

Lean cohomology computation for electromagnetic modeling

Paweł Dłotko¹, Bernard Kapidani², Ruben Specogna²

¹DataShape, Inria Saclay, Palaiseau 91120, France

²Polytechnic Department of Engineering and Architecture (DPIA), University of Udine, 33100 Udine, Italy
ruben.specogna@uniud.it

Physics-inspired algorithms as the Dłotko–Specogna (DS) has been shown to be extremely fast to perform the topological pre-processing required to solve challenging eddy current problems formulated by using a magnetic scalar potential in the insulator. Yet, they produce efficiently a set of *lazy* generators instead of a regular cohomology basis. A regular cohomology basis is favourable over lazy generators given that it reduces the number of unknowns in the linear system and also produces a full-rank system matrix.

This paper extends the DS algorithm in such a way that it provides a regular basis of the first cohomology group as output at a negligible computational cost. The obtained speedup with respect to the best alternative algorithm is more than two order of magnitudes on a challenging benchmark problem, demonstrating the potential impact of the proposed contribution in the low-frequency computational electromagnetics community.

Index Terms—eddy currents, magnetic scalar potential, cuts, cohomology, first de Rham cohomology group

I. INTRODUCTION

LET US assume that the computational domain D is a topologically trivial 3-manifold with boundary embedded in \mathbb{R}^3 . D is covered with a cell complex $\mathcal{K} = \mathcal{K}_a \cup \mathcal{K}_c$, where \mathcal{K}_a and \mathcal{K}_c are two sub-complexes of \mathcal{K} representing the insulating and conducting regions, respectively.

This contribution introduces an important extension in the Dłotko–Specogna (DS) algorithm [1], [2] to efficiently compute a regular cohomology $H^1(\mathcal{K}_a)$ basis of the insulating domain in negligible time. In order to introduce the novel algorithm we start by recalling the DS algorithm [1]:

- 1) The first cohomology group generators of the discrete surface $\mathcal{S} = \mathcal{K}_c \cap \mathcal{K}_a$ (see Fig. 1a) are computed with a linear time combinatorial algorithm, see [1].
- 2) The Hiptmair–Ostrowski (HO) technique [3] adapted as described in [1] is used to construct a set of cohomology generators on \mathcal{S} such that the dual on \mathcal{S} of half of them bound inside \mathcal{K}_c and the remaining ones whose dual bound in \mathcal{K}_a . Only the former half of generators are used in what follows.
- 3) *Thinned currents* are found by pre-multiplying the half generators of \mathcal{S} by the incidence matrix \mathbf{C}_c between face and edge pairs [1] restricted to \mathcal{K}_c . The support of the thinned current of a toric conductor is represented in Fig. 1a by dark faces.
- 4) Finally, a vectorialized version of the Extended Spanning Tree Technique (ESTT) algorithm [4] is run on the whole complex \mathcal{K} for all thinned currents at once. The ESTT algorithm is a general version of the Webb–Forghani (WF) iterative algorithm [5] to obtain a discrete field whose discrete curl is assigned (in our case to the curl of the thinned current). The output of the ESTT restricted to \mathcal{K}_a form the required cohomology generators (Fig. 1d).

The dual of the thinned current forms a 1-cycle \tilde{c} on the dual complex, see the thick edges in Fig. 1b. An important interpretation that is going to be used later is that the ESTT is

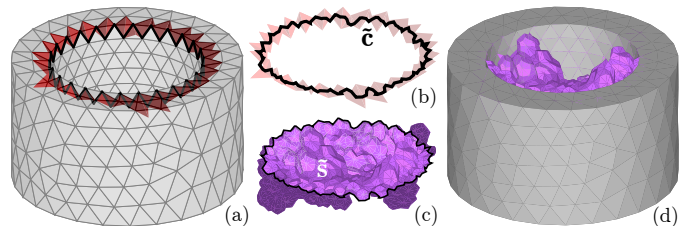


Fig. 1. (a) The thick edges represent the support of a cohomology generator of the boundary of \mathcal{K}_c , a solid 2-torus. The dark triangles represent the support of the thinned current. (b) The dual edges which are dual to thinned current faces form a 1-cycle \tilde{c} on the dual complex. (c) This dual 1-cycle \tilde{c} is the boundary of a 2-chain \tilde{s} on the dual complex. (d) \tilde{s} , restricted to \mathcal{K}_a , is the dual of the cohomology generator of \mathcal{K}_a .

computing a discrete surface \tilde{s} on the dual complex (possibly self-intersecting) having \tilde{c} as boundary, see Fig. 1c.

The HO technique finds the required change of cohomology basis by computing the null-space of a small and sparse matrix (the dimension is $2g \times 2g$, g being the genus of \mathcal{S}) [3]. The bottleneck of this process is the computation of the entries of this matrix, which are all the mutual linking numbers between the dual of the cohomology generators of \mathcal{S} and the dual of the thinned currents (\tilde{c} in Fig. 1b). In the proposed benchmark problem g is 1621, which means that more than 10 millions of linking numbers have to be computed. This yields to an unsurmountable bottleneck due to the HO technique.

This is why we proposed in the past to skip step 2 and use all generators of \mathcal{S} instead half of them [1], [2] in the following steps. In this case, the generators are called *lazy*: they span the needed cohomology group, but contain additional, dependent elements. Yet, a standard cohomology basis is attractive given that it reduces the number of unknowns in the linear system and also produces a full-rank system matrix. For this purpose, we introduce in this contribution a technique to remove this obstruction and thus speed up the computation of a regular cohomology basis.

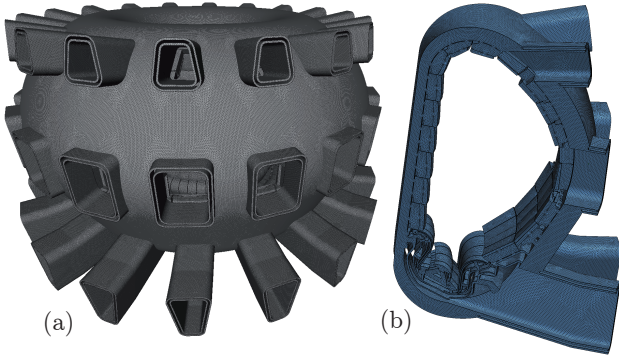


Fig. 2. (a) The geometry of the considered conductive structures of an ITER-like nuclear fusion reactor. (b) A 1/18 of the geometry.

II. LEAN DS ALGORITHM

The novel idea proposed in this contribution allows to eliminate the bottleneck in the linking number matrix computation by introducing a radically novel way to compute it. In fact, the linking number matrix may be computed (almost) for free, i.e. without the costly integrations used in [3], [1]. The lean DS algorithm is briefly described:

- 1) The first homology group generators of the discrete surface $\mathcal{S} = \mathcal{K}_c \cap \mathcal{K}_a$ (see Fig. 1a) are computed with a linear time combinatorial algorithm.
- 2) *Thinned currents* are found easily from all homology generators of \mathcal{S} by using a technique similar to the one described in [6].
- 3) ESTT is run for all thinned currents.
- 4) Here is the key idea of this contribution. The linking number between the discrete surface 1-cycles and the same cycles pushed inside \mathcal{K}_c (i.e. the dual of the thinned currents) may be evaluated as the number of times the cycles pierce the dual discrete surfaces produced by the ESTT. This is due to a reinterpretation of linking numbers (see for example [7]). This, in turn, is realized in our setting simply with a sparse dot product between the two sparse arrays that represent the homology generator of \mathcal{S} and the dual discrete surface.
- 5) The adapted HO technique is applied. The change of basis found in the previous step is applied to the output of the ESTT that, once restricted to \mathcal{K}_a , form the required cohomology generators.

III. NUMERICAL RESULTS

The authors have implemented in C++ the novel algorithm presented in this contribution inside the TOPOPROCESSOR code [8]. To demonstrate the performance of the novel DS algorithm we perform the topological preprocessing in the complement of the conductive structures of an ITER-like nuclear fusion device (see Fig. 2a) with respect to a box which represents the insulating region. The conductor which is considered in the benchmark is formed by gluing together 18 structures as the one in Fig. 2b. The number of mesh elements and the topological features render the topological pre-processing for this benchmark particularly challenging. In fact, we expect to

TABLE I
NUCLEAR FUSION REACTOR BENCHMARK

	Mesh
Number of tetrahedra in \mathcal{K}	26 648 351
Number of tetrahedra in \mathcal{K}_a	13 225 740
Meshing time GMSH [s]	1061
t_{GMSH} [s]	14 933
t_{TP} lazy [s]	202
t_{TP} regular [s]	209

extract 1621 generators of the first cohomology group of the insulating region, which corresponds to 3242 lazy generators. In below, t_{TP} denotes the total wall time (in seconds) needed by TOPOPROCESSOR code to compute a lazy and a regular cohomology basis, whereas t_{GMSH} represents the total time required by the software GMSH [9], which is an efficient implementation of the standard paradigm based on reducing the complex and computing the Smith Normal Form of the reduced matrix. All computations are performed on a workstation with a 12Core-Xeon E5-2687Wv4 processor equipped with 128 Gb of RAM. It is also interesting to note that GMSH, also has a higher memory consumption, roughly 60 Gb, while TOPOPROCESSOR would successfully terminate without swapping on a machine with as few as 20 Gb of RAM. TOPOPROCESSOR code has been able to compute all generators in about 3 minutes of total computing time, whereas GMSH terminated after more than four hours on the same workstation, see details on the mesh and on timings in Tab. I.

The obtained speedup of nearly two orders of magnitude with respect to the state-of-the-art competitor software well motivates in our opinion the interest in the algorithm proposed in this paper. In particular, we remark that the topological preprocessing is a small fraction of meshing time in the case of TOPOPROCESSOR, whereas it ends up being the bottleneck of the whole simulation chain in the case of GMSH.

REFERENCES

- [1] P. Dlotko, R. Specogna, Physics inspired algorithms for (co)homology computations of three-dimensional combinatorial manifolds with boundary, *Computer Phys. Commun.*, Vol. 184, No. 10, pp. 2257-2266, 2013.
- [2] P. Dlotko, R. Specogna, Lazy cohomology generators: a breakthrough in (co)homology computations for CEM, *IEEE Trans. Magn.*, Vol. 50, No. 2, 7014204, 2014.
- [3] R. Hiptmair, J. Ostrowski, Generators of $H_1(\Gamma_h, \mathbb{Z})$ for triangulated surfaces: construction and classification, *SIAM J. Comput.*, Vol. 31, No. 5, pp. 1405-1423, 2002.
- [4] P. Dlotko, R. Specogna, Efficient generalized source field computation for h-oriented magnetostatic formulations, *Eur. Phys. J. Appl. Phys.*, Vol. 53, 20801, 2011.
- [5] J.P. Webb, B. Forghani, A single scalar potential method for 3D magneto-statics using edge elements, *IEEE Trans. Magn.*, vol. 25, pp. 4126-4128, 1989.
- [6] B. Kapidani, P. Dlotko, P. Alotto, P. Bettini, R. Specogna, Computation of relative 1-cohomology generators from a 1-homology basis for eddy currents boundary integral formulations, *IEEE Trans. Magn.*, Vol. 52, No. 10, 7210006, 2016.
- [7] R.L. Ricca, B. Nipoti, Gauss' linking number revisited, *J. Knot Theory Ramifications*, Vol. 20, pp. 1325-1343, 2011.
- [8] P. Dlotko, B. Kapidani, R. Specogna, TOPOPROCESSOR: an efficient computational topology toolbox for h-oriented eddy current formulations, under review at *IEEE Trans. Magn.*.
- [9] M. Pellikka, S. Suuriniemi, L. Kettunen, C. Geuzaine, Homology and cohomology computation in finite element modeling, *SIAM J. Sci. Comput.*, Vol. 35, No. 5, pp. 1195-1214, 2013.