Fast halo currents computation in fusion reactors by electrokinetic complementary formulations

Paolo Bettini[1,2], Ruben Specogna[3]
[1]Università di Padova, Dipartimento di Ingegneria Industriale (DII), Via Gradenigo 6/A, 35131 Padova, Italy
[2]Consorzio RFX, EURATOM-ENEA Association, C.so Stati Uniti 4, 35127 Padova, Italy
[3]Università di Udine, Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica, Via delle Scienze 206, 33100 Udine, Italy
paolo.bettini@unipd.it

***Abstract*—This paper presents a tool to estimate the halo current forces generated by disruptions that occurr during fusion reactors operations. The domain of the elektrokinetic problem is so complicated that two complementary formulations are used to monitor the discretization error. It turns out that thousands of cohomology generators are needed by the electric vector potential formulation, that would require an enormous amount of memory and computing power to retrieve all of them even using state-of-the-art algorithms. To solve this challenging problem, we present a novel algorithm to generate the absolute second cohomology group generators exploiting the idea of lazy cohomology generators stored as sparse vectors. The new algorithm is able to save orders of magnitude computational time.**

***Index Terms*—fusion reactor design, tokamaks, computer simulation**

I. Introduction

The instabilities of plasma during fusion reactors operations may cause the plasma to hit the first wall of the vessel in such a way that enormous currents are injected in the passive conductive structures of the machine. These currents, delivered on a time scale of milliseconds in a region submerged in a huge magnetic flux density field needed for plasma confinement, give rise to tremendous forces that has to be taken into account when designing the mechanical structure of the machine. A fully self-consistent model that may be used as a predictive tool for these forces is not yet available due to the complexity of the physical magneto-hydrodynamic phenomena and the quite complicated domain of study.

In [1], to give a first estimate on halo forces, two codes have been benchmarked assuming a resistive distribution of halo currents. The first one is the well-known CARIDDI code based on a gauged electric vector potential formulation, whereas the second one is the CAFE code based on complementary formulations for stationary conduction [2]. The problem to be solved consists in an electrokinetic Neumann boundary value problem where the sources are exactly the currents injected by the plasma to the first wall (FW) of the vessel. In particular, we study a so-called vertical displacement event (VDE), in which the plasma move upwards impinging the FW. Even though this problem may be solved employing the classical electric scalar potential formulation, we use complementarity to have some estimate on the quality of the solution. This is important since the geometry of the problem is so complicated that a convergence study using only one formulation may be hard to achieve using hexahedral meshes.

The practical problem encountered using electric vector potential-based formulations is that they require a complicated topological pre-processing due to the fact that the domain under study is topologically non-trivial [3], [2]. For Neumann problems, it is possible to show that a $H^2(\mathcal{K}-\partial\mathcal{K})$ cohomology basis is needed [3], [2]. We note that any basis is suitable for this application, so issues related to basis selection [4] are not of interest in this context. We also note that this group is torsion-free [3].

There are various ways to produce a cohomology group basis. CARIDDI, for instance, computes directly the kernel of a matrix whose entries are real numbers [1]: this approach is practical, but time consuming (some hours on meshes formed by some hundred of thousands hexahedra) and prone to errors due to the finite precision of real numbers. This time can be slightly reduced by using state-of-the-art reductions for rigorous cohomology computations over integers as described in [5]. Nevertheless, the time remains relatively high (more than 1 hour) because cohomology computations for this application are challenging (a few thousands of generators have to be computed). With the plan to increase the mesh size to tens of millions hexahedra, this is an obstruction that prevents the exploitation of complementarity.

In this paper we first introduce the idea behind the novel approach for cohomology computation, then we present some preliminary results.

II. Computing $H^2(\mathcal{K}-\partial\mathcal{K})$ generators

In this paper we present a novel solution for computing $H^2(\mathcal{K}-\partial\mathcal{K})$ cohomology generators by exploiting the concept of *lazy cohomology generators* and by modifying the `DS` algorithm [6], [7], that originally generates $H^1(\mathcal{K})$ lazy generators, to compute $H^2(\mathcal{K}-\partial\mathcal{K})$ generators instead. Lazy generators are a set of generators that span the corresponding cohomology group but they are not a basis, some of them being linearly dependent or cohomologically trivial. The key idea is that, with ungauged formulations, lazy generators may be used as if they were a standard basis. In fact, the linear system is already overdetermined, so adding a few dependent but consistent equations does not change the property of

the system. Moreover, since the number of generators to retrieve in fusion engineering applications is huge, in the DS algorithm implemented CAFE code they are represented as sparse vectors. Using this approach, the computation of a few thousands of lazy cohomology generators requires a some minutes even on huge meshes.

The modified DS algorithm to retrieve the $H^2(\mathcal{K}-\partial\mathcal{K})$ lazy generators reads as follows:

1) Compute the first cohomology $H^1(\partial\mathcal{K},\mathbb{Z})$ generators $\mathbf{c}^1,\ldots,\mathbf{c}^{2g}$, where g denotes the genus of $\partial\mathcal{K}$. This can be performed in linear time worst-case complexity $O(card(\partial\mathcal{K})\,g)$ with the graph-theoretic algorithm:
   a) An edge spanning tree is found on $\partial\mathcal{K}$ by a breadth first search (BFS) technique.
   b) A dual edge spanning tree is found on the graph obtained by edges that are dual to edges in $\partial\mathcal{K}$ that are not contained in the edge spanning tree by a BFS technique.
   c) The "free" edges in $\partial\mathcal{K}$ that belong neither to the tree nor to the dual tree are found. Each free edge produces a $H^1(\partial\mathcal{K})$ generator. Once a free edge interpreted as a dual edge is added to the dual tree, exactly one cycle is produced on the dual complex. The edges dual to the dual edges in the cycle are the support of the generator. The coefficients of the 1-cocycle are easily found by orienting the dual cycle.
2) Find *second cohomology group lazy generators* $\mathbf{t}^1,\ldots,\mathbf{t}^{2g}$ corresponding to $\mathbf{c}^1,\ldots,\mathbf{c}^{2g}$ in $O(card(\partial\mathcal{K})\,g)$:
   for each 1-cell E with nonzero coefficient c_E in $\mathbf{c}^i$
   for each 2-cell $T\in\mathcal{K}$ with E in the boundary
   $\langle\mathbf{t}^i,T\rangle += c_E\ \kappa(T,E)$;
   The value of the cochain $\mathbf{t}$ on a cell E is $\langle\mathbf{t},E\rangle$, whereas $\kappa(A,B)$ denotes the incidence between cells A and B. Initially, set $\langle\mathbf{t}^i,T\rangle=0$ for all 2-cells $T\in\mathcal{K}$.

This algorithm is typically between one and two orders of magnitude faster than standard algorithms for cohomology computation. Nonetheless, when this algorithm is applied to problems that require thousands generators, the dual cycles retrieval (i.e. point c) of the algorithm) becomes a bottleneck. To retrieve the cycle relative to a given free edge E, the two surface elements F_1, F_2 on $\partial\mathcal{K}$ that have E in the boundary are found. F_1 and F_2 may be interpreted as dual nodes on the dual complex of $\partial\mathcal{K}$. Then, a discrete distance field from dual node F_1 is found by a BFS technique [8]. The distance field propagation stops when the F_2 dual node is reached. Finally, the cycle is retrieved starting from F_2 and following the predecessors until F_1 is reached, see [8] for more details.

It is intuitive that independently retrieving thousands of dual cycles becomes inefficient. That is why in this paper, as a further improvement of the DS algorithm [6], [7], we propose a technique to retrieve simultaneously all generators. The idea is to find just one distance field per connected component of $\partial\mathcal{K}$ starting from a random dual node on each connected component of $\partial\mathcal{K}$. Then, all the cycles can be retrieved by the only distance field available. In fact, the dual path between a pair of dual nodes F_1, F_2 may be found by considering the predecessors of F_1 and F_2 until a common predecessor is found. The detailed algorithm and its rigorous analysis will be presented in the full paper.

Decreasing complexity from $O(card(\partial\mathcal{K})\,g)$ to $O(card(\partial\mathcal{K})+P)$, where P is the sum of the cardinality of all the cycles, the computational time is further reduced to few seconds or few tens of seconds even on huge meshes.

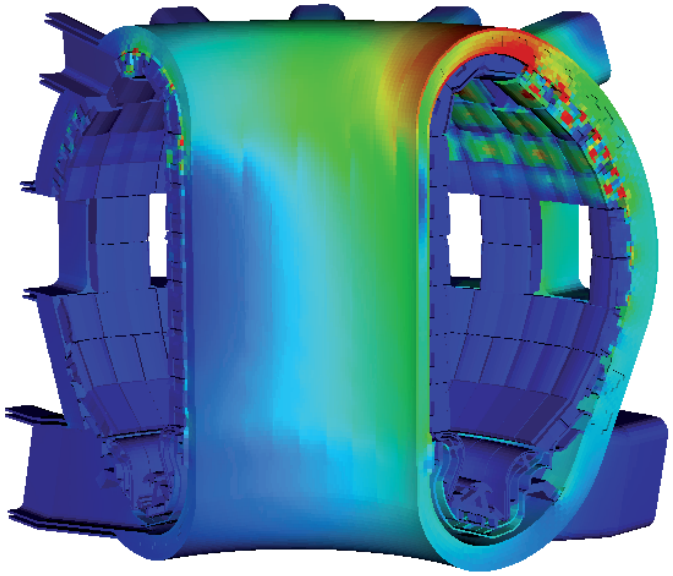

Figure 1: The module of the current density in a quarter of the machine (the simulation has been performed on the whole machine discretized with 1,132,020 hexahedra.) 3280 lazy cohomology generators are computed in less than 30 seconds.

III. Numerical results

Complementarity has been used to give an estimate of the halo currents. In Fig. 1, the typical behavior of an asymmetric VDE is shown. In the full paper more results will be presented.

References

[1] R. Albanese, P. Bettini, N. Marconato, M. Furno Palumbo, S. Peruzzo, G. Rubinacci, R. Specogna, S. Ventre, and F. Villone. Numerical modeling of 3d halo current path in iter structures. *Fusion Eng. Des.*, 2013, 10.1016/j.fusengdes.2012.11.017.

[2] R. Specogna. Complementary geometric formulations for electrostatics. *Int. J. Numer. Meth. Eng.*, 86:1041–1068, 2011.

[3] P.W. Gross and P. R. Kotiuga. *Electromagnetic Theory and Computation: A Topological Approach.* Cambridge University Press, 2004.

[4] M. Pellikka, S. Suuriniemi, and L. Kettunen. Powerful heuristics and basis selection bring computational homology to engineers. *IEEE Trans. Magn.*, 47:1226–1229, 2011.

[5] P. Dłotko and R. Specogna. Efficient cohomology computation for electromagnetic modeling. *CMES*, 60:247–278, 2010.

[6] P. Dłotko and R. Specogna. Physics inspired algorithms for (co)homology computation. *(arXiv:1212.1360)*, 2012.

[7] P. Dłotko and R. Specogna. Lazy cohomology generators: a breakthrough in (co)homology computations for cem. *submitted to COMPUMAG 2013, Budapest, Hungary*, 2012.

[8] R. Specogna. Optimal cohomology generators for 2d eddy-current problems in linear time. *IEEE Trans. Magn.*, DOI: 10.1109/TMAG.2012.2222905, 2013.