Fast Computation of Cuts With Reduced Support by Solving Maximum Circulation Problems

Paweł Dłotko[1], Bernard Kapidani[2], and Ruben Specogna[2]

[1]Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica, Università di Udine, Udine 33100, Italy
[2]Department of Mathematics, University of Pennsylvania, Philadelphia, PA 19104-6395 USA

We present a technique to efficiently compute optimal cuts required to solve 3-D eddy current problems by magnetic scalar potential formulations. By optimal cuts, we mean the representatives of (co)homology generators with minimum support among the ones with a prescribed boundary. In this paper, we obtain them by starting from the minimal (co)homology generators of the combinatorial two-manifold representing the interface between conducting and insulating regions. Optimal generators are useful because they reduce the fill-in of the sparse matrix and ease human-guided basis selection. In addition, provided that the mesh is refined enough to allow it, they are not self-intersecting. The proposed technique is based on a novel graph-theoretic algorithm to solve a maximum circulation network flow problem in unweighted graphs that typically runs in linear time.

Index Terms— **(Co)homology, eddy currents, maximum circulation network flow problem, thin and thick cuts.**

I. Introduction

When solving eddy-current problems containing topologically non-trivial conductors with a magnetic scalar potential formulation, the first cohomology group generators of the complex $\mathcal{K}_a$ that represents the insulators are required [1]– [3] to get a well-posed problem. The availability of unprecedented efficient numerical methods, as the Dłotko–Specogna (DS) algorithm based on lazy cohomology generators [3], [4], allows the practical use of cohomology in everyday industrial problems arising in computational engineering. The generality of these algorithms have been proved mathematically, therefore robustness is guaranteed for every possible input, no matter how complicated the problem is. Nonetheless, other requirements that the representatives of cohomology basis should fulfill have been pointed out recently [4]–[6]. These can be classified in three categories.

- *R1* (*Automatic or Human-Guided Basis Selection):* The user should select the basis that suits his needs. To this aim, the representatives should be simple enough to allow the user of the electromagnetic software to easily relate a representative to the relevant constraint on the boundary value problem (BVP).
- *R2* (*Minimization of Fill-In):* The representatives of cohomology generators with compact support usually reduce the fill-in of the linear system matrix, with obvious advantages in the solution stage.
- *R3* (*Avoid Self-Intersections):* Self-intersections are not a problem when using $H^1(\mathcal{K}_a, \mathbb{Z})$ cohomology generators (i.e., *thick cuts*) in formulations based on edge basis functions [1]–[4], [7], [8]. Yet, they go against requirement $R1$ in the sense that the constraints on the BVP due to self-intersecting cuts are in general complicated to grasp. On the contrary, we remark that avoiding self-intersections is a prerequisite for the construction of relative $H_2(\mathcal{K}_a, \partial\mathcal{K}_a, \mathbb{Z})$ homology generators that can be used with formulations based on nodal basis functions [9]–[11] (i.e., *thin cuts*.) A time consuming technique to avoid self-intersections based on solving one non-physical Poisson problem for each generator is proposed in [9], [10], and [11]. As a matter of fact, this render the formulations based on thin cuts practically unattractive.

Most of these requirements would be satisfied, if one were be able to minimize the support of representatives of generators, at least—as it happens in practice—if the mesh is refined enough to allow R3 to be fulfilled.[1] While for 2-D problems this is easy [6], in 3-D this seems to be—as mentioned in [5]—quite a hard problem.

There are various ways to minimize the support of the representatives of generators.

- *S1* (*Minimal (Co)Homology Basis):* In this case, one asks to minimize the whole basis, i.e., to find a set of representatives of generators, whose supports have a minimum total area or a minimum total number of (primal or dual [12]) faces. There are two reasons, why not to pursue this idea. First, minimal (co)homology basis is very often not an optimal choice. For example, to ease the source enforcement, it is wise to guarantee that the current flowing in each torus-shaped conductor is in one-to-one correspondence with a generator. In addition, $R1$ requires the basis to be fixed by the user not necessarily to the minimal one. Second, it has been proved to be a NP-hard problem [13], whose solution is likely not to be practical in the context addressed in this paper.
- *S2* (*Finding the Minimal Support of Each Representative Taking the (Co)Homology Basis As Fixed):* Even though this can be solved in polynomial time, for example, with standard optimization techniques based on linear

Manuscript received May 22, 2014; revised August 21, 2014; accepted September 19, 2014. Date of current version April 22, 2015. 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.2014.2359976

[1]Self-intersecting thin or thick cuts cannot be always avoided. In fact, the mesh could be too coarse to allow the surface to fold the necessary number of times. This is the reason why we advocated in the past the use of thick cuts [8], [1], that work on arbitrary meshes, in place of standard thin cuts that need some assumptions on the input mesh that are hard to check in practice.

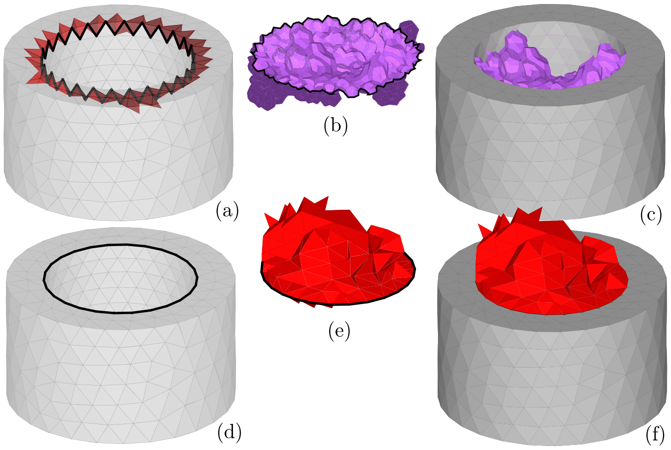

Fig. 1. (a) Thick edges: support of a cohomology generator of the boundary of $\mathcal{K}_c$, a solid two-torus. Dark triangles: support of the thinned current. (b) Dual edges dual to thinned current faces are the boundary of a two-chain on the dual complex. (c) Such two-chain, restricted to $\mathcal{K}_a$, is the dual of the cohomology generator of $\mathcal{K}_a$. (d) Thick edges: support of a homology generator of the boundary of $\mathcal{K}_c$, a solid two-torus. (e) Edges belonging to the support of the homology generator of $\partial\mathcal{K}_c$ are the boundary of a two-chain on the primal complex. (f) Such two-chain, restricted to the interior of $\mathcal{K}_a$, is the relative homology generator of $\mathcal{K}_a$.

programming [13], it still requires too much computational resources to be practical in the computational electromagnetics context.

S3 In this paper, we introduce, for the first time, quite a different technique with respect to the ones analyzed before. This novel technique, described in Section II, is able to reduce the supports of both $H^1(\mathcal{K}_a, \mathbb{Z})$ and $H_2(\mathcal{K}_a, \partial\mathcal{K}_a, \mathbb{Z})$ generators.

This paper is structured as follows. Section II proposes a novel idea to produce optimal thin or thick cuts suitable to solve eddy-current problems. Section III presents the idea behind the algorithm to minimize the support of the representatives of generators keeping their boundaries as fixed. Section IV introduces a novel algorithm to minimize the support of $H^1(\mathcal{K}_a, \mathbb{Z})$ generators. If one needs minimal $H_2(\mathcal{K}_a, \partial\mathcal{K}_a, \mathbb{Z})$ generators instead, then the complex dual to $\mathcal{K}$ should be provided at the input of the algorithm. The elements dual to the output form the desired $H_2(\mathcal{K}_a, \partial\mathcal{K}_a, \mathbb{Z})$ basis (see the Lefschetz duality in [2]). Section V shows the numerical results on a number of practical benchmarks. The conclusions are drawn in Section VI.

II. DS Algorithm and Optimal Cuts

The material presented in this paper strongly relies on concepts from algebraic topology. An informal presentation can be found in [2] and [3]. The idea behind the algorithm to find optimal thin or thick cuts is tightly linked on the use of the recently introduced DS algorithm [3], [4]. This algorithm finds (thin or thick) cuts in two steps. First, it finds in linear time worst case complexity the (co)homology generators of the combinatorial two-manifold $\mathcal{S} = \mathcal{K}_c \cap \mathcal{K}_a$, where $\mathcal{K}_c$ is the complex that models the conductors [Fig. 1(a) and (d)].[2] We assume $\mathcal{K} = \mathcal{K}_c \cup \mathcal{K}_a$ is homologically trivial by hypothesis.[3] For thin cuts, we put a unity current on relative homology generators of $\mathcal{S}$ [Fig. 1(d)], whereas for thick cuts, the thinned currents are found as described in [3] [Fig. 1(a)]. Then, a vectorialized version of the extended spanning tree technique (ESTT) algorithm [14] is run on the whole complex $\mathcal{K}$. The ESTT algorithm is a general version of the Webb–Forghani iterative algorithm [15] to obtain a field whose curl is assigned. Its typical complexity is linear (actually, for all tested problems), even though there is no proof yet that its worst case complexity is better than cubical. Once the ESTT algorithm is run on the primal complex with the cohomology generators of $\mathcal{S}$ as input, it outputs one-cochain on primal complex. If one runs it on the dual complex [12], one gets two-chains on the primal complex as output [16]. Their supports may be interpreted as (in general, self-intersecting) discrete surfaces [Fig. 1(b) and (e)]. In addition, the output contains the required (co)homology generators, as shown in Fig. 1(c) and (f), and demonstrated formally in [3].

The DS algorithm has various advantages over competing algorithms. First, it is easy to implement in every in-house finite-element-like software, being based on spanning trees and not on linear algebra over integers. Second, it outperforms standard approaches based on reducing the input complex and performing the Smith normal form computation [17], [18] both in memory consumption and computational time.

The idea of this paper is to use again minimal (co)homology generators on $\mathcal{S}$, as in [4], but also to minimize the support of the representatives of generators produced by the ESTT algorithm. This is performed by bringing into practice the ideas introduced in [19] about the computation of minimal discrete surfaces with a given boundary. We are not aware about any detailed pseudocode or implementation of these ideas. The same conclusion was reached in [20]: it is unclear if his (Ed. Sullivan) approach was ever implemented or compared with numerical methods. In addition, due to the amount of difficult machinery employed, this paper has remained somewhat inaccessible and unfortunately unappreciated.... Section IV of this paper describes in detail the graph-theoretic algorithm inspired in [19, Sec. II] to compute discrete minimal surfaces, while Section III outlines the very idea behind it.

III. Discrete Minimal Surfaces

Given a one-boundary b in $\mathcal{K}$, our aim is to find a two-chain s in $\mathcal{K}$ such that $\partial s = b$ and the support $|s|$ of s has a minimal area or the minimum number of faces. The idea, inspired from [19], is to consider a two-cochain $\mathbf{I}$ subject to two constraints. First, $\mathbf{DI} = \mathbf{0}$, where $\mathbf{D}$ is the volumes-faces incidence matrix. The cocycle $\mathbf{I}$ is then called a *flow*. Second, the coefficients of $\mathbf{I}$ are bounded by the capacity given by the area of the corresponding faces. Namely, the absolute value of the ith coefficient I_i of $\mathbf{I}$ is bounded by the area A_i of the ith face of $\mathcal{K}$: $|I_i| \leq A_i$. Since $\mathbf{I}$ is a two-cocycle, the flux $F = \langle \mathbf{I}, s \rangle$[4] is the same for all two-chains s sharing the same boundary b. The key observation is that the flux

[2] Namely, relative homology generators $H_1(\mathcal{S}, \partial\mathcal{S})$ for thin cuts and cohomology generators $H^1(\mathcal{S})$ for thick cuts.

[3] In computational electromagnetism $\mathcal{K}$ is usually topologically trivial, therefore we can assume this without loss of generality. This assumption can be relaxed as described in [3].

[4] $\langle a, b \rangle$ denotes the dot product of a and b.

F—because of previous constraint b)—cannot be greater than the area $\langle \mathbf{A}, \mathrm{abs}(s)\rangle$ of $|s|$, where $\mathbf{A}$ is a vector containing the area of the faces of $\mathcal{K}$. Therefore, one has

$$\langle \mathbf{I}, s\rangle \leq \langle \mathbf{A}, \mathrm{abs}(s)\rangle \tag{1}$$

where abs(s) simply means that one has to take the absolute value of the integers in the array that represents s.

Because of the well-known maxflow-mincut theorem [13], the equality in (1) holds when the left-hand side represents the maximum possible flow (or maxflow) fulfilling the aforementioned constraints and right-hand side represents the minimal surface spanned by b. The minimal surface is thus obtained by finding the minimal cut set (or mincut) that can be interpreted as the flow bottleneck, which is formed by saturated faces, i.e., the minimal set C of faces, whose flow I_i uses all the corresponding capacity A_i and such that C blocks any circulation.

What was just illustrated is the main contribution in [19]: produce a discrete minimal surface by solving a graph-theoretic maximum circulation problem. As far as we are aware, the details and implementation of algorithms to solve maximum circulation problems are not documented in any paper.

On the contrary, maximum flow problems between a source and sink nodes are standard, for example, the Ford–Fulkerson or the Edmonds–Karp algorithms in [13]. This paper covers this gap in the literature in Section IV.

The other aim of this paper is to apply the idea just exposed to minimize the support of thick cuts. To this aim, we assume that all edges have a unit weight (or the dual faces have a unit area). We do not minimize the total area of the support since it requires much more computational effort, [19]. In addition, a minimal support is more convenient in our context since it usually reduces the fill in of the system matrix.

IV. Solving Maximum Circulation Problems

As it was more convenient to explain the idea of this paper in Section III using $H_2(\mathcal{K}_a, \partial\mathcal{K}_a)$, it is easier to present the algorithm for $H^1(\mathcal{K}_a)$. We remind that, to minimize the support of $H_2(\mathcal{K}_a, \partial\mathcal{K}_a)$ generators, one can minimize the first cohomology basis on the dual complex (in which the external two-cells of $\mathcal{K}$ are joined with the unique external three-cell. Their capacity is 1.) In this case, the dual to the obtained cochains are the desired surfaces.

We now present Algorithm 1 that brings into practice Sullivan's ideas [19]. δ and ∂ are (co)boundary operators (see [2] for an informal introduction). The input of the presented algorithm is the one-cochain c obtained from the ESTT algorithm [Fig. 1(b)]. Let us take an edge $E = [v_0, v_1]$ such that E is non-zero in c. For an edge $E = [v_0, v_1]$, $\langle c, E\rangle$ denotes its value in the cocycle c. By the first node of E, we mean v_0 if $\langle c, E\rangle > 0$ and v_1 in the other case. The second node of E is the one which is not first. Note that the concept of first and second node depends on c, which changes during the algorithm run. An edge E is saturated if a flow through it cannot be increased. Analogously, a path p is said to be saturated if it contains at least one saturated edge. If $\langle c, E\rangle \neq 0$, by $\langle c, E\rangle[v_0, v_1]$, we mean $[v_0, v_1]$ if $\langle c, E\rangle > 0$ and $[v_1, v_0]$ if $\langle c, E\rangle < 0$.

The idea of the algorithm is as follows. As long as there is a non-saturated edge E in the support of c, pick it. Using

Algorithm 1 Maximum Circulation Network Flow

```
Input: Topologically trivial K, cochain c;
Output: Minimal support cochain c' such that δc' = δc;
 Set every edge in K as not saturated;
 while there exist not saturated edge E ∈ c do
   Queue Q; Q ← second node of E; bool s = false;
   while Q ≠ ∅ do
     node V = pop(Q); Mark V as visited;
     if V is the first node of E then
       s = true; Break;
     for every E' ∈ δV such that E' is not saturated and in
     ∂E' there is exactly one visited node do
       if ⟨E', c⟩ ≠ 0 and V is a second node of E' then
         continue;
       else
         Push the other node V' of E' to Q; Remember that
         the predecessor of V' is V;
   if s is true then
     Restore the path by using predecessors of vertices (start-
     ing from first node of E) and put an unit flow through
     it;
   else
     Modify c by adding the coboundary of all the visited
     vertices to it;
   Set all visited vertices as unvisited;
 return the current c;
```

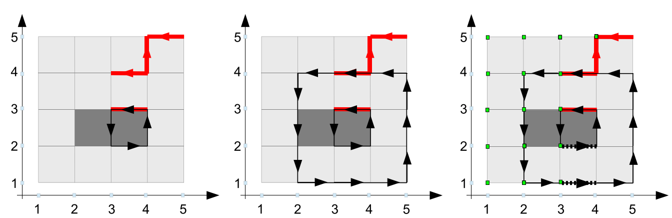

Fig. 2. (a) 2-D example of the maximal circulation problem solution. Gray rectangles: $\mathcal{K}_c$ region. Red lines: initial cochain c. Arrows: circulation from edge $[3, 4] \times [3]$. (b) Circulation from edge $[3, 4] \times [4]$. (c) Since one cannot saturate the flow through $[4] \times [4, 5]$ or $[4, 5] \times [5]$, green nodes represent the visited vertices and dashed black edges represent the new c. All edges of the new c are saturated, therefore the algorithm terminates.

a BFS algorithm [13], try to find a non-saturated path from the first node of E to its second node. Note that during BFS propagation, one cannot cross the support of the current c in a direction opposite to its orientation. If such a path from the first to the second node of E exists, increase the flow through it (note that this implies that the flow through E is also increased). When it does not exist, modify the surface by adding the coboundary of vertices one can reach from E.

The idea of Algorithm 1 is shown graphically in Fig. 2.

The proof that in this way one obtains the minimization of the surface can be found in [19]. Concerning the complexity analysis, let us point out that number of not saturated edges in the surface c always decreases in the algorithm run. The only place where it can potentially increase is when the surface is modified. But, since we are adding the coboundary of all the edges reachable from the edge $E \in c$, all the new edges in the surface c are saturated. Given this it is straightforward to demonstrate that the worst case complexity of the whole

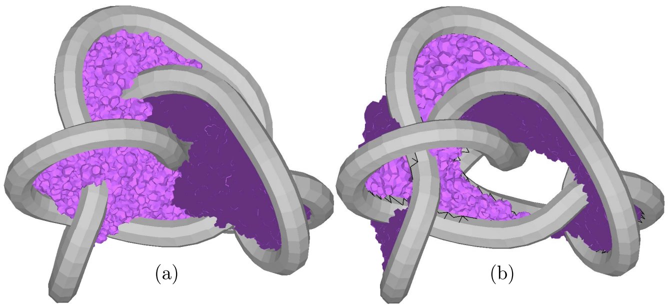

Fig. 3. (a) Support of the self-intersecting thick cut for the knot complement consisting of 87 000 tetrahedra is made by 5274 edges. (b) Thick cut support is reduced in ∼5 s to 3434 edges. The obtained surface is non-self-intersecting, thus it is a Seifert surface.

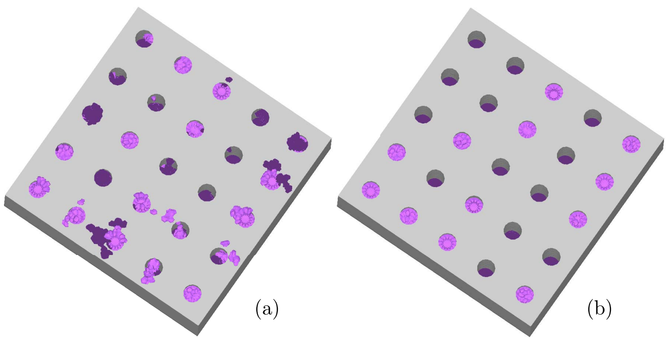

Fig. 4. (a) Support of the 25 thick cuts for a conducting plate consisting of 95 714 tetrahedra is made by 8394 edges. (b) Thick cut support is reduced in 4.2 s to 3138 edges.

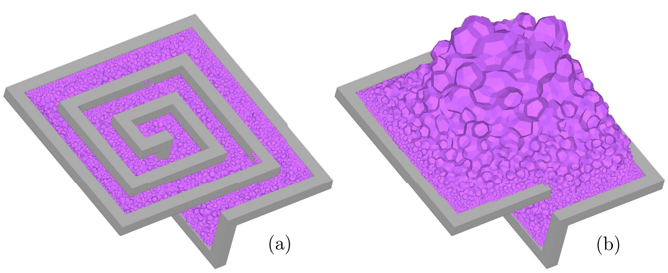

Fig. 5. (a) Support of the thick cut for a microinductor consisting of 306 804 tetrahedra is made by 19 089 edges. (b) Thick cut support is reduced in <1 min to 9126 edges.

Algorithm 1 is $O(Sn)$, where S is the number of non-zero elements in the initial c and n is the cardinality of the first skeleton of the complex. If we assume, as it happens in practice, that the number of faces in the support of the initial surface is small, the typical complexity is linear with respect to the cardinality of the complex.

V. Numerical Results

The practical applicability of the algorithm proposed in this paper has been tested on a number of benchmarks. In this paper, we present three of them: 1) a thick knot (Fig. 3); 2) a conducting plate with 25 holes (Fig. 4); and 3) a more complicated example representing a microinductor (Fig. 5).

VI. Conclusion

It has been already proved that combinatorial algorithms as DS outperform classical algebraic techniques based on the paradigm of reductions followed by Smith normal form computation as [17] and [18]. This paper shows how the DS algorithm can be further extended to produce thin or thick cuts with strongly reduced support, something that would be really hard to perform with classical algebraic methods.

References

[1] P. Dłotko, R. Specogna, and F. Trevisan, "Automatic generation of cuts on large-sized meshes for T-Ω geometric eddy-current formulation," *Comput. Methods Appl. Mech. Eng.*, vol. 198, nos. 47–48, pp. 3765–3781, Oct. 2009.

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

[3] 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, Oct. 2013.

[4] P. Dłotko and R. Specogna, "Lazy cohomology generators: A breakthrough in (co)homology computations for CEM," *IEEE Trans. Magn.*, vol. 50, no. 2, Feb. 2014, Art. ID 7014204.

[5] 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.

[6] 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.

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

[8] R. Specogna, S. Suuriniemi, and F. Trevisan, "Geometric T–Ω approach to solve eddy currents coupled to electric circuits," *Int. J. Numer. Methods Eng.*, vol. 74, no. 1, pp. 101–115, Apr. 2008.

[9] P. R. Kotiuga, "On making cuts for magnetic scalar potentials in multiply connected regions," *J. Appl. Phys.*, vol. 61, no. 8, pp. 3916–3918, 1987.

[10] P. R. Kotiuga, "An algorithm to make cuts for magnetic scalar potentials in tetrahedral meshes based on the finite element method," *IEEE Trans. Magn.*, vol. 25, no. 5, pp. 4129–4131, Sep. 1989.

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

[12] E. Tonti, *The Mathematical Structure of Classical and Relativistic Physics*. Basel, Switzerland: Birkhäuser, 2013.

[13] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.

[14] 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. 20801–20808, Feb. 2011.

[15] J. P. Webb and B. Forghani, "A single scalar potential method for 3D magnetostatics using edge elements," *IEEE Trans. Magn.*, vol. 25, no. 5, pp. 4126–4128, Sep. 1989.

[16] A. A. Rodriguez, E. Bertolazzi, R. Ghiloni, and R. Specogna. (2014). "Efficient construcion of homological seifert surfaces." [Online]. Available: http://arxiv.org/abs/1409.5487

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

[18] M. Pellikka, S. Suuriniemi, L. Kettunen, and C. Geuzaine, "Homology and cohomology computation in finite element modeling," *SIAM J. Sci. Comput.*, vol. 35, no. 5, pp. B1195–B1214, 2013.

[19] J. Sullivan, "A crystalline approximation theorem for hypersurfaces," Ph.D. dissertation, Dept. Math., Princeton Univ., Princeton, NJ, USA, 1990.

[20] D. Kirsanov and S. J. Gortler, "A discrete global minimization algorithm for continuous variational problems," Dept. Comput. Sci., Harvard Univ., Cambridge, MA, USA, Tech. Rep. TR-14-04, 2004.