ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-MOPS30

HIGH-PERFORMANCE MAGNET SIMULATION SOFTWARE

I. A. Zilberter†, S. A. Veitzer, Tech-X Corporation, Boulder, CO, USA
B. M. Cowan[1], Alturas Research, LLC, Boulder, CO, USA

Abstract

We present a high-performance solver for the magnetostatic equations. The solver can simulate nonlinear and anisotropic magnetic materials on a highly variable grid, enabling efficient resolution of fine features even in very large systems. It is built on the Tpetra parallel sparse linear algebra package, allowing it to handle problems with billions of degrees of freedom and employ hardware acceleration with Nvidia graphics processing units. Integration into the VSim electromagnetics software allows users to design magnetic systems using existing graphical interface features. Example simulations of nonlinear magnets, with application to particle accelerator magnet design, are shown.

INTRODUCTION

In this paper, we present magnet simulation software that can efficiently and accurately solve large magnet problems, such as the multi-bend achromat magnets required for the Advanced Photon source (APS) upgrade at ANL [1]. For instance, a proposed MBA dipole design is shown in Figure 1. This magnet is over 2 m long, but with spacings between iron segments and coil thicknesses of just 2 cm, and chamfered edges with a feature length scale of just 1mm. This large separation of length scales raises the problem of computational modeling of the entire structure while resolving the smallest features [2, 3]. In addition, the simulation involves solving a nonlinear problem because nonlinear magnetic materials are used. An engineer involved in the design of this magnet reported that the existing software used to simulate the magnetic fields failed on a problem with less than 10 million degrees of freedom, without reaching the desired 1 mm resolution of fields in the gap [4].

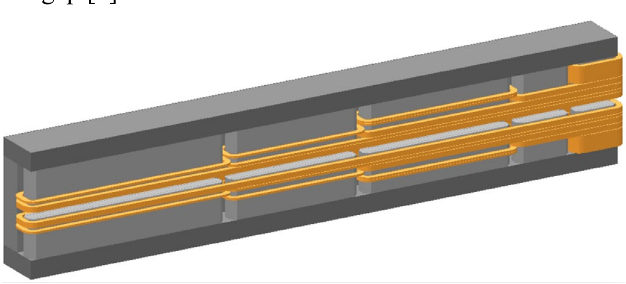

Figure 1: A Schematic of a proposed dipole magnet for the APS lattice upgrade, with exterior shields cut away.

The new magnet solver is incorporated into VSim software, a high-performance multiphysics simulation tool under development at Tech-X Corporation [5]. Its computational engine, Vorpal, is a high-performance code which runs highly efficiently on large-scale compute systems, such as supercomputers with as many as 100,000 cores [6], and is designed to work with hardware acceleration through graphics processing units (GPUs). In addition to its Vorpal physics engine, VSim includes a graphical user interface, called Composer, which provides geometry setup from CAD files, material selection, parameter adjustment, and visualization. Integrating our magnetic code into VSim allows us to leverage its existing capabilities including efficient parallelism, structured data architecture for fields and geometries, and the GUI to provide a user-friendly and efficient computational tool for magnet designers.

† izilberter@txcorp.com

METHODOLOGY

The Magnetostatic Problem

We solve the magnetic equations:

$$\nabla \times H = J \tag{1}$$
$$\nabla \cdot B = 0 \tag{2}$$

Here H and B have a nonlinear relationship, described by a vector function B(H). We reduce the nonlinearity in this system to a scalar equation as follows: First, we use a Helmholtz decomposition of H, letting $H = \nabla \times u + \nabla\varphi$. Then the magnetostatic equations become

$$\nabla \times (\nabla \times u) = J \tag{3}$$
$$\nabla \cdot B\,(\nabla \times u + \nabla\varphi) = 0 \tag{4}$$

Since Eq. (1) implies that $\nabla \cdot J = 0$, Eq. (3) becomes $-\nabla^2 u = J$, so u is solved with a linear equation. Then Eq. (4) is a nonlinear equation in φ only. To solve it, we use the Newton method, an iterative approach to solving nonlinear equations. Given an intermediate solution φ^n at iteration n, we compute a correction $\varphi^{n+1} = \varphi^n + \delta\varphi$ by linearizing Eq. (4). Letting $H^n = \nabla \times u + \nabla\varphi^n$, we approximate

$$0 = \nabla \cdot B\,(H^n + \nabla\delta\varphi) \approx \nabla \cdot B\,(H^n) + \partial B/\partial H^n\, \nabla(\delta\varphi) \tag{5}$$

where ∂B/∂H is the Jacobian of the magnetic constitutive relation. This yields the linear equation

$$-\nabla \cdot \partial B/\partial H^n \nabla(\delta\varphi) = \nabla \cdot B\,(H^n) \tag{6}$$

Equations (3) and (6) are sparse linear systems which can readily be solved with iterative methods such as GMRES without the need to form the matrix inverse. Because the operator matrices only depend on nearest cell neighbours in a structured grid, the number of non-zero matrix entries scales linearly with problem size, yielding to efficient computation of large systems. Convergence of the problem can be accelerated with the use of multigrid methods. In

ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-MOPS30

particular, we benefit from the use of the algebraic multigrid (AMG) preconditioner [7], which is designed for problems where operator matrix entries can vary by multiple orders of magnitude (in our case, caused by large variation in the ∂B/∂H term).

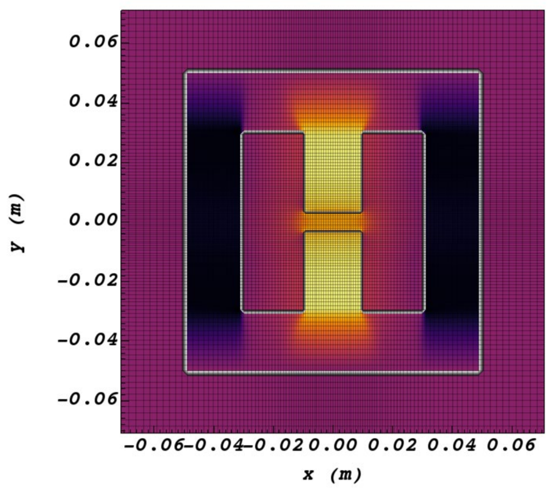

Figure 2: VSim simulation of a nonlinear, 2D H-Magnet on a variable mesh. Contours of vertical magnetic flux shown, Maximum B = 2.6T.

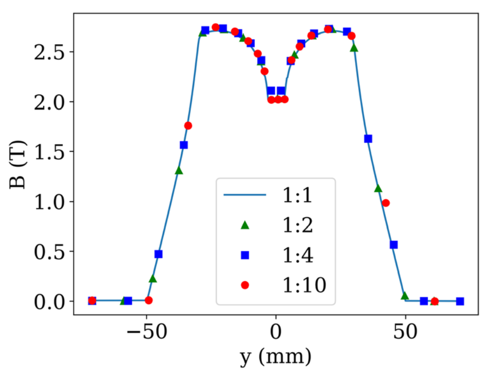

Figure 3: Centerline By of the H-Magnet shown above for varying ratios of maximum to minimum cell size.

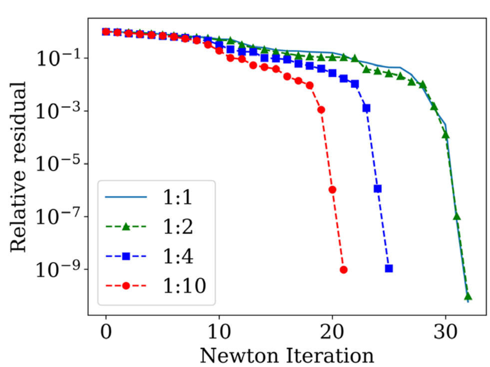

Figure 4: Relative residual as a function of nonlinear iteration steps for varying mesh ratios for the H-Magnet simulation.

To implement the solution of the above equations, we used Tpetra [8], a package within Trilinos that supports linear algebra for large, distributed systems using MPI, with options for hardware acceleration including GPUs. Tpetra includes classes for distributed vectors and sparse matrices, along with iterative linear solvers through the Belos package and algebraic multigrid preconditioners through MueLu. To solve the magnetostatic equations, we generate the B and J fields in Vorpal along with the grid and in/out fields for each magnetic object. The latter are used by the magnetic solver to determine where the magnetic permeability and flux values are to be applied for each magnetic material. The fields are translated into Tpetra vectors and the grid data is used to form the discrete divergence and curl operators in Eqs. (3) and (6). We then use the Belos pseudo block GMRES solver package to solve the linear systems in (3) and (6), repeating the Newton iteration step (6) until convergence of B.

RESULTS

Verification of the magnetic solver was performed with a 2D H-Magnet simulation. The magnet poles are surrounded by current-carrying coils represented by 62.5 A/m^2 current density fields oriented into and out of the simulation plane. The magnetic material was selected to be AISI 1006 carbon steel. The vertical magnetic field solution from the solver is shown in Fig. 2. A key feature of this work is enabling the solver to work on a Cartesian mesh with non-uniform cell spacing. This is critical for simulating large magnets with fine features, as selective resolution allows a dramatic decrease in overall problem size while maintaining physical accuracy. We demonstrated this for the magnetic solver by maintaining a fine resolution at the gap of the H-magnet while coarsening the grid outside. The vertical magnetic field profile is presented in Fig. 3 for varying ratios of minimum to maximum cell spacing, and compared with the results for a uniform high-resolution mesh. The magnetic field profiles in the gap are almost identical for all meshes. Figure 4 shows the nonlinear convergence for the H-Magnet under differing mesh size ratios, showing rapid convergence of the simulations as the outside mesh is coarsened.

Another motivation for using Tpetra as the basis for the magnetic solver is the ability to use double-precision integer indexing, allowing for problems with greater than 2^{31} (2.1 billion) elements. This enabled us to perform a magnetostatic solve with over 2^{31} field components, and in doing so, reach the desired resolution of 1mm for ANL's APS dipole magnet computation [9]. At that resolution, the problem had 2.26 billion degrees of freedom. Fig. 5 shows the vertical magnetic fields in the magnet; these are consistent with the lower resolution computations performed previously. We found convergence in 18 iterations of the nonlinear Newton solver, comparable to the convergence for much smaller problems, thus demonstrating excellent scaling of the magnetic algorithms for large systems. The entire simulation required 20.6 node-hours using 32 nodes / 1024 cores on the NERSC Cori cluster.

ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-MOPS30

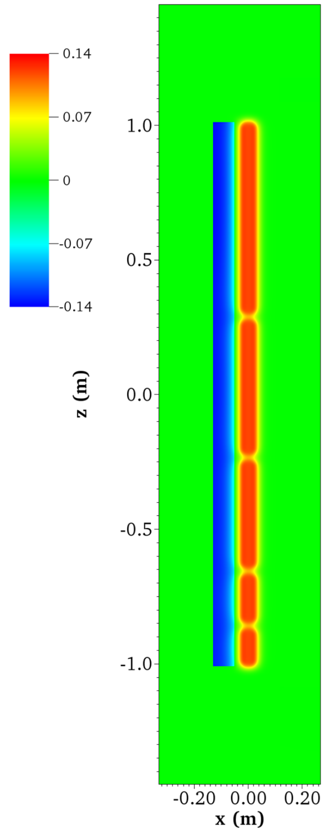

Figure 5: VSim simulation of the APS dipole magnet in Figure 1, showing vertical magnetic flux values in Tesla. Figure previously shown in [9].

GPU Acceleration

We performed GPU benchmark tests on a non-linear, anisotropic 3-D grain-oriented electrical steel (GOES) bar magnet problem. We first compared the timings of the nonlinear iteration loop on a single AMD EPYC 7302 CPU and NVIDIA A100 GPU as the domain size was increased from 50^3 (0.375M degrees of freedom) to 150^3 (10.1M degrees of freedom). On average, each simulation required 5 nonlinear iteration loops and 23 total Krylov iterations in the linear solves. Overall, the software is very performant; the entire nonlinear iteration loop for the 150^3 cell problem took 33.9 seconds on a single GPU. Figure 6 shows the total per-iteration timings as a function of problem size. The GPU provides approximately a factor of two speedup over the CPU for sufficiently large problems. The step time also increases linearly with problem size for both CPU and GPU, which is a key feature for efficient solves of very large problems. The GPU speedup can be further improved by a different choice of linear solver since Belos is not optimized for GPU [10]; moving to a more performant solver such as Hypre should provide overall speedups of 8x or beyond [11,12]. The evaluation of B(H) and $\partial B/\partial H$ takes up a small fraction of the overall solve time but also showcases the largest GPU speedup. The B-H interpolation for each cell is handled independently by a single GPU thread; on an NVIDIA A100 GPU, this enables the simultaneous evaluation of 221,000 cells. Figure 7 shows the per-DOF evaluation time of $\partial B/\partial H$ for the 3D problem. For the largest problem tested, the GPU provided a 24x speedup over the CPU for a time of 0.37 microseconds, or 0.005% of the total Newton iteration time.

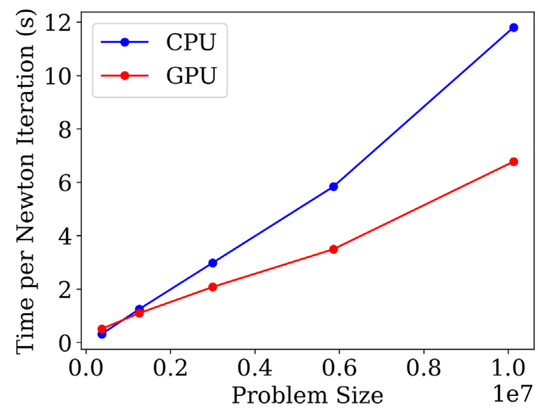

Figure 6: Wall clock time for each nonlinear iteration of the 3D magnet simulation as a function of problem size.

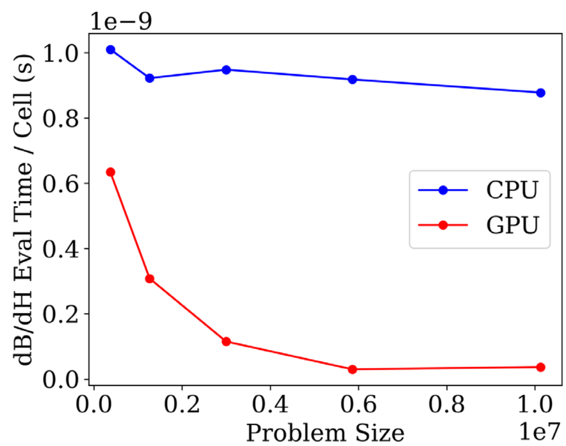

Figure 7: Global dB/dH evaluation time for 3D nonlinear magnet problem normalized by problem size.

CONCLUSION

This work demonstrated the implementation of an efficient, highly scalable magnetostatic solver into the VSim software. The solver is able to simulate highly nonlinear, anisotropic magnetic systems with billions of degrees of freedom while taking advantage of GPU acceleration through the Tpetra linear solver framework. As we continue development of the software, we intend to add additional magnetic physics features such as eddy currents and hysteresis, and pursue further GPU optimization.

ACKNOWLEDGEMENTS

This work supported by US DOE Office of Basic Energy Science Grant DE-SC0018515. Computing time was provided by the National energy Research Scientific Computing Center (NERSC).

ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-MOPS30

REFERENCES

[1] S. Henderson, "Status of the APS upgrade project," in *Proceedings of the 6th International Particle Accelerator Conference*, Richmond, VA, USA, May 2015, pp. 1791–1793.

[2] M. Borland *et al.*, "Lattice design challenges for fourth-generation storage-ring light sources," *J. Synchrotron Radiat.*, no. 21, p. 912, 2014.

[3] M. Jaski *et al.*, "Magnet designs for the multi-bend achromat lattice at the Advanced Photon Source," in *Proceedings of the 6th International Particle Accelerator Conference*, Richmond, VA, USA, 3–8 May 2015, pp. 3260–3263.

[4] M. Jaski, Private communication.

[5] Tech-X Corp. – VSim, https://www.txcorp.com/vsim

[6] C. Nieter and J. R. Cary, "VORPAL: A versatile plasma simulation code," *J. Comput. Phys*., vol. 196, p. 538, 2004.

[7] P. Vaněk, J. Mandel, and M. Brezina, "Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems," *Computing*, vol. 56, pp. 179–196, 1996.

[8] "Tpetra: Sparse Parallel Linear Algebra", `https://trilinos.github.io/tpetra.html`

[9] B. Cowan and M. de Messieres. "High-Performance Magnetostatic Solvers", presented at the 2019 North American Particle Accelerator Conference, Lansing, MI, USA, Sep. 2019, unpublished

[10] J. A. Loe, C. A. Glusa, I. Yamazaki, E. G. Boman, and S. Rajamanickam "Experimental Evaluation of Multiprecision Strategies for GMRES on GPUs", 2021. `doi:10.48550/arXiv:2105.07544v1`

[11] T. Powis, private communication, Apr. 2024

[12] HYPRE: Scalable Linear Solvers and Multigrid Methods, `https://computing.llnl.gov/projects/hypre-scalable-linear-solvers-multigrid-methods`