A Special Purpose Finite-Element Framework for High-Temperature Superconductor Applications

Christian Messe

Accelerator Technology and Applied Physics Division

Lawrence Berkeley National Laboratory

Berkeley, CA

cmesse@lbl.gov

***Abstract*—Significant progress has been made in the recent years on mixed finite-element formulations of the magnetostatic Maxwell equations. These formulations are crucial for the electromagnetic modeling of high-temperature superconductors (HTS) used in magnets and superconducting cables. We are developing the Berkeley Lab Finite Element Framework BELFEM in order to collect these new formulations into a community toolbox that is tailored to the needs of selected HTS applications. This abstract begins with a short summary of the considered methods, sets up the project goals and discusses the basic code architecture, as well as the current development status.**

Keywords—finite-element, high-temperature superconductors, magneto-static Maxwell equations *(key words)*

I. Introduction

One distinctive characteristic of the magnetostatic Maxwell equations compared to say thermal conduction or mechanic elasticity is that there are not one but two conservation equations, only one of which can be solved consistently at a time. If one chooses to solve for the Ampere-Maxwell equation, the magnetic flux density ($\mathbf{b}$) is conform, and the magnetic vector potential ($\mathbf{a}$) represents the unknowns the model is solved for. Choosing to solve for Faraday's law on induction on the other hand leads to a conform magnetic field ($\mathbf{h}$). Now the degrees of freedom are either expressed through the vector field ($\mathbf{h}$) or the scalar potential (ϕ). Experience has shown that a mixture of these formulations leads to the most efficient models, with the $\mathbf{h}$-$\mathbf{a}$ and $\mathbf{h}$-ϕ formulations being the most popular ones. The handling of the different types of degrees of freedom, the special treatment of interfaces between different domains, and specialties like domain cuts in $\mathbf{h}$-ϕ or the treatment of thin shells require the data structures to be significantly more flexible than for traditional finite element problems. Moreover, the system matrices produced by these formulations tend to be very ill-conditioned, non-symmetric and non-positive definite. These circumstances motivated us to start the development of the Berkeley Lab Finite Element Framework (BELFEM). This special purpose code aims to be tailored to the needs of HTS magnet and cable modeling.

II. State of the Art

A general overview over the $\mathbf{h}$-$\mathbf{a}$ and $\mathbf{h}$-ϕ formulation and relevant literature has been given by Dular et al. [1]. Besides the weak forms of both formulations, the paper also discusses the performances of various iteration techniques. It also recaps the general concept of domain cuts that have been introduced back in the 1990s [2], [3]. An improved cutting method using Lagrange multipliers has been introduced by both Smajic et al. [4] and Arsenault et al. [5], [6]. Coupling strategies for the $\mathbf{h}$-$\mathbf{a}$ formulation have been investigated by Brambilla et al. [7]. In a more recent work, Dular investigated numerical stabilities of selected formulations [8]. A thin-shell approach of the $\mathbf{h}$-$\mathbf{a}$ formulation has been presented by Brotot et al. [9]. A recent work of Alves et al. [10] introduced a thin-shell variant of the $\mathbf{h}$-ϕ formulation.

III. Motivation and Project Goals

Most of the formulations mentioned in Sec. II have been implemented on top of commercial toolboxes such as COMSOL and are not yet available to the general public. Moreover, it is in the very nature of closed source codes to limit the access the developer has to the data structure and the knowledge of underlying algorithms. The desire of having full control over the data structure motivated us to look into open-source alternatives. We found the library STRUMPACK [11] to be a very promising solver for the type of sparse matrices we encounter, but neither of the leading open source FEM codes such as MFEM or FreeFEM support this solver at this time. Having defined the project goals below, we decided to develop a new finite element framework from scratch to achieve them:

- Support $\mathbf{h}$-$\mathbf{a}$ and $\mathbf{h}$-ϕ formulations for 2D and 3D, as well as thin shells, both with first and higher order elements.
- Support multiphysics, specifically thermal and mechanical coupling, as well as current sharing.
- Have a text-based user interface that is tailored to the needs of HTS magnet and cable modeling.
- Use popular open-source data formats, such as HDF5, GMSH [12], and Exodus II (ParaView) [13].
- Link against modern sparse linear algebra solvers such as STRUMPACK, PETSc, PARDISO and MUMPS.
- Run in parallel using the MPI standard.
- Be readable, extendable and maintainable.

IV. Code Architecture

The code is written in C++ with a strong focus on modularity and flexibility. No external library is ever accessed directly, but through wrappers. This way, a unified, mode readable and therefore more maintainable source code is achieved. External libraries can be easily exchanged should the need arise in the future. The most essential modules of the code are:

A. Dense Linear Algebra

Dense linear algebra is used on the *element level* where the discretized physics are implemented. To simplify the implementation and make it more readable, the module aims to imitate the look and feel of MATLAB most scientists are familiar with. It can currently wrap against Armadillo [14], which has more support, or Blaze [15], which is faster.

B. Mesh

An external mesh generator provides node coordinates, element-to-node topology as well as block and sideset information. On top of these, the mesh modules generates the edges and faces that are needed for the Nédélec interpolations [16] in the conducting regions, as well as the cuts in the air domain that are needed for the $\mathbf{h}$-ϕ formulation. The geometry also defines processor ownerships that are required for parallel computing. Moreover, all computed field information is stored in the mesh class.

C. Degree of Freedom Management

Each computational domain has either $\mathbf{h}$, $\mathbf{a}$ or ϕ degrees of freedom, interfaces may also contain Lagrange multipliers. The purpose of the degree of freedom management system is to determine a numbering scheme for all degrees of freedom that is unique over all processors. With the numbering scheme in place, this module computes the adjacency graph that determines the sparsity pattern of the system matrix.

D. Integrated Weak Governing Equation

The equation object computes the element-wise contributions of the Jacobian matrix and the right hand side of the sparse linear system. Since individual blocks and sidesets represent different physical domains, each domain type relevant to the equation requires a specialized subroutine.

E. Finite Element Kernel

The kernel manages both the initialization of the calculation as well as the time stepping method. For each used geometry element on the mesh, a degree-of-freedom element is created. Once the sparsity pattern is computed the system matrix is initialized and the time stepping method is started. Within one iteration step, multiple equation objects can be stacked serially. One can for example first compute the electromagnetic problem and solve it using a direct solver, and then compute the thermal conduction problem with an iterative one. The user can specify criteria for the code to switch between Newton-Raphson iterations and Picard iterations during a timestep.

V. Current Development Status

Code development started in June 2020. We are working closely together with the STRUMPACK development team in order to get the maximum performance out of the solver.

A. Code Development and Validation

At the time this digest was written, the wrappers for mesh input and output, parallelization as well as dense and sparse linear algebra have been completed. Both node and edge interpolation functions for triangle and tetrahedron shape functions have been implemented and validated against non-physical unit tests. Both the $\mathbf{h}$-$\mathbf{a}$ and the $\mathbf{h}$-ϕ formulation passed first physical integration tests against the Biot-Savart law. We are currently finalizing a major code review and validate the code against analytical solutions from Brandt [17]. In the next development step, we will compare the solutions of the code against selected benchmark problems in COMSOL.

B. Theory Manual

Along with the code, we are developing a theory manual that summarizes the implemented formulations in a textbook-like fashion. The notations of the referred literature have been unified and transcribed into the classical FEM notation often found in standard literature such as Zienkiewicz [18], Bathe [19] and Belytscho [20]. We hope that this approach provides an easier access to the numerical aspects of the mixed formulations, and encourages scientists to develop their own formulations on top of our framework.

VI. Conclusion

A new finite element framework for high-temperature superconductor applications is being developed at the Lawrence Berkeley National Laboratory. The code provides highly flexible data structures sufficient for current and future mixed-field formulations. Popular open source tools such as the mesh generator GMSH and the post processor ParaView are incorporated into the workflow. The code links against several modern sparse linear algebra libraries and runs in parallel using the MPI standard. Once a sufficient state of development is achieved, it is planned to publish the source code under a BSD-like license.

References

[1] J. Dular, C. Geuzaine, and B. Vanderheyden, "Finite-Element formulations for systems with high-temperature superconductors," *IEEE Transactions on Applied Superconductivity*, vol. 30, no. 3, pp. 1–13, 2019, fundamental equations.

[2] P. Dular, F. Henrotte, F. Robert, A. Genon, and W. Legros, "A generalized source magnetic field calculation method for inductors of any shape," *IEEE Transactions on Magnetics*, vol. 33, no. 2, pp. 1398–1401, 1997, h-phi with cut: original paper 1.

[3] P. Dular, C. Geuzaine, and W. Legros, "A natural method for coupling magnetodynamic H-formulations and circuit equations," *IEEE Transactions on Magnetics*, vol. 35, no. 3, pp. 1626–1629, 1999, h-phi with cut: original paper 2.

[4] J. Smajic, M. K. Bucher, C. Jäger, and R. Christen, "Treatment of multiply connected domains in time-domain discontinuous galerkin $h - \phi$ eddy current analysis," *IEEE Transactions on Magnetics*, vol. 55, no. 6, pp. 1–4, 2019, lagrange multiplier interface.

[5] A. Arsenault, F. Sirois, and F. Grilli, "Implementation of the h-ϕ formulation in COMSOL Multiphysics for simulating the magnetization of bulk superconductors and comparison with the h-formulation," *IEEE Transactions on Applied Superconductivity*, vol. 31, no. 2, pp. 1–11, 2021-3.

[6] A. Arsenault, B. d. S. Alves, and F. Sirois, "COMSOL Implementation of the h-ϕ-formulation with thin cuts for modeling superconductors with transport currents," *IEEE Transactions on Applied Superconductivity*, vol. 31, no. 6, pp. 1–9, 2021, h-phi with cut: comsol.

[7] R. Brambilla, F. Grilli, L. Martini, M. Bocchi, and G. Angeli, "A Finite-Element Method framework for modeling rotating machines with superconducting windings," *IEEE Transactions on Applied Superconductivity*, vol. 28, no. 5, pp. 1–11, 2018, interface H-A.

[8] J. Dular, M. Harutyunyan, L. Bortot, S. Schops, B. Vanderheyden, and C. Geuzaine, "On the stability of mixed finite-element formulations for high-temperature superconductors," *IEEE Transactions on Applied Superconductivity*, vol. 31, no. 6, pp. 1–12, 2021, coupling h-a.

[9] L. Bortot, B. Auchmann, I. C. Garcia, H. D. Gersem, M. Maciejewski, M. Mentink, S. Schps, J. V. Nugteren, and A. P. Verweij, "A coupled AH formulation for magneto-thermal transients in high-temperature superconducting magnets," *IEEE Transactions on Applied Superconductivity*, vol. 30, no. 5, pp. 1–11, 2019, h-a thin.

[10] B. d. S. Alves, V. Lahtinen, M. Laforest, and F. Sirois, "Thin-shell approach for modeling superconducting tapes in the h-ϕ finite-element formulation," *arXiv*, 2021, h-phi thin.

[11] F.-H. R. S. W. Pieter Ghysels, Xiaoye S. Li and A. Napov, "An efficient multicore implementation of a novel hss-structured multifrontal solver using randomized sampling," *SIAM Journal on Scientific Computing*, vol. 38, no. 5, pp. 358–384, October 2016.

[12] C. Geuzaine 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*, vol. 79, no. 11, pp. 1309–1331, 2009.

[13] L. A. Schoof and V. R. Yarberry, "Exodus ii: a finite element data model," Sandia National Laboratories, Tech. Rep. SAND-92-2137, 1994.

[14] C. Sanderson and R. Curtin, "Armadillo: a template-based c++ library for linear algebra," *Journal of Open Source Software*, vol. 1, pp. 26–27, 2016.

[15] J. T. K. Iglberger, G. Hager and U. Rüde, "High performance smart expression template math libraries," in *Proceedings of the 2nd International Workshop on New Algorithms and Programming Models for the Manycore Era APMM 2012 and HPCE 2012*.

[16] J. C. Nedelec, "Mixed finite elements in $\mathbb{R}^3$," *Numerische Mathematik*, vol. 35, no. 3, pp. 315–341, 1980.

[17] E. H. Brandt, "Superconductors of finite thickness in a perpendicular magnetic field: Strips and slabs," *Physical Review B*, vol. 54, no. 6, pp. 4246–4264, 1996, analytical solution.

[18] O. C. Zienkiewicz, R. L. Taylor, and D. D. Fox, *The Finite Element Method for Solid and Structural Mechanics*, 7th ed. Oxford, UK: Butterworth-Heinemann, 2013.

[19] K.-J. Bathe, *Finite Element Procedures*, 2nd ed. Watertown, MA: Klaus-Jürgen Bathe, 2014.

[20] T. Belytschko, W. K. Liu, B. Moran, and K. I. Elkhodary, *Nonlinear Finite Elements for Continua and Structures*, 2nd ed. John Wiley & Sons, 2014.