

# HOMOLOGY AND COHOMOLOGY COMPUTATION IN FINITE ELEMENT MODELING

M. PELLIKKA<sup>†</sup>, S. SUURINIEMI<sup>†</sup>, L. KETTUNEN<sup>†</sup>, AND C. GEUZAINÉ<sup>‡</sup>

**Abstract.** A homology and cohomology solver for finite element meshes is represented. It is an integrated part of the finite element mesh generator *Gmsh*. We demonstrate the exploitation of the cohomology computation results in a finite element solver, and use an induction heating problem as a working example. The homology and cohomology solver makes the use of a vector-scalar potential formulation straightforward. This gives better overall performance than a vector potential formulation. Cohomology computation also clarifies the lumped parameter coupling of the problem and enables the user to obtain useful post-processing data as a part of the finite element solution.

**Key words.** homology computation, cohomology computation, finite element method, lumped parameter coupling, electromagnetics

**AMS subject classifications.** 57R19, 58Z05, 65M60, 78M10

**1. Introduction.** We present a tool for the homology and cohomology computation of domains tessellated with finite element meshes. The tool is an integrated part of the finite element mesh generator *Gmsh* [17]. Homology and cohomology computation can be exploited to exhaustively fix the so-called cohomology class of the solution of a boundary value problem that is solved with the finite element method. As a concrete application, we demonstrate how such computations greatly benefit the modeling of an induction heating machine.

In boundary value problems that involve the Hodge-Laplace operator, one often needs to choose the cohomology class of the solution. Such problems are usual in electromagnetics, which is why our working example is chosen from that field. The cohomology classes of the problem are generated by the choice of the boundary conditions and by the homology of the problem domain. Informally, homology is about the quantity and the quality of holes in an object, whether it has voids or tunnels or both. Relative homology captures whether the object has holes when one “disregards” a part of the model. In the finite element method, the disregarded part is a subdomain where the solution is fixed by a boundary condition. Cohomology can be characterized by saying that it assigns quantities to these holes. In boundary value problems such assignments fix the cohomology class of the solution. For the technical definitions of homology and cohomology spaces, see appendix A.

In the finite element method, typically only a bounded portion of the device and the surrounding space is modeled. The modeling domain may contain holes, and boundary conditions are assigned to confine the fields and couple them with external phenomena outside the domain. Further, the domain can be split into many coupled regions where different approximations and potential formulations are being employed. These modeling aspects give rise to homology and cohomology and their relative forms in numerical models, since one is required to assign source quantities to entities that are absent from the model.

For electrical engineers, an evident manifestation of homology and cohomology are Maxwell’s equations in their integral form. Classically, electromagnetic phenom-

---

<sup>2</sup>Tampere University of Technology, Dept. of Electrical Engineering, Electromagnetics, Finland

<sup>3</sup>University of Liege, Dept. of Electrical Engineering and Computer Science, Montefiore Institute, Belgium

ena are assumed to take place in the entire 3-dimensional Euclidean space that has no holes. However, once boundaries or holes are introduced to the modeling domain, the coupling with the external electromagnetic phenomena can often be formalized with cohomology. For a more detailed discussion, see [18]. To employ so-called magnetic field conforming method [4] in our working example, cohomology computation is needed to make the use of an efficient vector-scalar potential formulation straightforward. Making it easy to use such a formulation instead of having to resort to a pure vector potential formulation is the main benefit of the cohomology computation. As other benefits, it helps to couple the problem with circuit models, provides efficient and accurate post-processing possibilities, and in general gives a structural insight into such problems. Methods that exploit homology and cohomology in electromagnetics are also presented in [11, 10, 32]. As heat conduction is in many respects analogous to electrostatics, the presented homology and cohomology solver can be exploited in heat conduction boundary value problems as well. The role of homology and cohomology have also been recognized in the field of linear elasticity [1, 37]. Thus, homology and cohomology computations might also have applications in that field.

Another related application for cohomology computations is the parametrization of surface meshes. Good quality surface meshes of objects are desirable in computer graphics and are a prerequisite for 3D-meshes of good computational quality. To produce a good surface mesh, a surface parametrization is often needed [17, 30]. A closed surface cannot be parametrized with a single coordinate chart, for a discontinuity is introduced at the coordinate patch boundary. To avoid this, cohomology computation can be exploited to find a global conformal parametrization for surfaces with an arbitrary genus [19]. For compact non-closed surfaces with boundary and holes, relative cohomology might be exploited in a similar manner.

The presented homology and cohomology solver is an integrated part of the finite element mesh generator *Gmsh* [17]. It can also be accessed through the application programming interface (API) of *Gmsh*. Thus, homology and cohomology computation can be embedded as a part of the mesh generation workflow. The working example is solved using the finite element solver *GetDP* [12] which is able to fluently exploit the results of the cohomology solver in its problem definition. However, any solver can be programmed to exploit the results, as they are saved along with the finite element mesh.

The design principles of the presented homology and cohomology solver are the following. First, it operates on large, up to 3 dimensional finite element meshes that are possibly non-simplicial. It is expected that the usual input is such that the *reduction techniques* are able to tremendously reduce the size of the algebraic problem. Second, it produces a basis representation of the homology and cohomology spaces in addition to *Betti numbers*. Such output is needed in finite element solvers as we will demonstrate. Examples of other software that perform homology and cohomology computation include *CHomP* [6], *jPlex* [33], and *GAP homology* [22] packages. The design objectives of these popular packages are slightly different from ours, with less emphasis on problem position in finite element modeling.

**2. The homology and cohomology solver for up to 3D meshes.** A finite element mesh is easily converted into a *regular cell complex*<sup>1</sup> [20] which is the

---

<sup>1</sup>A mesh that contains only triangles or tetrahedra can be converted into a special case called a *simplicial complex*, which is often used in the introductory expositions of homology computation. The regular cell complex is required to handle more general finite element meshes, for instance, containing hexahedra, prisms, and pyramids.

usual input for homology or cohomology computation. In the presented homology and cohomology solver, the extraction of the cell complex from the finite element mesh is the first stage of the computation. From the cell complex, integer matrix representations of so-called boundary operators can be obtained. They are basically the volumes-to-faces, faces-to-edges, and edges-to-vertices incidence matrices of the mesh.

The general algorithm for homology and cohomology group<sup>2</sup> computation is the Smith normal form integer matrix decomposition, but it suffers from high worst case computational complexity [36]. Thus, before the matrix decomposition is employed, the problem size is reduced using a variety of methods that usually run in linearithmic time [23, 34]. In the homology and the cohomology computation of the finite element meshes, the problem size is reduced by removing “homologically irrelevant” parts of the mesh and the mesh elements in the relevant parts are combined. These operations keep the homology of the mesh invariant [23]. Heuristically, the problem size is usually reduced by an order of  $10^3$  to  $10^5$  depending on both the geometry and the topology of the mesh. In the represented homology and cohomology solver, reduction algorithms described in Pellikka et al. [28] are employed. A short description on the essentials of the solver is given in the section 2.2.

The input for the solver is a mesh that is generated by *Gmsh* or imported into *Gmsh*. The solver only uses the connectivity information of the mesh elements, and ignores the coordinates of the mesh vertices. Therefore, the solver can be utilized with any data set that can be realized as an at most 3-dimensional simplicial complex, once it is converted to a *Gmsh*-readable mesh format. In section 2.1, we consider an example where a point cloud in  $\mathbb{R}^3$  is interpreted as a 2-dimensional simplicial complex.

Results of the homology and cohomology solver are the bases for the relative homology spaces  $H_k(M, S)$  and the relative cohomology spaces  $H^k(M, S)$  of the domain  $M$  and its subdomain  $S$  for each  $k$  from 0 to 3. The bases are represented by a set of  $k$ -chains and  $k$ -cochains whose *cosets*<sup>3</sup> form the actual bases. With the finite element mesh, the bases thus have matrix representations  $H_k = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{\beta_k}]$  and  $H^k = [\mathbf{z}^1, \mathbf{z}^2, \dots, \mathbf{z}^{\beta_k}]$ , respectively, where for  $\beta_k = \dim H_k(M, S) = \dim H^k(M, S)$  holds.<sup>4</sup> The sparse integer coefficient vectors  $\mathbf{z}_i$  and  $\mathbf{z}^i$  contain the information which  $k$ -dimensional mesh cells are a part of the corresponding  $k$ -(co)chain  $z_i$  or  $z^i$ .

The homology and cohomology solver produces such matrices given a finite element mesh of a computational domain  $M$  with a designated subdomain  $S$ . The basis representations are non-unique on two levels. First, the representative chains and cochains can be any from the corresponding coset. Second, the basis can be transformed to an another basis, with different basis cosets, by an unimodular transformation [36]. The basis representation issue is further discussed in Pellikka et al. [29].

**2.1. Example: surface reconstruction from a point cloud.** A technique to construct a surface mesh based on a point cloud in  $\mathbb{R}^3$  is called *alpha shapes* [15]. They aim to approximate the “shape” of a point cloud, where the level of detail is

<sup>2</sup>Inherently, homology and cohomology have the structure of an *abelian group*, which is also solved by the presented homology and cohomology solver. However, in the context of the finite element method, we are primarily interested on the homology and cohomology *vector spaces* instead. The *free* subgroup of an abelian group is interpreted as the vector space of interest.

<sup>3</sup>Two  $k$ -chains that are  $k$ -cycles belong to the same homology coset, if their difference is a  $k$ -boundary. This equivalence relation generalizes to the relative  $k$ -cycles and  $k$ -cocycles.

<sup>4</sup>If  $S$  is empty,  $\beta_k$  is called the  $k$ :th Betti number of  $M$ . Then,  $\beta_0$  is the number of connected components of  $M$ ,  $\beta_1$  is the number of tunnels through  $M$ , and  $\beta_2$  is the number of voids in  $M$ .

expressed with a parameter  $\alpha$ . As extrema, if  $\alpha = \infty$  holds, the resulting shape is the convex hull of the point cloud. If  $\alpha = 0$  holds, the resulting shape is empty.

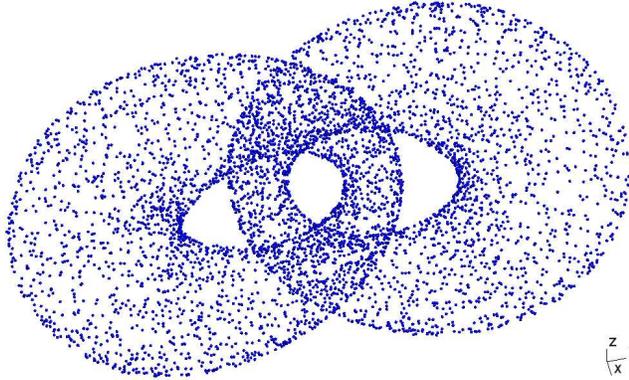


Fig. 2.1: A point cloud of 6000 points in  $\mathbb{R}^3$  whose shape is to be determined.

The usual problem is to choose such an  $\alpha$  that a shape with desired characteristics is recovered. The alpha shape is always a subcomplex of the Delaunay triangulation of the point cloud. Therefore, the homology solver of *Gmsh* can be applied to compute its topological features.

In this example, we use *a priori* knowledge of the shape to find a suitable value of  $\alpha$ . Namely, we insist that the shape consists of two tori. As a single torus has Betti numbers  $\beta_0 = \beta_2 = 1$  and  $\beta_1 = 2$ , we can use the homology solver to find such  $\alpha$  that the Betti number criteria for the shape is met. In Fig 2.1 we have depicted an example point cloud. Fig 2.2 depicts the shape the point cloud determined with different values for  $\alpha$  along with the Betti numbers of the shape. We used the program *Hull* [7] for the alpha shape computation.

**2.2. Homology and cohomology computation process.** The homology and cohomology solver in *Gmsh* relies on reduction of chain complexes [23] before the computation of the actual homology or cohomology groups. It is due to the reduction algorithms that the computed bases have somewhat small support. See Fig. 7.3 for example. While the details of the employed reduction algorithms are presented in Pellikka et al. [28], here we describe the overall strategy how the reduction algorithms are combined together to make up the solver in *Gmsh*.

The first step of homology and cohomology computation is to construct a *chain complex* [26, 20] from the finite element mesh. The chain groups  $C_k$  are formal sums of the mesh cells of each dimension  $k$ . That is, to construct the chain complex from an  $n$ -dimensional mesh, first all the  $n$ -dimensional mesh elements are enumerated. Then, the set of their unique  $n - 1$ -faces is constructed and the faces are oriented. Thereafter, the  $n - 2$ -faces of the  $n - 1$ -cells are processed similarly and so on, until the 0-dimensional cells are reached. To construct the *boundary operators* of the chain complex, the incidence relations of the cells are stored along the process. The chain complex construction has computational complexity bound  $O(n \log n)$ , where  $n$  is the number of most abundant  $k$ -cells.

Once the chain complex has been constructed, the chain complex reduction takes place. The reduction algorithms for homology and cohomology computation have sub-

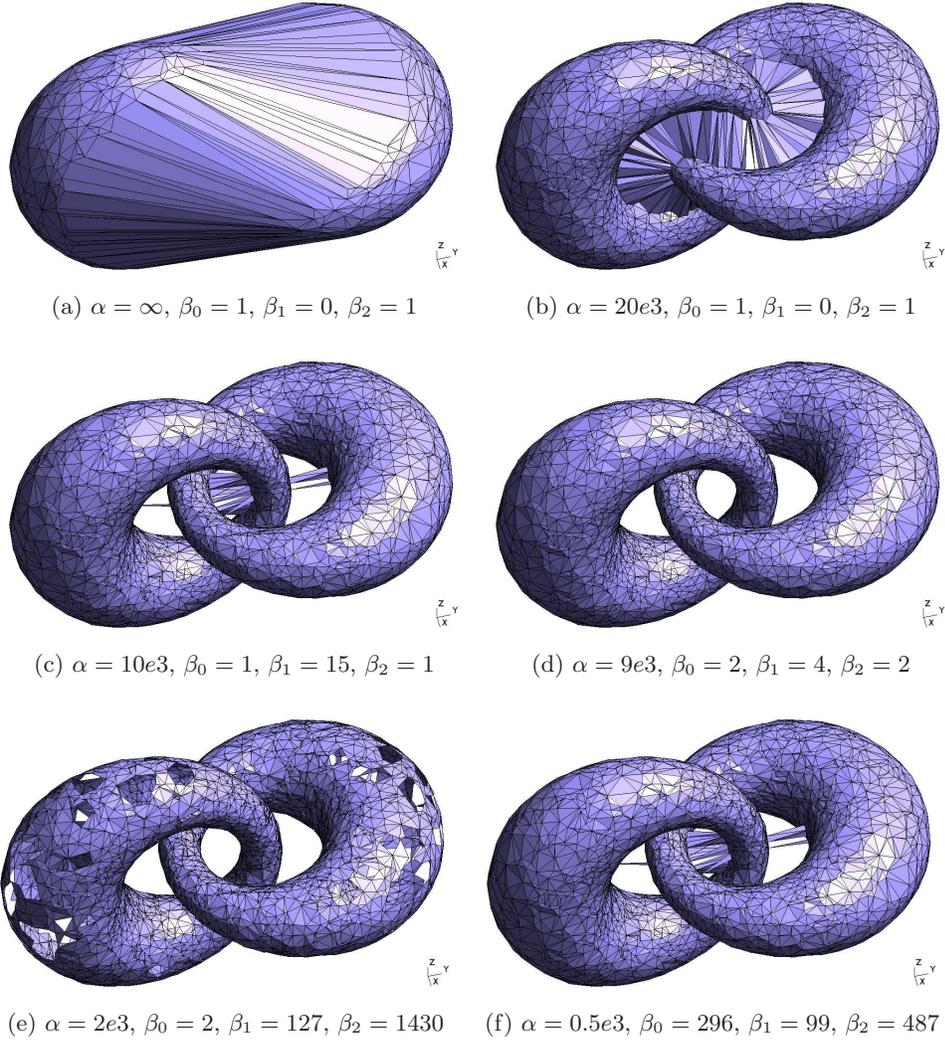


Fig. 2.2: Alpha shapes with different values for  $\alpha$  and their Betti numbers.

tle differences, but they all rely on *chain equivalences* [26]. A chain equivalence maps a chain complex to a chain complex with equivalent homology and cohomology. The reduction algorithms apply chain equivalences that reduce the number of chain group basis elements computationally efficiently. Effectively, this makes the mesh incidence matrices smaller in the last stage of the homology and cohomology computation.

The chain equivalences employed in *Gmsh* have geometric interpretations, that are depicted in Fig. 2.3. They appear in the following works: [23, 25]. Their computational complexities range from  $O(n \log n)$  to  $O(n^2)$ , where  $n$  is the size of a basis of a chain group that is being modified. The algorithms are applied in succession so that most of the work is done by algorithms with lowest computational complexities.

The reduction techniques are able to reduce the size of the problem by a large

factor when the input is a typical finite element mesh. That is, when the Betti numbers of the chain complex are much smaller than the size of the bases of the chain groups, i.e. the number of mesh elements. For example, with the mesh in Fig. 2.2f, which no longer resembles a typical finite element mesh, the reduction factor is only 94.4893 %. Compare that to the reduction factor of 99.9981 % in the computation of the results in Fig. 7.3. In conclusion, the reduction techniques are a good heuristic for homology computations within the finite element method.

The reduction algorithms are responsible for the somewhat geometrically smooth appearance of the computed homology and cohomology bases. They remove mesh cells from the same vicinity instead of in a random manner. This forces the actual homology and cohomology computation to operate on a chain complex that is geometrically localized. That is, the employed reduction algorithms are also a good heuristic for the geometric appeal of the results. Except for *local deformation* [29] of the homology basis representatives, we do not apply more advanced methods to attain appealing bases with respect to some criteria. However, there is a lot of research on such methods [8, 38, 9, 16].

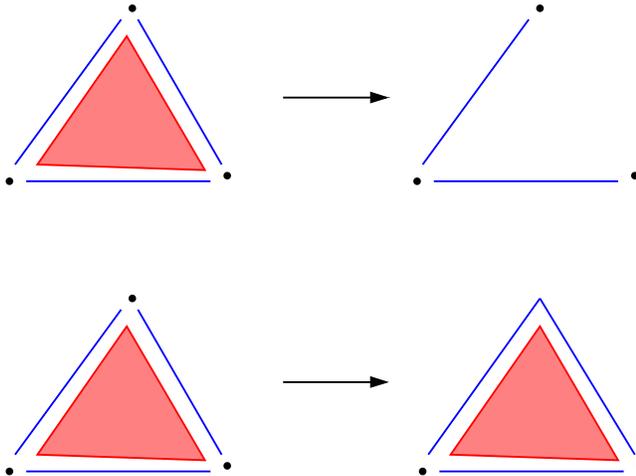


Fig. 2.3: Geometric interpretations of two chain equivalences used in the homology computation. In the upper one, a cell on the boundary of the domain is “pushed-in.” In the lower one, two cells are combined. The chain equivalences used in the cohomology computation have similar interpretations, but in the *dual mesh*.

Once the chain complex has been reduced, its homology or cohomology groups are computed using the Smith normal form integer matrix decomposition. The computation concentrates on the *short exact sequence* of abelian groups

$$0 \rightarrow B_k \xrightarrow{i_k} Z_k \xrightarrow{j_k} Z_k/B_k \rightarrow 0. \quad (2.1)$$

The group  $B_k$  is the codomain of the  $k + 1$ -cells to  $k$ -cells incidence matrix of the mesh, i.e. the group of  $k$ -boundaries. The group  $Z_k$  is the kernel of the  $k$ -cells to  $k - 1$ -cells incidence matrix of the mesh, i.e. the group of  $k$ -cycles. The *quotient group*  $Z_k/B_k$  is by definition the  $k$ :th homology group  $H_k$  of the chain complex: the group of equivalence classes of non-bounding  $k$ -cycles.

The Smith normal form is then used in three steps. First, to find out the kernels and the codomains of the incidence matrices. Second, to construct the matrix representation of the map  $i_k$ . Third, the matrix representation of  $i_k$  is decomposed to solve the structure of the  $k$ :th homology group  $H_k = Z_k/B_k \simeq Z_k/\text{im } i_k$ , where the last isomorphism results from the algebra of groups [26]. The procedure for cohomology computation is analogous.

**3. Working example: induction heating with magnetic field oriented formulation.** To introduce and motivate the homology and cohomology computations in the finite element method, we take an eddy current problem that models an induction heating machine as an example. We use magnetic field oriented formulation and consider both current- and voltage-driven variants.

We start the problem definition with the problem domain. Our primary interest in modeling sense are the eddy currents in the workpiece, while the model also needs to take the inductor coil configuration in the surrounding air into account. Due to a large non-conducting air region, a magnetic field oriented formulation that treats most of the air with a scalar potential only is likely more efficient than a magnetic flux density oriented one, where a vector potential is employed throughout the domain [13].

The topological aspects of the 3D modeling domain  $M \subset \mathbb{R}^3$  in the Euclidean space<sup>5</sup> are the main challenge of using a  $h$ -oriented formulation. The domain is divided into conducting and non-conducting subdomains  $M_c$  and  $M_a$ , respectively, depicted in Fig. 3.1, left.

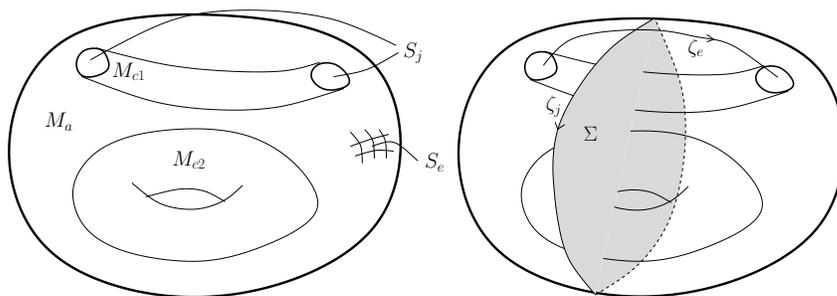


Fig. 3.1: Topology of the induction heating problem.

Topologically interesting aspects of the domain  $M$  are described as follows. The conducting domain  $M_c = M_{c1} \cup M_{c2}$  consists of two parts, the inductor  $M_{c1}$  and the workpiece  $M_{c2}$ . The workpiece has a tunnel through it, and the inductor pierces the non-conducting domain. Therefore, the non-conducting domain  $M_a$  has two tunnels through it. These features are captured by the homology and cohomology of the domains.

For clear and uniform notation, we use differential forms in this treatment. See the appendix B for a crash course on differential forms on the Euclidean space  $\mathbb{R}^3$ . Let  $\mathcal{F}^k(M)$  denote the space of differential  $k$ -forms on  $M$ . Moreover, let  $h, e \in \mathcal{F}^1(M)$  and  $j, b \in \mathcal{F}^2(M)$  denote the magnetic field, electric field, current density, and magnetic flux density, respectively. The tensor fields  $\mu$  and  $\rho$  are maps  $\mathcal{F}^1(M) \rightarrow \mathcal{F}^1(M)$  called

<sup>5</sup>More generally,  $M$  is a compact oriented 3-dimensional Riemannian manifold  $(M, g)$  with boundary that can be covered with a single chart. That is, the notation and the treatment in this paper remain valid under the change of the metric tensor  $g$  and under the change of charts of  $M$ .

permeability and resistivity, respectively. The operators  $d$ ,  $t$ , and  $\star$  are the exterior derivative, trace, and Hodge operators, respectively; see the appendix B for their definitions.

Our problem statement on  $M$  reads: find the fields  $h, e \in \mathcal{F}^1(M)$  and  $j, b \in \mathcal{F}^2(M)$  that satisfy the magnetoquasistatic approximation to the Maxwell's equations and material relations

$$de = -\partial_t b, \quad db = 0, \quad (3.1)$$

$$dh = j, \quad (3.2)$$

$$b = \star \mu h, \quad e = \rho \star j. \quad (3.3)$$

In addition, we require the following local constraints and boundary conditions to hold:

$$dh = 0 \text{ on } M_a, \quad tj = 0 \text{ on } (\partial M \cup \partial M_c)/S_j, \quad (3.4)$$

$$tb = 0 \text{ on } \partial M, \quad te = 0 \text{ on } S_j, \quad (3.5)$$

where  $S_j = \partial M \cap \partial M_c$  and  $S_e = \partial M \cap \partial M_a$  hold, that is  $\partial M = S_e \cup S_j$  holds. With such boundary conditions, the problem qualifies as a circuit element [35].

To couple the problem with a driving circuit, we need to pose a non-local constraint to take the source current or source voltage into account. Such constraints fix the cohomology class of the problem. For the source current  $I_s$  driven problem we require

$$\int_{\zeta_j} h = I_s, \quad [\zeta_j] \in H_1(S_e) \quad (3.6)$$

It follows from the *long exact homology sequence* [26] that  $\zeta_j$  is a boundary of any surface  $\Sigma$  in  $M$  that isolates the two terminals on  $S_j$ . The coset  $[\Sigma]$  of such surface belongs to the homology space  $H_2(M, S_e)$ . For example in Fig. 3.1, right,  $\partial\Sigma$  qualifies as such  $\zeta_j$ . Then,  $I_s = \int_{\zeta_j} h = \int_{\partial\Sigma} h = \int_{\Sigma} dh = \int_{\Sigma} j$  holds by Stokes' theorem. Alternatively, we can drive the problem by a source voltage  $V_s$  and require

$$\int_{\zeta_e} e = V_s, \quad [\zeta] \in H_1(S_e, \partial S_e) \quad (3.7)$$

That is,  $\zeta_e$  is any path along  $S_e$  between the two inductor terminals on  $S_j$ , for example the one in Fig. 3.1, right.

*Remark:* By the Lefschetz duality theorem the spaces  $H_1(S_e)$  and  $H_1(S_e, \partial S_e)$  are isomorphic [26, 18]. This induces a duality between current driven and voltage driven problems.

#### 4. Vector-scalar potential formulation of the eddy current problem.

In boundary value problems in 3-dimensional domains, the unknown field is often a vector field. The computational burden of such problem is relieved if the unknown vector field can be expressed as a gradient of a scalar potential. Such a potential is known to exist, but only *locally*, if the vector field is curl-free [3]. In the terminology of differential forms, one seeks for a potential 0-form of a *closed* 1-form. In the eddy current problem, the magnetic field 1-form  $h$  is closed in the non-conducting domain  $M_a$ . Thus,  $h$  can be locally expressed as an exterior derivative of a potential 0-form in  $M_a$ . However, in the conducting domain  $M_c$ , one still needs to solve for a 1-form.

Such formulation in eddy current problems is called  $T - \Omega$  formulation, which we will now employ.

The  $T - \Omega$  potential formulation in magnetodynamics draws its name from the mixed use of the electric current vector potential  $T$  and the magnetic scalar potential  $\Omega$  [31]. For a better insight, instead of writing the formulation in terms of  $T$  and  $\Omega$ , we first write it in terms of magnetic field  $h$ . This prevents the structure of the problem from being obscured by the excess of notation. Later, we describe a basis of the function space from which an approximate solution for  $h$  is sought. The basis will have terms resembling the potentials  $T$  and  $\Omega$ , and terms arising from the homology of the non-conducting subdomain  $M_a$ .

In our example problem, the function space from which we look for the solution  $h$  is

$$h \in \mathcal{H}(M) = \{ h \in \mathcal{F}^1(M) \mid \begin{aligned} dh = 0 \text{ on } M_a, \quad \text{td}h = 0 \text{ on } (\partial M \cup \partial M_c)/S_j, \\ \text{t}(\star\mu h) = 0 \text{ on } \partial M, \quad \text{t}(\rho \star dh) = 0 \text{ on } S_j \} \quad (4.1) \end{aligned}$$

together with (3.6) or (3.7) depending on the source quantity. The following weak formulation for the equations (3.1) to find  $h \in \mathcal{H}(M)$  is obtained by using the equations (3.2) and (3.3) and by taking the  $b$ -side constraints (3.5) in (4.1) into account:

$$\int_M \partial_t \star \mu h \wedge h' + \int_{M_c} \rho \star dh \wedge dh' + \int_{S_e} \text{t}(e \wedge h') = 0 \quad \forall h' \in \mathcal{H}(M), \quad (4.2)$$

$$- \int_M \star \mu h \wedge h' = 0 \quad \forall h' \in \mathcal{H}(M). \quad (4.3)$$

As  $dh = 0$  holds in  $M_a$ , we wished to express  $h$  exhaustively in terms of a magnetic scalar potential  $\phi$  in  $M_a$ , so that  $h = d\phi$  holds. This would be computationally efficient. However, we cannot, which we know from de Rham's first theorem [18]:

$$(dh = 0 \implies h = d\phi) \iff \int_z h = 0 \quad \forall z \in H_1(M_a). \quad (4.4)$$

In our case the integral condition fails to hold at two places. First, integrals of  $h$  over 1-cycles around the inductor should equal to the net current in the inductor. Second, integrals of  $h$  over 1-cycles that go through the tunnel in  $M_{c2}$  should equal the net current in the workpiece around the tunnel, see Fig. 5.1, left. Two cosets of such 1-cycles are a basis for the homology space  $H_1(M_a)$ .

With the aid of the homology and cohomology solver, one can automatically construct the missing part of an approximate subspace of  $\mathcal{H}(M)$  in  $M_a$ : the edge-based cohomology basis functions that are often called “thick-cuts” [24, 21]. The approximate subspace of  $\mathcal{H}(M)$  is still mainly spanned by the nodal finite element shape functions in the non-conducting domain  $M_a$ , but the two cohomology basis functions enables one to have non-zero circulations of the magnetic field  $h$  around the inductor and the workpiece. Typically, the computed cohomology basis functions have a small support, and thus do not excessively populate the system matrix.

Let us have a finite element mesh on the domain  $M$ . Let  $N(M)$  denote the set of mesh nodes on  $M$ , and let  $E(M_a)$  denote set of mesh edges on  $M_a$  and let  $E(M_c/M_a)$  denote the set of mesh edges on  $M_c$  that are not on  $M_a$ . For each node and edge we associate a Whitney 0-form  $n^i$  and Whitney 1-form  $e^i$ , respectively. These form a basis for Whitney spaces  $\mathcal{W}^0(M)$  and  $\mathcal{W}^1(M)$  [4].

Then, a field that will suffice to represent the approximate solution is of the form

$$h = \sum_{i \in N(M)} \phi_i \mathrm{d}n^i + \sum_{i \in E(M_c/M_a)} \mathbf{t}_i \mathbf{e}^i + I_1 \sum_{i \in E(M_a)} \mathbf{z}_i^1 \mathbf{e}^i + I_2 \sum_{i \in E(M_a)} \mathbf{z}_i^2 \mathbf{e}^i \in \mathcal{W}^1(M) \cap \mathcal{H}(M), \quad (4.5)$$

where the coefficients  $I_1$  and  $I_2$  of the cohomology basis functions correspond to the values of the integrals of  $h$  over two linearly independent 1-cycle cosets in  $H_1(M_a)$ , and  $\mathcal{W}^1(M) \cap \mathcal{H}(M)$  is the approximate subspace of  $\mathcal{H}(M)$ . The sparse coefficient vectors  $\mathbf{z}^1$  and  $\mathbf{z}^2$  represent the 1-cochains that represent a computed basis  $\{z^1, z^2\}$  of the space  $H^1(M_a)$ . Note that the interpretation of the unknown coefficients  $I_1$  and  $I_2$  depends on the basis of  $H^1(M_a)$ .

*Remark:* The  $h$ -oriented constraints in (4.1) are satisfied by the choice of basis for the function space  $\mathcal{H}(M)$ . The  $b$ -oriented constraints are satisfied “weakly” by the absence of boundary integrals over  $S_j$  and  $\partial M$  in equations (4.2) and (4.3), respectively [5].

**5. Choosing a useful basis for the cohomology space  $H^1(M_a)$ .** In the example problem, we would like that the coefficients  $I_1$  and  $I_2$  have an evident interpretation for the engineer. For example, in order make it easy to fix the source current or the source voltage, the coefficient  $I_1$  should correspond to the net current through the inductor  $M_{c1}$ . Also, if the coefficient  $I_2$  would correspond to the net current through the workpiece, one could read off useful post-processing information right from the solution, without any actual post-processing! For example, if our apparatus was a transformer, the coefficient  $I_2$  would indicate the net short-circuit current in the secondary winding. In order to make such connotations, the engineer needs to be involved in the choice of the cohomology basis.

However, in practice it is actually easier to choose the homology basis of the representative 1-chains and then adjust the cohomology basis accordingly. Let the matrix  $H_1 = [\mathbf{z}_1 \quad \mathbf{z}_2 \quad \dots \quad \mathbf{z}_{\beta_1}]$ ,  $\beta_1 = \dim H_1(M_a)$ , represent a computed basis for  $H_1(M_a)$ , where  $\mathbf{z}_i$  are coefficient vectors of the representatives of the basis 1-chains  $z_i$ . For any unimodular matrix  $U$ , the matrix  $H_1 U$  also represents a basis for the space  $H_1(M_a)$ . The intuitive basis representation for  $H_1(M_a)$  would be the one where the representatives  $z_i$  only loop around a single tunnel in  $M_a$  once, and loop around no other tunnels. Then, one can directly relate the circulation of the magnetic field  $h$  over the representative  $z_i$  to the net current  $I_i$  in the tunnel. That is, the basis would correspond to the source current  $I_s$  and net current  $I_i$  in the workpiece. Such a basis is represented in the Fig. 5.1, left. Unfortunately, the homology solver cannot distinguish such loops without help from the user: Effectively, the user needs to feed in a unimodular matrix  $U$  to produce such a desired basis from the computed one. For example, if the homology solver produced the basis representation on the right in Fig. 5.1, the basis representation on the left can be attained by the  $(2 \times 2)$ -matrix  $U = \begin{bmatrix} 1 & -1; & 0 & -1 \end{bmatrix}$ .

Since  $H_2(M) = \bar{H}_1(M) = 0$  holds in our example problem, an automated method to obtain such clear basis for the space  $H_1(M_a)$  does exist. We can infer from the *long exact homology sequence* [26]

$$0 = H_2(M) \xrightarrow{j_*} H_2(M, M_a) \xrightarrow{\partial_*} H_1(M_a) \xrightarrow{i_*} H_1(M) = 0 \quad (5.1)$$

that the spaces  $H_2(M, M_a)$  and  $H_1(M_a)$  are isomorphic<sup>6</sup>. Therefore, we obtain a basis representation for the space  $H_1(M_a)$  by applying the boundary operator

<sup>6</sup>Furthermore,  $H_2(M, M_a) \simeq H_2(M_c, \partial M_c \cap \partial M_a)$  holds and moreover

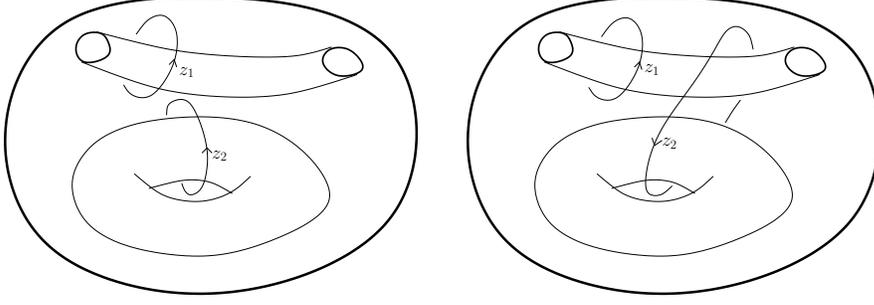


Fig. 5.1: Two basis representations for the homology space  $H_1(M_a)$ .

to the computed representative basis 2-chains of the space  $H_2(M, M_a)$ . That is,  $H_1 = D_2 H_2 = [D_2 \mathbf{z}^1 \ D_2 \mathbf{z}^2 \ \dots \ D_2 \mathbf{z}^{\beta_2}]$  holds, where  $H_2$  is the basis representation matrix for the space  $H_2(M, M_a)$  produced by the homology solver, and  $D_2$  is the faces-to-edges incidence matrix of the mesh. Such a basis is guaranteed to be equivalent to the one in Fig. 5.1, left.

Once we have obtained a desired basis representation  $\tilde{H}_1 := H_1 U$  for the space  $H_1(M_a)$ , it is possible to have a compatible cohomology space basis in the following sense. Let the matrix  $H^1 = [\mathbf{z}^1 \ \mathbf{z}^2 \ \dots \ \mathbf{z}^{\beta_1}]$  be a basis representation for the space  $H^1(M_a)$  produced by the cohomology solver. Then one can automatically obtain a basis representation  $\tilde{H}^1 = H^1 V$  for the space  $H^1(M_a)$  that satisfies

$$I = \tilde{H}_1^T \tilde{H}^1 = \tilde{H}_1^T H^1 V \iff V = (\tilde{H}_1^T H^1)^{-1} = ((H_1 U)^T H^1)^{-1} \quad (5.2)$$

once the  $\beta_1 \times \beta_1$ -matrix  $\tilde{H}_1^T H^1$  is known. Then,  $\tilde{H}^1$  is the *cobasis* of the basis  $\tilde{H}_1$ , since

$$\mathbf{z}_i^T \mathbf{z}^j = \int_{[z_i]} [z^j] = \delta_{ij}, \quad 1 \leq i, j \leq \beta_1 \quad (5.3)$$

holds for the bases. In *Gmsh*, one can request a cohomology cobasis for a computed homology basis at the post-processing stage of the (co)homology computation.

**6. Realization of the non-local constraints and the linear system.** So far, we have left the boundary term in equation (4.2) unspecified. It turns out that it can be related to the non-local constraints of the example problem. Consequently, we can write down the linear system that is either constrained by the source voltage or by the source current. The other will be a part of the solution to the system.

Let  $\{\mathbf{z}_1, \mathbf{z}_2\}$  represent the basis of the space  $H_1(M_a)$  in Fig. 5.1, left, and let  $\{\mathbf{z}^1, \mathbf{z}^2\}$  represent the cobasis of the space  $H^1(M_a)$ . Then, the cohomology basis functions  $\mathbf{E}^1 = \sum_{i \in E(M_a)} \mathbf{z}_i^1 \mathbf{e}^i$  and  $\mathbf{E}^2 = \sum_{i \in E(M_a)} \mathbf{z}_i^2 \mathbf{e}^i$  of the function space  $\mathcal{W}^1(M) \cap \mathcal{H}(M)$  correspond to the source current  $I_s$  and the net current  $I_i$  in the workpiece, respectively.

Let us take a closer look to the boundary term in (4.2). On the boundary  $\partial M$ , the field  $te$  can be expressed by a scalar potential. As  $dte = -\partial_t tb = 0$  holds by

$\overline{H_2(M_c, \partial M_c \cap \partial M_a)} \simeq H_2(M_{c1}, \partial M_{c1} \cap \partial M_a) \oplus H_2(M_{c2}, \partial M_{c2} \cap \partial M_a)$  holds. The first isomorphism holds by the *excision theorem* [26], and the second since  $M_{c1}$  and  $M_{c2}$  are not connected.

the boundary condition (3.5) and  $\dim H_1(\partial M) = 0$  holds, it follows from de Rham's first theorem (4.4) that  $d\varphi = te$  holds for a scalar field  $\varphi \in \mathcal{F}^0(\partial M)$ . Then for  $e, h' \in \mathcal{F}^1(M)$ ,

$$\int_{S_e} t(e \wedge h') = \int_{S_e} d\varphi \wedge th' \quad (6.1)$$

$$= \int_{S_e} \varphi \wedge tdh' - \int_{S_e} d(\varphi \wedge th') \quad (6.2)$$

holds. The first term vanishes, since for any  $h' \in \mathcal{H}(M)$  it holds that  $tdh' = tj = 0$  on  $S_e$ . The voltage  $\varphi$  is a constant on each connected component of  $S_j$ , with a voltage difference  $V_1 = \varphi_1 - \varphi_2$ .

Let the boundary  $\partial S_j = \partial S_e$  be represented by a 1-chain  $c_1 - c_2$ , where  $c_1$  and  $c_2$  are homologous in the subdomain  $M_a$ , and  $\int_{c_1} h = -\int_{c_2} h' = I_1$  hold for any  $h' \in \mathcal{H}(M)$ . That is,  $c_1$  and  $c_2$  belong to the same homology class of  $H_1(M_a)$ , represented by  $z_1$ . Because  $\mathbf{z}_1^T \mathbf{z}^1 = 1$  and  $\mathbf{z}_1^T \mathbf{z}^2 = 0$  hold, we obtain as the boundary term of the equation (4.2)

$$\int_{S_e} t(e \wedge h') = - \int_{\partial S_e} \varphi \wedge th' = - \left( \varphi_1 \int_{c_1} h' - \varphi_2 \int_{c_2} h' \right) = \begin{cases} -V_1 & \text{when } h' = \mathbf{E}^1 \\ 0 & \text{when } h' \neq \mathbf{E}^1 \end{cases}. \quad (6.3)$$

Similar developments without the connection to the (co)homology were made in [14, 13].

Now, by plugging in the approximation in equation (4.5) to the equations (4.2) and (4.3) and treating the problem as time harmonic, we obtain the following block linear system:

$$\begin{bmatrix} \mathbf{A}_{\text{dn,dn}} & \mathbf{A}_{\text{e,dn}} & \mathbf{A}_{\mathbf{E}^1,\text{dn}} & \mathbf{A}_{\mathbf{E}^2,\text{dn}} \\ \mathbf{A}_{\text{dn,e}} & \mathbf{A}_{\text{e,e}} & \mathbf{A}_{\mathbf{E}^1,\text{e}} & \mathbf{A}_{\mathbf{E}^2,\text{e}} \\ \mathbf{A}_{\text{dn},\mathbf{E}^1} & \mathbf{A}_{\text{e},\mathbf{E}^1} & \mathbf{A}_{\mathbf{E}^1,\mathbf{E}^1} & \mathbf{A}_{\mathbf{E}^2,\mathbf{E}^1} \\ \mathbf{A}_{\text{dn},\mathbf{E}^2} & \mathbf{A}_{\text{e},\mathbf{E}^2} & \mathbf{A}_{\mathbf{E}^1,\mathbf{E}^2} & \mathbf{A}_{\mathbf{E}^2,\mathbf{E}^2} \end{bmatrix} \begin{bmatrix} \phi \\ \mathbf{t} \\ I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ V_1 \\ 0 \end{bmatrix}, \quad (6.4)$$

where the blocks of the complex matrix correspond to the different kinds of basis functions of the space  $\mathcal{W}^1(M) \cap \mathcal{H}(M)$ .

Since  $z_1$  is homologous to  $\zeta_j$ , which was the boundary of a surface  $\Sigma_j$  that isolates the terminals on  $S_j$ , the current-driven problem is obtained by fixing  $I_1$  equal to the known source current  $I_s$ . Then, the equation involving  $V_1$  plays no role in the solution of the problem. However, once the problem has been solved, one can obtain the value of  $V_1$  by substitution. Analogously, since the 0-chain  $\partial\zeta_e$  lies on the distinct terminals on  $S_j$ , the voltage-driven problem is obtained by fixing  $V_1$  equal to the known source voltage  $V_s$ . Then, the net current  $I_1$  will be a part of the solution. Note that in both current- and voltage-driven problems, the net current  $I_2$  through the workpiece is obtained as a part of the solution.

*Remark:* The linear system (6.4) is actually an affine relation between the current  $I_1$  and the voltage  $V_1$ .

*Remark:* If the workpiece is interpreted as the secondary winding of a transformer, one could also drive the problem with either net current  $I_2$  through the winding or voltage  $V_2$  across the winding in addition to either  $I_1$  or  $V_1$ . This is done by insisting that there are two electrodes very close to each other in the workpiece  $M_{c_2}$ , that

creates a void in the domain  $M$ . The electrodes would belong to  $S_j$ , while the sides of the void would belong to  $S_e$ , see Fig. 6.1. Then again since  $\mathbf{z}_i^T \mathbf{z}^j = \delta_{ij}$  holds, the boundary term of the equation (4.2) would be

$$\int_{S_e} \mathbf{t}(e \wedge h') = - \int_{\partial S_e} \varphi \wedge \mathbf{t}h' = \begin{cases} -V_1 & \text{when } h' = \mathbf{E}^1 \\ -V_2 & \text{when } h' = \mathbf{E}^2 \\ 0 & \text{when } h' \neq \mathbf{E}^1 \text{ and } h' \neq \mathbf{E}^2 \end{cases} \quad (6.5)$$

However, then the current leads from the electrodes to the circuit are excluded from the model. Thus, the model assumes that negligible electromotive force is induced to the current leads and the contribution to the magnetic field by the current in the leads is negligible [35].

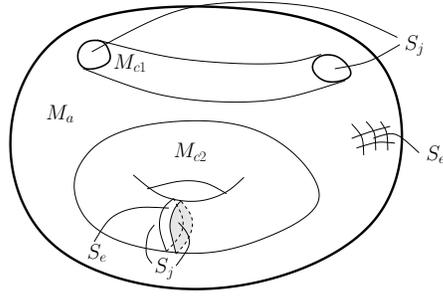


Fig. 6.1: A void in the workpiece  $M_{c2}$  that is excluded from the actual model. The electrodes are part of  $S_j$  while the insulating boundary of the void is a part of  $S_e$ .

**7. Numerical example.** We consider an example problem whose geometry is depicted in Fig. 7.1. For such a geometry, to imagine a suitable basis representation for the space  $H^1(M_a)$  with mind's eye is a difficult task. Thus, having a proven algorithm for the task is a considerable advantage. Note that from a topological point of view, the problem is still equivalent to the Fig. 3.1.

In our numerical example, the workpiece is made of aluminum, and the conductor is copper. AC current of 1 A at 50 Hz frequency is led to the conductor. The thickness of the workpiece is a bit larger than the skin depth at that frequency.

This example is available at a website [27] together with *Gmsh* and *GetDP* script files. They demonstrate how to perform the required homology and cohomology computations in *Gmsh* and how to input the approximation in the equation (4.5) of the function space  $\mathcal{H}(M)$  and the equations (4.2) and (4.3) to the finite element solver *GetDP*.

In Figures 7.2 and 7.3 are computed representations for the basis  $\{z_1, z_2\}$  of the space  $H_1(M_a)$  and for the cobasis  $\{z^1, z^2\}$  of the space  $H^1(M_a)$ . The bases are such that  $z_1$  and  $z^1$  correspond to the net current in the inductor, and  $z_2$  and  $z^2$  correspond to the net current through the workpiece. Therefore, the coefficients  $I_1$  and  $I_2$  in the equation (4.5) will be equal to them. Note how the coefficients of  $z^1$  equal 6 inside the workpiece to ensure that the incidence matrix of the bases is an identity matrix.

In Table 7.1 we compare the homology and the cohomology computation time for various meshes of the geometry. The computation time overhead added by the homology and cohomology computations are of the order of magnitude of the mesh

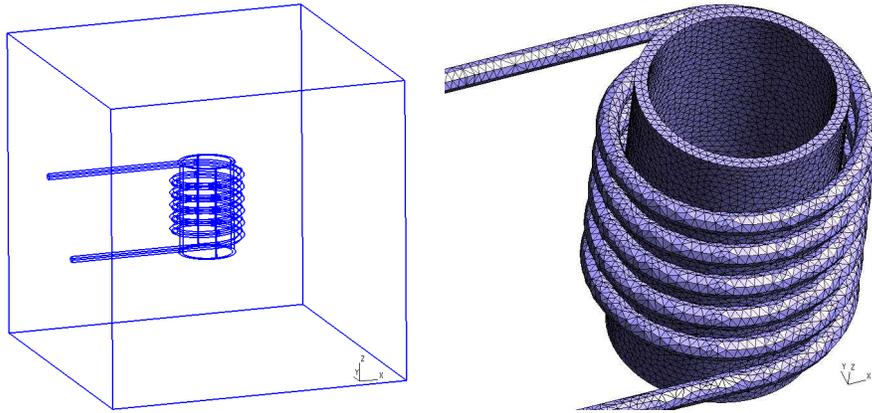


Fig. 7.1: Induction heating geometry and surface mesh of  $M_c$ .

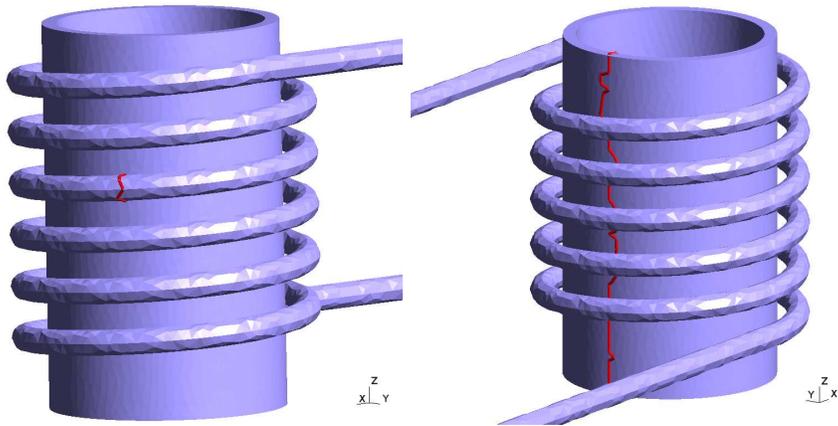


Fig. 7.2: A representation of the basis of the homology space  $H_1(M_a)$ .

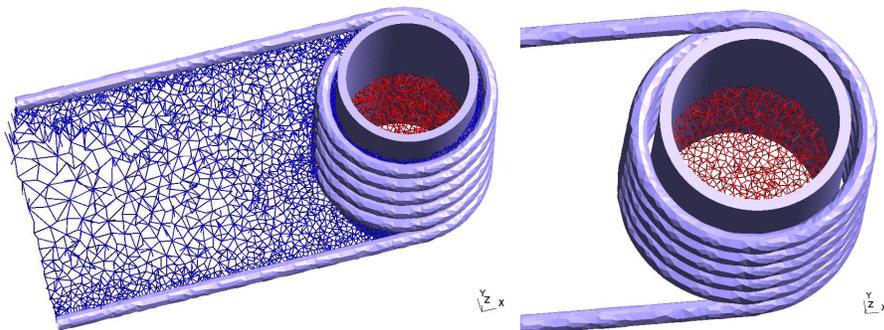


Fig. 7.3: A representation of the basis of the cohomology space  $H^1(M_a)$ .

Table 7.1: Comparing computation times with different meshes. Number of tetrahedra in the mesh; Time to generate the mesh; Number of tetrahedra in  $M_a$  and  $M_c$  which are the cohomology and homology computation domains, respectively; Time to compute the cohomology  $H^1(M_a)$  and the homology  $H_2(M_c, \partial M_c \cap \partial M_a)$ ; Degrees of freedom in the  $T - \Omega$ -formulation and in the  $A - V$ -formulation for the same problem; Solution time for the problems using a direct solver; Impedance  $Z = U_1/I_1$  of the  $T - \Omega$  solution.

	#Tet. $M$ Mesh time	#Tet. $M_a$ $H^1$ time	#Tet. $M_c$ $H_2$ time	$T - \Omega$ #DoF's time	$A - V$ #DoF's time	Re( $Z$ ) Im( $Z$ )
Mesh 1	106 141 2.2 s	88 849 5.9 s	17 292 2.0 s	61 568 301 s	257 148 3 823 s	0.532 m $\Omega$ 1.01 m $\Omega$
Mesh 2	148 962 3.4 s	123 977 8.2 s	24 985 3.2 s	88 406 666 s	366 252 –	0.485 m $\Omega$ 0.962 m $\Omega$
Mesh 3	268 238 8.1 s	232 227 17.9 s	36 011 9.2 s	179 514 2 126 s	649 332 –	0.494 m $\Omega$ 0.932 m $\Omega$
Mesh 4	392 218 13.8 s	299 328 26.7 s	84 534 15.3	279 490 4 920 s	949 848 –	0.494 m $\Omega$ 0.918 m $\Omega$

generation. Note that the cohomology computation domain is not the full mesh, but just the air subdomain  $M_a$ , and the homology computation domain is just the conducting subdomain  $M_c$ .

We also compare the number of degrees of freedom between the  $T - \Omega$ -formulation and the  $A - V$ -formulation [13] of the entire problem. In the  $A - V$ -formulation of the eddy current problem, one seeks for a vector field in the whole domain  $M$ : There's one DoF for each edge in the mesh of  $M$ , and one DoF for each node in the mesh of the conducting subdomain  $M_c$ . When the  $A - V$ -formulation is used, (co)homology computation typically isn't needed.<sup>7</sup>

As a conclusion of the results in Table 7.1, the benefits of being able to use  $T - \Omega$ -formulation outweigh the additional cost of the homology and cohomology computation.

In Fig. 7.4 we demonstrate the effect of the cohomology basis functions  $E^1$  and  $E^2$  to the scalar part  $\sum_{i \in N} \phi_i n_i$  of the function space. The values of the scalar part have a jump equal to  $I_1 \mathbf{z}_i^1$  and  $I_2 \mathbf{z}_i^2$  across the edges  $i \in E(M_a)$ . In Fig. 7.5 is the computed current density  $j = dh$  of the problem, where  $h \in \mathcal{W}^1(M) \cap \mathcal{H}(M)$ .

**8. Conclusions.** A tool for the homology and the cohomology computation for finite element meshes and exploitation of its results in computational electromagnetics was presented. The solver is an integrated part of the finite element mesh generator *Gmsh*. As such, the homology and cohomology computation step can be seamlessly incorporated to the finite element modeling workflow. The performance of the solver is found to be good when the input is a typical finite element mesh. The output of

<sup>7</sup>However, if we interpreted the workpiece as the secondary winding of a transformer, one could use cohomology basis functions arising from the space  $H^1(M_c, S_j)$  to drive it with voltage or net current.

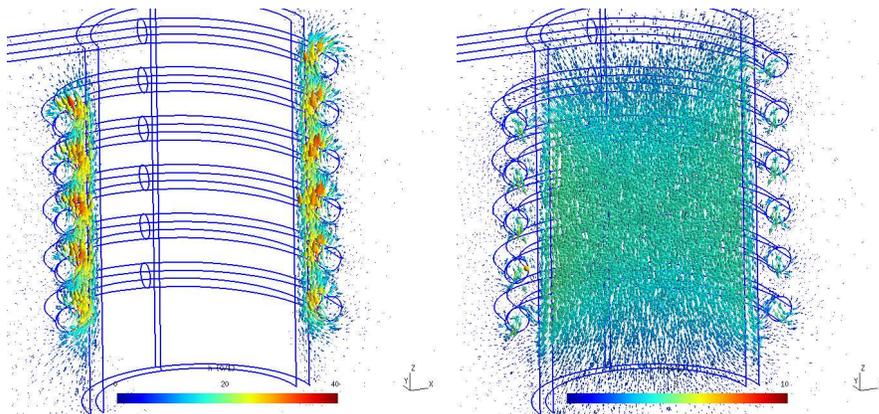


Fig. 7.4: Real and imaginary parts of the computed magnetic field  $h \in \mathcal{H}(M)$ .

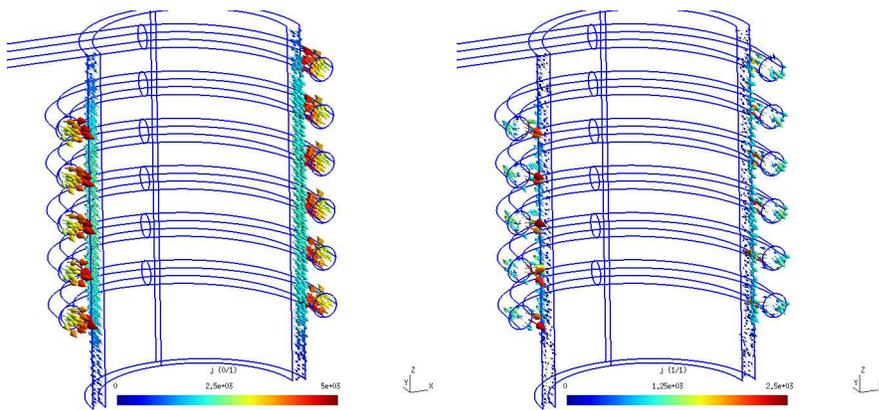


Fig. 7.5: Real and imaginary parts of the computed current density  $j = dh$ , where  $h \in \mathcal{H}(M)$ .

the solver is associated with the input mesh, and thus can be easily exploited within a finite element solver.

The cohomology solver can be used to produce cohomology basis functions for the finite element method. Such basis functions make the use of vector-scalar formulations of boundary value problems straightforward. The degrees of freedom associated with the cohomology basis functions provide useful post-processing information, as they can be related to the lumped parameter model of the problem.

In this paper, the exploitation of homology and cohomology computation was demonstrated by a  $T - \Omega$ -formulation of an eddy current problem, which provides a considerable performance increase when compared to the  $A - V$ -formulation of the problem. Without the cohomology computation, the usage of the  $T - \Omega$ -formulation is more cumbersome in many cases. Homology and cohomology computation also clarified the circuit coupling of the problem.

**Acknowledgments.** This work was supported in part by the Belgian Science Policy (IAP grant P7/02) and the Walloon region (WIST3 grant “DOMHEX”).

**Appendix A. Homology and cohomology spaces.** Briefly, the homology and the cohomology of a finite element mesh is described as follows. Based on a mesh on a domain  $M$ , *vector spaces*  $C_k(M)$  of  $k$ -chains  $c_k$  are defined to be formal sums of  $k$ -dimensional mesh cells  $\sigma^k$ . For example 1-chain  $c_1 = \sum_{\sigma_i^1 \in M} \mathbf{c}_i \sigma_i^1$ , where  $\mathbf{c}_i \in \mathbb{R}$ , is a formal sum of the edges in the mesh of  $M$ . Thus, on a given mesh a  $k$ -chain  $c_k$  can be represented by its coefficient vector  $\mathbf{c} = [\mathbf{c}_1 \quad \mathbf{c}_2 \dots \mathbf{c}_{N_k}]^T$ . Boundary operator  $\partial_k$  maps  $k$ -chains to  $(k-1)$ -chains. They can be represented as the incidence matrices  $D_k$  of the mesh. Then for example  $\partial_1 c_1 = \sum_{\sigma_i^0 \in M} (D_1 \mathbf{c})_i \sigma_i^0$  is a sum of nodes  $\sigma_i^0$  that constitute the endpoints of the 1-chain  $c_1$ .

A  $k$ -chain whose boundary is the zero element of  $C_{k-1}(M)$  is called a  $k$ -cycle and often denoted by  $z_k$ . A  $k$ -chain that is a boundary of a  $k+1$ -chain is called a  $k$ -boundary. That is,  $k$ -cycles constitute the space  $\ker(\partial_k)$  and  $k$ -boundaries constitute the space  $\text{im}(\partial_{k+1})$ . Two  $k$ -cycles are called *homologous* if their difference is a  $k$ -boundary. Such relation on  $k$ -cycles is an equivalence relation and thus can be used to construct a quotient space whose elements are cosets: equivalence classes of  $k$ -cycles.

Cochains are linear maps from chains to real (or complex) numbers. That is, they assign a scalar quantity to the given chain. Integration of a field over a domain is an example of a cochain when the field is fixed, but the integration domain varies.

A  $k$ -cochain  $c^k$  can be represented as a complex or real coefficient vector  $\mathbf{c}^k$  such that for a given  $k$ -chain  $c_k$ ,  $c^k(c_k) = (\mathbf{c}^k)^T \mathbf{c}_k$  holds. The coboundary operator  $\delta_k$  maps  $k$ -cochains to  $k+1$ -cochains and it is defined by  $\delta_k c^k(c_{k+1}) = c^k(\partial_{k+1} c_{k+1})$ . Therefore, their matrix representations are  $D_{k+1}^T$ . The elements of the spaces  $\ker(\delta_k)$  and  $\text{im}(\delta_{k-1})$  are called  $k$ -cocycles and  $k$ -coboundaries, respectively.

Homology and cohomology spaces  $H_k(M)$  and  $H^k(M)$  are the *quotient spaces*

$$H_k(M) = \ker(\partial_k) / \text{im}(\partial_{k+1}) = \{c_k + \text{im}(\partial_{k+1}) \mid c_k \in \ker(\partial_k)\}, \quad (\text{A.1})$$

$$H^k(M) = \ker(\delta_k) / \text{im}(\delta_{k-1}) = \{c^k + \text{im}(\delta_{k-1}) \mid c^k \in \ker(\delta_k)\}. \quad (\text{A.2})$$

For example, the elements of  $H_k(M)$  are represented by  $k$ -chains that are non-bounding  $k$ -cycles, such as loops around tunnels, or surfaces that enclose voids. One can add to such  $k$ -chain any bounding  $k$ -chain, and it still represents the same element of  $H_k(M)$ .

Cosets  $C_k(M, S) = \{c + C_k(S) \mid c \in C_k(M)\}$  are called  *$k$ -chains relative to a subdomain  $S \subset M$* . Usually, a choice of representative is made which zeroes the part of the chain that lies in  $S$ .<sup>8</sup> With such choice, the representations of the relative boundary operator matrices are obtained by removing rows and columns of the boundary operator matrices  $D_k$  that correspond to mesh cells lying on the subdomain  $S$ . From the definition of relative  $k$ -chains, spaces of relative  $k$ -cochains, relative homology spaces, and relative cohomology spaces can be induced. For example, a relative non-bounding 1-cycle would draw from  $S$  to  $S$  so that it wouldn't enclose a surface, part of whose boundary might lie in  $S$ .

**Appendix B. Differential forms on  $M \subset \mathbb{R}^3$ .** In the Euclidean space  $\mathbb{R}^3$ , the algebra and the analysis of the differential forms is very similar to the classical

<sup>8</sup>Such choice disregards *torsion* in the relative chain groups  $C_k(M, S)$ . In the finite element context, this is not a loss since boundary conditions are not typically assigned to a *multiple* of a boundary.

vector analysis. The main difference is the different point of view: Differential forms are functionals on vectors rather than vectors. However, once these functionals are given a component representation, the operations on the component vectors resemble the operations on the vector fields in the vector analysis. Due to this similarity, it may seem that the differential forms are just glorified vector fields. However, even in the Euclidean 3-space, differential forms have some virtues: While the 0-, 1-, 2-, and 3-forms have clearly different geometric nature, the operators acting on them are uniform and have a uniform notation. Therefore, differential forms make a good companion to the concepts of the homology and the cohomology.

A differential 0-form  $\omega^0 \in \mathcal{F}^0(M)$  on a subset  $M$  of the Euclidean space  $\mathbb{R}^3$  is a *differentiable* map  $\omega^0 : M \rightarrow \mathbb{F}$  to a scalar field  $\mathbb{F}$  of real or complex numbers.

A tangent vector  $v_p$  of  $M \subset \mathbb{R}^3$  is a vector  $\mathbf{v} \in \mathbb{R}^3 = T_pM$  that is “bound” to a point  $p \in M$ . The vector space  $T_pM$  is actually just  $\mathbb{R}^3$ , but denoted in such a way to emphasize that it is a vector space that is bound to a point  $p$  of  $M$ . Such tangent vector is often denoted  $v_p = \sum_{i=1}^3 \mathbf{v}_i \partial/\partial x^i$ , where  $\{\partial/\partial x^1, \partial/\partial x^2, \partial/\partial x^3\}$  is the standard basis of  $\mathbb{R}^3$  with Cartesian coordinates  $x^1, x^2$ , and  $x^3$ . Informally, a tangent vector can be considered as a “line element” at a point  $p$  of a curve lying in  $\mathbb{R}^3$ . A pair of tangent vectors  $u_p, v_p$  can be considered to form a “surface element” that has the coefficients of a vector normal to the surface they span:  $\mathbf{n} = \mathbf{u} \times \mathbf{v}$ . Similarly, three tangent vectors  $u_p, v_p, w_p$  form a “volume element” that is represented by a scalar volume  $V = \mathbf{u} \times \mathbf{v} \cdot \mathbf{w}$  of their span.

A differential 1-form  $\omega^1 \in \mathcal{F}^1(M)$  is a differentiable linear map  $\omega^1 : T_pM \rightarrow \mathbb{F}$ . That is, it takes a tangent vector as an argument and produces a scalar. A differential 1-form can be defined by providing three scalar functions  $\omega_i^1 : M \rightarrow \mathbb{F}$  that are the “components” the 1-form:  $\omega^1 = \sum_{i=1}^3 \omega_i^1(p) dx^i$  at  $p$ , where informally, the notation  $dx^i$  denotes that 1-forms are integrands of a line integral. The evaluation is then given by  $\omega^1(v_p) = \sum_{i=1}^3 \omega_i^1(p) \mathbf{v}_i = \omega^1(p) \cdot \mathbf{v} \in \mathbb{F}$ . Here and in the following paragraphs, we use the symbols  $\cdot$  and  $\times$  merely to express the computation rules of the components of the differential forms. That is, they do *not* denote any binary operators that have representation independent meaning.

A differential 2- and 3-forms are similar maps but they take as arguments two and three tangent vectors, respectively. Or informally, the arguments are surface and volume elements, respectively. A 2-form has three components:  $\omega^2 = \omega_1^2(p) dx^2 dx^3 + \omega_2^2(p) dx^3 dx^1 + \omega_3^2(p) dx^1 dx^2$ . It is evaluated by feeding in a surface element:  $\omega^2(u_p, v_p) = \sum_{i=1}^3 \omega_i^2(p) \mathbf{n}_i = \omega^2(p) \cdot \mathbf{n}$ . A 3-form has just one component  $\omega^3 = \omega^3(p) dx^1 dx^2 dx^3$ . It is evaluated for a volume element:  $\omega^3(u_p, v_p, w_p) = \omega_1^3(p) V$ .

The wedge product  $\wedge$  of a  $k$ -form and an  $l$ -form produces a  $k + l$ -form when  $k + l \leq 3$ , otherwise it produces zero. The components of  $\omega^0 \wedge \eta^k$  are  $\omega^0(p) \boldsymbol{\eta}(p)$  for any  $k$ , for 1-forms the components of  $\omega^1 \wedge \eta^1$  are  $\omega^1(p) \times \boldsymbol{\eta}^1(p)$ , for a 1-form and a 2-form the component of  $\omega^1 \wedge \eta^2$  is  $\omega^1(p) \cdot \boldsymbol{\eta}^2(p)$ .

The exterior derivative  $d$  maps a  $k$ -form to a  $k + 1$ -form. The components of  $d\omega^k$  for  $k = 0, 1, 2$  are  $\nabla \omega^0(p)$ ,  $\nabla \times \omega^1(p)$ , and  $\nabla \cdot \omega^2(p)$ , respectively.

The Hodge operator  $\star$  maps a 1-form  $\omega^1$  to a 2-form  $\eta^2 = \star \omega^1$ . The components of  $\eta^2$  are  $\boldsymbol{\eta}^2(p) = A(p) \omega^1(p)$ , where  $A(p)$  is the identity  $3 \times 3$ -matrix in the *Euclidean 3-space*. However, if the space was not Euclidean but a more general Riemannian manifold  $(M, g)$ , the expression for the positive definite matrix  $A(p)$  would be obtained from the *metric tensor*  $g$ .

The trace operator  $t$  “restricts” a differential form to a subdomain of the Euclidean space. Given a set of one or two tangent vectors at a point  $p \in M$ , the trace operator

picks the tangential part of a 1-form with respect to the line element or the surface element spanned by the tangent vectors. That is, the components of  $t\omega^1$  are  $(\omega^1(p) \cdot \hat{\mathbf{u}})\hat{\mathbf{u}}$  or  $\omega^1(p) - (\omega^1(p) \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}$ , respectively, where the vectors  $\hat{\mathbf{u}}$  and  $\hat{\mathbf{n}}$  are normalized with respect to the Euclidean norm. For 2-forms, the trace operator picks the normal part with respect to the surface element spanned by two tangent vectors:  $t\omega^2$  has the components  $(\omega^2(p) \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}$ . For a 0-form,  $t\omega^0$  is the ordinary restriction of a scalar function.

Finally, the integration of a differential  $k$ -form  $\omega^k$  is defined over a  $k$ -chain  $c_k = \sum_i \mathbf{c}_i \sigma_i^k$ . It is given by  $\int_{c_k} \omega^k = \sum_i \mathbf{c}_i \int_{\sigma_i^k} t\omega^k$ , where each  $k$ -dimensional cell  $\sigma_i^k$  is a  $k$ -dimensional bounded set in the Euclidean space  $\mathbb{R}^3$ . The Stokes' theorem is stated as  $\int_{c_k} d\omega^k = \int_{\partial c_k} \omega^k$ . This reveals that the exterior derivative  $d$  can be considered as the coboundary operator  $\delta$ . Indeed, the cohomology where the cochains are differential forms is called *de Rham cohomology*. Together with the Hodge operator and the wedge product, integration provides an inner product space structure for the function spaces  $\mathcal{F}^k(M)$  of differential  $k$ -forms. The inner product is defined by  $\langle \omega^k, \eta^k \rangle = \int_M \star \omega^k \wedge \eta^k$ . Thus, one can construct various Hilbert spaces of differential forms needed for a rigorous treatment of differential forms in the finite element method [2].

A tensor field  $C$  that maps a 1-form  $\omega^1$  to a 1-form  $\eta^1$  can be represented as a  $(3 \times 3)$ -matrix  $C(p)$ . The components of  $\eta^1$  are  $\boldsymbol{\eta}^1(p) = C(p)\boldsymbol{\omega}^1(p)$ . In the possible non-linear case, the elements of the matrix  $C(p)$  depend on the components of  $\boldsymbol{\omega}^1(p)$ . Then, the matrix  $C(p, \boldsymbol{\omega}^1(p))$  represents a linear approximation to the non-linear map  $C : \mathcal{F}^1 \rightarrow \mathcal{F}^1$  near  $\boldsymbol{\omega}^1(p)$  (in the sense of the inner product of the space  $\mathcal{F}^1$ ). To use such a (possibly non-linear) map in the finite element method, the map  $C$  is required to be strictly monotonous and symmetric with respect to the inner product of the differential 1-forms [18].

## REFERENCES

- [1] D. N. ARNOLD, R. S. FALK, AND R. WINTHER, *Finite element exterior calculus, homological techniques, and applications*, Acta Numer., 15 (2006), pp. 1–155. doi:10.1017/S0962492906210018.
- [2] ———, *Finite element exterior calculus: from Hodge theory to numerical stability*, Bull. Amer. Math. Soc. (N.S.), 47 (2010), pp. 281–354. doi:10.1090/S0273-0979-10-01278-4.
- [3] A. BOSSAVIT, *Magnetostatic problems in multiply connected regions: some properties of the curl operator*, Physical Science, Measurement and Instrumentation, Management and Education - Reviews, IEE Proceedings A, 135 (1988), pp. 179–187.
- [4] A. BOSSAVIT, *Computational Electromagnetism*, Academic Press, 1998.
- [5] A. BOSSAVIT, *How weak is the “weak solution” in finite element methods?*, Magnetics, IEEE Transactions on, 34 (1998), pp. 2429–2432. doi:10.1109/20.717558.
- [6] CHOMP GROUP, *CHomP*. <http://chomp.rutgers.edu/software/>, 2012. Accessed: 25.6.2013.
- [7] K. CLARKSON, *A program for convex hulls*. <http://www.netlib.org/voronoi/hull.html>, 1995. Accessed: 25.6.2013.
- [8] T. K. DEY, A. N. HIRANI, AND B. KRISHNAMOORTHY, *Optimal homologous cycles, total unimodularity, and linear programming*, in Proceedings of the 42nd ACM symposium on Theory of computing, STOC '10, ACM, 2010, pp. 221–230. doi:10.1145/1806689.1806721.
- [9] T. K. DEY, K. LI, AND J. SUN, *Computing geometry-aware handle and tunnel loops in 3d models*, ACM Trans. Graph., (2008), pp. 1–9.
- [10] P. DLOTKO AND R. SPECOGNA, *Efficient cohomology computation for electromagnetic modeling*, CMES: Computer Modeling in Engineering & Sciences, 60 (2010), pp. 247–278.
- [11] P. DLOTKO, R. SPECOGNA, AND F. TREVISAN, *Automatic generation of cuts on large-sized meshes for the  $T - \Omega$  geometric eddy-current formulation*, Computational Methods in Applied Mechanical Engineering, 198 (2009), pp. 3765–3781.
- [12] P. DULAR, C. GEUZAIN, F. HENROTTE, AND W. LEGROS, *A general environment for the treatment of discrete problems and its application to the finite element method*, Magnetics, IEEE Transactions on, 34 (1998), pp. 3395–3398. doi:10.1109/20.717799.

- [13] P. DULAR, P. KUO-PENG, C. GEUZAINÉ, N. SADOWSKI, AND J.P.A. BASTOS, *Dual magnetodynamic formulations and their source fields associated with massive and stranded inductors*, *Magnetics*, IEEE Transactions on, 36 (2000), pp. 1293–1299. doi:10.1109/20.877677.
- [14] P. DULAR, W. LEGROS, AND A. NICOLET, *Coupling of local and global quantities in various finite element formulations and its application to electrostatics, magnetostatics and magnetodynamics*, *Magnetics*, IEEE Transactions on, 34 (1998), pp. 3078–3081. doi:10.1109/20.717720.
- [15] H. EDELSBRUNNER AND E. P. MÜCKE, *Three-dimensional alpha shapes*, in Proceedings of the 1992 Workshop on Volume Visualization, ACM, 1992, pp. 75–82.
- [16] J. ERICKSON AND K. WHITTLESEY, *Greedy optimal homotopy and homology generators*, in Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '05, Society for Industrial and Applied Mathematics, 2005, pp. 1038–1046.
- [17] C. GEUZAINÉ AND J.-F. REMACLE, *Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*, *International Journal for Numerical Methods in Engineering*, 79 (2009), pp. 1309–1331. doi:10.1002/nme.2579.
- [18] P. W. GROSS AND P. R. KOTIUGA, *Electromagnetic Theory and Computation*, Cambridge University Press, 2004.
- [19] X. GU AND S.-T. YAU, *Global conformal surface parameterization*, in Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing, SGP '03, Eurographics Association, 2003, pp. 127–137.
- [20] A. HATCHER, *Algebraic Topology*, Cambridge University Press, 2002.
- [21] F. HENROTTE AND K. HAMEYER, *An algorithm to construct the discrete cohomology basis functions required for magnetic scalar potential formulations without cuts*, *Magnetics*, IEEE Transactions on, 39 (2003), pp. 1167–1170. doi:10.1109/TMAG.2003.810404.
- [22] J.-G. DUMAS, F. HECKENBACH, B. D. SAUNDERS AND V. WELKER, *GAP Homology*. <http://www.eecis.udel.edu/~dumas/Homology/>, 2011. Accessed: 25.6.2013.
- [23] T. KACZYNSKI, M. MROZEK, AND M. ŚLUSAREK, *Homology Computation by Reduction of Chain Complexes*, *Computers Math. Applic.*, 35 (1998), pp. 59–70.
- [24] L. KETTUNEN, K. FORSMAN, AND A. BOSSAVIT, *Formulation of the eddy current problem in multiply connected regions in terms of  $h$* , *Int. J. Numer. Meth. Engng.*, 41 (1998), pp. 935–954. doi:10.1002/(SICI)1097-0207(19980315)41:5<935::AID-NME321>3.0.CO;2-F.
- [25] M. MROZEK AND B. BATKO, *Coreduction homology algorithm*, *Discrete & Computational Geometry*, 41 (2009), pp. 96–118. doi:10.1007/s00454-008-9073-y.
- [26] J. R. MUNKRES, *Elements of Algebraic Topology*, Perseus Books, 1984.
- [27] M. PELLIKKA, *Magnetodynamics with cohomology conditions*. [http://onelab.info/wiki/Magnetodynamics\\_with\\_cohomology\\_conditions](http://onelab.info/wiki/Magnetodynamics_with_cohomology_conditions). Accessed: 25.6.2013.
- [28] M. PELLIKKA, S. SUURINIEMI, AND L. KETTUNEN, *Homology in Electromagnetic Boundary Value Problems*, *Boundary Value Problems*, 2010 (2010). Article ID 381953.
- [29] M. PELLIKKA, S. SUURINIEMI, AND L. KETTUNEN, *Powerful heuristics and basis selection bring computational homology to engineers*, *Magnetics*, IEEE Transactions on, 47 (2011), pp. 1226–1229. doi:10.1109/TMAG.2010.2091109.
- [30] J.-F. REMACLE, C. GEUZAINÉ, COMPÈRE, G., AND E. MARCHANDISE, *High-quality surface remeshing using harmonic maps*, *Int. J. Numer. Meth. Engng.*, 83 (2010), pp. 403–425. doi:10.1002/nme.282.
- [31] Z. REN,  *$t-\omega$  formulation for eddy-current problems in multiply connected regions*, *Magnetics*, IEEE Transactions on, 38 (2002), pp. 557–560. doi:10.1109/20.996146.
- [32] R. SPECOGNA, *Complementary geometric formulations for electrostatics*, *International Journal for Numerical Methods in Engineering (IJNME)*, 86 (2011), pp. 1041–1068.
- [33] STANFORD UNIVERSITY, *JPlex*. <http://comptop.stanford.edu/u/programs/jplex/>, 2008. Accessed: 25.6.2013.
- [34] S. SUURINIEMI, *Homological Computations in Electromagnetic Modeling*, PhD thesis, Dept. Elect. Eng., Tampere Univ. Technology, Tampere, Finland, 2004.
- [35] S. SUURINIEMI, J. KANGAS, AND L. KETTUNEN, *Driving a coupled field-circuit problem*, *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 26 (2007), pp. 899–909.
- [36] C. K. YAP, *Fundamental Problems of Algorithmic Algebra*, Oxford University Press, 2000.
- [37] A. YAVARI, *On geometric discretization of elasticity*, *Journal of Mathematical Physics*, 49 (2008). doi:10.1063/1.2830977.
- [38] A. ZOMORODIAN AND G. CARLSSON, *Localized homology*, *Computational Geometry*, 41 (2008), pp. 126–148. doi:http://dx.doi.org/10.1016/j.comgeo.2008.02.003.