

## *Review Article*

# **Homology in Electromagnetic Boundary Value Problems**

**Matti Pellikka, Saku Suuriniemi, and Lauri Kettunen**

*Tampere University of Technology, Electromagnetics, P.O. Box 692, 33101 Tampere, Finland*

Correspondence should be addressed to Matti Pellikka, matti.pellikka@tut.fi

Received 4 March 2010; Revised 2 July 2010; Accepted 14 July 2010

Academic Editor: Veli Shakhmurov

Copyright © 2010 Matti Pellikka et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We discuss how homology computation can be exploited in computational electromagnetism. We represent various cellular mesh reduction techniques, which enable the computation of generators of homology spaces in an acceptable time. Furthermore, we show how the generators can be used for setting up and analysis of an electromagnetic boundary value problem. The aim is to provide a rationale for homology computation in electromagnetic modeling software.

## **1. Introduction**

Setting up a well-posed electromagnetic boundary value problem encompasses setting up constraints that are related to the problem domain: the boundary conditions, field sources, and material parameters. These constraints usually surreptitiously involve the topology of the field problem domain. The division of the domain boundary to be governed by different boundary conditions, imposed source fields or circuit quantities, the division of the domain to conducting and nonconducting subdomains, and the use of field potentials bring up topological aspects that cannot be overlooked when defining the boundary value problem [1–6].

Failure to provide the above information in a consistent manner often leads to an over- or underdetermined system, and worse than that, the user as well as the software might be clueless where the problem lies. In this light, an automated, fast and definite procedure to find out the keystones of problem topology would save human effort and supersede problem-specific methods and heuristics in EM modeling software.

The tools of homology, capturing the topological nature of Maxwell's equations, can provide what is needed. Therefore, the efficient computation of absolute and relative homology groups has been under active study in computational electromagnetics community during recent years [7, 8]. The absolute homology links to topologically complicated

boundary value problem domain, while the relative homology is needed when the boundary values are imposed for the unknown fields, or the boundary value problem is coupled to external systems.

In this paper, we represent a succession of algorithms that, when given a cellular mesh, will lead to computation of generators of homology groups in an acceptable time and memory, establishing the viability of homology computation as an integral part of an electromagnetic modeling software. We also discuss how the information provided by homology computation can be exploited.

## 2. Homology

The study of homology is sewed around the concept of *boundary* of a *chain*, which is an intrinsic part of Maxwell's integral equations and Stokes' theorem. We define a *boundary operator* on chains, which maps chains to their boundary chains. A chain with zero boundary is called a *cycle*, and the boundary of the boundary of any chain is empty. Therefore, all boundaries are cycles. However, the converse is not true, there can be cycles that are not boundaries. This makes a dramatic difference in the integration of *closed fields* over cycles. By Stokes' theorem, the integrals of closed fields vanish over boundaries, but not necessarily over cycles. Therefore, if two cycles differ only by a boundary, integrals of closed fields over them are equal, and they can be considered equivalent in this sense. The *homology spaces* (they are generalizable into *moduli*, but we restrict to spaces with coefficient field of  $\mathbb{R}$  or  $\mathbb{C}$ , which are sufficient to our application) of a *chain complex* provide reliable means to classify cycles depending on their contribution to the integrals of closed fields. The elements of homology spaces are thus *cosets* of cycles.

To algebraize the concepts of boundary and chain, we form a finitely generated chain complex  $C(K)$  based on a cellular mesh  $K$  of the problem domain  $\Omega$ . The cellular mesh  $K$  is an instance of the *regular cell complex*, a finite element mesh for example. The domain  $\Omega$  is taken to be an *orientable manifold with boundary*. Now, the finite spaces of  $p$ -chains constitute a chain complex (Chain complex:  $\text{im}(\partial_{p+1}) \subset \ker(\partial_p)$  for each  $p$ ). under the boundary operator. It is represented by an incidence matrix  $\partial_p$ , one for each dimension  $p$ . If we chose to disregard part of the chains that lie in a subdomain  $\Gamma \subset \partial\Omega$ , covered by cell subcomplex  $L$ , we get so-called *relative chain complex*  $C(K, L)$  [5, 7]. (In a case boundary values of the unknown field are imposed  $\Gamma \subset \partial\Omega$ , we are not interested about the field integrals there.) This is done by forming an inclusion map  $i : C(L) \rightarrow C(K)$  and quotienting to get  $C(K)/i(C(L))$ . In boundary value problem, the quotient can be formed by dropping the rows and columns from  $\partial_p$  corresponding to elementary  $p$ -chains that lie in  $L$  (As there are no boundary conditions concerning a multiplicity of a boundary, this is a practical choice of representatives for the relative chains of  $C_p(\Omega, \Gamma)$ .) If  $L$  is empty, we call the chain complex *absolute*. Our discussion applies to absolute homology as well. Also, the distinction between the domains  $\Omega, \Gamma$  and the FE meshes  $K, L$  is disregarded, and  $\Omega$  and  $\Gamma$  are used throughout. (Chain space  $C_p(K, L)$  is finite dimensional whereas  $C_p(\Omega, \Gamma)$  is not, while  $H_p(K, L) \simeq H_p(\Omega, \Gamma)$ .)

### 2.1. Decomposition of Chains and Cohomology

It holds that for each  $p$ , the relative  $p$ -chain space has the nonunique decomposition [5]

$$C_p(\Omega, \Gamma) \simeq B_p(\Omega, \Gamma) \oplus H_p(\Omega, \Gamma) \oplus C_p^\partial(\Omega, \Gamma). \quad (2.1)$$

That is, a nonzero relative chain is a relative boundary of some  $p + 1$  chain, that it belongs to an equivalence class of chains that are neither boundaries of  $p + 1$  relative chains and do not have a relative boundary in the chain space  $C_{p-1}(\Omega, \Gamma)$ , or it is a relative chain that has a nonzero relative boundary. Spaces  $B_p(\Omega, \Gamma)$  and  $H_p(\Omega, \Gamma)$  constitute a space of  $p$ -chains whose boundaries are empty, the relative  $p$ -cycles  $Z_p(\Omega, \Gamma)$ .

It turns out that the *relative homology spaces, moduli* in general,  $H_p(\Omega, \Gamma)$  are interesting when posing an electromagnetic boundary value problem. (Restriction to space disregards *torsion*, which may occur in relative homology of orientable manifolds [7]. Our implementation uses integer computations to detect it.) They are isomorphic with *cohomology spaces*  $H^p(\Omega, \Gamma)$  of the (relative) *cochains*  $C^p(\Omega, \Gamma)$ . Cohomology is constructed on the *exterior derivative*, in our context of the de Rham cohomology,  $d$  in the similar manner homology concerns the boundary operator [1, 9]. (In general, this operator is called *coboundary operator*.) Cochains can be used to model the electromagnetic fields, and the integration of fields can be regarded as a bilinear mapping  $\int : C_p(\Omega, \Gamma) \times C^p(\Omega, \Gamma) \rightarrow \mathbb{R}$  of chains and cochains [1, 9]. The subdomain  $\Gamma \subset \partial\Omega$  is typically a region where a boundary condition is imposed upon an unknown field.

Cohomology and homology are related by de Rham's theorem. A field  $\omega \in C^p(\Omega, \Gamma)$  is *closed* if  $d\omega = 0$ , and *exact* if  $\omega = d\alpha$  holds. By de Rham's theorem, a field  $\omega \in C^p(\Omega, \Gamma)$  is exact if and only if

$$d\omega = 0, \quad \int_c \omega = 0, \quad \forall [c] \in H_p(\Omega, \Gamma). \quad (2.2)$$

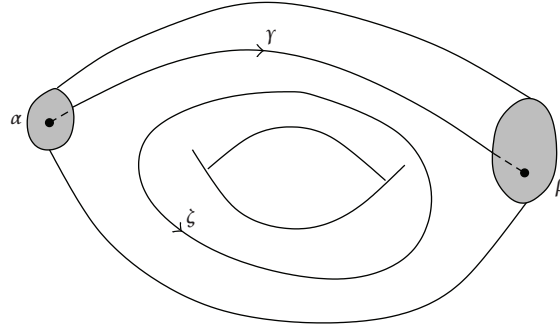
Fields that are closed but fail to be exact belong to the cohomology class  $H^p(\Omega, \Gamma)$ . Now, such fields can only exist if the homology space  $H_p(\Omega, \Gamma)$  contains nonzero elements. Otherwise by (2.2) and Stokes' theorem, all closed fields are also exact. This is *de Rham isomorphism* between homology and cohomology:  $H_p(\Omega, \Gamma) \simeq H^p(\Omega, \Gamma)$  for all  $p$  [9]. Therefore, computation of the relative homology of the boundary value problem domain gives us full understanding of how the boundary conditions fix the cohomology classes of the unknown fields, what is left to be determined because of tunnels or voids in the domain.

## 2.2. Some Properties and Examples of Homology Spaces

The relation between absolute and relative homology spaces can be summarized with the exact sequence of homology spaces [1]:

$$\cdots \longrightarrow H_p(\Gamma) \xrightarrow{i_*} H_p(\Omega) \xrightarrow{j_*} H_p(\Omega, \Gamma) \xrightarrow{\partial_p} H_{p-1}(\Gamma) \longrightarrow \cdots. \quad (2.3)$$

Here  $i_*$  and  $j_*$  are an inclusion and a projection induced by the corresponding mappings of chain spaces [5, 7]. That is, all the elements of  $H_p(\Omega, \Gamma)$  can be decomposed into such nonunique parts that either have a nonzero preimage in  $H_p(\Omega)$  or image in  $H_{p-1}(\Gamma)$ . This twofold structure of  $H_p(\Omega, \Gamma)$  emerges when the domain  $\Omega$  contains tunnels or voids, where field sources might be present in electromagnetic boundary value problem. Such sources can be taken into account with the aid of generators of  $H_p(\Omega, \Gamma)$ , which have a counterpart in  $H_p(\Omega)$ .



**Figure 1:** Two representatives of generators of  $H_1(\Omega, \Gamma)$ . One generator has a nonzero preimage in  $H_1(\Omega)$ , while the other has image in  $H_0(\Gamma)$ .

*Example 2.1.* A toroid  $\Omega$  where  $\Gamma$  consists of the two shaded walls in is shown Figure 1. 1-chain cosets  $[\gamma]$  and  $[\zeta]$  generate  $H_1(\Omega, \Gamma)$ . Generator  $[\zeta]$  has a nonzero preimage in  $H_1(\Omega)$ , while generator  $[\gamma]$  has image in  $H_0(\Gamma)$ :  $[\beta - \alpha]$ .

Let  $\Omega$  be an  $n$ -dimensional manifold. Then there exists an isomorphism [1, 9]

$$H_p(\Omega) \simeq H_{n-p}(\Omega, \partial\Omega). \quad (2.4)$$

If the boundary  $\partial\Omega$  is divided into two complementary nonoverlapping parts  $\Gamma_1$  and  $\Gamma_2$ , there exists an isomorphism [1, 9]

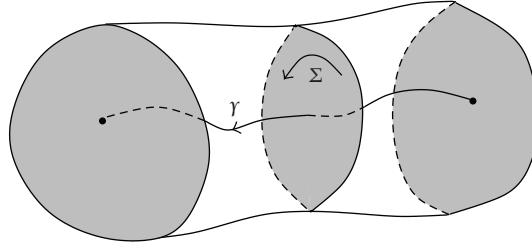
$$H_p(\Omega, \Gamma_1) \simeq H_{n-p}(\Omega, \Gamma_2). \quad (2.5)$$

These are so-called duality theorems, actually, compositions of the Lefschetz and de Rham duality theorems [1, 9], (It is crucial that the coefficient fields are the same on the homology and cohomology side.), which turn out to be useful together with the isomorphism between chains and cochains when considering dual formulations of an electromagnetic boundary value problem.

*Example 2.2.* Dual generators of a bar  $\Omega$  in are shown Figure 2.  $\Gamma_1$  consists of the opposite shaded ends of the tube, and  $\Gamma_2$  consists of the jacket. 1-chain coset  $[\gamma]$  generates  $H_1(\Omega, \Gamma_1)$ , while 2-chain coset  $[\Sigma]$  generates the dual space  $H_2(\Omega, \Gamma_2)$ .

### 3. Computation

From the data structure representing a cellular mesh, that is a regular cell complex, we can deduce the incidence matrices representing the boundary operators. The  $p$ th matrix contains the information about the boundary  $p-1$ -cells of  $p$ -cells. If a  $p-1$ -cell is on the boundary of a  $p$ -cell, the corresponding incidence index is either 1 or  $-1$ , depending whether their orientations match or not [5, 7]. Otherwise the incidence index is 0. The boundary operator matrices and knowledge about the cells belonging to  $K$  and  $L$  hold all the information we need.



**Figure 2:** Representatives of dual generators.

The computation of the  $p$ :th homology space is about finding the quotient space  $\ker(\partial_p)/\text{im}(\partial_{p+1})$ . In other words, the  $p$ :th relative homology-space is  $H_p(\Omega, \Gamma) = Z_p(\Omega, \Gamma)/B_p(\Omega, \Gamma)$ . The computation of a basis of this quotient space using the bases for  $C_p$  and  $C_{p+1}$  can be done with integer matrix decompositions of the boundary operator matrices, as discussed in [7, 8]. The decompositions are called Hermite normal form and Smith normal form. The problem is divided into three parts: finding bases for  $\ker(\partial_p)$  and  $\text{im}(\partial_{p+1})$ , finding an inclusion map  $i : \text{im}(\partial_{p+1}) \rightarrow \ker(\partial_p)$ , and finding a basis for quotient  $\ker(\partial_p)/i(\text{im}(\partial_{p+1}))$ .

Unfortunately, the computation of the relevant integer matrix decompositions has worse than  $O(\max(N_p, N_{p+1})^3)$  upper bound for computational time complexity, where  $N_p$  is the number of  $p$ -cells in the complex [10]. This is unacceptable, as finite element solution of a boundary value problem is a lot cheaper. However, the number of generators of  $H_p(\Omega, \Gamma)$  is practically always substantially smaller than  $N_p$ , which suggests that we need to look for means to reduce the size of the bases of  $p$ -chain spaces without altering the relative homology spaces. Only after that, the integer matrix decompositions of the reduced boundary operator matrices are used to compute the generators of homology spaces.

### 3.1. Problem Size Reduction

Homology spaces are topological invariants and they are isomorphic under *weak homotopy equivalences* [7, 11, 12]. Those properties provide that homeomorphisms and homotopy invariant mappings between topological spaces also preserve the homology spaces. Therefore, it is possible to apply certain homotopical operations to the problem domain  $\Omega$ , which keep the homology spaces  $H_p(\Omega, \Gamma)$  invariant. When the domain  $\Omega$  is covered by a regular cell complex  $C$ , two operations of this kind can easily be exploited [7, 13] as follows.

- (i) First is removing a cell that bounds only one cell, together with the cell it bounds. This makes a dent in the domain but not any holes. The corresponding homotopy invariant mapping of the underlying space  $\Omega$ , is called deformation retraction [12]. In [13], this is referred to as *exterior face collapsing*.
- (ii) Second is combining neighboring  $p$ -cells that share on their boundary a common  $p - 1$ -cell that is not bounding any other  $p$ -cells. This mapping leaves  $\Omega$  intact. It only alters the chain complex  $C$ , which still covers the same domain. Therefore, the topology of  $\Omega$  does not change. In [13], this is referred to as *interior face reduction*.

The aim is to perform and implement these two operations in such manner that as many cells as possible get removed with as little computational effort as possible. We have avoided

```

C := cell complex
Repeat
  for all  $p-1$ -cells  $c_{p-1} \in C$  do
    if  $c_{p-1}$  is on the boundary of exactly one  $p$ -cell  $c_p$  and
      ( $c_p, c_{p-1} \in \Omega \setminus \Gamma$  or  $c_p, c_{p-1} \in \Gamma$ ) then
       $C := C \setminus \{c_{p-1}, c_p\}$ 
    end if
  end for
until no cell has been removed

```

**Algorithm 1:**  $p$ -Reduce.

nonlocal reduction methods in order to avoid algebraic computations until the sizes of the bases of  $p$ -chain spaces are modest. The reason we split the reduction techniques in [13] in two separate algorithms is that the former alone is found more computationally efficient, and one should only after that resort to the latter.

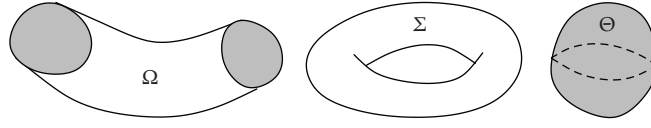
### 3.1.1. Reduction of Cells

If a  $p$ -cell  $c_p$  has a boundary  $p-1$ -cell  $c_{p-1}$  that does not bound any other  $p$ -cell, both of these cells can be removed from the cell complex without altering the relative homology spaces  $H_p(\Omega, \Gamma)$  [7, 13]. These cell pairs  $(c_{p-1}, c_p)$  satisfying the above condition are called *elementary reduction pairs*. Let  $N_p$  be the number of  $p$ -cells in the cell complex  $C$ .

An estimate for the computational complexity Algorithm 1 is in our interest. The order in which the cells are removed affects the efficiency of the algorithm, which ultimately depends on the index numbering of the cells. It is possible to construct examples where the repeat-until-loop must be ran  $\max(N_p, N_{p-1})$  times, resulting to at least quadratic worst case computational complexity. However, for a typical manifold of a boundary value problem, the loop runs only a couple of times. That gives us more promising estimate when the cell complex  $C$  is implemented as a binary search tree and the cells carry the information about their boundary cells. The computational complexity is then essentially due to the linear-time for-loop and the logarithmic-time operation of removing items from a binary search tree. This estimate is  $O(N_{p-1} \log(\max(N_p, N_{p-1})))$ .

The removal of a  $p$ - and a  $p-1$ -cell corresponds to the removal of a row and a column from the matrix of  $\partial_p$  and the removal of a row from the matrix of  $\partial_{p-1}$ . The matrix of  $\partial_{p+1}$  does not change as the removed  $p$ -cell cannot be on the boundary of any  $p+1$ -cell. In such case, no  $p-1$ -cell can be found that satisfies the if-clause in the algorithm. The algorithm can be applied to the cell complex in succession from the dimension  $n$  of the cell complex down to  $p = 1$ . Note though, that the matrix operations are not definitely the most efficient way of implementation.

The number of pairs  $(c_{p-1}, c_p)$  the algorithm is able to remove from the cell complex ranges from zero to all of them. Usually one successful removal triggers an avalanche of removals. Therefore, the cases where no cells can be removed are the most dramatic, and we enumerate the three possible cases (Figure 3) for each connected component  $\Omega_i$  of the problem domain  $\Omega$  when  $p$  is equal to the dimension  $n$  of the cell complex:



**Figure 3:** Three manifolds: a 3-dimensional tube  $\Omega$ , the surface of a toroid  $\Sigma$  ( $\partial\Sigma = \emptyset$ ), and a solid  $\Theta$  ( $\partial\Theta = \Gamma$ ), where  $\Gamma$  consists of the shaded regions. Reduction is able to reduce cells from cell complex  $C(\Omega, \Gamma_\Omega)$ , but not from complexes  $C(\Sigma)$  and  $C(\Theta, \Gamma_\Theta)$ .

- (1)  $\partial\Omega_i \neq \emptyset$ ,  $\partial\Omega_i \not\subseteq \Gamma$ : elementary reduction pairs  $(c_{n-1}, c_n)$  can be found. There exist  $n-1$ -cells in  $\partial\Omega_i \setminus \Gamma$  that bound only one  $n$ -cell of  $\Omega_i$ ;
- (2)  $\partial\Omega_i = \emptyset$ : no elementary reduction pairs  $(c_{n-1}, c_n)$  can be found. A manifold without a boundary: there can be no  $n-1$ -cells bounding only one  $n$ -cell, otherwise they would belong to the boundary;
- (3)  $\partial\Omega_i \subset \Gamma$ : no elementary reduction pairs  $(c_{n-1}, c_n)$  can be found. This is similar to the first case, but all the  $n-1$ -cells in possible elementary reduction pair candidates for removal belong to  $\Gamma$ .

In the last two cases no  $n-1$ -cells can be found that satisfy the if-clause in the  $n$ -Reduce algorithm. The inability to remove any pairs  $(c_{n-1}, c_n)$  implies that no lower dimensional pairs can be removed either (true for cell complexes whose underlying space is a manifold.) It seems that  $p$ -Reduce algorithm is toothless in many situations, but fortunately we can resort to an approach derived from the *Coreduction homology algorithm* [14].

### 3.1.2. The Reduce-Omit Algorithm

Coreduction, as opposed to reduction, is the procedure of removing  $p$ -cell and  $p-1$ -cell pairs  $(c_p, c_{p-1})$  when the  $p$ -cell  $c_p$  has only one  $p-1$ -cell  $c_{p-1}$  on its boundary. This pair  $(c_p, c_{p-1})$  is called an *elementary coreduction pair*. As no such pairs can be found from a complete cellular mesh, the algorithm is started by arbitrarily removing a single 0-cell, leaving pairs  $(c_1, c_0)$  which satisfy the condition. The algorithm goes on by removing one 0-cell each time there are no elementary coreduction pairs  $(c_p, c_{p-1})$  left, while there are still 0-cells left. The number of arbitrary removals needed to be done in order to banish all 0-cells equals the Betti number  $\beta_0$ , the number of connected components of the domain [14].

The 0-cell removal is justified by treating the cell complex as the *reduced cell complex* (constituting together with the *augmentation map*  $e : C_0 \rightarrow \mathbb{Z}$  an *augmented chain complex* [11].), where there is one additional  $-1$ -cell in each connected component of  $\Omega$ . Thus, the removal of 0-cell is interpreted as the removal of the coreduction pair  $(c_0, c_{-1})$ . For the *reduced homology* (homology for the augmented chain complex, an example of an *extraordinary homology theory* [11]).  $\widetilde{H}(C) \simeq \widetilde{H}(C \setminus \{c_0, c_{-1}\})$  holds [14]. For the reduced cell complex  $\widetilde{H}_0(\Omega_i) = 0$  holds for each connected component, and  $\widetilde{H}_p(\Omega) = H_p(\Omega)$  holds for all  $p \neq 0$  [11]. The information lost about  $H_0(\Omega)$  due to computing the reduced homology instead can be recovered by storing the arbitrarily removed 0-cells: they are the generators of  $H_0(\Omega)$ .

The algebraic dual of a cellular mesh is a mesh where the roles of the cells in the boundary relation of cells have been interchanged. We say  $c_1$  is on the coboundary of cell  $c_2$ , if  $c_2$  is on the boundary of  $c_1$ . So in the dual mesh, boundary cells of a cell become coboundary cells and vice versa. Also the dimensions of the cells get flipped over: if the dimension of



the manifold is  $n$ , the dual cell of a  $p$ -cell is an  $n - p$ -cell. For the dual mesh, the boundary operator matrices are  $\tilde{\partial}_p = \partial_{n-p}^T$  for each  $p$ , that is, the transposes of the boundary operators of the primal mesh [5].

Coreduction is a dual algorithm to reduction in the sense that, instead of checking whether a  $p - 1$ -cell is on the boundary of a single  $p$ -cell only, it checks whether a  $p$ -cell is on the coboundary of a single  $p - 1$ -cell only. *So the Coreduction can be interpreted as a reduction algorithm for the dual mesh in terms of primal mesh cells.* Due to the isomorphism between primal and dual cells, the coreductions can be regarded as weak homotopy equivalences.

A dual version of the launch of the Coreduction can be used to devise an extension to  $p$ -Reduce algorithm, but *only* in the case  $p$  equals  $n$ : Applying the arbitrary removal to  $n$ -cells from the cell complex each time it notices no elementary reduction pairs  $(c_{n-1}, c_n)$  could be found in the previous run, yet there still are  $n$ -cells left. The three cases enumerated earlier tell us that this is the case only when  $\partial\Omega_i$  is either empty or a subset of  $\Gamma$ . The  $p$ -Reduce algorithm detects this nonexistence of elementary reduction pairs in linear time. Similarly to the Coreduction, the number of arbitrary removals needed to banish all the  $n$ -cells from the cell complex tells us the dimension of  $H_n(\Omega, \Gamma)$ , the Betti number  $\beta_n$ . By duality theorem  $H_n(\Omega_i, \partial\Omega_i) \simeq H_0(\Omega_i)$ ,  $\beta_n$  is equal to the number of connected components  $\Omega_i$  of  $\Omega$ , for which either  $\partial\Omega_i = \emptyset$  or  $\partial\Omega_i \subset \Gamma$  holds.

Therefore, each arbitrary removal of a  $n$ -cell leads to a removal of all the  $n$ -cells in the given connected component of  $\Omega$  and the removed  $n$ -cells together with the “trigger”  $n$ -cell constitute a generator of  $H_n(\Omega, \Gamma)$ . This procedure of course leaves the  $n$ -generators out of the final algebraic problem and without orientation, but they can be recovered afterwards. The operation also clearly alters the topology of the domain, but in a controlled fashion. It is known exactly how many and which  $n$ -generators were omitted: one for each connected component  $\Omega_i$  of  $\Omega$ , for which either  $\partial\Omega_i = \emptyset$  or  $\partial\Omega_i \subset \Gamma$  holds. We call Algorithm 2 Reduce-Omit.

This strategy cannot be applied further down in the dimension  $p < n$ , as there is no guarantee that the  $p$ -generators have separate supports. Therefore, the arbitrary removal of a  $p$ -cell could lead to a removal of such set of  $p$ -cells, from which no representative of the generator of  $H_p(\Omega, \Gamma)$  can be constructed. The dimension  $n$  was exceptional, as all the  $n$ -generators had separate supports, and therefore all the removed  $n$ -cells constituted one  $n$ -generator. For this reason, it seems that the power of the Coreduction algorithm in Betti number computations does not carry over to homology generator computation.

### 3.1.3. Combining Cells

If a  $p - 1$ -cell resides on the boundary of exactly two  $p$ -cells, it is possible to combine these two  $p$ -cells to form single  $p$ -cell and remove the  $p - 1$ -cell from the cell complex without altering the covered domain  $\Omega$ . [7, 13].

Algorithm 3 has worst-case computational complexity of  $O(N_p N_{p-1})$ , where  $N_p$  is the number of  $p$ -cells in the cell complex. This is due to the fact that eventually most of the  $p - 1$ -cells get enqueued in  $Q$ . Another option would be a faster version without a queue, but experience on test runs suggests that the end-state of such algorithm will leave more cells in the cell complex. This trade-off gets worse as the size of the problem is increased, as one needs to face the algebraic decomposition in the end. Furthermore, accessing the boundary information of a cell is a constant-time operation, if memory is sacrificed to carry this information. The check whether a queue already contains an element, can be done in



```

C := cell complex
n := dimension of the cell complex C
 $\beta_n := 0$ 
for  $p = n$  to 1 do
   $p$ -Reduce ( $C, p$ )
end for
while cell complex C has  $n$ -cells  $c_n$  such that  $c_n \in \Omega \setminus \Gamma$  do
  choose  $n$ -cell  $c_n \in C$  such that  $c_n \notin \Gamma$ 
   $C := C \setminus \{c_n\}$ 
   $\beta_n := \beta_n + 1$ 
  for  $p = n$  to 1 do
     $p$ -Reduce ( $C, p$ )
  end for
  store  $c_n$  and  $n$ -cells removed by  $p$ -Reduce ( $C, n$ ) in previous loop.
end while

```

Algorithm 2: Reduce-Omit.

```

C := cell complex
Q := empty queue of  $p-1$ -cells
for all  $p$ -cells  $c_p \in C$  do
  enqueue boundary  $p-1$ -cells of  $c_p$  into Q
  while Q is not empty do
    dequeue  $c_{p-1}$  from Q
    If  $c_{p-1}$  is on the boundary of exactly two  $p$ -cells  $c_p^1, c_p^2$  and
      ( $c_p^1, c_p^2, c_{p-1} \in \Omega \setminus \Gamma$  or  $c_p^1, c_p^2, c_{p-1} \in \Gamma$ ) then
         $C := C \setminus \{c_{p-1}, c_p^1, c_p^2\}$ 
         $c_p^1 := c_p^1 \pm c_p^2$ , the sign chosen such that the orientations match
        enqueue boundary  $p-1$ -cells of  $c_p^1$  not already in Q to Q
      end if
    end while
  end for

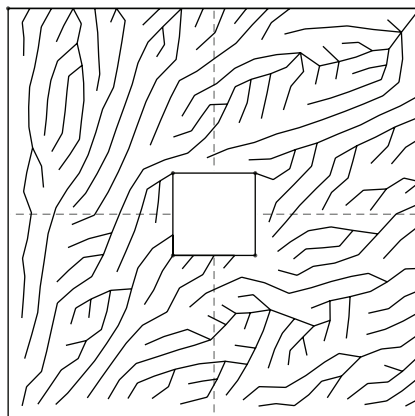
```

Algorithm 3:  $p$ -Combine.

logarithmic time by maintaining a binary search tree of the elements in the queue. Again, the end-state of the algorithm ultimately depends on the index numbering of the mesh vertices.

The combining of two  $p$ -cells and removal of a  $p - 1$ -cell correspond to an addition row-operation of two rows followed by removal of a row and a column in the matrix of  $\partial_p$ , an addition column operation of two columns followed by removal of a column in the matrix of  $\partial_{p+1}$  and a removal of a row in the matrix of  $\partial_{p-1}$ . Again, matrix operations are not an efficient way to reduce the mesh.

$p$ -Combine can be run for each dimension of the cell complex, starting from  $n$  down to 1. In addition, after each run of  $p$ -Combine,  $p$ -Reduce ( $C, p - 1$ ) can be run to get rid of “loose tails” that  $p$ -Combine leaves behind; see Figure 4.



**Figure 4:** A mesh of a square with a hole after cell combining. Loose tails are left behind, which can be removed with  $p$ -Reduce algorithm.

### 3.2. General Procedure for Homology Computation

Let us now propose a general procedure to find the generators of homology spaces  $H_p(\Omega, \Gamma)$  for all  $p$ . Let  $n$  be the dimension of the chain complex.

- (1) Run Reduce-Omit.
- (2) Run  $p$ -Combine followed by  $p$ -Reduce ( $C, p - 1$ ) for each  $p = n - 1, \dots, 1$ .
- (3) Find the generators of  $H_p(\Omega, \Gamma)$  using the reduced boundary operator matrices  $\partial_p$  for each  $p$  by HNF and SNF.
- (4) Recover the possible generators of  $H_n(\Omega, \Gamma)$  from the cells stored by Reduce-Omit (optional).

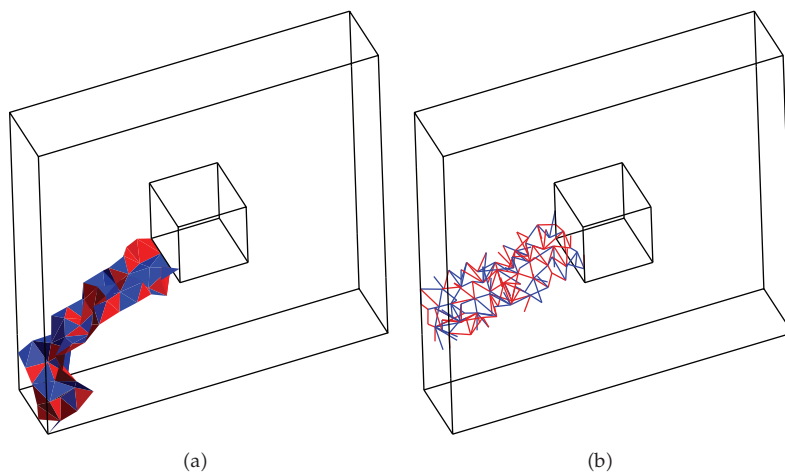
An implementation of this procedure is available in the open source finite element pre- and postprocessor Gmsh [15].

#### 3.2.1. Performance Discussion

The total computational complexity of the procedure above quite plausibly depends on the sizes and the ranks of the homology spaces, along with the index numbering of the mesh. Therefore, it is hard to give any definite estimates other than the complexities for individual algorithms in it. However, the upper and lower bounds can be given.

The ultimate upper bound is the computational complexity bound of the Smith normal form decomposition applied to the nonreduced problem [10]. However, it is highly unlikely that such situation will ever occur in practice with finite element mesh and homology groups without torsion. The worst-case end-states of  $p$ -Reduce and  $p$ -Combine are extremely difficult to discover, as they are homotopy invariances, and homotopy groups are known to be computationally challenging. In case the ranks of the homology spaces are zero for all  $p = n, n - 1, \dots, 1$ , the procedure has the computational complexity of  $p$ -Reduce.

In cases where  $p$ -Reduce is unable to ignite, the use of Reduce-Omit significantly increases the efficiency of this procedure, as it eliminates the need to give large inputs to



**Figure 5:** (a): A representative of the generator of  $H_2(\Omega, \partial\Omega)$  of a toroid. (b): A representative of the same generator but in terms of primal mesh cells, so-called thick-cut, computed by applying coreduction to  $C(\Omega)$ .

*p*-Combine. In case only the ranks of the homology spaces are of interest, the Coreduction homology algorithm can be applied [14]. It has the superior computational complexity of  $O(N \log N)$ , where  $N$  is the total number of cells in the cell complex.

### 3.2.2. A Strategy for Thick-Cuts

Coreduction was found to be a powerful tool to compute the Betti numbers of a cellular mesh, but the answer to whether it is suitable for computation of the generators of homology spaces, seems to be unfortunate. When  $p < n$  holds, the supports of the generators may overlap, because the recovery of the orientations of the generators is *complexity theoretically intractable* in general [7]. However, a modified algorithm can be applied to compute the generators, but in no less computational complexity than the reduction algorithm can.

One can apply Coreduction to the cellular mesh so that only 0-cells are arbitrarily removed in a similar but dual manner as in the Reduce-Omit algorithm. Then use straightforward dual versions of *p*-Combine and *p*-Reduce. Replace “boundary” with “coboundary” and “ $p-1$ -cell” with “ $p+1$ -cell” in Algorithms 3 and 1. After that, the homology computation with the reduced and *transposed* boundary operator matrices will find out the dual generators of the homology spaces, but in terms of primal cells. An example of this is so-called thick-cut, which we shall discuss more later.

**Example 3.1.** The established method for finding the generators of  $H_p(\Omega, \partial\Omega)$  is to apply Reduce-Omit and *p*-Combine to  $C(\Omega, \partial\Omega)$  and then compute the decompositions of  $\partial_p$  and  $\partial_{p+1}$ . Alternatively, one can apply this modified coreduction technique to  $C(\Omega)$ , as  $H_p(\Omega) \simeq H_{n-p}(\Omega, \partial\Omega)$  holds. Then compute the decompositions of  $\partial_{n-p}^T$  and  $\partial_{n-p-1}^T$  to find the same generators, but represented by the primal mesh cells; see Figure 5.

## 4. Uses

In this section we motivate the interest of computational electromagnetics community in homology computation. The purpose is to show its benefits in an electromagnetic modeling software.

Homology often relates to the electromagnetic phenomena outside the domain  $\Omega$ , when we want to couple them with the electromagnetic fields defined on the domain. In contrast, modeling decisions are also made to barricade certain kinds of coupling with the outside world. Nevertheless, the interface is always present, and its homological character is an intrinsic part of an electromagnetic boundary value problem.

Before the understanding of the role of homology and cohomology in electromagnetic modeling, case-specific and limited solutions have been used as substitutes. The genuine use of homology can offer a generally applicable and systematic framework for electromagnetic modeling and computation.

### 4.1. Setting up an Electromagnetic Boundary Value Problem

Setting up an electromagnetic boundary value problem with unique solution involves a variety of questions that, in general, are about the sources of the electromagnetic fields modeled. How does one set the field sources in  $\Omega$  together with the boundary conditions consistently? What does imposing a voltage between terminals actually mean? How does one impose a net current through a conducting domain that has terminals on the domain boundary  $\Gamma$ ? Given multiple terminals, how many and which voltages/currents does one need to impose in order to get a unique solution? When is a field potential sufficient to describe alone the unknown field in electromagnetic boundary value problem?

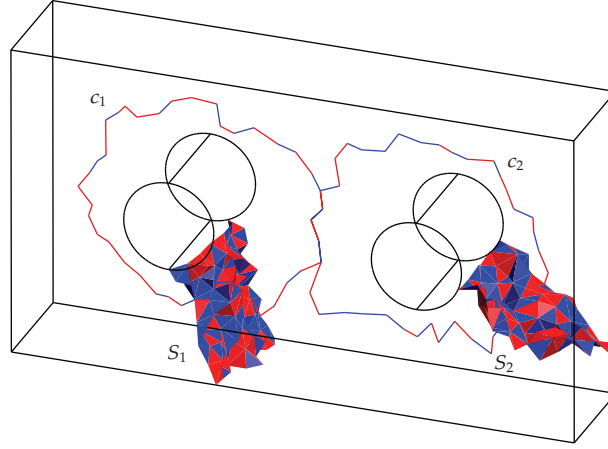
In simple problems, the above questions can be answered with reasoning based on knowledge about Maxwell's integral equations and circuit theory. But what if the problem at hand was so complex that simple reasoning would not help or if we wanted to automate the problem posing and validation in order to save human effort. Of course, the software cannot tell what the user wants, but it can help the user to do what he wants as soon as he tells it.

With the aid of homology computation, the software's answer to all of the above questions would be something like "Impose  $N$  of these integrals of these field quantities over these generators of homology spaces in order to get a unique solution." Next we will present some tangible examples.

### 4.2. Cuts for Scalar Potential

The use of scalar potential where possible is preferable in field computation, as it reduces the size of the system matrix: there are far less nodes than edges in a typical mesh. Let a closed field  $h \in Z^1(\Omega) = \{h \in C^1(\Omega) \mid dh = 0\}$  be the field of interest. Due to de Rham's theorem, field  $h$  can be exhaustively represented with a scalar potential  $\phi \in C^0(\Omega)$  such that  $h = d\phi$  holds, if and only if  $\int_c h = 0$  holds for all  $c \in Z_1(\Omega)$  [9]. That is, the circulation of  $h$  must vanish on all closed chains.

Whether this holds for all closed chains  $c$  depends on the homology of the problem domain. It holds by Stokes' theorem for all chains, that are boundaries, but not all cycles are necessarily boundaries. What is left to be determined are the integrals over nonbounding cycles, values  $I_i = \int_{c_i} h, [c_i] \in H_1(\Omega)$ , that is, the field circulations around the generators of the



**Figure 6:** Representatives  $c_1$  and  $c_2$  of generators of  $H_1(\Omega)$  and the corresponding representatives  $S_1$  and  $S_2$  of the dual generators of  $H_2(\Omega, \partial\Omega)$ .

homology space  $H_1(\Omega)$ , which is nonempty when the problem domain  $\Omega$  contains tunnels. However, this fails for a scalar potential alone, as  $\int_{c_i} d\phi = \int_{\partial c_i} \phi = 0$  for all  $[c_i] \in H_1(\Omega)$ . Therefore, the field  $h$  is decomposed into two parts,  $h = d\phi + T$ , where  $T$  satisfies  $dT = 0$ , and  $\int_{c_i} T = I_i$  for all  $[c_i] \in H_1(\Omega)$ . A systematic approach to fix these circulations for  $T$  and thereby for  $h$ , is to produce so-called thick-cuts, where the support of  $T$  entirely lies. This will guarantee sparse system matrix, and the problem is computationally cheap to solve.

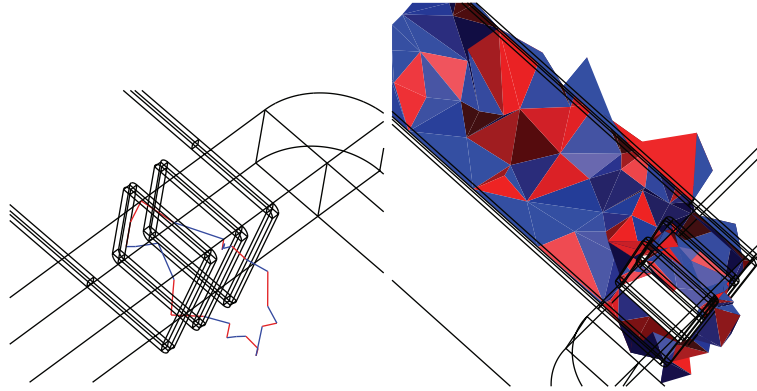
There is a multiplicity of methods for finding the thick-cuts. They often mix in homotopy, or rely on some strict assumptions about the domain  $\Omega$ . However, the cut surfaces are actually the corresponding duals for the generators of  $H_1(\Omega)$ , elements of  $H_2(\Omega, \partial\Omega)$ , reliably computable with our methods. The thick-cuts are the dual 1-chains of the 2-chain generators of  $H_2(\Omega, \partial\Omega)$ .

*Example 4.1.* Consider a magnetic core  $\Omega$  shaped like Figure 8, with coils wound around its branches. The cut surfaces  $S_1$  and  $S_2$ , the generators of  $H_2(\Omega, \partial\Omega)$ , are shown in Figure 6.

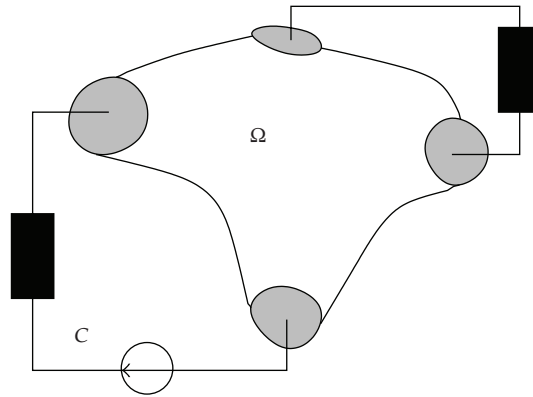
*Example 4.2.* Consider a wire wound around a magnetic core that is surrounded by air. Let  $\Omega$  consist of the core and air, see Figure 7. Now, the single generator of  $H_1(\Omega)$  is a simple loop around the wire. Its dual generator of  $H_2(\Omega, \partial\Omega)$  does not share this virtue, as it “cuts” all such loops.

### 4.3. Detection of Contradicting Source Quantities and Boundary Conditions

Field sources inside the problem domain  $\Omega$  and the boundary conditions on domain boundary  $\partial\Omega$  are often interrelated: the user can set source quantities and boundary conditions in such a fashion that Maxwell’s equations cannot hold in the problem domain and a solution is unattainable. Typically, the only symptom is the linear solver’s failure to converge. In such cases, a simple sanity check can be made using homology computation



**Figure 7:** On the left: a representative of the generator of  $H_1(\Omega)$ . On the right, the representative of the generator of  $H_2(\Omega, \partial\Omega)$ .



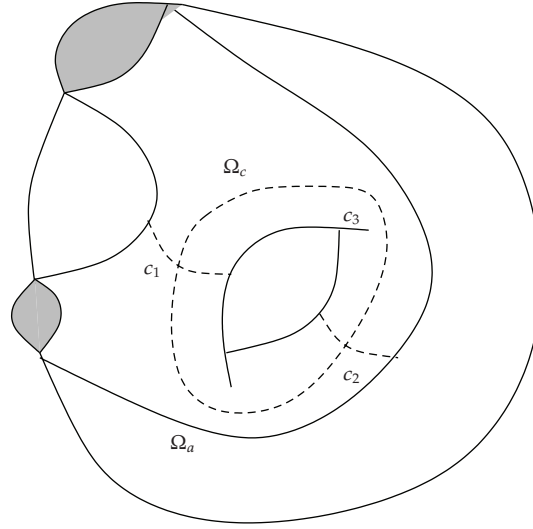
**Figure 8:** Coupled field-circuit problem. The circuit and field domains are isolated everywhere else except on the shaded region  $\Gamma \subset \partial\Omega$ .

before the solver is launched, to inform the user about possible clashes. For a more detailed discussion, see [16].

*Example 4.3.* Consider a magnetostatic problem where the domain  $\Omega$  is a current-carrying toroid. The problem is governed by the equations  $dh = j$  and  $db = 0$ . The source term  $j \neq 0$  is set inside the domain, and the boundary conditions  $th = f$  and  $tb = g$  are set on nonoverlapping regions  $\Gamma_h, \Gamma_b \subset \partial\Omega$ , which cover the whole domain boundary. Now, the magnetostatic equations require that

$$\begin{aligned} \int_{\partial S} h &= \int_S j, \quad \forall S \in C_2(\Omega), \\ \int_{\partial V} b &= 0, \quad \forall V \in C_3(\Omega) \end{aligned} \tag{4.1}$$





**Figure 9:** Multiformal field problem. It is often economical to use different field problem formulations in the conducting domain  $\Omega_c$  and in the nonconducting domain  $\Omega_a$ . 1-chain cosets  $[c_1]$ ,  $[c_2]$ , and  $[c_3]$  generate  $H_1(\partial\Omega_c \cap \partial\Omega_a)$ .

hold. In order to check whether our boundary conditions comply to those requirements, it is sufficient to check that [16]

$$\begin{aligned} \int_{\partial s} th &= \int_s j, \quad \forall [s] \in H_2(\Omega, \Gamma_h), \\ \int_{\partial v} tb &= 0, \quad \forall [v] \in H_3(\Omega, \Gamma_b), \end{aligned} \quad (4.2)$$

where  $s$  and  $v$  are representatives of the generators of the homology spaces, readily computable with our methods. If the space  $H_2(\Omega, \Gamma_h)$  is not empty and the user has for example, set  $th = 0$  on  $\Gamma_h$  and nonzero net current through  $s$ , then a clash will occur.

This procedure generalizes to all electromagnetic boundary value problems because all of the Maxwell's equations are of the form

$$\int_{\partial c} \alpha = \int_c \omega, \quad (4.3)$$

where  $\omega$  is the field source, set in the domain  $\Omega \setminus \Gamma$ , and  $\alpha$  is the unknown field, subject to a boundary condition  $t\alpha = f$  on  $\Gamma$ . A clash will occur, if  $\int_{\partial c} t\alpha \neq \int_c \omega$  for some  $[c] \in H_p(\Omega, \Gamma)$ .

#### 4.4. Setting Up Coupled Circuit-Field Problems

Consider a coupled electric circuit-field problem, where Kirchhoff's laws hold in the circuit domain  $C$  and Maxwell's equations hold in the field domain  $\Omega$  (Figure 8). To make sure that the field problem constitutes a circuit element, we restrict to the case where the domain  $\Omega$

contains no tunnels and subject it to the Kirchhoff's current law:  $\int_{\partial\Omega} j = 0$ . The state variables of the circuit problem are voltages and currents, which describe the power in the system. The circuit problem and the field problem are isolated in such manner that the field problem constitutes a circuit element. The interfaces between the circuit and the field problem are the conductor terminals on  $\Gamma \subset \partial\Omega$ . All the power flux by voltages and currents between  $C$  and  $\Omega$  is accounted by this interface only, on which voltage and current must have a precise meaning. The isolation is built by boundary conditions

$$\begin{aligned} te &= 0, & \text{on } \Gamma, \\ td &= 0, & \text{on } \partial\Omega, \\ tb &= 0, & \text{on } \partial\Omega, \\ tj &= 0, & \text{on } \partial\Omega \setminus \Gamma. \end{aligned} \tag{4.4}$$

That is, no magnetic or electric flux passes through the boundary  $\partial\Omega$ , and current can only pass through the terminals  $\Gamma$ . By  $tb = 0$  on  $\partial\Omega$  we are guaranteed to have scalar potential there. Terminals  $\Gamma$  must be equipotential surfaces to be eligible nodes of the circuit. In order to get an unique solution for the field problem, one needs to set a number of voltages between the terminals and/or currents through them. These constraints are either the state variables of the circuit problem, or preset variables [17].

The generators of the relative homology spaces can be used to interpret the coupling through the terminals on the field problem side. Imposing field integrals along the generators corresponds to providing enough information to construct an impedance matrix of the field problem. We will represent two examples where this is intuitively clear. However, homology computation can bring this "intuition" to the computer, and aid the human intuition in more complex problems. It can tell exactly how many "nonlocal" constraints are needed, and the visualizations of the generators help to understand their meaning.

*Example 4.4.* Consider a DC resistor model in Figure 2 with conducting region  $\Omega$ , two terminals on  $\Gamma$  and an insulating surface  $\partial\Omega \setminus \Gamma$ . By intuition and experience, one can tell that a unique solution is found by imposing a voltage difference  $V$  between the terminals. In terms of homology, that is, the integral of electric field  $e$  along the single generator  $[\gamma]$  of  $H_1(\Omega, \Gamma)$ :  $\int_{\gamma} e = V$ . This reveals that we are actually interested in the EMF between the terminals, and the familiar scalar potential is just one way to impose it. This widens the understanding of the situation beyond the cases where the existence of a scalar potential can be guaranteed everywhere in  $\Omega$ . However, in coupled field-circuit problems, the scalar potential must exist on the surface  $\partial\Omega$ .

Alternatively, instead of the voltage between the terminals, one can set the net current  $I$  through the resistor. Again, in the language of homology, this is the integral of current density  $j$  over the single generator  $[\Sigma]$  of  $H_2(\Omega, \Omega \setminus \Gamma)$ , which is the dual generator of  $[\gamma]$ . So the user needs to set  $\int_{\Sigma} j = I$ . This together with  $dj = 0$  guarantees that  $\int_{\Gamma} j = 0$  will hold. If neither or both the voltage and the net current are set, one is left with an under/over-determined system with no unique solution [2], about which homology-aware software is able to inform the user.

*Example 4.5.* Consider a general problem with the full set of Maxwell's equations. Let the domain  $\Omega$  have four terminals on  $\Gamma \subset \partial\Omega$ , as in Figure 8, and it may consist of conducting

and nonconducting regions. Not all terminals are necessarily galvanically connected to each other. Now we can find three generators of  $H_1(\Omega, \Gamma)$  and  $H_2(\Omega, \partial\Omega \setminus \Gamma)$ , respectively. Imposing voltages between the terminals on  $\Gamma$  corresponds to imposing EMFs along the generators of  $H_1(\Omega, \Gamma)$ .

More interestingly, imposing integral of  $j + \partial_t d$  over generators of  $H_2(\Omega, \partial\Omega \setminus \Gamma)$  corresponds to imposing currents through terminals, as no displacement current  $\partial_t d$  escapes from  $\Omega$  and the continuity equation  $\int_{\partial v} (j + \partial_t d) = 0$  must hold for all volumes  $v$  inside  $\Omega$ . Conversely, imposing currents through terminals may impose displacement currents in nonconducting parts of the domain  $\Omega$ , as well as currents in conducting parts. Homology-aware software would be able to help the user to perceive what has been imposed. In total, three EMFs or fluxes of  $j + \partial_t d$  must be imposed to get an unique solution to the problem.

#### 4.5. Multiformalational Field Problem

In many problems, it is possible to use computationally more efficient formulations in some parts the computational domain  $\Omega$ . For example, the conducting and nonconducting parts of the domain can be treated separately. However, the price for the computational efficiency is the increased effort to set up the problem, as the coupling between the subdomains must be taken into account.

*Example 4.6.* Let the domain  $\Omega$  consist of a conducting subdomain  $\Omega_c$  and nonconducting subdomain  $\Omega_a$ . In  $\Omega_c$ , we can either have an imposed time-varying source current, or it can be coupled to an external circuit through terminals on  $\Gamma \subset \partial\Omega_c$ , as described in previous section. However, let  $\Omega_a$  have tunnels, through which the conducting domain  $\Omega_c$  extends.

Now, in the air region  $\Omega_a$ , we seek for the solution of the magnetic field  $h$  only, for which  $dh = 0$  holds. We are able to use a scalar potential formulation with the thick-cut technique, with expression  $h = T_a + d\phi_a$ . But what should be the total currents  $I_i$  through the holes in region  $\Omega_a$ , fixing the circulations  $\int_{c_i} h = \int_{c_i} T_a$ ,  $[c_i] \in H_1(\partial\Omega_c \cap \partial\Omega_a)$ ?

In the conductor region  $\Omega_c$ , in addition to the magnetic field, we are interested in the electric field and the corresponding currents. Therefore, we look for the solution of the (coupled) magnetoquasistatic problem defined there. In other words,  $dh = j$ ,  $db = 0$ , and  $de + \partial_t b = 0$  must hold, where  $j$  consists of imposed source current and eddy currents generated by the time-varying magnetic flux, and  $tj = 0$  on  $\partial\Omega_c \cap \partial\Omega_a$ . The problem is solved by decomposition  $h = T_c + d\phi_c$  in  $\Omega_c$ , and on  $\partial\Omega_c \cap \partial\Omega_a$  by setting  $\phi_a = \phi_c$  and outside the thick-cuts  $tT_c = 0$  [6]. We note that the required currents

$$I_i = \int_{s_i} j = \int_{s_i} dT_c = \int_{\partial s_i} T_c = \int_{\partial s_i} T_a, \quad [s_i] \in H_2(\Omega_c, \partial\Omega_c \cap \partial\Omega_a), \quad (4.5)$$

which are also the independent currents of the coupled problem, as  $\partial\Omega_c \cap \partial\Omega_a = \partial\Omega_c \setminus \Gamma$ .

## 5. Conclusions

A compilation of mesh reduction methods was presented, leading to a general procedure for practical homology computation. The procedure was implemented within this work, and

the benefits of homology computation to electromagnetic modeling were discussed. It was suggested that the homology computation, when efficiently implemented, can be a useful preprocessing aid in electromagnetic modeling software. It can be used as a problem posing companion as well as a guardian of the solver, reducing the human effort put to the problem.

Many engineering problems, solved with methods of computational electromagnetism, fall into predefined problem categories, for which templates are readily available in the software. This is helpful for the engineer, but it also narrows down the class of problems a software is able to solve. A minor variant of a commonplace problem can make it unsolvable because of some needless but guiding restriction within the software. Homology computation can help the software designer to give freer hands to the user without reducing the software's usability.

## References

- [1] P. R. Kotiuga, *Hodge decompositions and computational electromagnetics*, Ph.D. thesis, Electrical and Computer Engineering, McGill University, Montreal, Canada, 1984.
- [2] A. Bossavit, "Most general "non-local" boundary conditions for the Maxwell equation in a bounded region," *COMPEL*, vol. 19, no. 2, pp. 239–245, 2000.
- [3] L. Kettunen, "Fields and circuits in computational electromagnetism," *IEEE Transactions on Magnetics*, vol. 37, no. 5, pp. 3393–3396, 2001.
- [4] S. Suuriniemi, T. Tarhasaari, and L. Kettunen, "Generalization of the spanning-tree technique," *IEEE Transactions on Magnetics*, vol. 38, no. 2, pp. 525–528, 2002.
- [5] J. Kangas, *Algebraic decompositions in analysis of quasistatic electromagnetic problems*, Ph.D. thesis, Department of Electrical Engineering, Tampere University of Technology, Tampere, Finland, 2007.
- [6] R. Specogna, S. Suuriniemi, and F. Trevisan, "Geometric  $T$ - $\Omega$  approach to solve eddy currents coupled to electric circuits," *International Journal for Numerical Methods in Engineering*, vol. 74, no. 1, pp. 101–115, 2008.
- [7] S. Suuriniemi, *Homological computations in electromagnetic modeling*, Ph.D. thesis, Department of Electrical Engineering, Tampere University of Technology, Tampere, Finland, 2004.
- [8] T. Kaczynski, K. Mischaikow, and M. Mrozek, *Computational Homology*, vol. 157 of *Applied Mathematical Sciences*, Springer, New York, NY, USA, 2004.
- [9] P. W. Gross and P. R. Kotiuga, *Electromagnetic Theory and Computation: A Topological Approach*, vol. 48 of *Mathematical Sciences Research Institute Publications*, Cambridge University Press, Cambridge, Mass, USA, 2004.
- [10] C. K. Yap, *Fundamental Problems of Algorithmic Algebra*, Oxford University Press, New York, NY, USA, 2000.
- [11] J. R. Munkres, *Elements of Algebraic Topology*, Perseus Books, Cambridge, Mass, USA, 1984.
- [12] A. Hatcher, *Algebraic Topology*, Cambridge University Press, Cambridge, Mass, USA, 2002.
- [13] T. Kaczyński, M. Mrozek, and M. Ślusarek, "Homology computation by reduction of chain complexes," *Computers & Mathematics with Applications*, vol. 35, no. 4, pp. 59–70, 1998.
- [14] M. Mrozek and B. Batko, "Coredution homology algorithm," *Discrete and Computational Geometry*, vol. 41, pp. 96–118, 2000.
- [15] C. Geuzaine and J.-F. Remacle, "Gmsh: a 3-D finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.
- [16] S. Suuriniemi, J. Kangas, and L. Kettunen, "Formalization of contradicting source quantities and boundary conditions in quasi-statics," *IEEE Transactions on Magnetics*, vol. 43, no. 4, pp. 1253–1256, 2007.
- [17] S. Suuriniemi, J. Kangas, L. Kettunen, and T. Tarhasaari, "State variables for coupled circuit-field problems," *IEEE Transactions on Magnetics*, vol. 40, no. 2, pp. 949–952, 2004.