

Parallel Finite Volume solver to compute electromagnetic fields

W. Poirier¹, M. Boubekeur², I. Kissami³, A. Kameni¹

¹Laboratory Group of Electrical Engineering of Paris, CentraleSupélec, UMR 8507 CNRS, Paris-Saclay University, Sorbonne University, Plateau de Moulon, 11 rue Joliot Curie, 91190, Gif-sur-Yvette, France

²LAGA, UMR 7539 CNRS, Sorbonne Paris Nord University 99 Avenue J. B. Clément, 93430, Villetaneuse, France

³College of Computing, Mohammed VI Polytechnic University, Lot 660, Ben Guerir, 43150, Morocco

In this paper, we present the implementation of a DDFV solver on a massively parallel architecture to compute electromagnetic fields. This work relies on three standard scientific libraries, Metis for mesh partitioning, MPI to manage process communications and PETSc for linear algebra operations on multi-core supercomputers. A discrete basis renumbering technique is also implemented to reduce synchronization slowdown and memory bandwidth limitation. A 2D example solving a system with a few million Degrees Of Freedom (DOF) on a supercomputer with 480 cores is proposed to show the efficiency of our implementation.

Index Terms—Finite volume methods, Modeling, Parallel architectures, Supercomputers

I. INTRODUCTION

The global decarbonization of our society is highly dependent on electrical power to succeed. To overcome this challenge, new power generation devices such as electrical transformers or generators are being studied in electrical engineering. Their design requires computations of electromagnetic fields on increasingly complex systems. The differential systems treated are often derived from Maxwell's equations. They may be linear, for example when considering classical conductive metals such as copper or aluminum, or nonlinear when considering specific materials such as ferromagnetic or superconducting materials.

In electrical engineering, the Finite Element Methods (FEM) are the widely used techniques to compute electromagnetic fields. In the case of complex systems, parallel computing techniques are needed to reduce computational time. Parallel matrix solvers have been developed in recent years and are used in the most FEM software such as FreeFEM, COMSOL. They are based on sequential computations. They have shown their efficiency in solving complex systems [4] but still remain costly to solve large systems with a few million DOF. Domain Decomposition Methods (DDM) have been introduced to achieve high performance FEM computations by breaking down the geometry into smaller parts [5]. However, for complex geometries such as coils for high field magnets or superconducting twisted cables, DDM methods can become trickier to implement.

This paper focuses on parallel Finite Volume solvers which are widely spread in the fluid mechanics domain. They have proven to be effective on Computational Fluid Dynamics problems [6], [14]. These methods are based on mesh partitioning which is an asset to manage process communications. They are rarely used in electrical engineering to compute electromagnetic fields. In this paper, we propose to use a parallel Dual Discrete Finite Volume Solver (DDFV) to solve the H-formulation of electromagnetic problem that can be encountered in many numerical models in electrical engineering [7].

Modern computing architecture provides hundreds of nodes

and GPU [8], [9], and libraries have been developed to better exploit their capabilities. In this work, the Message Passing Interface (MPI) protocol is used to handle communication between processes [10], Metis library for mesh partitioning [12] and PETSc library for linear algebra operations [13]. However, those libraries cannot overpass hardware limitations. Thus, programmers need to take care of the amount of synchronization between processes as well as memory bandwidth usage. For that, a discrete basis renumbering is implemented.

II. LIBRARIES INTEGRATION

First we use Metis to partition the mesh as in the example presented in Fig. 1, where the mesh is divided into 4 parts. Note that each mesh partition is in a different process. In

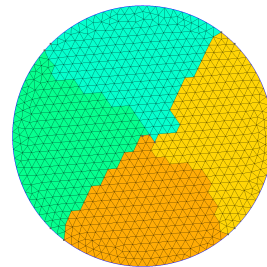


Fig. 1. Example of a mesh partitioning in 4 groups with Metis.

a Finite Volume scheme, unknowns are defined on cells but key expressions are the fluxes defined on the faces. When the faces are in the partition, computations of flux expressions are performed by the same process. When a face is on the boundary of two partitions, communications between the two concerned processes become time consuming. To better optimize this communication time it is usual to implement a preprocessing step. This allows us to identify the faces common to at least two partitions. It is realized only once after Metis mesh partitioning. As the data needed to compute matrix coefficients for these faces are in different processes, this preprocessing step should define which process will perform the computations and thus avoid unnecessary communication and synchronization halts.

As a remark, it should be noticed that partitioning into N -groups, on sufficiently large meshes, divides the number of faces per partition by approximately N . The calculation of the matrix coefficients should follow the same improvement. Depending on the application, this can have a significant impact. Nonlinear problems, where matrices are often recalculated are a case in point.

By default, the numbering of the discrete basis has no consistency with Metis partitioning. Thus, their position on the problem matrix doesn't follow the locality rule expressed previously. This misorganization costs more memory bandwidth for matrix operations. This problem is solved by renumbering cells and reorganizing them by partitions as shown by Fig. 2.

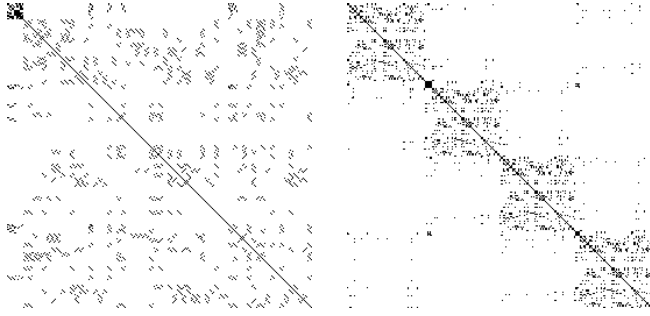


Fig. 2. View of the matrix coefficient position for a mesh with 4 partitions. (Left) without renumbering (right) with renumbering.

III. BENCHMARK

This section focuses on the integration of PETSc. It is one of the most efficient and popular toolkit for scientific computation and is compatible with MPI libraries.

The partial differential equation (PDE) is the H -formulation in resistive materials, obtained from Maxwell's equations (1):

$$\mu_0 \frac{\partial H}{\partial t} + \nabla \times (\rho \nabla \times H) = 0 \quad (1)$$

where the unknown is the vector magnetic field H and ρ the material resistivity.

Let's consider a 2D model consisting of two concentric circles of diameter $d_1 = 2 \text{ mm}$ and $d_2 = 20 \text{ mm}$. Their resistivities are respectively $\rho_1 = 10^1 \Omega.mm^{-1}$ and $\rho_2 = 10^3 \Omega.mm^{-1}$. The boundary condition is an applied magnetic field in the z direction $H_a = H_0 \hat{z}$ with $H_0 = \frac{1}{\mu_0}$ where $\mu_0 = 4\pi 10^{-7} \text{ H.m}^{-1}$. The time step is $\delta t = 10^{-3} \text{ s}$ and calculation duration is $T_f = 50\delta t$.

First, we consider a mesh of 47k cells to test the effect of matrix reorganization. The server used was an AMD Ryzen Threadripper Pro 5965WX 24-Cores. A good scale factor is achieved when the renumbering is applied Table I. Without renumbering, the scale factor obtained is lower than 3. To test the scaling capability over a higher number of cores, we use a 4M cells mesh. Computations are performed on a cluster with dual socket Intel Xeon Gold 6230 20-Cores server. The computation with 40 cores is used as reference and the scale factor presented in Fig. 3. Until 320 cores, the scale factor is close to the perfect one. Beyond that, it decreases due to MPI implementation [11].

TABLE I
SCALING COMPARISON OF THE SOLVING ITERATION WITH AND WITHOUT
RENUMBERING OF THE MATRIX COEFFICIENT

No. of MPI	With renumbering		Without renumbering	
	Time (s)	Scale factor	Time (s)	Scale factor
1	21.3	1.0	21.4	1.0
2	10.4	2.0	17.1	1.2
4	4.8	4.4	11.5	1.8
6	3.1	7.0	11.1	1.9
8	2.3	9.3	9.2	2.3
12	1.6	13.7	8.0	2.7
16	1.3	16.9	7.4	2.9
18	1.1	19.5	7.8	2.7
24	0.9	24.4	8.8	2.4

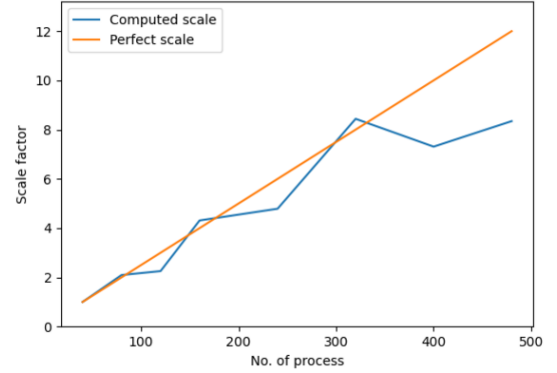


Fig. 3. Scaling performance over a large number of cores

REFERENCES

- [1] E. Nilsson *et al.*, "Design of the Superconducting AC and DC Distribution for the ASCEND Demonstrator at Airbus", *IEEE Transactions on Applied Superconductivity*, vol. 33, no. 5, pp. 1-6, Aug. 2023.
- [2] A. Allais *et al.*, "SuperRail-World-First HTS Cable to be Installed on a Railway Network in France", *IEEE Trans. on App. Sup.*, Vol. 34, no. 3, pp. 1-7, May 2024.
- [3] P. Dular, C. Geuzaine, A. Genon, W. Legros, "An Evolutive Software Environment for Teaching the Finite Element Method in Electromagnetism", *IEEE Transactions on Magnetics*, Vol. 35, no. 3, 1999.
- [4] C. Fu, "Parallel Computing for Finite Element Structural Analysis on Workstation Cluster," *2008 International Conference on Computer Science and Software Engineering*, pp. 291-294, Wuhan, China, 2008.
- [5] W.-J. Wang, X.-J. Chen, H.-Y. Li, H.-J. Zhou, W. -Y. Yin, "A Multilevel Method With Novel Correction Strategy for Parallel Finite-Element Analysis of Electromagnetic Problems," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 7, pp. 3787-3791, July 2018
- [6] J. Hawkes, G. Vaz, A. Phillips, S. Cox, S. Turnock, Stephen, "On the Strong Scalability of Maritime CFD", *Journal of Marine Science and Technology*, 2017.
- [7] W. Poirier, M. Boubekeur, I. Kissami, A. Kamani, "Discrete Duality Finite Volume scheme to model Superconductor", *submitted to COMPUMAG 2025*.
- [8] B. Venu, "Multi-core processors - An overview", Oct 2011.
- [9] Mitch Leslie, "First Supercomputer Breaks Exascale Barrier, with More Expected Soon", *Engineering*, pp. 10-12 2023.
- [10] Message Passing Interface Forum, "MPI: A Message-Passing Interface Standard Version 4.1", Nov 2023.
- [11] Sai P. Chenna, M. Steyer, N. Kumar, M. Garzaran, P. Thierry, "Modeling and Simulation of Collective Algorithms on HPC Network Topologies using Structural Simulation Toolkit"
- [12] G. Karypis, V. Kumar, "METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices", Retrieved from the University Digital Conservancy
- [13] L. Dalcin, R. Paz, P. Kler, A. Cosimo, "Parallel distributed computing using Python", *Adv Water Resour.*, Vol. 34, pp. 1124-1139, Sept 2011.
- [14] I. Kissami, "Manapy: an MPI-based Python framework for solving Poisson's equation using finite volume on unstructured-grid", *AIP Conference Proceedings*, Vol. 3034, 2024.