L_E = \sum_k \alpha_k h_k + \partial b$ for some $b \in C_2(\mathcal{K})$.
2. `return` α_i, where α_i is the coefficient of h_i in the representation of the cycle L_E in the homology basis.

FIG. 4.1. *Algorithm to construct the right-hand side array* $\mathbf{b}^i$.

fundamental cycle L_E, the value is obtained by means of the algorithm in Figure 4.1. The cochains $\mathbf{b}^i$ and $\mathbf{t}^i$ can be represented in the fixed cochains basis as vectors, which are also referred to as $\mathbf{b}^i$ and $\mathbf{t}^i$. It is easy to see that the linear system

$$\mathbf{B}\mathbf{t}^i = \mathbf{b}^i \tag{4.1}$$

is a maximal set of independent equations.[4] The Eth component of the vector $\mathbf{b}^i$, for $E \in \mathcal{C}$, corresponds to the desired evaluation of $\mathbf{t}^i$ over the corresponding fundamental cycle L_E. For this section only, let us permutate the fixed basis of 1-chains and a dual basis of 1-cochains in such a way that the 1-simplices that belong to the tree $\mathcal{T}$ come before the 1-simplices that belong to the cotree $\mathcal{C}$. The matrix $\mathbf{B}$ in the new base, denoted also as $\mathbf{B}$, can be consequently partitioned as

$$\mathbf{B} = [\mathbf{B}_{\mathcal{T}} \ \mathbf{I}_d], \tag{4.2}$$

where $\mathbf{B}_{\mathcal{T}}$ is the block of the matrix $\mathbf{B}$ relative to 1-simplices belonging to the tree $\mathcal{T}$ and $\mathbf{I}_d$ is the identity matrix, since the orientation of the fundamental cycle L_E is inherited from the one of the 1-simplex E. Also the vector $\mathbf{t}^i$, in the new basis, is partitioned as $\mathbf{t}^i = \left[\mathbf{t}^i_{\mathcal{T}}, \mathbf{t}^i_{\mathcal{C}}\right]^T$. Equation (4.1) can be written as

$$[\mathbf{B}_{\mathcal{T}} \ \mathbf{I}_d] \begin{bmatrix} \mathbf{t}^i_{\mathcal{T}} \\ \mathbf{t}^i_{\mathcal{C}} \end{bmatrix} = \mathbf{b}^i. \tag{4.3}$$

Again, it is possible to fix the value over the 1-simplices that belong to $\mathcal{T}$ to obtain a unique solution to (4.1) and (4.3). Let us fix these values $\mathbf{t}^i_{\mathcal{T}} = \mathbf{0}$. Then the solution has the following simple form:

$$\begin{array}{ll} (\mathbf{t}^i_{\mathcal{C}})_E = (\mathbf{b}^i)_E & \forall E \in \mathcal{C}, \\ (\mathbf{t}^i_{\mathcal{T}})_E = 0 & \forall E \in \mathcal{T}. \end{array} \tag{4.4}$$

One can easily prove the following theorem.

THEOREM 4.3. *The cochains obtained as the solutions of* (4.4) *are exactly the cochains defined in section* 4.1.

It should be noted that the computations in the algorithm in Figure 4.1 may be done in theory, provided that the representatives of a basis of the homology group is given. However, this technique is extremely time consuming, since it is necessary to find the representation in the homology basis for each fundamental cycle, which involves the solution of a linear system over integers. Thus, this solution is not suitable in practice. This is the reason why this technique has been referred to as naïve at the beginning of this section. The *belted tree*, described in the next section, has been developed as an attempt to avoid such a computationally expensive procedure.

[4]A heuristic demonstration exploits the fact that each equation on fundamental cycles involves the value on a cotree 1-simplices, which is not used in any other equation. Thus the equations have to be independent. They are also maximal, since the fundamental cycles form a basis for $Z_1(\mathcal{K})$.

4.3. Belted tree. In the following sections, a different technique with respect to the algorithm presented in the section 4.2 is used to construct the family of 1-cocycles $\{\mathbf{t}^i\}_{i=1}^{\beta_1(\mathcal{K})}$ defined in section 4.1. At the beginning, the evaluation of the 1-cocycle $\mathbf{t}^i$ on every 1-cycle h_j representing the first homology group generator in the given homology basis is fixed to δ_{ij}. Then the `while` loop in the GSTT algorithm is used to compute the values corresponding to the remaining 1-simplices in $\mathcal{K}_1$ by enforcing $\mathbf{t}^i$ to be a cocycle. Due to the Lemma 2.7, to do so it is enough to set $\langle \mathbf{t}^i, \partial T\rangle = 0$ for every 2-simplex $T \in \mathcal{K}_2$. It is clear that the cocycles obtained in this way are exactly the cocycles defined in the section 4.1.

To set $\langle \mathbf{t}^i, h_j\rangle = \delta_{ij}$, the concept of the belted tree is used.

DEFINITION 4.4. *A belted tree $\mathfrak{B}$ is a spanning tree $\mathcal{T}$ together with a set of 1-simplices $\{E_i\}_{i=1}^{\beta_1(\mathcal{K})}$, $E_i \in \mathcal{K}_1$, such that the set of graph-theoretic cycles $\{c_i\}_{i=1}^{\beta_1(\mathcal{K})}$, where c_i is the only cycle in $\mathcal{T} \cup E_i$, are exactly the supports of the chains $\{h_i\}_{i=1}^{\beta_1(\mathcal{K})}$. Each cycle c_i is referred to as a belt, and the 1-simplex E_i is referred to as a belt fastener.*

The algorithmic way to obtain a belted tree will be described in the section 4.4.

In the GSTT algorithm, during construction of the cocycle $\mathbf{t}^i$, the belted tree is used to set $\langle \mathbf{t}^i, h_j\rangle = \delta_{ij}$. This is obtained by setting $\langle \mathbf{t}^i, E\rangle = 0$ for $E \in \mathfrak{B} \setminus E_i$ and $\langle \mathbf{t}^i, E_i\rangle = 1$.

4.4. Automatic construction of a belted tree. In this section, the automatic construction of a belted tree for a simplicial complex $\mathcal{K}$ is addressed. At the beginning, the representatives of the first homology group generators are computed. Then, based on them, the belted tree is constructed.

In order to obtain the first homology basis, one of the libraries [43], [44], [45], [46] may be used. Before the algebraic Smith normal form computations, various original reduction techniques are applied to make the complex as small as possible [30], [31]. [24], Let us assume that, as the output of the homology computation algorithm, a set of chains $\{h_i\}_{i=1}^{\beta_1(\mathcal{K})}$ is obtained such that the $h_i = \sum_{E \in \mathcal{K}_1} \alpha_i^E E$ represent the generators of the first homology group.

The algorithm to construct a belted tree is now presented. For a set $\mathfrak{B} \subset \mathcal{K}_1$, let us define the concept of $\mathfrak{B}$-connected component. Two vertices $V_1, V_2 \in \mathcal{K}_0$ belongs to the same $\mathfrak{B}$-connected component if they can be joined with the 1-simplices in the set $\mathfrak{B}$. The algorithm presented in Figure 4.2 is used to obtain a belted tree.

The 1-simplices in the set $\mathfrak{B}$ are the belted tree 1-simplices. The 1-simplices in the set $\mathfrak{C}$ are cotree[5] 1-simplices.

LEMMA 4.5. *The only graph-theoretic cycles present in the belted tree $\mathfrak{B}$ are those which belong to the supports of $\{h_i\}_{i=1}^{\beta_1(\mathcal{K})}$.*

Proof. Suppose, by contrary, that there exists a set of 1-simplices $F \subset \mathfrak{B}$ that forms a graph theoretic cycle. Moreover, suppose that $F \not\subset \bigcup_{j=1}^{\beta_1(\mathcal{K})} |h_j|$. It follows that there exists a 1-simplex $E \in F$ that has been added to the set F in the `while` loop in the algorithm presented in Figure 4.2. Let $E' \in F$ denote the last 1-simplex added to the set F in the above `while` loop. Let V_1 and V_2 denote the 0-simplices in the boundary of E'. All the 1-simplices in F, except for E', are already in the set $\mathfrak{B}$ when E' is considered by the algorithm. Consequently, when E' is considered by the algorithm, V_1 and V_2 belong to the same $\mathfrak{B}$-connected component, since they can be joined by the 1-simplices in $(F \backslash E') \subset \mathfrak{B}$. In this case, from the algorithm, it follows that $E' \in \mathfrak{C}$, which gives a contradiction. ∎

[5] It should be noted that, in this section, the cotree is the complement of the belted tree.

- Let $\mathfrak{B} := \{E \in \mathcal{K}_1 \;\; \exists_{i\in\{1,\dots,\beta_1(\mathcal{K})\}} \;\; h_i = \sum_{S\in\mathcal{K}_1} \alpha_i^S S \text{ and } \alpha_i^E \neq 0\}$;
- $\mathfrak{C} := \emptyset$;
- `for` every $E \in \mathcal{K}_1 \backslash (\mathfrak{B} \cup \mathfrak{C})$
  - Let $V_1, V_2 \in |\partial E|$;
  - `if` V_1 and V_2 belong to the same $\mathfrak{B}$-connected component[a],
  - `then` $\mathfrak{C} := \mathfrak{C} \cup E$;
  - `else` $\mathfrak{B} := \mathfrak{B} \cup E$;
- `return` $\mathfrak{B}$, $\mathfrak{C}$;

[a]This can be effectively done by using find-union data structure; see [47].

FIG. 4.2. *Construction of a belted tree.*

From the first point of the algorithm in Figure 4.2 it follows that all the 1-simplices with nonzero coefficient in the representatives of the homology generators $\{h_i\}_{i=1}^{\beta_1(\mathcal{K})}$ are in $\mathfrak{B}$. From Lemma 4.5 it follows that those are the only graph-theoretic cycles in $\mathfrak{B}$, and from the algorithm in Figure 4.2 it is clear that no edge can be added to $\mathfrak{B}$ without closing a cycle. This implies that $\mathfrak{B}$ is a belted tree.

4.5. The GSTT algorithm. In section 4.1, the value of the cochain $\mathbf{t}^i$ is assumed to be set in the way that its evaluation on every homologically trivial cycle is zero. In this section, a technique to enforce this condition is presented.

A linear system of equations is solved for each $i \in \{1, \dots, \beta_1(\mathcal{K})\}$. Let $\mathfrak{B}$ be a belted tree created as in section 4.4. Then, for the fixed basis of the chain and cochain group, the following system is considered:

$$\mathbf{M}_{\delta^1}\mathbf{t}^i = 0, \;\; \mathbf{t}^i|_{\mathfrak{B}\backslash E_i} = 0, \;\; \mathbf{t}^i|_{E_i} = 1. \tag{4.5}$$

For this system, the following lemma holds.

LEMMA 4.6. *The system* (4.5) *has at most one solution.*

Proof. Let $\{h_j\}_{j=1}^{\beta_1(\mathcal{K})}$ be the representatives of the fixed homology basis. If the system is not consistent, then the lemma holds. Suppose that the system is consistent. It remains to show that, in this case, a unique solution exists. Suppose, by contrary, that two different solutions $\mathbf{t}^i$ and $\mathbf{t'}^i$ of (4.5) exist. It follows that there exists a cotree 1-simplex E such that $\langle \mathbf{t}^i, E\rangle \neq \langle \mathbf{t'}^i, E\rangle$. However, the 1-simplex E together with the 1-simplices $\mathfrak{B}\backslash(\bigcup_{j=1}^{\beta_1(\mathcal{K})} E_j)$ forms a unique fundamental cycle (as in section 3) C_E. Since the $\{h_j\}_{j=1}^{\beta_1(\mathcal{K})}$ form a homology basis, there exists a uniquely determinate set of integers $\{a_1, \dots, a_{\beta_1(\mathcal{K})}\}$ such that the cycle $\sum_{j=1}^{\beta_1(\mathcal{K})} a_j h_j$ is in the same homology class as the cycle C_E. From the Lemma 2.8 it follows that $\langle \mathbf{t}^i, C_E\rangle = \langle \mathbf{t}^i, \sum a_j h_j\rangle = \sum a_j \langle \mathbf{t}^i, h_j\rangle$ and $\langle \mathbf{t'}^i, C_E\rangle = \langle \mathbf{t'}^i, \sum a_j h_j\rangle = \sum a_j \langle \mathbf{t'}^i, h_j\rangle$. But the values $\langle \mathbf{t'}^i, h_j\rangle = \langle \mathbf{t}^i, h_j\rangle$, for each $j \in \{1, \dots \beta_1(\mathcal{K})\}$, have been fixed in the system. It follows that $\langle \mathbf{t}^i, C_E\rangle = \langle \mathbf{t'}^i, C_E\rangle$. From the assumptions in (4.5), the values associated to 1-simplices in $\mathfrak{B} \setminus E_i$ are set to zero. It is straightforward to see that $\langle \mathbf{t}^i, C_E\rangle = \langle \mathbf{t}^i, E\rangle$ and $\langle \mathbf{t'}^i, C_E\rangle = \langle \mathbf{t'}^i, E\rangle$, which implies $\langle \mathbf{t}^i, E\rangle = \langle \mathbf{t'}^i, E\rangle$. This gives a contradiction. ∎

Due to Lemma 4.6, the system has at most one solution, although the matrix is not square. In principle one can solve (4.5) by using an integer arithmetic system solver.

```
for i = 1 to β1(K)
    1. let 𝔅 be a belted tree and E_i be the belt fastener in h_i, chosen as described in
       section 4.4.
    2. L := K_2;
    3. for every E ∈ K_1 set ⟨t^i, E⟩ := UNDEFINED;
    4. set ⟨t^i, E_i⟩ := 1 and ⟨t^i, E⟩ := 0 for E ∈ 𝔅 \ E_i;
    5. while( L ≠ ∅ )
       (a) Lsize := card(L);
       (b) for every T ∈ L
           i. if for every E ∈ |∂T|, ⟨t^i, E⟩ ≠UNDEFINED, then
              A. if ⟨t^i, ∂T⟩ = 0 then remove T from L;
              B. else return FAILURE;
           ii. if there exist unique E ∈ |∂T| such that ⟨t^i, E⟩ = UNDEFINED then
              A. set ⟨t^i, E⟩ to get ⟨t^i, ∂T⟩ = 0;
              B. remove T from L;
       (c) if Lsize = card(L) then return INFINITE_LOOP;
return {t^i}_{i=1}^{β1(K)};
```

FIG. 4.3. *The GSTT algorithm.*

However, if a real-sized simplicial complex is used, this may take an unacceptable amount of time or memory. The GSTT algorithm is introduced as an attempt to solve (4.5), for every $i \in \{1, \dots, \beta_1(\mathcal{K})\}$, by using back-substitutions only.

The GSTT algorithm is presented in Figure 4.3.

The idea of the GSTT algorithm (for $\beta_1(\mathcal{K}) = 1$) is shown in Figure 4.4, on the same simplicial complex used in section 2.3. In the first picture, the edges belonging to the belted tree are shown. The thicker edge with coefficient 1 is the belt fastener edge. All the values on the other edges belonging to the belted tree are set to zero. Then the iteration process starts. On each iteration, the darker 2-simplices are the ones with 2 edges in their boundary already set. Thus the value on the third edge can be determined by setting a zero evaluation on the boundary of the considered 2-simplex. The dotted edges represent the edges whose value is determined in the considered iteration.

4.6. Conditions for GSTT error-free termination. This section summarizes some conditions required for the GSTT to terminate without errors and some conditions that arise from features that appear to cause difficulties for the GSTT algorithm. The conditions are strongly related to the counterexamples presented in section 5. No formal proof that GSTT returns errors when one or more of the described conditions do take place are known so far.

Let us first present the conditions related with the way belt fasteners are chosen:

1. Let $h_i = \sum_{E \in \mathcal{K}_1} \alpha_i^E E$ be the given representatives of the homology basis, and let $\alpha_i^{E_i}$ be the coefficient of belt fastener E_i in h_i. Due to the assumptions about $\mathbf{t}^i$, one has $1 = \langle \mathbf{t}^i, h_i \rangle = \langle \mathbf{t}^i, \sum_{E \in \mathcal{K}_1} \alpha_i^E E \rangle = \langle \mathbf{t}^i, \alpha_i^{E_i} E_i \rangle = \alpha_i^{E_i} \langle \mathbf{t}_i, E_i \rangle$. It follows that in order to set $\langle \mathbf{t}^i, h_i \rangle = 1$ in a belted tree one needs to have $|\alpha_i^{E_i}| = 1$.
2. For each representant of the homology basis $h_i = \sum_{E \in \mathcal{K}_1} \alpha_i^E E$, a belt fastener 1-simplex E_i has to be chosen in the way that a chain $\hat{C}_i = \sum_{E \in \mathcal{K}_1} \beta_i^E E$ such that

$$\beta_i^E := \begin{cases} \alpha_i^E & \text{if } E \neq E_i, \\ 0 & \text{if } E = E_i, \end{cases}$$

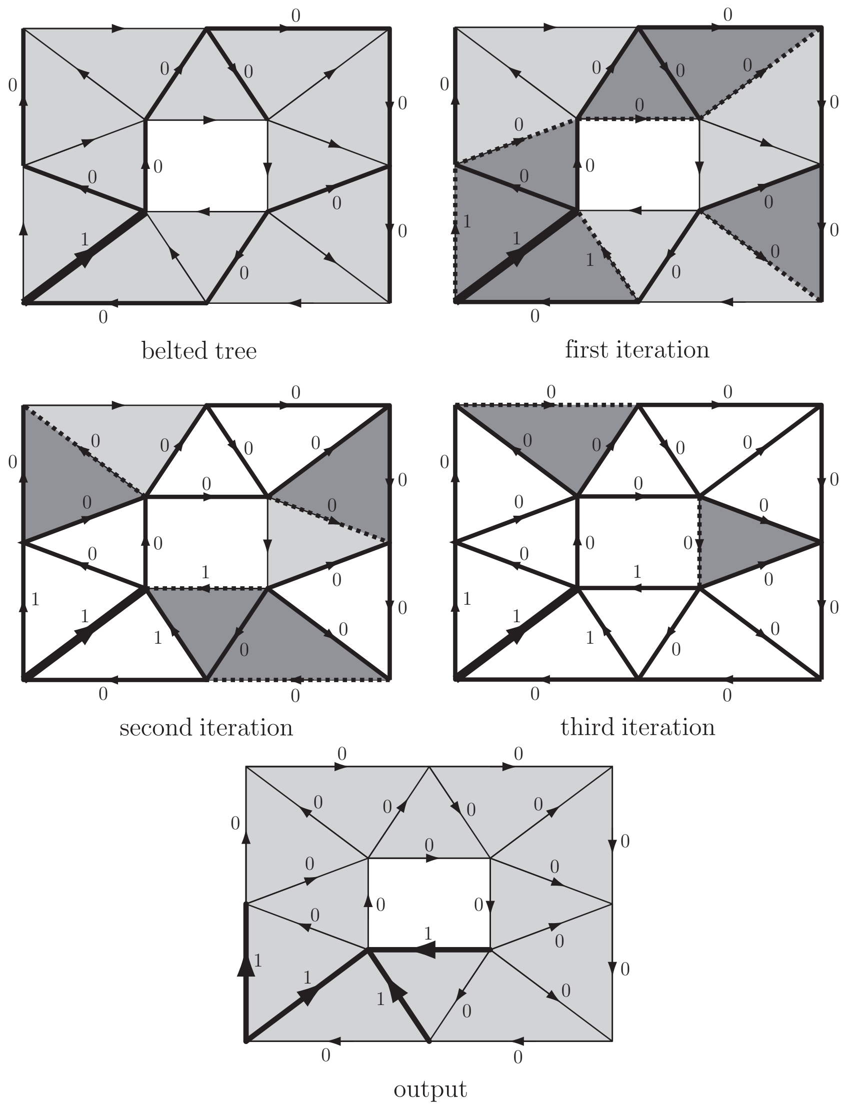

Fig. 4.4. *Illustration of the GSTT algorithm iterations.*

does not contain subchains[6] that are homologically nontrivial cycles.

It is shown in sections 5.2.1 and 5.3.1 that these are critical requirements for the belt fastener. However, in general, it is not easy in practice to algorithmically verify this assumption in an effective way.

Now let us present the conditions related with belts:

1. As stated in section 5.2.2, the belt cannot be a knot. A similar situation takes place when the considered domain is a complement of a knot; see section 5.4.1. The authors believe that GSTT will never terminate without errors in those cases.

[6]Let $c = \sum_{E \in \mathcal{K}_1} \beta^E E, d = \sum_{E \in \mathcal{K}_1} \gamma^E E$, with $c, d \in C_1(\mathcal{K})$. d is a subchain of c if the following implication holds: $\beta^E = 0 \Rightarrow \gamma^E = 0$.

2. Unfortunately, valid (but not minimal in some sense) generators that do not have intersections nor self-intersections and which do not form any knot may cause a problem for correct GSTT termination as well, as reported in section 5.3.2.

Also "knotted paths" in the belted tree, as in sections 5.1.1 and 5.1.2, may prevent STT/GSTT error-free termination. In fact, since STT and GSTT use the same propagation strategy, the STT counterexamples reported in sections 5.1.1, 5.1.2 and 5.1.3 hold for GSTT as well.

All these counterexamples indicate that it may be very hard to try to present necessary and sufficient conditions for the STT/GSTT error-free termination.

4.7. Formal description of the GSTT output. It is assumed that the algorithm presented in Figure 4.3 did not return `FAILURE` or `INFINITE_LOOP`. For each chain h_i representing the $H_1(\mathcal{K})$ generator, let $\mathbf{t}^i$ denote the cochain obtained as the output of the algorithm. It is demonstrated in the following that the $\{\mathbf{t}^i\}_{i=1}^{\beta_1(\mathcal{K})}$ returned by the GSTT algorithm represents an $H^1(\mathcal{K})$ basis. First, the universal coefficients theorem for cohomology is recalled.

THEOREM 4.7 (see [35, Theorem 3.2]). *If a simplicial complex $\mathcal{K}$ has (integer) homology groups $H_n(\mathcal{K})$, then the cohomology groups $H^n(\mathcal{K}, G)$ are determined by the exact sequences*

$$0 \to Ext(H_{n-1}(\mathcal{K}), G) \to H^n(\mathcal{K}, G) \xrightarrow{h} Hom(H_n(\mathcal{K}), G) \to 0.$$

In this paper, there is no need to go into the definition of the Ext functor. The key property is that $Ext(Q, G) = 0$ if Q is a free group. For further details and proof of this property consult [35, p. 195]. $Hom(H_n(\mathcal{K}), G)$ is denoted by the group of homomorphisms $H_n(\mathcal{K}) \to G$ with addition.

From the definition of a coboundary operator for a class $[d] \in H^n(\mathcal{K}, G)$, since d is a cocycle, one has $0 = \langle \delta d, z \rangle = \langle d, \partial z \rangle$ for each $z \in C_2(\mathcal{K})$. From the above equality it follows that $d|_{B_n(\mathcal{K})} = 0$. Let us define the restriction $d_0 = d|_{Z_n(\mathcal{K})}$. Since $d_0|_{B_n(\mathcal{K})} = 0$, then $d_0 \in Hom(H_n(\mathcal{K}), G)$. This shows the correctness of definition of the map $h([d]) = d_0 \in Hom(H_n(\mathcal{K}), G)$ used in the exact sequence in Theorem 4.7. In our case the group G is the group of integers, and the universal coefficient theorem for cohomology is used in the case $n = 1$. In this case the exact sequence from Theorem 4.7 has the form

$$0 \to Ext(H_0(\mathcal{K}), \mathbb{Z}) \to H^1(\mathcal{K}, \mathbb{Z}) \xrightarrow{h} Hom(H_1(\mathcal{K}), \mathbb{Z}) \to 0.$$

Now, the main theorem of this section is presented.

THEOREM 4.8. *The output of the GSTT algorithm, $\{\mathbf{t}^i\}_{i=1}^{\beta_1(\mathcal{K})}$, consists of the cocycles representing a basis of the first cohomology group $H^1(\mathcal{K})$.*

Proof. Since in our case the simplicial complex $\mathcal{K}$ is nonempty and connected, from Theorem 2.3 it follows that $H_0(\mathcal{K}) = \mathbb{Z}$. This provides that $H_0(\mathcal{K})$ is a free group. From the cited property of the Ext functor, it follows that $Ext(H_0(K), \mathbb{Z}) = 0$. From exactness of the sequence, one has that $h : H^1(\mathcal{K}, \mathbb{Z}) \to Hom(H_1(\mathcal{K}), \mathbb{Z})$ is an isomorphism. Let $\{h_i\}_{i=1}^{\beta_1(\mathcal{K})}$ be the cycles representing the $H_1(\mathcal{K})$ basis that has been used as the input to the GSTT algorithm. Let us define $\phi_i \in Hom(H_1(\mathcal{K}), \mathbb{Z})$ such that $\phi_i([h_j]) = \delta_{ij}$. Since $\mathcal{K}$ is a three-dimensional simplicial complex, then, due to Theorems 2.2 and 2.3, all the homology groups of $\mathcal{K}$ are free groups. Consequently $\{\phi_i\}_{i=1}^{\beta_1(\mathcal{K})}$ form a basis of $Hom(H_1(\mathcal{K}), \mathbb{Z})$. Since h is an isomorphism, $\{h^{-1}(\phi_i)\}_{i=1}^{\beta_1(\mathcal{K})}$

form a basis of $H^1(\mathcal{K},\mathbb{Z})$. Let us denote $(\mathbf{t}'^i) = h^{-1}(\phi_i)$. Due to the definition of the isomorphism h, the cocycles $\mathbf{t}'^i$ verify $\langle \mathbf{t}'^i, h_j\rangle = \delta_{ij}$, as the cocycle $\mathbf{t}^i$ obtained from the GSTT algorithm presented in Figure 4.3. It remains to show that $\mathbf{t}'^i$ and $\mathbf{t}^i$ are in the same cohomology class for every $i \in \{1,\dots,\beta_1(\mathcal{K})\}$. Suppose, by contrary, that $\mathbf{t}'^i - \mathbf{t}^i$ is a nonzero element in $H^1(\mathcal{K},\mathbb{Z})$. Then the image $h(\mathbf{t}'^i - \mathbf{t}^i)$ is a nonzero element in $Hom(H_1(\mathcal{K}),\mathbb{Z})$. Therefore, there exists a nonzero element $[e] \in H_1(\mathcal{K})$, $[e] = [\sum_{j=1}^{\beta_1(\mathcal{K})} \alpha_j h_j]$, such that $\alpha_j \neq 0$ for some $j \in \{1,\dots,\beta_1(\mathcal{K})\}$, which implies $\langle \mathbf{t}'^i - \mathbf{t}^i, e\rangle \neq 0$. However, $0 \neq \langle \mathbf{t}'^i - \mathbf{t}^i, e\rangle = \langle \mathbf{t}'^i - \mathbf{t}^i, \sum_{j=1}^{\beta_1(\mathcal{K})} \alpha_j h_j\rangle = \sum_{j=1}^{\beta_1(\mathcal{K})} \alpha_j(\langle \mathbf{t}'^i, h_j\rangle - \langle \mathbf{t}^i, h_j\rangle) = 0$. This gives a contradiction. Hence, the cocycles $\{\mathbf{t}^i\}_{i=1}^{\beta_1(\mathcal{K})}$ obtained by the GSTT algorithm represent a basis of $H^1(\mathcal{K})$. ∎

5. Termination issues.

Despite that the STT and the GSTT are considered to be general, in this section we show that quite a big number of limitations of these techniques exist. Let us present these limitations with concrete counterexamples. To this aim, the STT and GSTT algorithm have been implemented by the authors using MATLAB® [48]. The meshes and homology generators employed in all the presented counterexamples are provided with the implementation.

5.1. STT termination errors.

5.1.1. Counterexample A: A coarse ball.

The simplest counterexample is formed by a three-dimensional homologically trivial complex. It contains eight 3-simplices, 22 2-simplices, 21 1-simplices, and eight 0-simplices. An exploded view of the 3-simplices is visible on the left of Figure 5.1. A spanning tree is formed by considering the thick 1-simplices on the right of Figure 5.1. If the STT is run, no 1-simplex can be set, and the STT algorithm returns `INFINITE_LOOP`. In fact, each 2-simplex has one and only one tree 1-simplex in its boundary.

Remark 1. This counterexample demonstrates that there exist trees for which the STT terminates with an error.

5.1.2. Counterexample B: A cube.

A three-dimensional complex which represents a cube and is homeomorphic to the three-dimensional ball is introduced. A tree is constructed first on the boundary of the complex. Then the tree is constructed in the interior, using as graph only the 1-simplices that have no node on the boundary of the cube. Moreover, the tree in the interior contains a "knotted path;" see Figure 5.2. Finally, a tree on the whole cube is obtained by adding one 1-simplex that joins the tree on the cube's boundary and the tree in the interior. Such a procedure

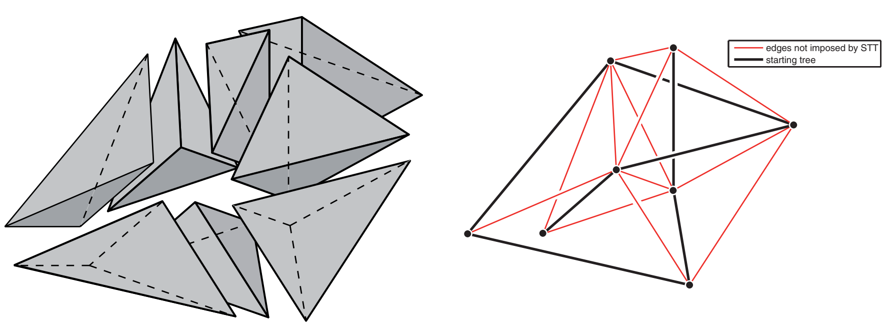

FIG. 5.1. *The complex considered in counterexample A.*

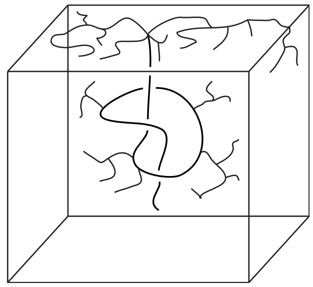

FIG. 5.2. *A three-dimensional complex which represents a cube is considered in counterexample B. A subset of the chosen tree is shown in addition.*

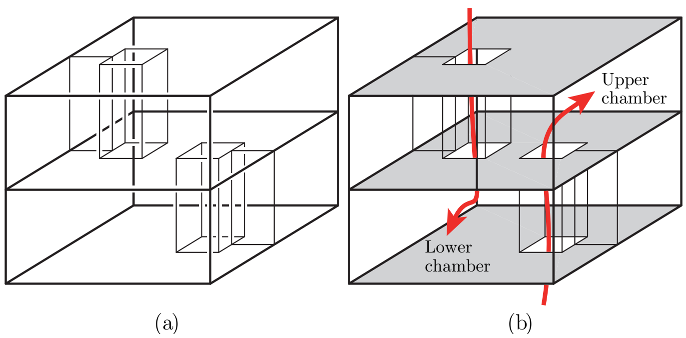

FIG. 5.3. *The sketch of the complex considered in counterexample C.*

to build a tree is frequently used in computational electromagnetics (see, for example, [15]), to reduce the support of the 1-cochain $\mathbf{h}^s$. In this counter-example the STT returns `INFINITE_LOOP`, it being not possible to set all the 1-simplices in the cube.

5.1.3. Counterexample C: The Bing's house. A Bing's house [49] is now considered. The complex, homeomorphic to the three-dimensional ball, can be obtained by replacing every surface in the polyhedron represented in Figure 5.3(a) by a "thick wall" made of 3-simplices. At the end of this procedure, one obtains the polyhedron in Figure 5.4. The two views are obtained by cutting the polyhedron with a vertical plane (Figure 5.4(a)) and a horizontal plane (Figure 5.4(b)). The obtained polyhedron can be considered informally as a "house" with two "chambers." In fact, the polyhedron is made in such a way that one can enter the two chambers by following the paths shown in Figure 5.3(b). It can be demonstrated that the Bing's house is homeomorphic to the tree-dimensional ball.

Also with this complex the STT returns `INFINITE_LOOP`.

Remark 2. This counterexample demonstrates that, for a given mesh of the Bing's house and for a randomly chosen spanning tree, the STT terminates with an error. We conjecture that this is the case for every spanning tree of the Bing's house.

5.2. GSTT termination errors arising with one generator.

5.2.1. Counterexample D: Self-intersecting generator. The complement of a torus with respect to a cylinder is covered with the complex $\mathcal{K}$. On the left of Figure 5.5, the boundary of the complex $\mathcal{K}$ is shown. On the right, the internal boundary of $\mathcal{K}$ is depicted. The 1-simplices with nonzero coefficient in the $H_1(\mathcal{K})$ generator

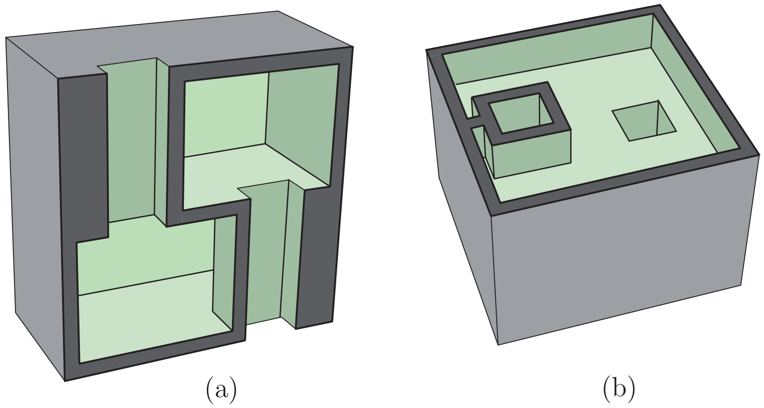

FIG. 5.4. *Two views of the Bing's house polyhedron obtained by cutting it with a vertical plane* (a) *or a horizontal plane* (b).

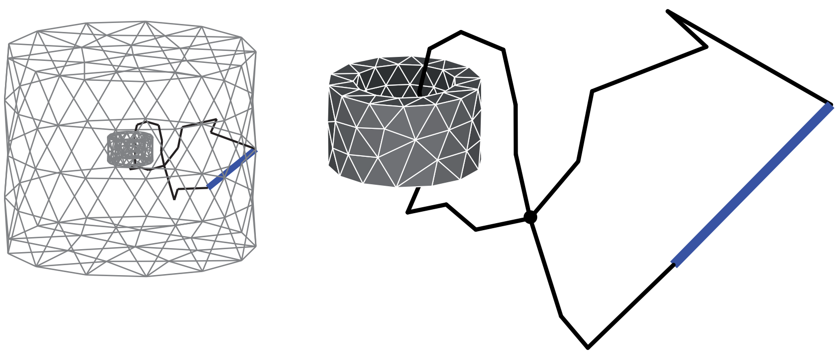

FIG. 5.5. *The complex used in the counterexample D.*

1-cycle are represented in the figure as thick edges. The thickest of all edges is the belt fastener. This homology generator may be obtained from a homology computation. In this counterexample, the belt fastener 1-simplex is chosen to enforce a nonzero circulation on a trivial cycle. Therefore, due to the Lemma 2.6, an inconsistent value is forced, and the GSTT returns `FAILURE`.

Remark 3. This class of problems in principle can be solved by selecting the appropriate belt fastener. Nonetheless, it should be noted that the procedure to select the appropriate belt fastener necessarily requires us to check whether a cycle is homologically trivial or not, which is computationally costly.

5.2.2. Counterexample E: Knotted generator. A simple torus complement, as in counterexample D, is considered; see Figure 5.6. A knotted generator, which may be obtained by an automatic homology computation, is used in this counterexample. For a random tree containing the knotted generator, the GSTT returns `INFINITE_LOOP`. We conjecture that this is the case for every tree when the generator forms a knot.

Remark 4. This counterexample demonstrates that, for a given knotted generator and randomly chosen belted tree, the GSTT terminates with an error. Moreover, we note that this situation is difficult to detect.

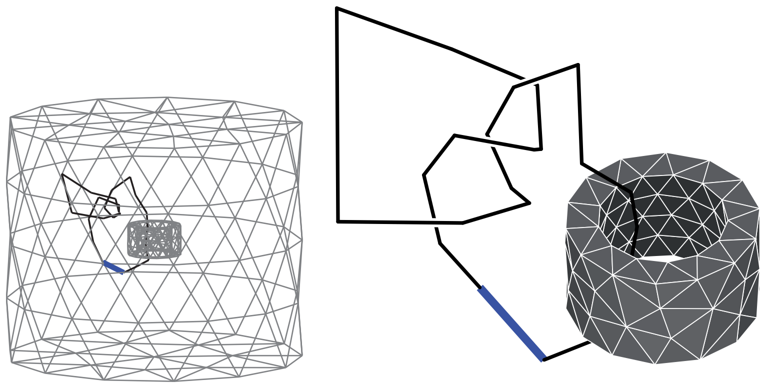

Fig. 5.6. *The complex used in the counterexample E.*

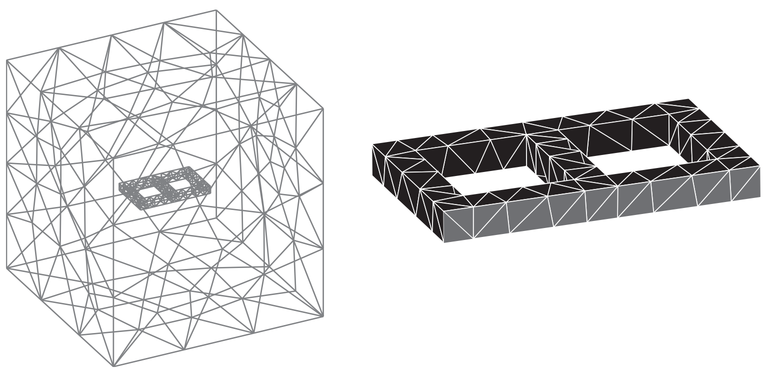

Fig. 5.7. *The complex used in the counterexamples F and G.*

5.3. GSTT termination errors arising with more than one generator. The complement of a double torus with respect to a cube is covered with the simplicial complex $\mathcal{K}$; see Figure 5.7. On the left of Figure 5.7, the boundary of the complex $\mathcal{K}$ is shown. On the right, the internal boundary of $\mathcal{K}$ is depicted. This complex is used in the next two counterexamples.

5.3.1. Counterexample F: Intersecting generators. A double torus complement complex is considered; see Figure 5.7. In the considered counterexample, the two homology generators are produced by an automatic homology computation. Such generators intersect each other (see Figures 5.8(a) and 5.8(b)), which is quite common in practice. If the first belt fastener is selected as in this counterexample, a nonzero evaluation is set on a trivial cycle; see Figure 5.8(c). This inconsistency induces the GSTT to return a `FAILURE`.

Remark 5. The same remark as the one in counterexample D holds.

5.3.2. Counterexample G: Complicated generators. A double torus complement complex is considered; see Figure 5.7. Let us denote the generators as in Figure 5.9(a) as g_1 and g_2. Let us also denote by $e_1 = g_1 + ng_2$ and $e_2 = g_1 + (n-1)g_2$. Then, of course, $e_1 - e_2 = g_2$. Since $ng_2 = n(e_1 - e_2)$, $e_1 = g_1 + ng_2 = g_1 + n(e_1 - e_2)$ holds. This implies $e_1 - n(e_1 - e_2) = g_1$. So one can get both g_1 and g_2 as a combination of e_1 and e_2. g_1 and g_2 is a valid homology basis; hence also e_1 and e_2 is a valid

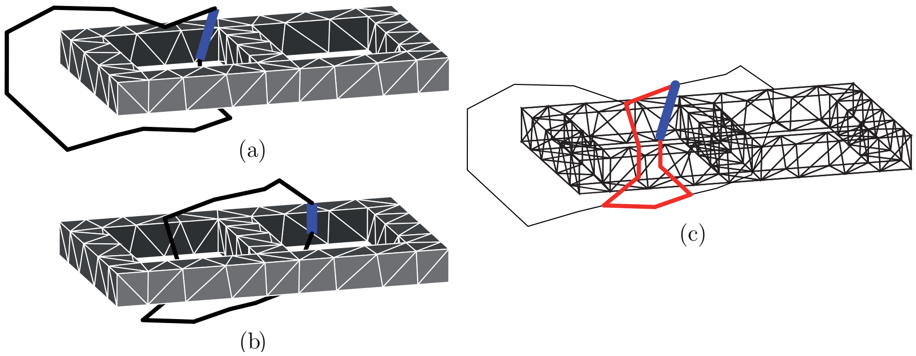

FIG. 5.8. *The complex used in the counterexample F.*

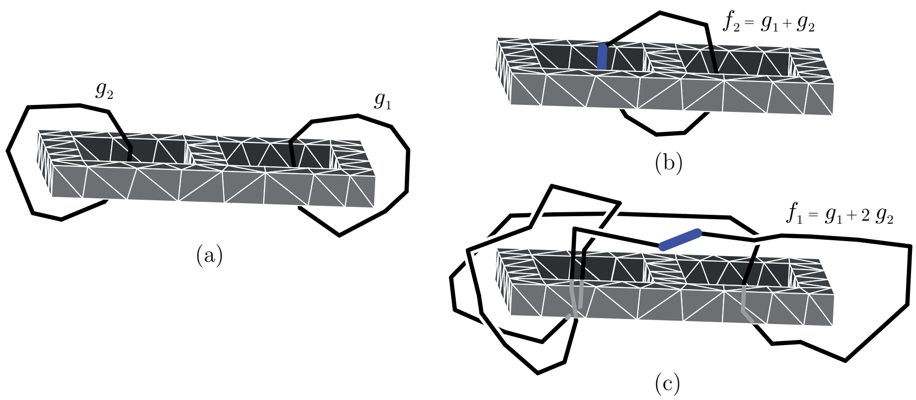

FIG. 5.9. *The complex used in the counterexample G.*

basis. In this example, let us focus on the case of $n = 2$. Let us define $g_1 + 2g_2 = f_1$ and $g_1 + g_2 = f_2$. It follows that $f_1 - f_2 = g_2$ and $f_2 - (f_1 - f_2) = 2f_2 - f_1 = g_1$. So f_1 and f_2 are a valid basis for the first homology group, which means that such a generators can be obtained by an automatic homology computation. If the generators f_1 and f_2 are used as input for the GSTT, the algorithm returns `INFINITE_LOOP`.

Remark 6. This counterexample demonstrates that, considering f_1 and f_2 as input generators and for a randomly chosen belted tree, the GSTT terminates with an error. We conjecture that this is the case for every spanning tree. Moreover, we note that this situation is difficult to detect, since the generators do not intersect each other and are not self-intersecting and nonknotted.

5.4. GSTT termination errors arising with knot's complements.

5.4.1. Counterexample H: Complement of a knot. A trefoil knot complement with respect to a cube is considered and covered by the complex $\mathcal{K}$; see Figure 5.10. A nonself-intersecting and nonknotted homology generator is used. The GSTT algorithm returns `INFINITE_LOOP`. We conjecture that this counterexample holds always when a knot's complement is considered.

Remark 7. This counterexample demonstrates that, for a given mesh of a knot's complement and for a random belted tree, the GSTT terminates with an error.

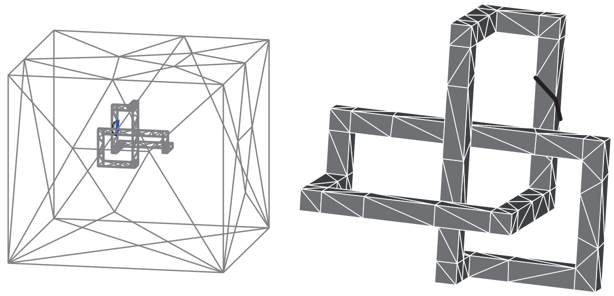

FIG. 5.10. *The complex used in the counterexample H.*

6. Conclusions. The STT and the GSTT have been analyzed in the light of algebraic topology. The number of counterexamples shows that many problems may arise both with the STT and the GSTT, preventing their termination without errors. Moreover, it seems not trivial to find a general solution to all of these problems. Therefore, the answer proposed by the authors to the open question raised by Bossavit in [25, p. 238] is that the direct computation of the first cohomology group generators appears to be a better option than GSTT.

Acknowledgments. We would like to thank the anonymous referees for some valuable comments which helped to improve the presentation of the paper.

REFERENCES

[1] G. KIRCHHOFF, *Ueber die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung Galvanischer Ströme geführt wird*, Poggendorf's Ann. Phys. Chemie, 72 (1847), pp. 497–508.

[2] G. KIRCHHOFF, *On the solution of the equations obtained from the investigation of the linear distribution of galvanic currents*, IRE Trans. Circuit Theory, 5 (1958), pp. 4–7.

[3] N. BALABANIAN AND T.A. BICKART, *Electrical Network Theory*, Wiley, New York, 1969.

[4] H. WEYL, *Reparticiòn de corriente en una red conductora: Introducciòn al analysis situs combinatorio*, Rev. Math. Ispano-Americana, 5 (1923), pp. 153–164.

[5] O. VEBLEN, *Analysis Situs*, Amer. Math. Soc. Coloq. Pub. 5, AMS, Providence, RI, 1931.

[6] J.P. ROTH, *An application of algebraic topology to numerical analysis: On the existence of a solution to the network problem*, Proc. Natl. Acad. Sci., 41 (1955), pp. 518–521.

[7] G. KRON, *Numerical solution of ordinary and partial differential equations by means of equivalent circuits*, J. Appl. Phys., 126 (1945), pp. 172–186.

[8] 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, Brooklyn, NY, 1966, pp. 453–491.

[9] P.J. GIBLIN, *Graphs, Surfaces and Homology*, Chapman and Hall, London, UK, 1977.

[10] P. BAMBERG AND S. STERNBERG, *A Course in Mathematics for Students of Physics: Vol.* 2, Cambridge University Press, Cambridge, UK, 1991.

[11] R. ALBANESE AND G. RUBINACCI, *Integral formulation for* 3*D eddy-current computation using edge elements*, IEE Proc. A, 135 (1988), pp. 457–462.

[12] Z. REN, *Influence of the R.H.S. on the convergence behaviour of the curl-curl equation*, IEEE Trans. Magn., 32 (1996), pp. 655–658.

[13] J.P. WEBB AND B. FORGHANI, *A single scalar potential method for* 3*D magnetostatics using edge elements*, IEEE Trans. Magn., 25 (1989), pp. 4126–4128.

[14] Y. LE MÉNACH, S. CLÉNET, AND F. PIRIOU, *Determination and utilization of the source field in* 3*D magnetostatic problems*, IEEE Trans. Magn., 34 (1998), pp. 2509–2512.

[15] F. HENROTTE AND K. HAMEYER, *An algorithm to construct the discrete cohomology basis functions required for magnetic scalar potential formulations without cuts*, IEEE Trans. Magn., 39 (2003), pp. 1167–1170.

[16] T. HENNERON, S. CLENET, P. DULAR, AND F. PIRIOU, *Discrete finite element characterizations of source fields for volume and boundary constraints in electromagnetic problems*, J. Comput. Appl. Math., 215 (2008), pp. 438–447.

[17] R. Specogna, S. Suuriniemi, and F. Trevisan, *Geometric T-Ω approach to solve eddy-currents coupled to electric circuits*, Internat. J. Numer. Methods Engrg., 74 (2008), pp. 101–115.

[18] P. Dłoko, R. Specogna, and F. Trevisan, *A homological algorithm for the automatic generation of cuts suitable for T-Ω eddy-current geometric formulation*, in Proceedings of the 5th Workshop on Advanced Computational Electromagnetics (ACE'09), Accademia dei Lincei, Rome, Italy, 2009, pp. 780–801.

[19] P. Dłoko, R. Specogna, and F. Trevisan, *Automatic generation of cuts on large-scaled meshes suitable for the T-Ω geometric eddy-current formulation*, Comput. Methods Appl. Mech. Engrg., 198 (2009), pp. 3765–3781.

[20] J.C. Maxwell, *A Treatise on Electricity and Magnetism, Volume* 1, Clarendon Press, Oxford, UK, 1891.

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

[22] P.R. Kotiuga, *Toward an algorithm to make cuts for magnetic scalar potentials in finite element meshes*, J. Appl. Phys., 63 (1988), pp. 3357–3359.

[23] 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), pp. 4129–4131.

[24] P.W. Gross and P.R. Kotiuga, *Electromagnetic Theory and Computation: A Topological Approach*, Math. Sci. Res. Inst. Monogr. Ser. 48, Cambridge University Press, Cambridge, UK, 2004.

[25] A. Bossavit, *Computational Electromagnetism*, Academic Press, San Diego, 1998; also available online at http://butler.cc.tut.fi/~bossavit/.

[26] L. Kettunen, K. Forsman, and A. Bossavit, *Formulation of the eddy current problem in multiply connected regions in terms of h*, Internat. J. Numer. Methods Engrg., 41 (1998), pp. 935–954.

[27] L. Kettunen, K. Forsman, and A. Bossavit, *Discrete spaces for div and curl-free fields*, IEEE Trans. Magn., 34 (1998), pp. 2551–2554.

[28] L. Kettunen and A. Bossavit, *Gauging in Whitney spaces*, IEEE Trans. Magn., 35 (1999), pp. 1466–1469.

[29] S. Suuriniemi, T. Tarhasaari, and L. Kettunen, *Generalization of the spanning-tree technique*, IEEE Trans. Magn., 38 (2002), pp. 525–528.

[30] M. Mrozek and B. Batko, *Coreduction homology algorithm*, Discrete Comput. Geom., 41 (2009), pp. 96–118.

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

[32] J. Schöberl, *NETGEN—An advancing front 2D/3D-mesh generator based on abstract rules*, Comput. Vis. Sci., 1 (1997), pp. 41–52.

[33] W.S. Massey, *Singular Homology Theory*, Grad. Texts Math. 70, Springer-Verlag, Berlin, Germany, 1980.

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

[35] A. Hatcher, *Algebraic Topology*, Cambridge University Press, Cambridge, UK, 2002; also available online at http://www.math.cornell.edu/~hatcher/AT/AT.pdf.

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

[37] V.V. Prasolov, *Elements of Combinatorial and Differential Topology*, Grad. Stud. Math. 74, AMS, Providence, RI, 2006.

[38] E. Tonti, *On the Formal Structure of Physical Theories*, Monogr. Ital. Nat. Res. Council (CNR), 1975; also available online at http://www.dic.univ.trieste.it/perspage/tonti/papers.htm.

[39] J. Dodziuk, *Finite difference approach to the Hodge theory of harmonic forms*, Amer. J. Math., 98 (1976), pp. 79–104.

[40] G. Strang, *Linear Algebra and its Applications*, 3rd ed., Brooks/Cole, Pacific Grove, CA, 2003.

[41] K. Preis, I. Bardi, 0. Biro, C. Magele, G. Vrisk, and K.R. Richter, *Different finite element formulations of 3D magnetostatic fields*, IEEE Trans. Magn., 28 (1992), pp. 1056–1059.

[42] P. Dular, F. Henrotte, F. Robert, A. Genon, and W. Legros, *A generalized source magnetic field calculation method for inductors of any shape*, IEEE Trans. Magn., 33 (1997), pp. 1398–1401.

[43] *Computer Assisted Proofs in Dynamics*, capd.wsb-nlu.edu.pl.

[44] *Chomp Library*, chomp.rutgers.edu.

[45] *JPlex Library*, comptop.stanford.edu/programs/jplex.

[46] *LinBox Library*, www.linalg.org.
[47] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed., MIT Press, Cambridge, MA, 2001.
[48] P. Dłotko and R. Specogna, Matlab *Implementation of the STT and GSTT Algorithm*, http://www.diegm.uniud.it/elettrotecnica/web/tc/ or http://www.comphys.com/tc/.
[49] R.H. Bing, *Some aspects of the topology of* 3*-manifolds related to the Poincaré conjecture*, in Lectures on Modern Mathematics II, T.L. Saaty, ed., Wiley, New York, 1964, pp. 93–128.
[50] P.R. Kotiuga, *Erratum: Toward an algorithm to make cuts for magnetic scalar potentials in finite element meshes*, [J. Appl. Phys., 63 (1988), pp. 3357–3359], J. Appl. Phys., 64 (1988), p. 4257.