Optimal Cohomology Generators for 2-D Eddy-Current Problems in Linear Time

Ruben Specogna

Università di Udine, Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica (DIEGM), Udine, Italy

The aim of this paper is to present an automatic and efficient algorithm to find cohomology generators suitable for 2-D eddy-current problems formulated by means of complementary formulations. The algorithm is general, straightforward to implement, exhibits a linear worst-case computational complexity and produces optimal representatives of generators. By optimal we mean the representatives that minimize in practical cases the fill-in of the system of equations matrix and guarantee that the current flowing in each conductor is in one-to-one correspondence with a generator. As a numerical example, the complementary formulations are used to compute the frequency-dependent per-unit-length impedance in integrated circuits.

***Index Terms*—Cohomology generators, complementarity, eddy-currents, finite element method (FEM), source field, thick cuts.**

I. Introduction

It is well established that the use of complementary formulations to solve eddy-current problems offers several advantages [1]–[7]. The results of the two formulations, in fact, may be used as a robust error estimator for mesh adaptivity [8]–[10] and they usually provide upper and lower bounds for impedance. The last statement hasn't been proved yet, but numerical experiments conducted in literature seem to indicate that this is the case in most practical cases [7], [11]–[13].

The obstruction to use complementary formulations is due to the fact that the one based on the magnetic scalar potential requires the so-called *cuts* to render the problem well defined. A vast literature in computational electromagnetics have been devoted to defining and retrieving them [14]–[28]. The fact that this issue has been considered unsolved for so many years indicates that computing cuts is not straightforward.

Let us cover the computational domain D of the eddy-current problem, which is assumed connected and simply connected, by a finite element mesh. From the incidences of the mesh elements, the cell complex $\mathcal{K}$ is obtained. Two subcomplexes $\mathcal{K}_c$ and $\mathcal{K}_a$ of $\mathcal{K}$ are introduced that contain elements belonging to the conducting and insulating regions, respectively. Each of the N_c connected components of $\mathcal{K}_c$ is called conductor and denoted as $\mathcal{C}_i$, with $i \in \{1, \ldots, N_c\}$.

To define the magnetic scalar potential consistently, it is popular to construct some cutting surfaces, known as (thin) cuts, that render $\mathcal{K}_a$ simply connected [16], [23]. Then, a jump in the magnetic scalar potential across cuts is obtained by doubling the nodes in the cutting surface. The problem is that this method becomes quite complicated when cuts intersect [22]. That is the reason why recently the formulation based on edge elements have been considered [18], [19], [21], [24], [25], [29]. In the latter case, in fact, intersecting or even self-intersecting cuts do not bring any additional difficulty neither in the theory nor in the implementation. Therefore, in this paper as cuts we mean the so-called *thick cuts*, that are defined as representatives of first cohomology group generators over integers of the insulating region [24], [25], [29]. For an informal introduction to the relevant concepts of algebraic topology, the potential definition for eddy-current problems and a physical interpretation of thick cuts, the reader is invited to consult [24], [29].

The algorithms for thick cut computation introduced in literature may be classified as follows. First, there are algorithms that compute the first cohomology group over integers $H^1(\mathcal{K}_a)$ as [25], [29]. These algorithms are not analyzed in this paper because for 2-D problems they are too computationally costly and too complicated to implement. Second, there are a number of algorithms based on the popular idea of "cutting" the complex $\mathcal{K}_a$ by removing some of its elements in such a way that what remains becomes simply connected. In practice, this is performed by growing as much as possible an acyclic subcomplex in $\mathcal{K}_a$ [20], [21]. Its complement with respect to $\mathcal{K}_a$ is supposed to be the union of all thick cuts. In 3-D, homotopy-based algorithms are patently incorrect [29], but in 2-D they potentially may be useful thanks to the ease of implementation. Third, the Generalized Spanning Tree Technique (GSTT) [19], [24] attempts to construct the thick cuts starting from a basis of the first homology group. GSTT is attractive for 2-D problems because—as we are going to see—homology generators are readily available and because, contrarily to the 3-D setting [26], its termination is guaranteed. Finally, an algorithm called Thinned Current Technique (TCT) has been recently introduced in [28]. It is based on a thinning of conductors followed by the application of the Extended Spanning Tree Technique (ESTT) [30].

The aim of this paper is to present a novel algorithm to obtain cohomology generators for 2-D problems that does not belong to any of the classes presented in the survey. The algorithm is completely automatic, provably general and exhibits a linear worst case computational complexity. Moreover, the algorithm is optimal in the sense that it provides the minimization of the fill-in of the linear system of equation sparse matrix, at least for practical examples.

The paper is structured as follows. Section II is devoted to a survey of existing methods for computing thick cuts. In

Manuscript received August 03, 2012; revised September 27, 2012; accepted September 27, 2012. Date of publication October 05, 2012; date of current version March 20, 2013. 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.2012.2222905

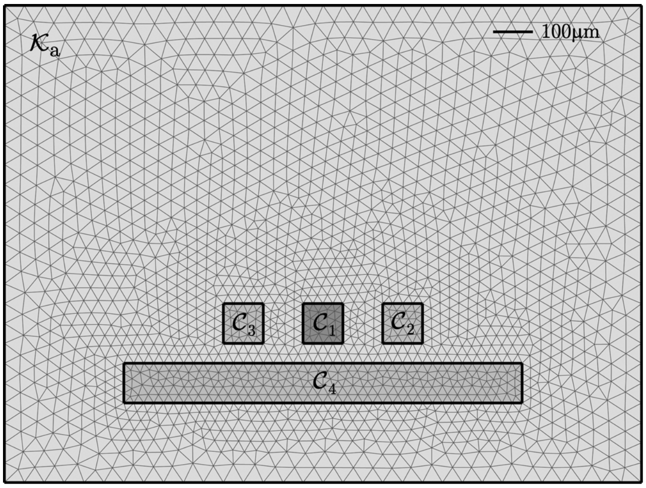

Fig. 1. The example used to illustrate the output of the various techniques for thick cuts generation.

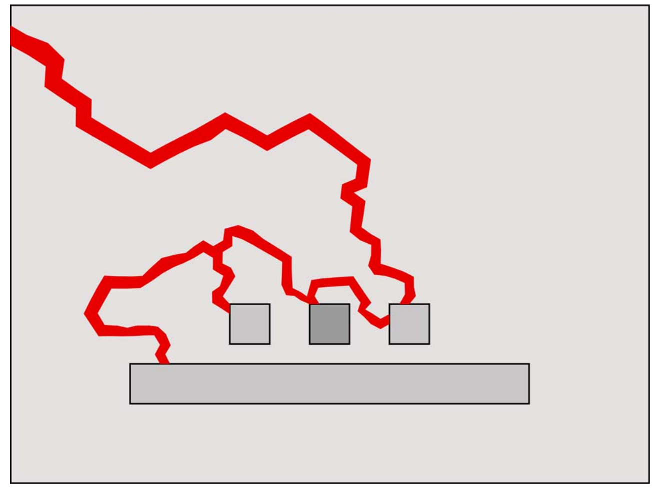

Fig. 2. The dark region (red in the color version) represents triangles belonging to the acyclic subcomplex complement, which is the union of the supports of all thick cuts.

Section III the novel algorithm to compute optimal generators of the cohomology group is presented. Section IV is devoted to the presentation of the numerical results while, in Section V, the conclusions are drawn.

II. Literature Survey on 2-D Cohomology Computations

A. Benchmark Problem

Let us consider as a benchmark the eddy-current problem consisting in three conducting microstrip lines above a lossy substrate, see Fig. 1. For the sake of clarity in the pictures, the coarse mesh represented in Fig. 1 is used.

B. Acyclic Sub-Complex

The idea of the acyclic subcomplex method is to remove from $\mathcal{K}_a$ the largest possible acyclic subcomplex $\mathcal{A}$. At the end of this process, its complement with respect to $\mathcal{K}_a$ is the support of the union of all thick cuts. Since $\mathcal{A}$ is homologically trivial by construction, the magnetic scalar potential only is required in that region. An example of such a support for the benchmark problem is represented in Fig. 2.

To present an algorithm, a number of problems have to be faced. Namely, a) how to construct the acyclic subcomplex; b) how to extract a set of thick cuts from its complement with respect to $\mathcal{K}_a$; c) how to find the integer coefficients of the thick cuts to assign to each edge; d) how to implement a basis selection [27] or automatically relate the current flowing in each conductor to the *independent currents* associated to thick cuts [29]. Last point is fundamental for imposing source current constraints or to perform some postprocessing.

Two classes of algorithms to find the acyclic subcomplex have been presented, one based on growing the acyclic subcomplex from a seed element, the other reducing the initial complex as much as possible. In [20], [21] it is proposed to start from a random seed element belonging to $\mathcal{K}_a$ and adding "an element to the existing acyclic set if both this element and this set share at least one face and that all their common nodes are linked by edges existing also in this set," see [21]. It is easy to realize that this algorithm hardly can succeed. In fact, if the algorithm is run on the complex represented in Fig. 3(a) starting from the seed element indicated in the picture, then no other element can be added after two iterations, leading to a premature termination of the algorithm with a wrong result.

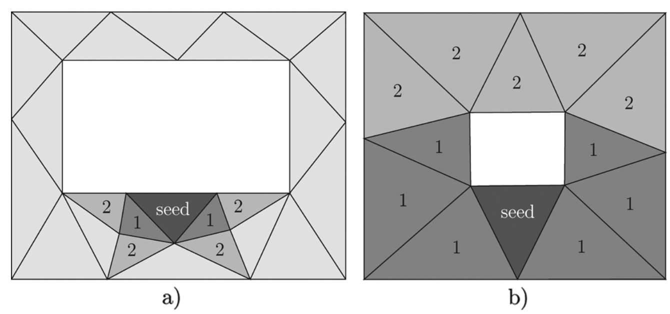

Fig. 3. (a) Counter-example for the [20], [21] algorithm. (b) Counter-example for the algorithm [22], [23].

Regarding the second class, two general and efficient algorithms to construct an acyclic subcomplex based on *lookup tables* and *coreduction* have been proposed in [25] (even though we remark that in that paper they are only used as efficient *reduction* techniques [25].)

A technique to generate a set of independent thick cuts and to produce the integer coefficients has been introduced in [21]. We remark that the basis obtained is in most cases not the natural one, i.e., the current flowing in each conductor is not in one-to-one correspondence with a generator. Therefore, additional effort is needed to perform a basis selection [27] or to relate the current flowing in each conductor to the set of cuts, for example as described in [21, Section 2.3]. As a final remark we note that algorithms based on the acyclic sub-complex usually produce thick cuts that do not have a compact support.

1) From Thin Cuts to Thick Cuts: Two other algorithms have been presented to find thin cuts with the acyclic subcomplex idea. Then, a thick cut may be obtained by growing a layer of elements on one side of the thin cut and finding the edge coefficients appropriately. The first one, based on simple reductions [33], has been presented in [16]. The second has been presented in [22], [23], but it seems not general, at least following the steps contained in those papers. In fact, by considering the complex in Fig. 3(b), no cut is found, leading to wrong results. In both cases, a procedure to remove dangling pieces of the cutting surfaces—called *filtering* in [22], [23]—is needed, which is the most time consuming part of the whole algorithm [23].

C. GSTT and TCT

The GSTT algorithm introduced in [19], and further investigated in [24] and [26], produces a cohomology basis given a homology basis as input. While for 3-D problems finding a homology basis is not easy and many problems may arise in the algorithm [26], for 2-D problems it seems to be a valid alternative. The reason is that the boundaries of the conductors form a canonical basis for the $H_1(\mathcal{K}_a)$ homology group that, therefore, does not need to be computed.

The following algorithm has to be run for each conductor $\mathcal{C}_i$, $i \in \{1, \ldots, N_c\}$. First, the edges in the boundaries of all conductors are marked as tree edges (even if in this way the tree contains cycles) to start forming the so-called *belted tree* [17], [26]. Then, to complete the belted tree, a tree is formed in the remaining part of $\mathcal{K}_a$. At the end, the only cycles present in the belted tree are the boundaries $\partial\mathcal{C}_i$, $i \in \{1, \ldots, N_c\}$ of the conductors. Then, a zero value is set to all belted-tree edges except one random edge belonging to $\partial\mathcal{C}_i$ whose value is set to one. Then, iterating over triangles, the coefficients of the ith thick cut on the belted cotree edges (i.e., the complement of the belted tree in $\mathcal{K}_a$) are found by means of the discrete Ampère's law [19]. In [18], thick cuts are computed as in [32], which presents an alternative algorithm for the construction of a belted tree. For 2-D problems belted trees are more effectively constructed as described in this paper.

Let us now present a novel algorithm based on the idea behind the TCT algorithm [28]. First, build a (unconstrained) tree in $\mathcal{K}$. Then, a random element belonging to the ith conductor $\mathcal{C}_i$ is selected. We set the current of that element to one, zero on the others. Physically, this corresponds to compress the conductor as much as possible. Finally, by applying the standard STT algorithm [19], [26] to $\mathcal{K}$ and restricting the output to $\mathcal{K}_a$, it is possible to demonstrate that a basis of the first cohomology group of $\mathcal{K}_a$ is obtained [28].

The thick cuts produced by the GSTT algorithm for the benchmark problem are represented in Fig. 4.

Both GSTT and TCT algorithms terminate always (this is related to the fact that a triangulation of a plane disk always collapses to a point [33, p. 90]). Their advantage is that they render useless any basis selection algorithm [21], [27], since there is a one-to-one correspondence between a conductor and a cohomology generator. The disadvantage of these algorithms is that the support of the thick cuts is in general far from minimal. The novel algorithm proposed in this paper allows the most compact family of thick cuts to be obtained at an even cheaper price.

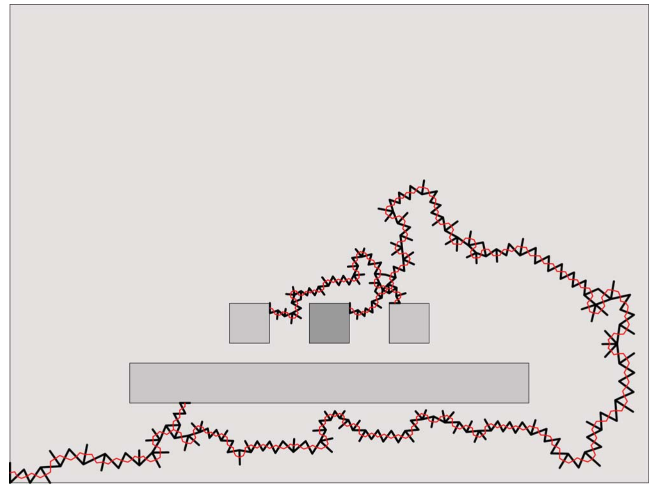

Fig. 4. Thick edges represent the support of thick cuts for the benchmark problem produced by the GSTT algorithm. Thin edges represent *dual edges* [17], [24] in $\mathcal{K}_a$ that are dual to thick edges. These dual edges form paths on the dual complex from $\partial\mathcal{K}$ to each conductor boundary.

III. Optimal Cohomology Generators in Linear Time

A. (Co)Homology Computations as Shortest Path Problems

The idea of the novel algorithm stems from noticing that a possible way to construct the thick cut for $\mathcal{C}_i$ is to chose as support the edges that are dual to *dual edges* [17], [24] that form paths from the outer surface (i.e., $\partial\mathcal{K}$) to $\partial\mathcal{C}_i$. At the end, this technique would produce thick cuts similar to those obtained by the GSTT algorithm in Fig. 4, that indeed can be interpreted as paths on the dual complex starting in $\partial\mathcal{K}$ (on the bottom left of the picture) and reaching each conductor boundary.

However, we want to produce *optimal cohomology generators*, where by optimal we mean the set of thick cuts with minimum support, that usually provide the minimization of the fill-in of the sparse matrix of the linear system of equations[1]. The most compact family of representatives may be obtained by retrieving shortest paths[2] made by dual edges between $\partial\mathcal{K}$ and each conductor boundary $\partial\mathcal{C}_i$. In principle, a multisource multidestination shortest path problem should be solved to determine such short paths between all possible pairs of dual nodes belonging to $\partial\mathcal{C}_i$ the ones belonging to $\partial\mathcal{K}$. To find the shortest path between two nodes in a graph, the well-known Dijkstra algorithm [34] may be used, which exhibits a worst-case complexity for the computation of each thick cut ranging from $O(V^2)$ to $O(E + V\,logV)$ with the most sophisticated implementation based on a Fibonacci heap [34], where V and E are the number of nodes and edges of the mesh, respectively. The number of elements in the mesh is denoted as T.

[1]We remark that generators with minimal support do not necessarily imply that the minimization of the fill-in is always achieved. There may be some examples built on a purpose that disprove this fact. Nonetheless, the proposed method typically produces the minimum fill-in in practical problems and nearly the minimum fill-in in the other cases.

[2]Shortest in the number of dual edges, not in Euclidean norm sense.

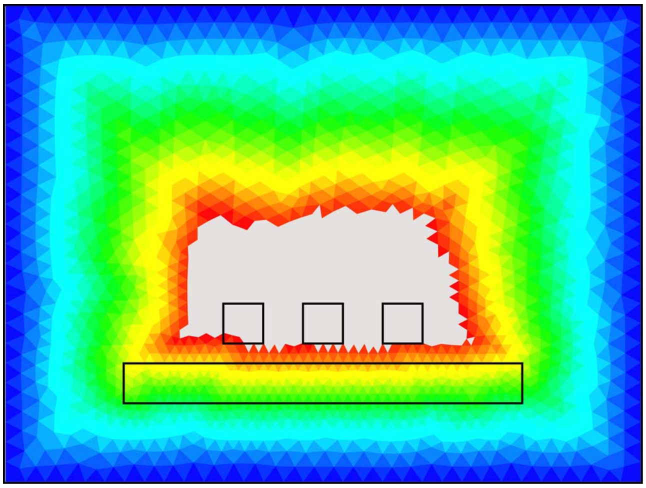

Fig. 5. Distance field obtained by propagating in $\mathcal{K}$ from $\partial\mathcal{K}$.

B. Algorithm for Computing Optimal Cohomology Generators

The main novelty of this paper is to show that optimal cohomology generators can be obtained automatically in linear time worst-case complexity just by solving a standard single-source single-destination shortest path problem. The proposed algorithm does not require any intervention of the user and it is even straightforward to implement. The algorithm, based on a breadth-first search (BFS) [34] single-source shortest path algorithm executed on an undirected and unweighted graph, is presented in Algorithm 1. From line 4 to 7 of Algorithm 1, the elements of $\mathcal{K}$ that touch $\partial\mathcal{K}$ with at least one edge are enqueued and their distance to $\partial\mathcal{K}$ is set to zero. From line 8 to 19, the distances of the other triangles in $\mathcal{K}$ are found by using a BFS algorithm. The propagation stops when the boundaries of all conductors are reached. The distance field obtained for the benchmark problem is represented in Fig. 5. The paths from each conductor boundary that define the thick cuts are easily found from line 20 to 32 by using the predecessor information. The integer edge coefficients of the thick cuts are found by enforcing the discrete Ampère's law on each element. Matrix $\mathbf{C}$ contains the incidence numbers $\mathbf{C}(f, e)$ between all possible pairs of faces f and edges e, whereas the colon notation as in Matlab or Fortran 90 is used to access an entire row or column of a matrix. Finally, from line 33 to 35, the coefficients of the thick cuts relative to edges not in $\mathcal{K}_a$ are set to zero.

Algorithm 1 `2-D_optimal_thick_cuts_generation`

1: integer $E \times N_c$ matrix $\texttt{thick\_cuts}$;

2: vector of T integers $\texttt{distance}$ initially set to ∞. $\texttt{distance}[i]$ indicates the distance of the ith triangle T_i with respect to $\partial\mathcal{K}$;

3: vector of T integers $\texttt{path}$. $\texttt{path}[i]$ indicates the triangle from which T_i is discovered;

4: **for** $\mathrm{i} = 1$ to T **do**

5: **if** some of the edges of T_i belong to $\partial\mathcal{K}$ **then**

6: $\texttt{distance}[i] \leftarrow 0$;

7: queue $Q \leftarrow T_i$;

8: **while** Q is not empty **do**

9: triangle $T_p \leftarrow \mathrm{pop}(Q)$;

10: **for** all triangles T_j that share an edge $\{E_i\}_{i=1}^{E}$ with T_p **do**

11: **if** $\texttt{distance}[j] = \infty$ **then**

12: $\texttt{distance}[j] \leftarrow \texttt{distance}[p] + 1$;

13: $\texttt{path}[j] \leftarrow T_p$;

14: $Q \leftarrow T_j$;

15: **if** T_j has an edge $E_k \in \partial\mathcal{C}_n$, $n \in \{1, \ldots, N_c\}$, and $\texttt{destination}[n] = 0$ **then**

16: $\texttt{destination}[n] \leftarrow T_j$;

17: $\texttt{thick\_cuts}[k, n] \leftarrow \mathbf{C}(T_j, E_k)$;

18: **if** $\texttt{destination}[n] \neq 0, \forall n \in \{1, \ldots, N_c\}$ **then**

19: break;

20: **for** $\mathrm{n} = 1$ to N_c **do**

21: $T_i \leftarrow \texttt{destination}[n]$;

22: **while** true **do**

23: **if** $\texttt{distance}[\mathrm{i}] = 0$ **then**

24: $d_i \leftarrow \mathbf{C}(T_i, :) \cdot \texttt{thick\_cuts}[:, \mathrm{n}]$;

25: $\texttt{thick\_cuts}[k, n] \leftarrow -d_i\mathbf{C}(T_i, E_k)$, where E_k is the edge of T_i on $\partial\mathcal{K}$;

26: break;

27: **else**

28: $T_j \leftarrow \texttt{path}[\mathrm{i}]$;

29: $E_k \leftarrow$ common edge between triangles T_i and T_j;

30: $d_i \leftarrow \mathbf{C}(T_i, :) \cdot \texttt{thick\_cuts}[:, n]$;

31: $\texttt{thick\_cuts}[k, n] \leftarrow -d_i\mathbf{C}(T_i, E_k)$;

32: $T_i \leftarrow T_j$;

33: **for** $\mathrm{k} = 1$ to E **do**

34: **if** $E_k \notin \mathcal{K}_a$ **then**

35: $\texttt{thick\_cuts}[k, n] \leftarrow 0, \forall n \in \{1, \ldots, N_c\}$;

36: **return** $\texttt{thick\_cuts}$;

It is easy to see that if the number of conductors is bounded by a constant $O(1)$, as it happens always in practice, then the worst-case computational complexity of the algorithm is linear w.r.t. the cardinality of the complex $\mathcal{K}$.

The result of the algorithm in case of the benchmark is visible in Fig. 6. We finally remark that the proposed algorithm does not avoid conductors during the distance field propagation in such a way that the dual path can pass through them (as happens in Fig. 6). As demonstrated rigorously in [35], one can safely find

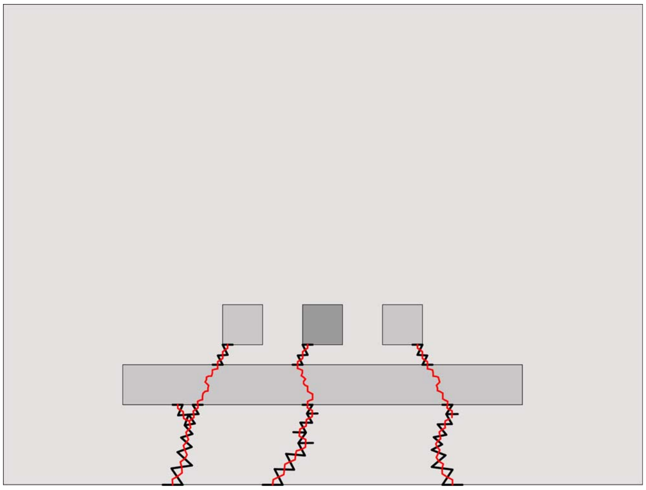

Fig. 6. Thick cuts for the benchmark problem produced by the BFS algorithm presented in this paper.

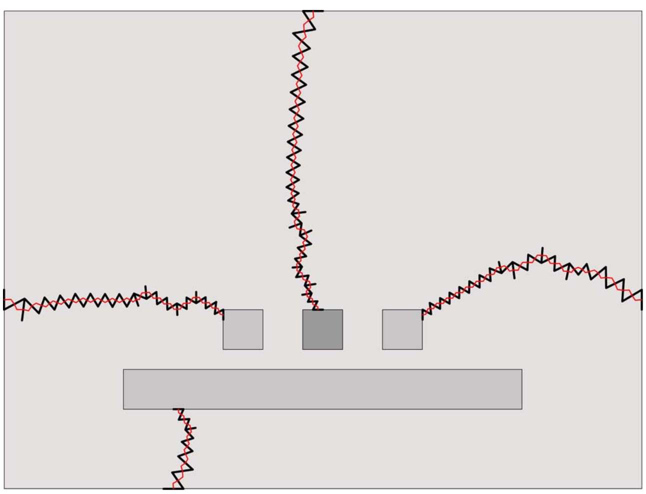

Fig. 7. Thick cuts for the benchmark problem produced by the BFS algorithm avoiding propagation in $\mathcal{K}_c$ during distance field computation.

the generators $H^1(\mathcal{K} \setminus \mathcal{C}_i)$ independently for each conductor $\mathcal{C}_i$, that is by considering the other conductors as insulators. Then, one obtains a cohomology $H^1(\mathcal{K}_a)$ basis by restricting to $\mathcal{K}_a$ all the cocycles previously obetained. If one avoids propagating inside $\mathcal{K}_c$ when computing the distance field, a set of thick cuts with non minimal support is obtained, see Fig. 7.

C. Neumann Boundary Conditions

The same algorithm may be used when a part of $\partial\mathcal{K}$ is subject to Neumann boundary conditions. The typical case is when one takes advantage of some symmetry in the problem. As an example, let us consider one of the two halves of the benchmark problem cut in the symmetry plane as in Fig. 8. The only modification that needs to be accounted for is that the $\partial\mathcal{K}$ in Alg. 1 has to be substituted only with the part of $\partial\mathcal{K}$ subject to Dirichlet boundary conditions. The generators in this case are visible in Fig. 8.

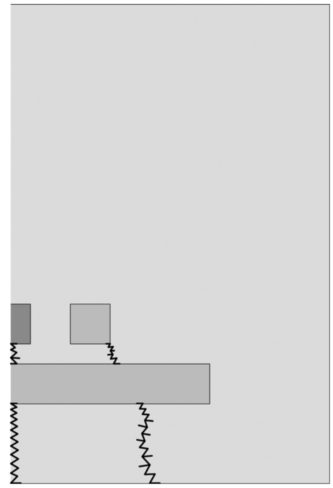

Fig. 8. Thick cuts for the benchmark problem reduced to half with Neumann boundary conditions.

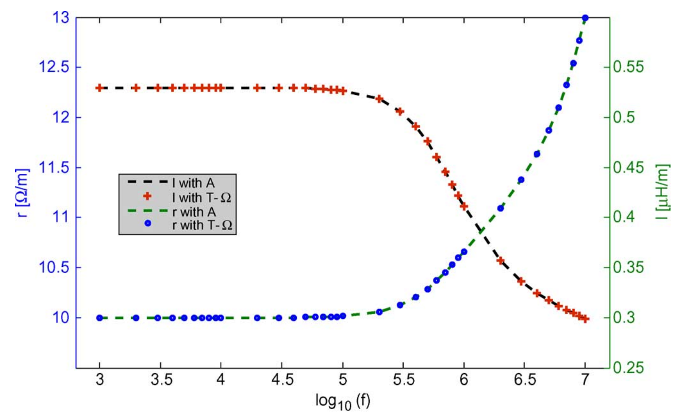

Fig. 9. Frequency-dependent per-unit-length resistance and inductance obtained with $\mathbf{A}$ and $\mathbf{T} - \mathbf{\Omega}$ formulations.

IV. Thin and Thick Links Computation

This algorithm may be used as is to produce optimal *thick links* for 2-D electrostatic problems solved by an electric vector potential formulation [31] and with slightly modifications to produce optimal thick links for 3-D electrostatic computations [35]. Also optimal thin links (or thin cuts for 2-D problems) may be obtained by computing paths on the primal complex instead of the dual complex. The trivial modifications needed in these cases are left to the reader.

V. Numerical Results

The algorithms described in this paper have been integrated into the CDICE research code [36]. A conductivity of $10^7\ \mathrm{S/m}$ has been considered for all conductors. As an example, frequency-dependent per-unit-length (p.u.l.) parameters of $\mathcal{C}_1$ are computed. The frequency course of p.u.l. resistance and inductance obtained by the $\mathbf{A}$ and $\mathbf{T} - \mathbf{\Omega}$ formulations are shown in Fig. 9. The two formulations show very good agreement and the cohomology generators are computed in near real-time.

VI. Conclusion

An efficient and general algorithm to produce cohomology generators for 2-D problems has been proposed. For the first time, this algorithm uses a shortest path algorithm applied on the dual complex to produce the cohomology basis. This technique has the virtue to be optimal in terms of speed and quality of the thick cuts obtained. In particular, the obtained thick cuts typically minimize the fill-in of the linear system matrix and they are in one-to-one correspondence with conductors, that enables to get rid of any issue related to basis selection. We note that compact thick cuts cannot be obtained by the acyclic subcomplex technique in general, being the complement of the support of thick cuts not connected.

Acknowledgment

This work was partially supported by the Italian Ministry of Education, University and Research (MIUR) project PRIN 2009LTRYRE.

References

[1] P. Hammond and J. Penman, "Calculation of eddy currents by dual energy methods," *IEE Proc.-A*, vol. 125, no. 7, pp. 701–708, 1978.

[2] A. Bossavit, "Two dual formulations of the 3D eddy currents problem," *COMPEL*, vol. 4, pp. 103–116, 1984.

[3] R. L. Ferrari, "Complementary variational formulations for eddy-current problems using the field variables E and H directly," *IEE Proc.-A*, vol. 132, no. 4, pp. 157–164, 1985.

[4] J. A. H. Rikabi, C. F. Bryant, and E. M. Freeman, "Error-based derivation of complementary formulations for the eddy-current problem," *IEEE Trans. Magn.*, vol. 29, no. 1, pp. 98–107, Jan. 1988.

[5] P. Hammond, "Upper and lower bounds in eddy current calculation," *IEEE Proc.-A*, vol. 136, no. 4, pp. 207–216, 1989.

[6] A. Bossavit, "On complementary formulations in steady-state eddy-currents theory," *IEE Proc.-A*, vol. 139, no. 6, pp. 265–272, 1992.

[7] C. Li, Z. Ren, and A. Razek, "Complementarity between the energy results of H and E formulations in eddy-current problems," *IEE Proc.-A*, vol. 141, no. 1, pp. 25–30, 1994.

[8] J. Penman and M. D. Grieve, "Self-adaptive mesh generation technique for the finite-element method," *IEE Proc.-A*, vol. 134, no. 8, pp. 634–650, 1987.

[9] C. Li, Z. Ren, and A. Razek, "An approach to adaptive mesh refinement for three-dimensional eddy-current computations," *IEEE Trans. Magn.*, vol. 30, no. 1, pp. 113–117, Jan. 1994.

[10] M. Bensetti, Y. Choua, L. Santandrea, Y. Le Bihan, and C. Marchand, "Adaptive mesh refinement and probe signal calculation in eddy current NDT by complementary formulations," *IEEE Trans. Magn.*, vol. 44, no. 6, pp. 1646–1649, Jun. 2008.

[11] R. Specogna and F. Trevisan, "Eddy-currents computation with $T-\Omega$ discrete geometric formulation for a NDE problem," *IEEE Trans. Magn.*, vol. 44, no. 6, pp. 698–701, Jun. 2008.

[12] Z. Ren and H. Qu, "Investigation of the complementarity of dual eddy current formulations on dual meshes," *IEEE Trans. Magn.*, vol. 46, no. 8, pp. 3161–3164, Aug. 2010.

[13] P. Dłotko, R. Specogna, and F. Trevisan, "Voltage and current sources for massive conductors suitable with the $A-\chi$ geometric formulation," *IEEE Trans. Magn.*, vol. 46, no. 8, pp. 3069–3072, Aug. 2010.

[14] P. W. Gross and P. R. Kotiuga, "Electromagnetic theory and computation: A topological approach," in *MSRI Monograph Series*. Cambridge, U.K.: Cambridge Univ. Press, 2004, vol. 48.

[15] C. S. Harold and J. Simkin, "Cutting multiply connected domains," *IEEE Trans. Magn.*, vol. 21, no. 6, pp. 2495–2498, Nov. 1985.

[16] P. J. Leonard, H. C. Lai, R. J. Hill-Cottingham, and D. Rodger, "Automatic implementation of cuts in multiply connected magnetic scalar region for 3-D eddy current," *IEEE Trans. Magn.*, vol. 29, no. 2, pp. 1368–1371, Mar. 1993.

[17] A. Bossavit, *Computational Electromagnetism*. San Diego, CA: Academic, 1998.

[18] Z. Ren, " $T-\Omega$ formulation for eddy-current problems in multiply connected regions," *IEEE Trans. Magn.*, vol. 38, no. 2, pp. 557–560, Mar. 2002.

[19] 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.*, vol. 39, no. 3, pp. 1167–1170, May 2003.

[20] J. Simkin, S. C. Taylor, and E. X. Xu, "An efficient algorithm for cutting multiply connected regions," *IEEE Trans. Magn.*, vol. 40, no. 2, pp. 707–709, Mar. 2004.

[21] P. Dular, "Curl-conform source fields in finite element formulations: Automatic construction of a reduced form," *COMPEL*, vol. 24, pp. 364–373, 2005.

[22] A.-T. Phung, P. Labie, O. Chadebec, Y. Le Floch, and G. Meunier, "On the use of automatic cuts algorithm for $T_0-T-\Omega$ formulation in nondestructive testing by eddy current," in *Intelligent Computer Techniques in Applied Electromagnetics*. Berlin: Springer, 2008.

[23] A. T. Phung, O. Chadebec, P. Labie, Y. Le Floch, and G. Meunier, "Automatic cuts for magnetic scalar potential formulations," *IEEE Trans. Magn.*, vol. 41, no. 5, pp. 1668–1671, May 2005.

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

[25] P. Dłotko and R. Specogna, "Efficient cohomology computation for electromagnetic modeling," *CMES*, vol. 60, no. 3, pp. 247–278, 2010.

[26] P. Dłotko and R. Specogna, "Critical analysis of the spanning tree techniques," *SIAM J. Numer. Anal.*, vol. 48, pp. 1601–1624, 2010.

[27] 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.

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

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

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

[31] Z. Ren, "2-D dual finite-element formulations for the fast extraction of circuit parameters in VLSI," *IEEE Trans. Magn.*, vol. 39, no. 3, pp. 1590–1593, May 2003.

[32] Z. Ren and A. Razek, "Boundary edge elements and spanning tree technique in three-dimensional electromagnetic field computation," *Int. J. Numer. Meth. Eng.*, vol. 36, pp. 2877–2893, 1993.

[33] P. Giblin, *Graphs, Surfaces and Homology*, 3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[34] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA: The MIT Press, 2001.

[35] R. Specogna, "Complementary geometric formulations for electrostatics," *Int. J. Numer. Meth. Eng.*, vol. 86, pp. 1041–1068, 2011.

[36] R. Specogna, CDICE: Complementarity and duality in computational science and engineering.