A Fast Procedure to Assemble Interaction Matrices in Equivalent Magnetic Charges Methods

Andrea G. Chiariello, Alessandro Formisano, Francesco Ledda, Raffaele Martone and Francesco Pizzo

Università degli Studi della Campania “Luigi Vanvitelli”, Aversa (CE), I-81031, Italy, Alessandro.Formisano@unina2.it

The computation of magnetic fields in presence of magnetic materials calls for methods able to cope with the complex interactions between the material characteristics and the field, yet guaranteeing satisfactory computational promptness and accuracy. The availability of new analytical formulas to compute fields from equivalent magnetic charges in the magneto-quasi-static limit allowed to set up a method providing accurate field representation with coarse discretization, but the assembly times of matrices from those complex formulas and their dense nature constitute a limit of the method. In this paper, High Performance Computing (HPC) approaches are proposed to speed up assembly, and multipole-based methods are proposed to simplify matrix structure.

***Index Terms*—Magnetic materials, Semi-Analytical methods, Multipole expansion, GPU's.**

I. Introduction

The use of magnetic materials in engineering applications drove the development of numerical methods capable of achieving satisfactory accuracy within reasonable computational times. Finite Elements Method (FEM) offers a relative ease of implementation and a simple structure of the involved matrices, but the discretization of all volumes, including air, is required. Alternative formulations as Boundary Elements Method (BEM) can exempt from air meshing, at the expense of obtaining dense matrix. A different class of methods models the magnetic materials as a set of “additional” equivalent sources contributing to the magnetic field. The magnetic parts are discretized into elementary sources, and their magnitude is obtained by suitably imposing the constitutive equations [1], [2]. Even using simple bases to discretize the sources (e.g. uniform magnetization over hexahedral small elements), the method suffers from complex formulas for the computation of the field, and the assembly results in full matrices. These drawbacks are common to a number of numerical methods [3]. Many solutions were suggested either to speed up the assembly process (including massive use of *High Performance Computing*, HPC) and to ease the field computation taking benefit from the different weight of interactions between neighboring sources and field points with respect to far interactions (e.g. multipole expansions, hierarchical matrices). The common strategy beneath these acceleration methods is splitting the contributions from each equivalent source into near and far field interactions, in such a way to guarantee a uniform accuracy. The elements of “near field” blocks need considerable computational effort; in these cases, using parallel procedures running in HPC environments is very useful to reduce the filling time. The “far field” blocks, on the other hand, can be approximated using multipole expansions (e.g. [3]), or using a QR matrix decomposition (e.g. [4]). This paper, in the framework of equivalent sources approaches, presents a method to combine HPC computation of near field interactions and simplified computations of far field elements. This short version reports just a comparison of assembly times using standard sequential coding and HPC implementation for near field blocks. The full paper will introduce the far field approximation approach and will assess the method performance on a complex geometry example from thermonuclear fusion reactors.

II. Overview of the Method

Let's consider the structure in Fig. 1, where a magnetically permeable cube is immersed in the field generated by a circular coil. Simple geometry, steady state currents and current drive assumptions allow easing exposition, but such hypotheses can be easily removed.

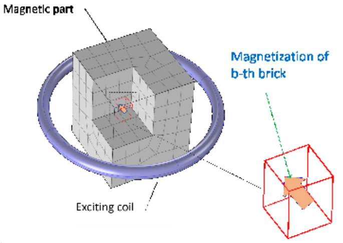

Fig. 1 – Example of geometry

The cube is discretized using a hexahedral mesh, and the effect of the magnetic material is described using the magnetization density **M**, here assumed uniform inside each element. This allows using closed form expressions for the magnetic field **H** produced by a “uniformly magnetized brick”, once its magnetization is known [5]:

$$\mathbf{H}_\mathrm{b}(\mathbf{x}_\mathrm{d}) = -\frac{1}{4\pi}\sum\nolimits_{\mathrm{S_f}\in\partial\mathrm{V_b}}(\mathbf{M}\times\hat{\mathrm{n}}_\mathrm{f})\times\nabla\mathrm{W_f}(\mathbf{x}_\mathrm{d}) \qquad (1)$$

with $\mathrm{W_f}(\mathbf{r})$:

$$\mathrm{W_f}(\mathbf{x}) = \sum\nolimits_{\mathrm{e_j}\in\partial\mathrm{S_f}}\hat{\mathrm{n}}_\mathrm{f}\times\left(\mathbf{x}_{e_j}-\mathbf{x}\right)\cdot\hat{\mathrm{t}}_\mathrm{j}\,\mathrm{w_j}(\mathbf{x}) - [(\mathbf{x}_\mathrm{f}-\mathbf{x})\cdot\hat{\mathrm{n}}_\mathrm{f}]\Omega_\mathrm{f}(\mathbf{x}) \qquad (2)$$

where e_j is one of the edges on a face S_f belonging to boundary $\partial\mathrm{V_b}$ of the brick volume $\mathrm{V_b}$; $\mathbf{x}_{\mathrm{e_j}}$ and $\mathbf{x}_\mathrm{f}$ are the position vectors of any two points along e_j and on S_f respectively (any point being equivalent due to cross and dot-product); $\hat{t}_j$ is the tangent vector of e_j; $\hat{n}_f$ the normal vector of S_f; Ω_f is the solid angle subtending face S_f from the origin, and, finally, $\mathrm{w_j}$ is:

$$\mathrm{w_j}(\mathbf{x}) = \ln\frac{|\mathbf{x}-\mathbf{x_e}|+|\mathbf{x}-\mathbf{x_s}|+|\mathbf{x_e}-\mathbf{x_s}|}{|\mathbf{x}-\mathbf{x_e}|+|\mathbf{x}-\mathbf{x_s}|-|\mathbf{x_e}-\mathbf{x_s}|} \qquad (3)$$

where $\mathbf{x}_s$ and $\mathbf{x}_e$ are starting and ending tips of the edge.

To compute the direction and the amplitude of **M**, it is necessary to evaluate not only the field due to the source coil, but also the contribution due to all the iron sources inside the mesh hexahedra. The assembly process consists then in creating the matrix $\boldsymbol{\mathcal{H}}$ providing the field inside the k-th brick generated by an elementary magnetization in the b-th brick. The matrix can be assembled component-wise, and then it will have a number of columns equal to three times the number of bricks. The number of rows depends on the method used to determine the actual magnetization. If using a one-point collocation method (used here for the sake of exposition, although this approach drives to oscillations in the **M** distribution), and assembling again element-wise, a square matrix $\boldsymbol{\mathcal{H}}$ is obtained. Similar considerations, limitedly to the number of rows, apply for the matrix $\boldsymbol{\mathcal{G}}$ providing the magnetic field in the k-th brick generated by a unit current in the source coil. As concerning the determination of the actual value of **M** inside each brick, we enforce the constitutive relationship **M**=f(**H**) in each brick using an under relaxed Picard approach, similar to what described in [6] and already used in [1]; in such a way the (i+1)-th estimate of **M** is:

$$\begin{aligned}
&\mathbf{H}^{\mathrm{i+1}}(\mathbf{x}_{\mathrm{b}}) = \mathbf{H}_{\mathrm{src}}(\mathbf{x}_{\mathrm{b}}) + \boldsymbol{\mathcal{H}}\,\mathbf{M}_{\mathrm{b}}^{\mathrm{i}} + \boldsymbol{\mathcal{G}}\,\underline{\mathrm{I}}_{c} \\
&\mathbf{M}_{\mathrm{b}}^{*} = f\left(\mathbf{H}^{\mathrm{k+1}}(\mathbf{x}_{\mathrm{b}})\right) \qquad \mathrm{b}=1,..,\mathrm{N_b} \qquad (4) \\
&\mathbf{M}_{\mathrm{b}}^{\mathrm{i+1}} = \left(1-\beta_{\mathrm{b}}^{\mathrm{i}}\right)\mathbf{M}_{\mathrm{b}}^{\mathrm{i}} + \beta_{\mathrm{b}}^{\mathrm{i}}\mathbf{M}_{\mathrm{b}}^{*}
\end{aligned}$$

where $\mathbf{x}_b$ are the central points of bricks and f is the constitutive relationship in the magnetic material relating **H** and **M** (verified in the center of bricks only with a collocation method). The relaxation factor β_b^i is adaptively adjusted in each brick and at each iteration of Picard procedure as $\beta_b^i = 1/\left[1-\left(1-\mu_b^i\right)\lambda\right]$, where μ_b^i is the estimate of relative permeability at i-th step in the b-th brick, and λ is the smallest eigenvalue of $\boldsymbol{\mathcal{H}}$**.** Elements of $\boldsymbol{\mathcal{H}}$ and $\boldsymbol{\mathcal{G}}$ are computed using analytical expressions [5] involving rather complex expressions, very expensive to be evaluated if using standard sequential coding on desktop-class computer.

III. HPC Approaches to Matrix Assembly

Systems based on Graphic Processor Units (GPU) architecture are getting an increasing success in HPC. Contrary to standard sequential Central Processor Units (known as "CPU"), designed to execute very different codes, GPU are optimized to execute just simple but highly parallel codes,. Due to this limitation, the GPU achieve the best performance when all the cores execute the same tasks, and all the tasks are independent each from the others [6]. In the $\boldsymbol{\mathcal{H}}$ assembly process, in order to provide the GPU homogeneous tasks, all the entries are grouped in near field and far field blocks during a preprocessing step. The full paper will present a different grouping, based on the **M** representation inside blocks (e.g. uniform or linear). Due to the particular structure of the GPU memory, in the preprocessing phase the geometrical information needed to calculate the interactions is structured in a coalescent way, so that each core can access to a dedicated GPU memory containing each data (see [6] for more details). Since usually GPU inner memory is in the order of GigaBytes, and the typical matrix dimension does not fit this limitation, the final reduction step has to be done in the CPU. Details about GPU implementation and its advantages will be given in the full version.

IV. Assessment of GPU evaluation

In order to show advantages of GPU computation for near-field blocks, we have considered a 10×10×30 cm parallelepiped, with constant permeability μ_{rel}=1000. A decomposition in an increasing number of uniformly magnetized bricks is used, and all contributions are considered "near interactions", thus computed using Eq. (1). In the full paper, indications about the optimal choice for the number of bricks will be given. The considered architecture comprises two Intel Xeon E52690 @2.90 GHz, equipped with a Nvidia Kepler K20 GPU (2496 cores, 6 GB of Global Memory). In order to ease understanding, a $\boldsymbol{\mathcal{H}}$ multiplication times the **M** values array takes approximatively 10^{-1} s. A comparison of the assembly times using CPU version of the code and GPU coding is reported in Table I for increasing discretization levels.

Table I – Speed ups for increasing number of bricks

N° of bricks	1040	2944	4992	6016	7040
CPU Time [s]	4	34	97	140	191
Speed up	12	34	39	40	41

V. Conclusions and Outlook

The paper presents an approach to speed-up assembly of iteration matrices involved in equivalent magnetic sources methods. The method bases on the splitting of matrix into near and far field interaction blocks, and on the use of suited measures to speed up computation of each block. The full version will provide a complete description of the method, including the parentage with other source-based methods, the interplay with a global numerical strategy for the field computation in the magneto-quasi-static limit and the block-structure of the matrices. An application to real cases with 3D geometries will also be presented.

Acknowledgment

This work has been partly supported by ENEA/Euratom/CREATE association.

References

[1] A. G. Chiariello, A. Formisano and R. Martone, "A Fast, Semi-Analytical Method for Field Computation in Presence of Magnetic and Conductive Materials", *Proc. of CEFC 2016*, Miami (FL), November 2016.

[2] I. Mayergoyz, "Iteration Methods for the Calculation of Steady Magnetic Fields in Nonlinear Ferromagnetic Media," *COMPEL*, Vol. 1 n° 2, pp. 89-110, March 1982.

[3] R. Albanese G. Rubinacci A. Tamburrino S. Ventre F. Villone, (2001),"A fast 3D eddy current integral formulation", *COMPEL*, Vol. 20 Iss 2 pp. 317 – 331.

[4] M. Bebendorf, *Hierarchical Matrices* (Lecture Notes in Computational Science and Engineering). New York, NY, USA: Springer-Verlag, 2008.

[5] M. Fabbri, "Magnetic flux density and vector potential of uniform polyhedral sources", *IEEE Trans. on Magnetics*, Vol. 44 n° 1, pp. 32-36, Jan. 2008.

[6] D. Kirk, W. Hwu, *Programming Massively Parallel Processors, Second Edition: A Hands-on Approach*, Morgan Kaufmann, Burlington, MA (2012)