INL/JOU-23-72203-Revision-0

The MOOSE electromagnetics module

February 2024

Casey T Icenhour, Alexander D Lindsay, Cody J Permann, Richard C Martineau, David L Green, Steven C Shannon

Changing the World's Energy Future

INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, LLC

INL/JOU-23-72203-Revision-0

The MOOSE electromagnetics module

Casey T Icenhour, Alexander D Lindsay, Cody J Permann, Richard C Martineau, David L Green, Steven C Shannon

February 2024

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517, DE-AC07-05ID14517

SoftwareX 25 (2024) 101621

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

The MOOSE electromagnetics module

Casey T. Icenhour [a,b,*], Alexander D. Lindsay [a], Cody J. Permann [a], Richard C. Martineau [c], David L. Green [d], Steven C. Shannon [b]

[a] Idaho National Laboratory, Computational Frameworks Department, Idaho Falls, ID 83415, United States of America
[b] North Carolina State University, Department of Nuclear Engineering, Raleigh, NC 27695, United States of America
[c] Sawtooth Simulation, LLC, Idaho Falls, ID 83402, United States of America
[d] Commonwealth Scientific and Industrial Research Organization, Melbourne, Australia

ARTICLE INFO

Keywords:
Electromagnetics
Simulation
Multiphysics
MOOSE

ABSTRACT

The Multiphysics Object-Oriented Simulation Environment (MOOSE) electromagnetics module has been developed to increase MOOSE physics module capabilities, enabling standalone and coupled computational electromagnetics within the MOOSE multiphysics ecosystem. The module is actively being utilized in the areas of plasma physics and advanced manufacturing, and it currently provides initial demonstrated capability in multi-dimensional, complex-valued electromagnetic wave propagation, electrostatic contact, reflection and transmission, and electromagnetic eigenvalue problems. Two-dimensional wave propagation and one-dimensional wave reflection and transmission are showcased as examples in this work. The modularity, parallelism, and plug-in infrastructure for custom future development is inherited from MOOSE itself, and the module can be used with both MOOSE-based and external codes, giving great flexibility.

Code metadata

Current code version	2023–11–08
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-23-00308
Legal Code License	GNU LGPL 2.1
Code versioning system used	git
Software code languages, tools, and services used	C++, python
Compilation requirements, operating environments & dependencies	C++17 compliant compiler Arch: 64-bit x86 or Apple Silicon Memory: 16GB+ Disk: 30GB+ OS: macOS 10.13+, Linux (POSIX) Deps: MPI, PETSc, Hypre, libMesh
If available Link to developer documentation/manual	https://mooseframework.inl.gov/modules/electromagnetics/
Support email for questions	https://github.com/idaholab/moose/discussions

1. Motivation and significance

Electromagnetism plays a critical role in energy, communications, advanced manufacturing, plasma physics, and many other disciplines. Further, accurate and flexible computational electromagnetics (CEM) tools are vital to both model and understand cutting-edge technologies in these technical areas. Models and simulations requiring CEM can be standalone, such as when assessing an antenna geometry for far-field signal transmission and performance. However, many applications of CEM are inherently multiphysics problems, where there is non-linear feedback between the various physics relevant to the performance or function of the system being modeled. As an example, within a low temperature plasma physics simulation, multiple interconnected

* Corresponding author at: Idaho National Laboratory, Computational Frameworks Department, Idaho Falls, ID 83415, United States of America.
E-mail address: casey.icenhour@inl.gov (Casey T. Icenhour).

https://doi.org/10.1016/j.softx.2023.101621
Received 8 November 2023; Accepted 18 December 2023
Available online 26 December 2023

physics of interest exist: electromagnetic power coupling to the feed gas and creating ionized species, diffusion of ionic and neutral species of interest, interactions of those species (such as collisions and chemical reactions), heat conduction, erosion, and thermo-mechanical effects on and within surrounding structures, etc. Within the low temperature plasma community, predictive capabilities (including CEM) have been identified in multiple decadal roadmaps performed by the National Academies as "the highest level of challenge and the highest potential return" [1]. Further, these institutions found that it is important to make "state-of-the-art computations accessible to researchers who are not computational experts" [2].

It was this—the importance of having a robust and flexible CEM solver that retains accessibility—that prompted the creation of the MOOSE electromagnetics module (MOOSE-EMM). MOOSE is an open source, C++-based multiphysics framework designed to solve highly-coupled systems of equations describing various kinds of physical processes [3]. Furthermore, multiphysics simulations can be performed tightly-coupled within the same input file or more loosely coupled within connected sub-applications (with in-memory transfers of information as necessary to perform calculations) [4]. This approach enables multiscale simulation, as well as use of physics-based acceleration methods [5].

The MOOSE project was started to support the modeling and simulation needs of the nuclear energy and fuels community [4,6], but has since developed capability in plasma liquid interactions [7], molten salt reactors [8], geochemistry [9], and sintering [10], amongst many other technical areas. However, the wider MOOSE ecosystem was unable to take easy advantage of CEM models and methods which were native to the framework and also directly couplable to established MOOSE-based codes. The MOOSE-EMM serves to fill that gap. The module is intended to be a general-purpose toolkit for CEM needs inside of the entire MOOSE ecosystem that can also be utilized as a standalone CEM solver in its own right.

2. Software description

The MOOSE-EMM is contained within the MOOSE repository and therefore follows its system design. MOOSE contains basic interfaces to the various components of a simulation, such as partial differential equation (PDE) terms (known as "Kernels"), boundary conditions, initial conditions, interface conditions, executioners/solvers, preconditioners, and postprocessors [3]. The module operates within this set of interfaces for its capabilities. It is developed to a high software quality standard alongside MOOSE and uses an agile development methodology. All proposed changes are reviewed by a designated change control board and tested across multiple operating systems (Linux, MacOS) and architectures (x86 and ARM64) using a custom continuous integration (CI) system, called the Continuous Integration, Verification, Enhancement, and Testing (CIVET) system [11]. Code documentation is stored alongside the source code and is built and posted online during the CI process. The version of the MOOSE-EMM discussed here corresponds to MOOSE release 2023-11-08.

2.1. Software architecture

The MOOSE-EMM utilizes the base infrastructure of the MOOSE framework, and thus inherits its core dependencies, architecture, and workflows. A flowchart of MOOSE (outlining the location of MOOSE-EMM functionality between the module and framework) is shown in Fig. 1. As core MOOSE dependencies, PETSc [12] provides the systems related to matrix construction, solution methods, and preconditioning. Also, libMesh [13] provides finite element (FE) basis function support, the Templated Interface to Message Passing Interface (TIMPI) system, automatic differentiation [14], and mesh input/output (I/O). The MOOSE-EMM also utilizes higher-level components of each dependency as necessary to solve problem types relevant to CEM (such as the eigenvalue solver within PETSc or the vector FE basis functions within libMesh).

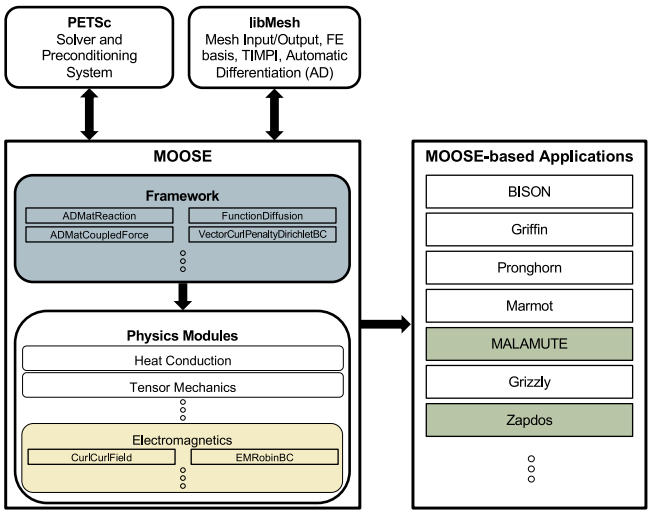


Fig. 1. Flowchart for MOOSE, MOOSE-based applications, and principal dependencies. Code objects used in the MOOSE-EMM exist in the framework (blue) and the module itself (yellow), combining to provide full capability in applications that have enabled the module (green). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The capabilities and code components that make up the MOOSE-EMM are split between those in the framework and those in the module itself. Those components placed within the framework were deemed sufficiently generic during module development to serve as possible building blocks for future framework and physics module capability or were present in the framework as a result of prior simulation needs. All MOOSE-EMM standalone simulations can utilize all objects within the core framework, and, similarly, all MOOSE-based applications that enable the MOOSE-EMM can utilize both framework capability as well as those provided by the module (via activation in their `Makefile`). The ability to combine and share objects between disparate physics modules is a great benefit of unified frameworks in code maintenance and flexibility.

2.2. Software functionalities

Due to the split in the MOOSE-EMM code base, this section will outline module capabilities that derive from the framework itself, and capabilities that solely exist within the module as of this publication.

2.2.1. Capability within the MOOSE framework

New shared framework capabilities included a new type of forcing term, *ADMatCoupledForce*, containing the ability to impose a forcing term in the PDE of interest corresponding to the contribution of a coupled scalar variable. In an example of code-reuse and code flexibility, another new framework capability included the *ADMatReaction* object, which enables reaction-style consumption/production terms given an arbitrary material coefficient. This object existed in a modified form in the MOOSE phase field module [15], as well as independently in the MOOSE-EMM. Finally, the Laplacian operator was specialized in a new way to take a diffusivity coefficient arbitrarily defined by a function. The function can be defined with respect to spatial position and time as well as coupled simulation variables. Other modifications to MOOSE code to facilitate module functionality included promoting a "test" boundary condition, *VectorCurlPenaltyDirichletBC*, to a regular member class of MOOSE so that the desired value of $\nabla\times\vec{u}$ at a boundary could be imposed using a penalty method. This was particularly useful when seeking a high level of code coverage and testing. For users, this can serve as an intermediate boundary condition when building more complicated simulations.

2.2.2. Capability within the electromagnetics module

The remainder of the module, and those components most closely related to CEM problems, resides within the repository at `moose/modules/electromagnetics`. Notable components of module capability (particularly those related to our illustrative examples in Section 3) are described here. At the core are terms associated with a vector wave equation, shown in transient form in Eq. (1) with a generic complex vector field, $\vec{u}$, and scalar coefficients, a and b (dependent on coupled spatial, temporal, and user-defined parameters given by the vector $\vec{r}$).

$$\nabla \times \nabla \times \vec{u} + a(\vec{r})\frac{\partial^2 \vec{u}}{\partial t^2} = -\left(b(\vec{r})\vec{F}\right) \tag{1}$$

This expression can be modeled in MOOSE-EMM using the *CurlCurlField*, *VectorSecondTimeDerivative*, and *VectorCurrentSource* objects. This form of the vector wave equation is used in the dipole antenna illustrative example presented in Section 3.2. A time-harmonic form of this equation is also available for frequency domain CEM simulations.

Another notable capability includes scalar and vector Robin-style boundary condition formulations, *EMRobinBC* and *VectorEMRobinBC*, for use as ports for field launching and absorbers for truncating time-harmonic simulation domains. Transient problems also have access to the focused *VectorTransientAbsorbingBC* object, which provides a pure absorber. Use of this object can be seen in Section 3.2 when truncating an open simulation domain.

Due to research interest in the area of advanced manufacturing, the *ElectrostaticContactCondition* object exists to impose current continuity and contact conductance across a boundary formed between two dissimilar materials, where a potential discontinuity would occur. This contact condition was based on the work of Cincotti [16], and is utilized in the MOOSE-based code MALAMUTE for electric field assisted sintering (EFAS) problems [17].

3. Illustrative examples

As described in Section 2, the MOOSE electromagnetics module can be operated as a standalone electromagnetics code in multiple dimensions, in both time-harmonic (frequency domain) or transient modes, and as an electromagnetic eigenvalue solver. The following examples will describe use cases for time-harmonic and transient modes in both 1-D and 2-D.

3.1. Reflection coefficient of a 1-D metal-backed dielectric slab in the frequency domain

The first example of module usage focuses on a scalar field scenario in 1-D, that of a uniform plane wave impinging on the surface of a nonhomogeneous dielectric slab bonded to a metal backplane, as described by Jin [18]. The goal of the benchmark is to accurately determine the power reflected by the slab due to the interaction with the electric plane wave, as compared to an analytic solution. The geometry for this example is shown in Fig. 2.

The slab has a thickness L, an electric permittivity $\epsilon = \epsilon_r \epsilon_0$ (where the subscript r denotes the relative permittivity coefficient and 0 the quantity in free space), and a relative magnetic permeability $\mu = \mu_r \mu_0$. Both material properties are functions of position within the slab region. The medium beyond the slab and metal backplane is free space ($\epsilon_r = \mu_r = 1$). An E_z-polarized plane wave is the incoming wave, and can be represented in a general sense by a complex exponential form given by Eq. (2). There, E_0 is the magnitude of the incident field, k_0 is the wavenumber ($2\pi/\lambda$, where λ is the wavelength), $j = \sqrt{-1}$, and θ is the incidence angle of the wave.

$$E_z^{\text{inc}}(x, y) = E_0 e^{jk_0 x\cos\theta - jk_0 y\sin\theta} \tag{2}$$

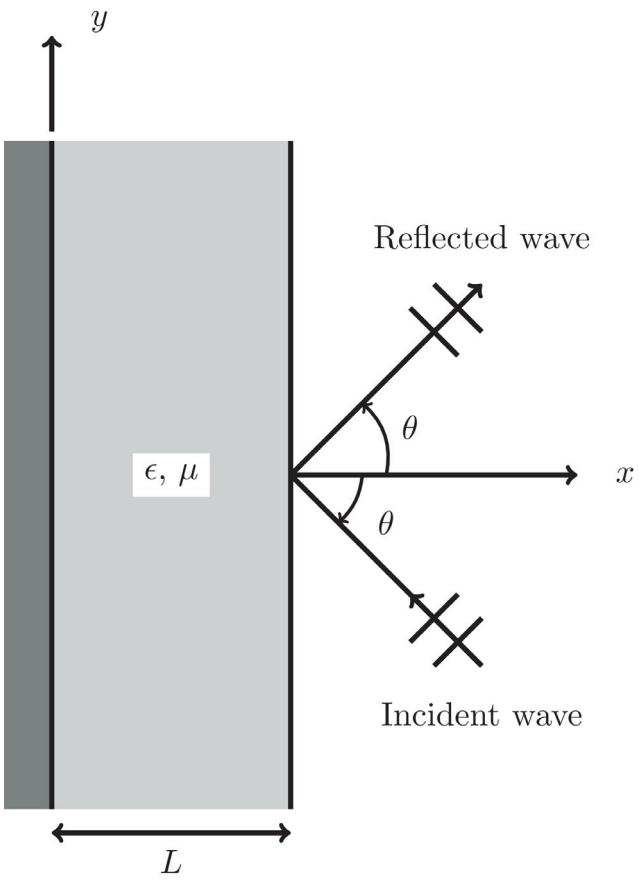


Fig. 2. Geometry for 1-D slab example.

Table 1
Constant model parameters for the slab reflection example.

Parameter (unit)	Value
Wave frequency (MHz)	20
Wavelength, λ (m)	15
Wavenumber, $k_0 = 2\pi/\lambda$ (1/m)	0.4189
Slab thickness, $L = 5\lambda$ (m)	75
Incident wave magnitude, E_0 (V/m)	1

The simplified scalar wave equation governing the electric field in the frequency domain considering the incoming wave form, is given by Eq. (3).

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{\mu_r}\frac{\mathrm{d}E_z}{\mathrm{d}x}\right) + k_0^2\left(\epsilon_r - \frac{1}{\mu_r}\sin^2\theta\right)E_z = 0 \tag{3}$$

Boundary conditions on the electric field within the dielectric are given for the conducting backplane as $E_z(0, y) = 0$ and for the slab side of the external interface (denoted at $x = L - 0$) by Eq. (4). The former condition is implemented by the MOOSE *DirichletBC* object with the latter implemented by the electromagnetics module *EMRobinBC* object.

$$\left[\frac{1}{\mu_r}\frac{\mathrm{d}E_z}{\mathrm{d}x} + jk_0\cos\theta E_z(x)\right]_{x=L-0} = 2jk_0\cos\theta E_0 e^{jk_0 L\cos\theta} \tag{4}$$

The relative electric permittivity in the slab is defined as $\epsilon_r = 4 + (2 - j0.1)\left(1 - \frac{x}{L}\right)^2$ with the relative magnetic permeability defined as $\mu_r = 2 - j0.1$. Other constant model parameters are defined in Table 1.

The reflection coefficient is calculated within MOOSE using the resulting 1-D electric field variable as a *Postprocessor* object, named *ReflectionCoefficient*. This object evaluates the expression in Eq. (5) and assumes an E_z-polarized plane wave as used in this benchmark. Because

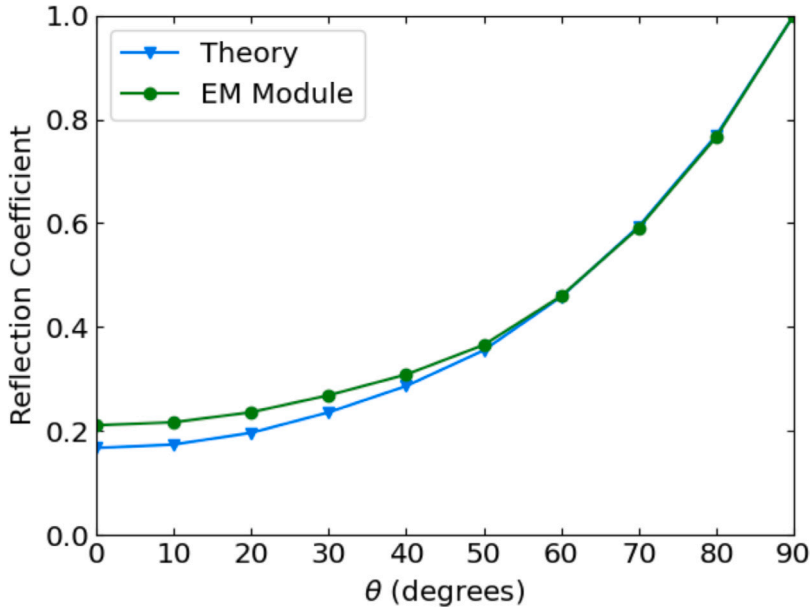


Fig. 3. Reflection coefficient (power) as a function of wave incident angle for the slab reflection example.

wave power is proportional to the magnitude of the wave squared, the reflection coefficient reported in Fig. 3 representing the percentage of reflected power is $R_{\text{power}} = |R_{\text{wave}}|^2$.

$$R_{wave} = \frac{E_z(x = L) - E_0 e^{jk_0 L\cos\theta}}{E_0 e^{-jk_0 L\cos\theta}} \tag{5}$$

The model was swept over a range of angles from $\theta = 0°$ to $\theta = 90°$, and the result compared to the analytic solution presented in Jin [18] is shown in Fig. 3. There is good agreement at higher incident angles of incidence but deviation at smaller ones, which was also seen by Jin. They attributed this to the magnitude of the coefficient in the second term of Eq. (3) decreasing as θ increases. In other words, there will be slower field variation in the x-direction as we approach $\theta = 90°$, giving more consistent results between the code and the analytic result. A 1-D model might not be sufficient to fully capture all effects here.

3.2. Transient half-wave dipole antenna: 2-D radiation intensity pattern

Next is a vector field example that focuses on a well-known benchmark antenna structure: the half-wave dipole antenna. In this scenario, a vertically-oriented antenna structure is excited by a 1 GHz signal in infinite vacuum. The transmitted complex electric field is emitted in a characteristic radiation pattern before being absorbed by a first-order representation of the Sommerfeld radiation condition [18,19]. To optimize transmission, the boundary of the problem domain is placed five wavelengths away from the antenna centerpoint.

A half-wave dipole antenna has a geometry shown in Fig. 4. The oscillating voltage signal between the two antenna conductors produces a current standing wave distribution along the length of the dipole structure. The antenna resonates (and transmits) when the applied voltage has a wavelength equal to two times the dipole length. Thus, the resonant frequency is given by $f_r = c/2L$, where c is the speed of light and L is the length of the antenna structure. The parameters for the antenna geometry being modeled is shown in Table 2.

To confirm that the simulation is performing as intended, the result is compared to the far-field radiation intensity pattern for the antenna under inspection. The far-field radiation pattern for a half-wave dipole antenna is given in [21], and a representation of the directional power intensity is shown in Fig. 5(a) for a vertically-oriented antenna. This shows that a half-wave dipole antenna has null, or zero intensity, regions in either direction parallel to its length, with maximum intensity at $\theta = 90°$ and $\theta = 270°$. The beamwidth, or the region outside of which has a signal which is below 3 dB, or half power, compared to the peak intensity, is around 90°.

Table 2
Half-wave dipole antenna geometry parameters.

Parameter (unit)	Value
Resonant frequency (GHz)	1
Wavelength, λ (m)	0.3
Antenna length, L (m)	0.15
Antenna feed gap, $\lambda/20$ (m)	0.015
Domain radius, 5λ (m)	1.5

Using the transient complex electric vector field wave equation in the frequency domain, an exciting complex electric field ($|E_y| = \sqrt{2}$ V/m) was applied to the surface of the antenna and the radiation field was simulated using the EM module. The steady state results are shown in Fig. 5(b), with good qualitative agreement compared to the theoretical far-field radiation pattern.

4. Impact

The MOOSE-EMM serves to provide a general CEM code for both coupled and standalone physics simulation requirements. It seeks to be flexible, modular, expandable, and accessible to scientists and engineers that are not computationally focused by providing sensible interfaces and comprehensive documentation. By building on the MOOSE framework, the MOOSE-EMM inherits the computational efficiency and parallelism of MOOSE [3], and can natively be used in any MOOSE-based application with the simple change of an application Makefile. The following discussion describes the initial impact of MOOSE-EMM in two areas: EFAS advanced manufacturing and low-temperature plasma physics.

In the MALAMUTE code [17], the MOOSE-EMM was used to compute the electrostatic contact conditions for a direct current-driven (DC) EFAS system designed after the work of Cincotti [16]. In this scenario, the EFAS system contains a combination of steel alloy rams and graphite forms within which a powdered alloy is pressed and then sintered using the applied electrical current. The MOOSE-EMM allows the constraint of the potential variable such that the current is continuous across each steel–graphite interface but the potential is allowed

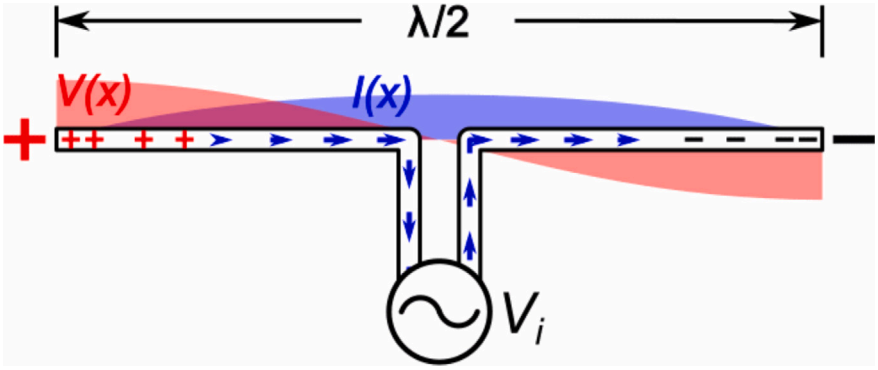
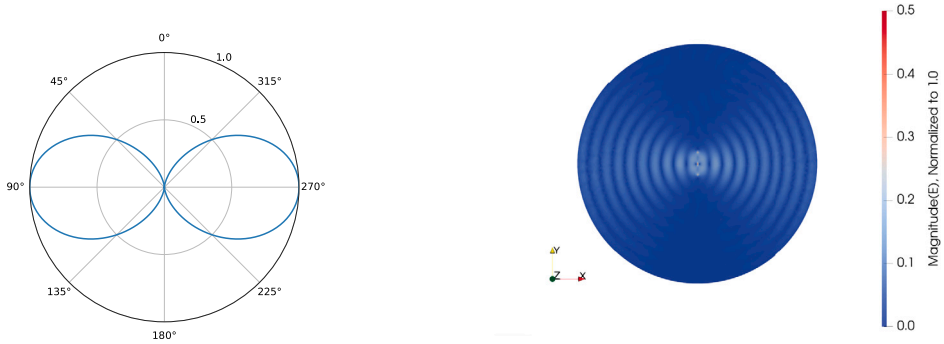


Fig. 4. Half-wave dipole antenna diagram showing voltage and current [20].



(a) Theoretical

(b) Simulated (normalized to 1.0, and saturated for ease of view)

Fig. 5. Theoretical vs. simulated steady state radiation intensity pattern of a vertically-oriented half-wave dipole antenna driven at 1 GHz.

to remain discontinuous, consistent with the electrical properties of each material. In Ref. [22], the MOOSE-EMM is being used to expand the capability of a MOOSE-based code, Zapdos, developed to study electrostatic, capacitively-coupled, low temperature plasma discharges. By adding Maxwell's equation solve capabilities to Zapdos, a greater number of plasma and plasma processing scenarios can be studied, including high frequency and inductively coupled discharges. The modularity of the MOOSE-EMM allows for two independent solvers—the legacy electrostatic solver packaged alongside Zapdos and the CEM solver contained within the MOOSE-EMM—to be used as desired by the end user.

Beyond these two applications, the MOOSE-EMM development team intends to target physics simulations that require expanded CEM capabilities, such as those related to fusion energy (magnet coil safety evaluation, plasma kinetics and plasma material interaction) and advanced manufacturing more broadly (microwave material treatment and radio-frequency EFAS). It is anticipated that the proven multiphysics capability of MOOSE, now with the MOOSE-EMM, can provide an attractive option for researchers working in these domains.

5. Conclusions

The MOOSE-EMM provides a base level of electromagnetics simulation capability to the MOOSE framework, usable in a standalone configuration as well as a library alongside MOOSE-based or external codes. Principles of flexibility, modularity, and code re-use allow the MOOSE-EMM to be an easily navigable code for computational novices, and extensive documentation provides clarity and examples for developing custom simulations. Further, the core software is developed to a nuclear quality assurance software standard and tested constantly, which provides even more confidence in code stability [11]. As an open source software project, the MOOSE-EMM is also a living tool, open and welcoming to the community to change and further develop.

In this work, the module has demonstrated capability for both open wave propagation problems as well as current flow within solid materials. Complex-valued field calculations, electromagnetic eigenvalue problems, and reflection and transmission calculations can also be performed, with documentation contained within the code repository. Steady state and transient modes of operation are available, and the code is developed agnostic of spatial dimension. This flexibility is leading to promising impact in the fields of low temperature plasma physics and advanced manufacturing, where the MOOSE-EMM is expanding legacy capability and enabling the study of prospective experimental designs and material configurations.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Casey Icenhour reports financial support was provided by United States Department of Energy Office of Science. Casey Icenhour reports financial support was provided by National Science Foundation.

Data availability

No data was used for the research described in the article.

Acknowledgments

The authors would like to thank Stephanie Pitts, David Andrš, Larry Aagesen, Guillaume Giudicelli, Logan Harbour, Derek Gaston, and John Mangeri for their support, thoughtful discussions, and constructive third-party code and documentation review during the production of the MOOSE electromagnetics module.

Funding

This work was funded by the United States Department of Energy (U.S. DOE) Office of Science Graduate Student Research program, administered by the Oak Ridge Institute for Science and Education for the U.S. DOE under contract number DE-SC0014664; the National Science Foundation program in Software Infrastructure for Sustained Innovation under Grant No. 1740300; the Idaho National Laboratory (INL) Graduate Fellowship Program; and the INL Laboratory Directed Research and Development program. The latter two sources are supported by the U.S. DOE under Contract No. DE-AC07-05ID14517.

This work made use of INL's High Performance Computing systems located at the Collaborative Computing Center and supported by the U.S. DOE Office of Nuclear Energy and the Nuclear Science User Facilities under Contract No. DE-AC07-05ID14517.

The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

References

[1] National Research Council. Plasma science: advancing knowledge in the national interest. Washington, DC: The National Academies Press; 2007, http://dx.doi.org/10.17226/11960.

[2] National Academies of Sciences, Engineering, and Medicine. Plasma science: enabling technology, sustainability, security, and exploration. Washington, DC: The National Academies Press; 2020, http://dx.doi.org/10.17226/25802.

[3] Lindsay AD, Gaston DR, Permann CJ, Miller JM, Andrš D, Slaughter AE, et al. 2.0 - MOOSE: Enabling massively parallel multiphysics simulation. SoftwareX 2022;20:101202. http://dx.doi.org/10.1016/j.softx.2022.101202.

[4] Gaston DR, Permann CJ, Peterson JW, Slaughter AE, Andrš D, Wang Y, et al. Physics-based multiscale coupling for full core nuclear reactor simulation. Ann Nucl Energy 2015;84:45–54. http://dx.doi.org/10.1016/j.anucene.2014.09.060.

[5] DeChant C, Icenhour C, Keniley S, Lindsay A, Curreli D, Shannon S. Verification and validation of the open-source plasma fluid code: Zapdos. Comput Phys Comm 2023;291:108837. http://dx.doi.org/10.1016/j.cpc.2023.108837.

[6] Williamson RL, Hales JD, Novascone SR, Pastore G, Gamble KA, Spencer BW, et al. BISON: A flexible code for advanced simulation of the performance of multiple nuclear fuel forms. Nucl Technol 2021. http://dx.doi.org/10.1080/00295450.2020.1836940.

[7] Lindsay AD, Graves DB, Shannon SC. Fully coupled simulation of the plasma liquid interface and interfacial coefficient effects. J Phys D: Appl Phys 2016;49(23):235204. http://dx.doi.org/10.1088/0022-3727/49/23/235204.

[8] Lindsay A, Ridley G, Rykhlevskii A, Huff K. Introduction to moltres: An application for simulation of molten salt reactors. Ann Nucl Energy 2018;114:530–40. http://dx.doi.org/10.1016/j.anucene.2017.12.025.

[9] Wilkins A, Green CP, Ennis-King J. PorousFlow: A multiphysics simulation code for coupled problems in porous media. J Open Source Softw 2020;5(55):2176. http://dx.doi.org/10.21105/joss.02176.

[10] Biswas S, Schwen D, Singh J, Tomar V. A study of the evolution of microstructure and consolidation kinetics during sintering using a phase field modeling based approach. Extreme Mech Lett 2016;7:78–89. http://dx.doi.org/10.1016/j.eml.2016.02.017.

[11] Slaughter AE, Permann CJ, Miller JM, Alger BK, Novascone SR. Continuous integration, in-code documentation, and automation for nuclear quality assurance conformance. Nucl Technol 2021;207(7):923–30. http://dx.doi.org/10.1080/00295450.2020.1826804.

[12] Balay S, Abhyankar S, Adams MF, Benson S, Brown J, Brune P, et al. PETSc/TAO users manual. Tech. rep. ANL-21/39 - Revision 3.18, Argonne National Laboratory; 2022, http://dx.doi.org/10.2172/1893326, URL https://www.mcs.anl.gov/petsc.

[13] Kirk BS, Peterson JW, Stogner RH, Carey GF. libMesh: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations. Eng Comput 2006;22(3–4):237–54. http://dx.doi.org/10.1007/s00366-006-0049-3.

[14] Lindsay A, Stogner R, Gaston D, Schwen D, Matthews C, Jiang W, et al. Automatic differentiation in MetaPhysicL and its applications in MOOSE. Nucl Technol 2021;1–18. http://dx.doi.org/10.1080/00295450.2020.1838877.

[15] Schwen D, Aagesen L, Peterson J, Tonks M. Rapid multiphase-field model development using a modular free energy based approach with automatic differentiation in MOOSE/MARMOT. Comput Mater Sci 2017;132:36–45. http://dx.doi.org/10.1016/j.commatsci.2017.02.017.

[16] Cincotti A, Locci AM, Orrù R, Cao G. Modeling of SPS apparatus: Temperature, current and strain distribution with no powders. AIChE J 2007;53(3):703–19. http://dx.doi.org/10.1002/aic.11102.

[17] Icenhour C, Lindsay A, Pitts S, Aagesen L, Jiang W. idaholab/malamute: MOOSE application library for advanced manufacturing utilities (MALAMUTE). 2021, http://dx.doi.org/10.11578/dc.20230313.3, URL https://github.com/idaholab/malamute.

[18] Jin J-M. The finite element method in electromagnetics. 3rd Ed.. Hoboken, New Jersey, USA: John Wiley & Sons; 2014.

[19] Sommerfeld A. Die Greensche Funktion der Schwingungslgleichung. Jahresbericht der Deutschen Mathematiker-Vereinigung 1912;21:309–52, URL http://eudml.org/doc/145344.

[20] Chetvorno. Animation of standing waves on a dipole antenna. 2023, Wikimedia, Available at https://w.wiki/6Vkw. [Accessed 26 March 2023].

[21] Silver S. Microwave antenna theory and design. Stevenage, UK: The Institute of Engineering and Technology; 1984, http://dx.doi.org/10.1049/PBEW019E.

[22] DeChant C, Icenhour C, Keniley S, Lindsay A, Gall G, Hizon KC, et al. Verification methods for drift-diffusion reaction models for plasma simulations. Plasma Sources Sci Technol 2023;32(4):044006. http://dx.doi.org/10.1088/1361-6595/acce65.