Lazy cohomology generators: a breakthrough in (co)homology computations for CEM

Paweł Dłotko[♭], Ruben Specogna[♯]
[♭]Jagiellonian University, Institute of Computer Science, Lojasiewicza 6, 30348 Kraków, Poland
[♯]Università di Udine, Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica, Via delle Scienze 206, 33100 Udine, Italy
ruben.specogna@uniud.it

***Abstract*—Computing the first cohomology group received great attention in computational electromagnetics as a theoretically sound and safe method to produce *cuts* required when eddy-current problems are solved with magnetic scalar potential formulations. In this paper we introduce the novel concept of *lazy cohomology generators*, that are cocycles which span the first cohomology group but are not a basis. After showing that they fit nicely in ungauged h-oriented eddy-current formulations, we dwell upon a fast 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. We are persuaded that this is the definitive solution to this long-standing problem.**

***Index Terms*—magneto-quasistatics, magnetic scalar potential, thick cuts, (co)homlogy, first cohomology group lazy generators.**

I. Introduction

Besides many attempts to circumvent it, (co)homology is recognized to be the only safe tool to define potentials in the nontrivial cases [1], [2]. In this paper we focus on eddy-current problems, even though the novel idea of lazy cohomology generators is much broader in scope. When solving eddy-current problems involving topologically non-trivial conductors with an efficient h-oriented formulation, to render the problem well defined, the first cohomology group generators have to be computed [1], [2]. Being impossible in practice to construct them by hand, it is natural to search for an algorithm to do it automatically.

For 2d problems, it has been recently shown in [3] that an optimal graph-theoretic algorithm exists that exhibits linear complexity and produces optimal cohomology generators. For 3d 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) [4] of the coboundary matrix. However, this approach is not feasible in practice as its complexity is hyper-cubical using the best implementation. The exploitation of sparsity together with a number of reductions of the input complexes [1], [5], [6] 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 cases takes 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 few state-of-the-art softwares [5], [6].

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 (TCT) [7]. This algorithm is easy to implement and order of magnitude faster than its competitors being its complexity linear with the number of elements in the mesh. Nonetheless, it assumes to deal with conductors that retract to a graph. If this is not the case in the actual eddy-current problem, it automatically switches to the algorithm described in [5]. This prerequisite may be too stringent for some topologically complicated conductors.

The aim of this paper is to cover this gap by introducing an extremely fast, general and graph-theoretic algorithm to solve this problem called the Dłotko–Specogna (`DS`) algorithm [8]. This algorithm, in its simpler version, does not produce a standard cohomology basis since the output 1-cocycles may be linearly dependent or cohomologically trivial. However, since they are still able to span the first cohomology group, we call them *lazy cohomology generators*. It is certainly possible to produce a cohomology basis with the `DS` algorithm (see details in [8]), but this requires additional computational time and coding efforts. Instead doing so, the very idea of this paper is to show that lazy cohomology generators may be directly employed in ungauged eddy-current formulations.

In this paper we present the `DS` algorithm to produce lazy generators together with timings that indeed indicate the superiority of the new algorithm. Later we show how this kind of generators are used in ungauged eddy-current formulations. We leave more details and insights on the algorithm together with a more extensive comparison for the full paper.

II. `DS` algorithm

Let $\mathcal{K}$ be a homologically trivial polyhedral cell complex in $\mathbb{R}^3$ representing the domain under study. Let us consider the two sub-complexes $\mathcal{K}_c$ and $\mathcal{K}_a$ that contain elements belonging to the conducting and insulating regions, respectively.

1) Compute the 1st cohomology $H^1(C, \mathbb{Z})$ generators $\mathbf{c}^1, \ldots, \mathbf{c}^{2g}$, where C is $\partial\mathcal{K}_c \setminus \partial\mathcal{K}$ and g denotes the

genus of C. This can be performed in linear time worst-case complexity $O(card(\partial\mathcal{K}_c)\, g)$ with the graph-theoretic algorithm presented in [9].

2) Find the $\mathbf{t}^1, \ldots, \mathbf{t}^{2g}$ corresponding to $\mathbf{c}^1, \ldots, \mathbf{c}^{2g}$ in $O(card(\partial\mathcal{K}_c)\, g)$ with the following algorithm:
   for each 1-cell E with nonzero coefficient c_E in $\mathbf{c}^i$
   for each 2-cell $T \in \mathcal{K}_c$ with E in the boundary
   $\langle \mathbf{t}^i, T\rangle += c_E \kappa(T, E)$;
   The value of the cochain $\mathbf{t}$ on a cell E is $\langle \mathbf{t}, E\rangle$, whereas $\kappa(A, B)$ denotes the incidence between cells A and B. Initially, set $\langle \mathbf{t}^i, T\rangle = 0$ for all 2-cells $T \in \mathcal{K}_c$.
3) Solve the integer systems $\delta\, \mathbf{h}^j = \mathbf{t}^j$, $j \in \{1, \ldots, 2g\}$, δ being the coboundary matrix, to find the $2g$ 1-cocycles $\mathbf{h}^1, \ldots, \mathbf{h}^{2g}$ in $\mathcal{K}_a$. This can be performed without solving any system by a simultaneous application of the ESTT algorithm [10], [7]. Simultaneous means that the ESTT algorithm is applied to all $\mathbf{t}^1, \ldots, \mathbf{t}^{2g}$ thinned currents at the same time. Algorithmically this can be easily achieved by changing a real number to a vector of $2g$ real numbers in the ESTT algorithm.
4) Store the restrictions of $\mathbf{h}^1, \ldots, \mathbf{h}^{2g}$ to $\mathcal{K}_a$. The average computational effort required is $O(card(\mathcal{K})\, g)$.

If the genus g is bounded by a constant $O(1)$, as it happens always in practical problems, the average complexity of the DS algorithm is linear $O(card(\mathcal{K}))$.

III. Lazy cohomology generators and ungauged formulations

Lazy cohomology generators are employed in a MQS formulation, for example the $\mathbf{T}$-$\mathbf{\Omega}$ [2], as if they were a set of standard $H^1(\mathcal{K}_a, \mathbb{Z})$ generators. Namely, a nonlocal Faraday's equation [2] is written on the support of the j-th cohomology generator as $\langle \tilde{\mathbf{U}}, \partial\tilde{h}_j\rangle = -i\,\omega\,\langle \tilde{\mathbf{\Phi}}, \tilde{h}_j\rangle$, where $\tilde{\mathbf{U}}$ is the electro-motive force 1-cochain on the dual complex, $\tilde{\mathbf{\Phi}}$ is the magnetic flux 2-cochain on the dual complex and $\tilde{h}_j = D(\mathbf{h}^j)$, D being the *dual map* [4] that maps elements of the original complex to elements of the dual complex. Lazy cohomology generators contain a $H^1(\mathcal{K}_a, \mathbb{Z})$ basis and generators that are dependent to the basis. Therefore, adding the dependent equations is not a problem considering that the system of equations to solve is already overdetermined. This is due to the fact that algebraic Faraday's equations [2] enforced in $\mathcal{K}_c$ are also dependent. Even though a full-rank system may be obtained by a tree-cotree gauging (i.e. set the electric vector potential on a tree of 1-cells in $\mathcal{K}_c$ to zero), it is widely known that with iterative linear solvers it is much more efficient to use an *ungauged* formulation. What is important from the modeling point of view is that even if the potentials are not unique, the fields are. Therefore, the use of linearly dependent cocycles in the physical modeling does not introduce either any inconsistency in the formulation of the boundary value problem or any penalties in the computational time employed by the simulation due, for example, to a hypothetical increase of the condition number of the linear system matrix or to the use of twice as many cohomology generators as needed.

As a final observation about lazy generators, let us now consider a lazy generator belonging to the trivial class of $H^1(\mathcal{K}_a, \mathbb{Z})$. Given an arbitrary 1-cycle $c \in Z_1(\mathcal{K}_a)$, the dot product of the lazy generator with c is zero. Therefore, trivial generators verify trivially the nonlocal algebraic Ampère's law and, in this case, the current i_j does not represent the current linked by the dual homology generator. Therefore, the value of the independent current relative to a trivial generator is not unique and it is determined by the solution of the system of equations. This is not surprising, since the independent current in this case does not have a physical meaning.

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

Benchmark	tetrahedra	$H^1(\mathcal{K}_a, \mathbb{Z})$ [5]	TCT [7]	DS lazy
trefoil knot	199,208	23	0.6	0.3
spiral	1,842,070	(612)	10.1	1.7
micro-inductor	2,197,192	(> 70000)	24.5	2.4
micro-transformer	2,582,830	(> 70000)	32.8	3.6
micro-coaxial line	4,861,655	(6128)	86.1	10.6
toroidal shell	2,769,200	(> 70000)	(> 70000)	3.4

IV. Numerical experiments

Table I shows the comparison in term of computational timing of the best algorithms available in literature in computing generators for six different eddy-current problems. As one can see from this results, the DS algorithm outperforms all its competing algorithms demonstrating its extreme usefulness in computational electromagnetics.

We are persuaded that lazy generators will be seen as a major step forward in the state-of-the-art of (co)homology computations for electromagnetic modeling since they provide what we consider the definitive solution to the long-standing open problem of computing cohomology generators for low-frequency electrodynamics.

References

[1] P.W. Gross and P. R. Kotiuga. *Electromagnetic Theory and Computation: A Topological Approach*. MSRI Vol. 48, Cambridge University Press, 2004.

[2] P. Dłotko and R. Specogna. Cohomology in 3d magneto-quasistatic modeling. *Commun. Comput. Phys. (arXiv:1111.2374)*, 14:48–76, 2013.

[3] R. Specogna. Optimal cohomology generators for 2d eddy-current problems in linear time. *IEEE Trans. Magn.*, DOI: 10.1109/TMAG.2012.2222905, in press, 2013.

[4] JR. Munkres. *Elements of algebraic topology*. Perseus Books, Cambridge, MA, 1984.

[5] P. Dłotko and R. Specogna. Efficient cohomology computation for electromagnetic modeling. *CMES*, 60:247–278, 2010.

[6] M. Pellikka, S. Suuriniemi, and L. Kettunen. Powerful heuristics and basis selection bring computational homology to engineers. *IEEE Trans. Magn.*, 47:1226–1229, 2011.

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

[8] P. Dłotko and R. Specogna. Physics inspired algorithms for (co)homology computation. *(arXiv:1212.1360)*, 2012.

[9] P. Dłotko. A fast algorithm to compute cohomology group generators of orientable 2-manifolds. *Pattern Recogn. Lett.*, 33:1468–1476, 2012.

[10] P. Dłotko and R. Specogna. Efficient generalized source field computation for *h*-oriented magnetostatic formulations. *Eur. Phys. J.-Appl. Phys.*, 53:20801, 2011.