Lazy Cohomology Generators: A Breakthrough in (Co)homology Computations for CEM

Paweł Dłotko[1,2] and Ruben Specogna[3]

[1]University of Pennsylvania, Department of Mathematics, Philadelphia, PA 19104 USA
[2]Jagiellonian University, Institute of Computer Science, Krakow 31-007, Poland
[3]Università di Udine, Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica, Udine 33100, Italy

Computing the first cohomology group generators received great attention in computational electromagnetics as a theoretically sound and safe method to produce cuts required when eddy-current problems are solved with the magnetic scalar potential formulations. This paper exploits the novel concept of lazy cohomology generators and a fast and general algorithm to compute them. This graph-theoretic algorithm is much faster than all competing ones being the typical computational time in the order of seconds even with meshes formed by millions of elements. Moreover, this paper introduces the use of minimal boundary generators to ease human-based basis selection and to obtain representatives of generators with compact support. We are persuaded that this is the definitive solution to this long-standing problem.

Index Terms— **Cohomology basis selection, (co)homology, first cohomology group lazy generators, magnetic scalar potential, magneto-quasi-statics, minimal cohomology basis, thick cuts.**

I. Introduction

BESIDES many attempts to circumvent it, (co)homology is recognized to be the only safe tool to define electromagnetic potentials in the nontrivial cases [1], [2]. A remarkable example is the potential definition for eddy-current problems involving topologically nontrivial conductors formulated with the magnetic scalar potential. In this case, the first cohomology group generators have to be considered [2], [3]. Being impossible in practice to construct them by hand, it is natural to search for an algorithm to do it automatically.

For 2-D problems, it has been recently shown in [4] that an optimal graph-theoretic algorithm exists that exhibits linear complexity and produces optimal cohomology generators. For 3-D problems, it seems to be much harder. In principle, cohomology generators over integers—unlike the real and complex ones—can be rigorously computed in polynomial time with the aid of the so-called Smith normal form (SNF) [5] of the coboundary matrix. However, this approach is not practical as its complexity is hypercubical. The exploitation of sparsity together with a number of reductions of the input complexes [1], [3], [6], [7] before applying the SNF, allows the practical solution of the problem even on meshes with millions of elements. What is not appealing is that this process in most of the cases takes much more than the time required by the remaining part of the simulation. This fact does not help to impose (co)homology as the best practice to defining potentials in computational electromagnetics and encourages naïve and patently incorrect solutions (as the ones surveyed in [2]). Moreover, the implementation of topological procedures is a rather complicated issue that usually confine this research to a few state-of-the-art softwares [6], [7].

The need for a dramatic speed up and the dream of a completely graph-theoretic algorithm to get rid of the SNF core, have lead us to an algorithm called thinned current technique [8]. This algorithm is easy to implement and orders of magnitude faster than its competitors, but it assumes to deal with the conductors that retract to a graph. If this is not the case as, for example, considering the region between the two coaxial toroids as a conductor, it loses all of its advantages.

The first aim of this paper is to present an extremely fast, general, and graph-theoretic algorithm to solve this problem called the Dłotko–Specogna (DS) algorithm [9]. This algorithm, in its simpler version, does not produce a standard cohomology basis since the output 1-cocycles are linearly dependent (and some of them may be even cohomologically trivial). However, since they span the first cohomology group, they are referred to as lazy cohomology generators [9]. It is certainly possible to produce a cohomology basis with the DS algorithm (see details in [9]), but this requires an additional computational time and coding effort. Instead of doing so, the very idea of lazy cohomology generators is that they may be directly employed in eddy-current formulations [9].

The second aim of this paper is to introduce the use of minimal boundary generators in the DS algorithm to ease human-based basis selection and to obtain representatives of generators with compact support. This way, the user of the electromagnetic software would be able to easily relate a representative of the cohomology basis to the relevant constraint on the boundary value problem [7]. Therefore, basis selection is fundamental, for example, to impose source current constraints or to perform some postprocessing.

This paper is structured as follows. In Section II, the DS algorithm is recalled. In Section III, it is intuitively explained why one may safely use lazy generators instead of cohomology basis in the EM computations. Section IV addresses how to obtain minimal generators on the boundary of conductor. Finally, in Section V, some numerical results are presented. They clearly show the superiority of the proposed technique over any other method.

II. DS Algorithm

Let $\mathcal{K}$ be a homologically trivial polyhedral cell complex in $\mathbb{R}^3$ representing the domain of the eddy-current problem. Let us consider two subcomplexes $\mathcal{K}_c$ and $\mathcal{K}_a$ that contain elements belonging to the conducting and insulating regions, respectively. Both $\mathcal{K}_c$ and $\mathcal{K}_a$ are combinatorial manifolds with boundary [9]. We denote the value of the cochain $\mathbf{t}$ on a cell

Manuscript received June 23, 2013; revised August 6, 2013; accepted September 4, 2013. Date of current version February 21, 2014. Corresponding author: R. Specogna (e-mail: ruben.specogna@uniud.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMAG.2013.2281076

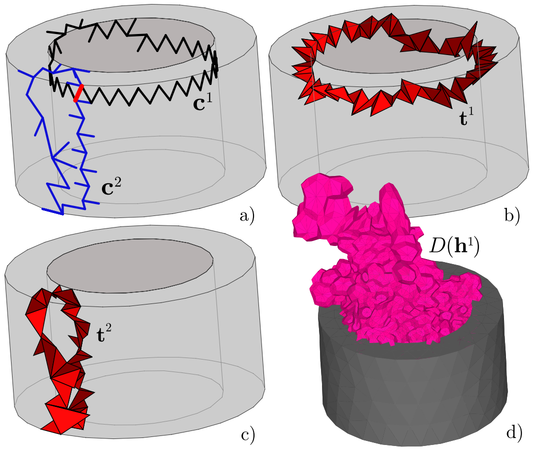

Fig. 1. (a) Two cohomology generators $\mathbf{c}^1$ and $\mathbf{c}^2$ for the boundary of a solid two-torus. Support of the thinned currents (b) $\mathbf{t}^1$ and (c) $\mathbf{t}^2$ corresponding to $\mathbf{c}^1$ and $\mathbf{c}^2$, respectively. (d) Two-chain $D(\mathbf{h}^1)$ on the complex dual to $\mathcal{K}_a$ is the dual with respect to the cohomology generator $\mathbf{h}^1$.

E by $\langle \mathbf{t}, E \rangle$, the genus of a surface $\mathcal{C}$ by g and the incidence between the cells A and B by $\kappa(A, B)$. The DS algorithm is presented in Algorithm 1. For a more detailed explanation, please refer to [9].

Fig. 1(a) shows the representatives $\mathbf{c}^1$, $\mathbf{c}^2$ of the two cohomology $H^1(\mathcal{C}, \mathbb{Z})$ generators, $\mathcal{C}$ being the boundary of a solid two-torus. The corresponding $\mathbf{t}^1$, $\mathbf{t}^2$, called thinned currents [9], are shown in Figs. 1(b) and (c). Fig. 1(d) shows the two-chain $D(\mathbf{h}^1)$ on the complex dual to $\mathcal{K}_a$ is the dual with respect to the cohomology generator $\mathbf{h}^1$, where by $D(\cdot)$, we denote the dual map [9].

It is clear that the cocycles obtained by Algorithm 1 are not a cohomology $H^1(\mathcal{K}_a, \mathbb{Z})$ basis, since they are twice the cardinality of such a basis. Since the cocycles obtained from the DS algorithm are indeed able to span $H^1(\mathcal{K}_a, \mathbb{Z})$, they are lazy cohomology generators [9]. It is clear from the example in Fig. 1 that the thinned current $\mathbf{t}^1$ provides the standard generator $\mathbf{h}^1$, whereas the thinned current $\mathbf{t}^2$ may be eliminated since its corresponding lazy generator $\mathbf{h}^2$ is cohomologically trivial. Since decide whether to remove or not a lazy generator is costly and complicated,[1] we bypass this additional step.

Let n denote the number of connected components of $\mathcal{K}_c \cap \mathcal{K}_a$. Finding $\mathbf{c}^1, \dots, \mathbf{c}^{2g}$ is performed in $O(\text{card}(\mathcal{K}_c \cap \mathcal{K}_a)g)$ time.[2] In the practical implementation, the ESTT algorithm is *vectorialized*, i.e., the ESTT algorithm is applied to all $\mathbf{t}^1, \dots, \mathbf{t}^{2g}$ thinned currents produced for all n connected components of $\mathcal{K}_c \cap \mathcal{K}_a$ at the same time. Algorithmically, it means performing the operations with a vector of $2gn$ real numbers instead of a scalar in the ESTT algorithm. In this case, the typical complexity of ESTT algorithm is $O(\text{card}(\mathcal{K})gn)$. If the genus g and the number of connected components n is bounded by $O(1)$, as it happens always in practical problems, the average complexity of the DS algorithm is linear $O(\text{card}(\mathcal{K}))$ when standard surface generators are used and $O(\text{card}(\mathcal{K}) \log(\text{card}(\mathcal{K})))$ when minimal surface generators are used, as described in Section IV.

Algorithm 1 DS Algorithm

```
Input: Topologically trivial complex $\mathcal{K} = \mathcal{K}_c \cup \mathcal{K}_a$;
Output: Lazy cohomology generators of $\mathcal{K}_a$;
  $L$ — list of 1-cocycles being lazy generators of $H^1(\mathcal{K}_a)$;
  for every $\mathcal{C}$, connected component of $\mathcal{K}_c \cap \mathcal{K}_a$ do
    Compute $\mathbf{c}^1, \dots, \mathbf{c}^{2g}$, the generators of $H^1(\mathcal{C}, \mathbb{Z})$, as described in [11] (or in Section IV once minimal generators are needed);
    $\mathbf{t}^1, \dots, \mathbf{t}^{2g}$ — 2-cocycles in $\mathcal{K}$ initialized to zero;
    for $i = 1$ to $2g$ do
      for each 1-cell $E \in C$ such that $\langle \mathbf{c}^i, E \rangle \neq 0$ do
        for each 2-cell $T \in \mathcal{K}_c$ with $E$ in the boundary do
          $\langle \mathbf{t}^i, T \rangle += \langle \mathbf{c}^i, E \rangle \kappa(T, E)$;
    Solve in $\mathcal{K}$ the integer systems $\mathbf{C}\,\mathbf{h}^j = \mathbf{t}^j$, $j \in \{1, \dots, 2g\}$, $\mathbf{C}$ being the face-edge incidence matrix, using a vectorialized version of ESTT algorithm [12]. In this way, $2g$ 1-cocycles $\mathbf{h}^1, \dots, \mathbf{h}^{2g}$ in $\mathcal{K}$ are obtained;
    Restrict $\{\mathbf{h}^1, \dots, \mathbf{h}^{2g}\}$ to $\mathcal{K}_a$;
    $L = L \cup \{\mathbf{h}^1, \dots, \mathbf{h}^{2g}\}$;
  Return $L$;
```

III. Lazy Cohomology Generators in Eddy-Current Formulations

Surprisingly, the use of lazy cohomology generators in electromagnetic modeling has not been explored in the literature. On the contrary, the procedure of removing redundant generators has been always deemed as necessary. Instead, lazy cohomology generators may be safely used in FEM or geometric formulations as if they were a standard cohomology basis [9].

In this paper, an intuitive explanation is presented by showing the relationship between the lazy cohomology generators and gauging. The system of equations to solve before adding nonlocal Faraday's equations is already rank deficient [2], [3], [13], but consistent. Indeed, adding an arbitrary 1-coboundary $\mathbf{GW}$ to the electric vector potential $\mathbf{T}$ does not alter the current $\mathbf{I}$, since $\mathbf{I} = \mathbf{CT} = \mathbf{C}(\mathbf{T} + \mathbf{GW})$ and $\mathbf{CG} = \mathbf{0}$ [3], where $\mathbf{W}$ is an arbitrary 0-cochain and $\mathbf{G}$ is the edge-node incidence matrix. Even though a full-rank system may be obtained by a tree-cotree gauging (i.e., by setting the electric vector potential on a tree of one-cells in the interior of $\mathcal{K}_c$ to zero [13] and by eliminating the corresponding local Faraday's equations), it is widely known that with iterative linear solvers ungauged [13] formulations are much more efficient. Ungauged formulations work because, even if the vector potential is not unique, its curl—i.e., the current density—is. Since the solution in terms of induced currents does not depend on the representatives of the fixed cohomology basis, one may use any representative of the given basis. Therefore, any cohomologically trivial lazy generator may be expressed by a 1-coboundary $\mathbf{GW}$ in $\mathcal{K}$ that does not alter the solution in term of induced currents for the same reasons why ungauged formulations work.

Let us now see the other possible case, namely when lazy generators are dependent but nontrivial in $H^1(\mathcal{K}_a, \mathbb{Z})$. To this aim, let us assume, for example, that $\hat{\mathbf{c}}^1 = \mathbf{c}^1$ and $\hat{\mathbf{c}}^2 = \mathbf{c}^1 + \mathbf{c}^2$ in Fig. 1(a) are the representatives of $H^1(\mathcal{K}_c \cap \mathcal{K}_a, \mathbb{Z})$ generators. Thus, by starting from the corresponding thinned currents, a standard $H^1(\mathcal{K}_a, \mathbb{Z})$ generator $\hat{\mathbf{h}}^1$ is obtained together with

[1] A cohomology basis may be obtained by computing linking numbers and a SNF of a small matrix. Since routines to perform these two steps are not available in FEM codes, they require a considerable coding effort.

[2] When using not minimal boundary generators. The complexity is $O(\text{card}(\mathcal{K}_c \cap \mathcal{K}_a) \log(\text{card}(\mathcal{K}_c \cap \mathcal{K}_a))g)$ when using minimal generators starting from a constant number of boundary triangles chosen with random or maxmin strategy. For details, refer to Section IV.

another lazy generator $\hat{\mathbf{h}}^2$ in the same class. Then, since the coboundary of the magnetomotive force $\mathbf{F} = \mathbf{T} + \mathbf{G}\Omega$ is the current, $\mathbf{I} = \mathbf{CF} = \mathbf{C}(\mathbf{T} + \mathbf{G}\Omega + \hat{i}_1\hat{\mathbf{h}}^1 + \hat{i}_2\hat{\mathbf{h}}^2) = \mathbf{C}(\mathbf{T}+(\hat{i}_1+\hat{i}_2)\mathbf{h}^1)$, where we used the standard generator $\mathbf{h}^1$ in place of the two lazy generators in the same class. Since with the standard basis $\mathbf{I} = \mathbf{C}(\mathbf{T} + i_1\mathbf{h}^1)$ and from Ampère's law $i_1 = \hat{i}_1 + \hat{i}_2$ implicitly, the two solutions in term of induced currents are again the same.

To conclude, the use of lazy cohomology generators does not introduce any inconsistency in the formulation and does not yield to any penalties in the computational time required to solve the eddy-current problem by the linear system solver.

IV. Globally Minimal Cohomology Generators for Human-Based Basis Selection

This section introduces an algorithm to compute minimal (with respect to length of dual cycles) representatives of $H^1(\mathcal{K}_c \cap \mathcal{K}_a)$ generators on the conductor boundary. The Algorithm 2 presented in this paper is inspired by [14], but some new ideas introduced in this paper provide a significant improvement.

The algorithm iterates through all the triangles of every connected component C of $\mathcal{K}_c \cap \mathcal{K}_a$. For each triangle v, the minimal distance spanning tree on $D(C)$ rooted at v is provided by Dijkstra algorithm [15]. Then the algorithm computes the $h^1, \ldots, h^{2g}$ being a $H^1(C)$ basis. Since for large meshes, it is not possible to store the $h^1, \ldots, h^{2g}$ obtained for every triangle, for every cohomology generator h^k we store only:

1) the length of h^k;
2) its coordinates, i.e., the index of the triangle v in the connected component C and k, the number of the generator h^k rooted in v;
3) dot product of h^k with a fixed $H_1(C)$ basis.

One can prove using the universal coefficient theorem for cohomology [10] that the intersection of h^k with the fixed $H_1(C)$ basis uniquely determines the cohomology class of the cocycle.

Finally, the set of all triples representing generators (obtained from every triangle in $\mathcal{K}_c \cap \mathcal{K}_a$) is sorted from the shortest to the longest one. Then, by starting from a void list G of shortest generators, all generators are considered from the shortest to the longest. A generator is added to the list G only if it is independent from the generators already in G. This may be performed by testing the rank of the nonsquare matrix $\mathbf{M}$ containing the evaluations on the homology basis of the generators in G plus the ones of the considered generator, or by testing if one of the eigenvalues of the square symmetric matrix $\mathbf{MM}^T$ is zero. Since the dimensions of these matrices are quite small, this part takes a negligible time in practice.

We remark that there is a lot of room for parallelization of the presented algorithm. First of all, each connected component C of $\mathcal{K}_c \cap \mathcal{K}_a$ may be processed independently. Second, provided suitable data structures are used to store primal and dual trees on C, each triangle in C may be processed independently from any other triangle in C.

The presented algorithm provides the globally shortest set of $H^1(\mathcal{K}_c \cap \mathcal{K}_a)$ generators. Of course, in practice it is enough to consider in Algorithm 2 just a small, uniformly distributed, subset of all triangles in C (instead of using all of them.) Doing so, probably one would not get the absolute minimal generators, but a set of generators quite close to the minimal ones that are surely good enough for the purpose of ease the basis selection. In the experiments, $5g$ triangles are usually sufficient, where g is the genus of C. To choose the subset of triangles one may employ the following two methods. Both are inspired by strategies of choosing the so-called landmark points in topological data analysis [16].

Algorithm 2 Computing Minimal Length Cohomology Generators of $\mathcal{K}_c \cap \mathcal{K}_a$

```
Input: $\mathcal{K}_c \cap \mathcal{K}_a$;
Output: Shortest basis of $H^1(\mathcal{K}_c \cap \mathcal{K}_a)$;
  $A = \emptyset$ — constructed minimal length basis of $H^1(\mathcal{K}_c \cap \mathcal{K}_a)$;
  for every connected component $C$ of $\mathcal{K}_c \cap \mathcal{K}_a$ do
    Compute $h_1, \ldots, h_{2g}$ — a basis of $H_1(C)$ with a tree-cotree technique presented in [14];
    Let $M$ be a matrix of the length $m \times 2g$ ($m$ is the number of triangles in $C$). Each entry is a triple: a real number (called $dist$), a pair of integers called $coef$ and a vector of $2g$ integers called $intersect$;
    $i = 0$;
    for every triangle $v$ in $C$ do
      With Dijkstra algorithm construct minimal distance spanning tree $T'$ on $D(\mathcal{K}_c \cap \mathcal{K}_a)$ rooted in $v$;
      Construct any spanning tree $T$ on $\mathcal{K}_c \cap \mathcal{K}_a$ such that $T \cap T' = \emptyset$;
      Based on $T$ and $T'$ construct $h^1, \ldots, h^{2g}$, a $H^1(\mathcal{K}_c \cap \mathcal{K}_a)$ basis as described in [11];
      for $j = 1$ to $2g$ do
        for $k = 1$ to $2g$ do
          $M[i][j].intersect[k] = \langle h^j, h_k \rangle$;
        $M[i][j].dist =$length of $D(h^j)$;
        $M[i][j].coef = (i, j)$;
      $i = i + 1$;
    Let $SM$ be a vector of all elements of $M$;
    Sort all the generators in $SM$ according to $dist$ parameter from the shortest to the longest one;
    $G = \emptyset$ — a vector of triples (as above) representing the constructed minimal length $H^1(\mathcal{K}_c \cap \mathcal{K}_a)$ basis;
    for $i = 1$ to $2gm$ do
      Let $N$ be the matrix constructed based on $G[k].intersect$ for $k \in \{1, \ldots, \mathrm{card}(G)\}$;
      Let $N'$ be the matrix constructed based on $G[k].intersect$ for $k \in \{1, \ldots, \mathrm{card}(G)\}$ and $SM[i].intersect$;
      if $rank(N) \neq rank(N')$ then
        $G = G \cup SM[i]$;
      if $\mathrm{card}(G) = 2g$ then
        Break;
    for $k = 1$ to $2g$ do
      $(i, j) = G[k].coef$;
      $h^k =$ recompute the $j$th cocycle for $i$th triangle of $C$;
      $A = A \cup h^k$;
  Return $A$;
```

1) *Random Selection:* Aims at randomly picking $5g$ triangles from the set of all triangles in C. This simple strategy usually gives good results and it is easy to implement.
2) *Maxmin Selection:* Aims at obtaining a set of triangles evenly distributed in the boundary of the conductors. Algorithm 3 shows the details of this technique.

Algorithm 3 Maxmin Selection

```
Input: $\mathcal{T}$ — the set of all triangles in $C$;
Output: Set of $5g$ triangles uniformly distributed in $C$;
  Let $M$ be empty set of triangles;
  Pick a random triangle $T \in \mathcal{T}$. $M = \{T\}$;
  while cardinality of $M \leq 5g$ do
    Choose $T \in \mathcal{T}$ such that $T \notin M$ and $T$ maximizes the function $T' \to min_{T'' \in M}\{d(T', T'')\}$, where distance between two triangles is the Euclidean distance of their barycenters;
    $M = M \cup T$;
  Return $M$;
```

TABLE I

Time Required (in Seconds) for Cohomology Computation

Benchmark	millions of tetrahedra	$H^1(\mathcal{K}_a, \mathbb{Z})$ [6]	TCT [8]	DS lazy
trefoil knot	0.199	23	0.6	0.3
spiral	1.842	(612)	10.1	1.7
μ-inductor	2.197	(> 70000)	24.5	2.4
μ-transformer	2.582	(> 70000)	32.8	3.6
μ-coaxial line	4.861	(6128)	86.1	10.6
toroidal shell	2.769	(> 70000)	(> 70000)	3.4

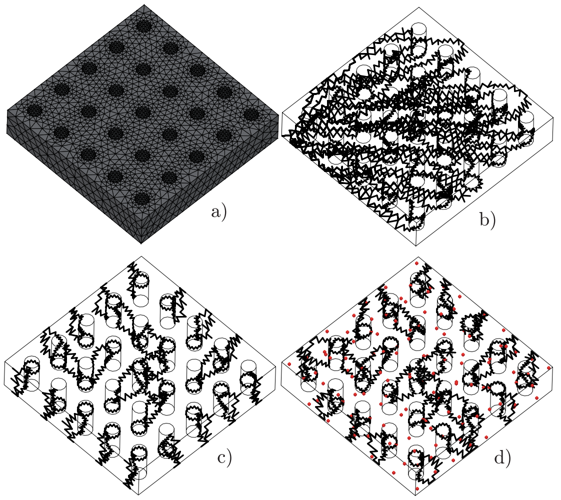

Fig. 2. (a) Mesh of a conductor with 25 holes. Support of the cohomology $H^1(\mathcal{K}_c \cap \mathcal{K}_a)$ generators representatives obtained with (b) one random basepoint, (c) globally minimal length cohomology basis, and (d) $5g = 125$ basepoints (shown in the picture) selected with maxmin strategy.

V. Numerical Experiments

Table I shows the comparison in term of computational timing of the best algorithms available in the literature in computing generators for six different eddy-current problems (For more information about the benchmarks, please refer to [8]). As one can observe from the timings, the DS algorithm outperforms all of its competing algorithms.

To test the idea of using minimal boundary generators in problems requiring human-based basis selection, a conductor with 25 holes is considered [see Fig. 2(a)]. Fig. 2(b) shows the support of $H^1(\mathcal{K}_c \cap \mathcal{K}_a)$ generators representatives obtained using one random basepoint in a few tens of milliseconds (this is the default of the DS algorithm). It is easy to observe that the basis is quite complicated and this may be a problem in case one requires to select a particular generator, for example, for the sake of enforcing some current. In Figs. 2(c) and (d), it is possible to observe the globally minimal generators—retrieved in less than 5 min—and the ones—computed in 2 s—obtained using 125 basepoints selected with Algorithm 3, respectively.

VI. Conclusion

The DS algorithm, even though general and straightforward to implement and parallelize, outperforms all competing state-of-the-art algorithms for first cohomology group computations, being the typical computational time in the order of seconds even for huge meshes. Therefore, we expect that the DS algorithm and the lazy cohomology generators will be observed as a major step forward in low-frequency computational electromagnetics since they allow to effectively solve what has been considered an open problem for more than 25 years.

References

[1] P. W. Gross and P. R. Kotiuga, *Electromagnetic Theory and Computation: A Topological Approach*, vol. 48. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[2] P. Dłotko and R. Specogna, "Cohomology in 3D magneto-quasistatic modeling," *Commun. Comput. Phys.*, vol. 14, no. 1, pp. 48–76, 2013.

[3] P. Dłotko, R. Specogna, and F. Trevisan, "Automatic generation of cuts on large-sized meshes for $T - \Omega$ geometric eddy-current formulation," *Comput. Meth. Appl. Mech. Eng.*, vol. 198, pp. 3765–3781, Jan. 2009.

[4] R. Specogna, "Optimal cohomology generators for 2-D eddy-current problems in linear time," *IEEE Trans. Magn.*, vol. 49, no. 4, pp. 1299–1304, Apr. 2013.

[5] J. Munkres, *Elements of Algebraic Topology*. Cambridge, MA, USA: Perseus Books, 1984.

[6] P. Dłotko and R. Specogna, "Efficient cohomology computation for electromagnetic modeling," *Comput. Model. Eng. Sci.*, vol. 60, no. 3, pp. 247–278, 2010.

[7] M. Pellikka, S. Suuriniemi, and L. Kettunen, "Powerful heuristics and basis selection bring computational homology to engineers," *IEEE Trans. Magn.*, vol. 47, no. 5, pp. 1226–1229, May 2011.

[8] P. Dłotko and R. Specogna, "A novel technique for cohomology computations in engineering practice," *Comput. Meth. Appl. Mech. Eng.*, vol. 253, pp. 530–542, Jan. 2013.

[9] P. Dłotko and R. Specogna, "Physics inspired algorithms for (co)homology computations of three-dimensional combinatorial manifolds with boundary," *Comput. Phys. Commun.*, vol. 184, no. 10, pp. 2257–2266, 2013.

[10] A. Hatcher, *Algebraic Topology*. Cambridge, U.K.: Cambridge Univ. Press, 2001.

[11] P. Dłotko, "A fast algorithm to compute cohomology group generators of orientable 2-manifolds," *Pattern Recognit. Lett.*, vol. 33, no. 11, pp. 1468–1476, 2012.

[12] P. Dłotko and R. Specogna, "Efficient generalized source field computation for h-oriented magnetostatic formulations," *Eur. Phys. J., Appl. Phys.*, vol. 53, no. 2, pp. 20801–20807, 2011.

[13] Z. Ren, "Influence of the RHS on the convergence behaviour of the curl-curl equation," *IEEE Trans. Magn.*, vol. 32, no. 3, pp. 655–658, May 1996.

[14] J. Erickson and K. Whittlesey, "Greedy optimal homotopy and homology generators," in *Proc. 16th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2005, pp. 1038–1046.

[15] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA, USA: MIT Press, 2001.

[16] V. de Silva and G. Carlsson, "Topological estimation using witness complexes," in *Proc. 1st Eurograph. Conf. Point-Based Graph.*, 2004, pp. 157–166.