Lawrence Berkeley National Laboratory

Title
A massively parallel time-domain coupled electrodynamics–micromagnetics solver

Authors
Yao, Zhi
Jambunathan, Revathi
Zeng, Yadong
et al.

A Massively Parallel Time-Domain Coupled Electrodynamics-Micromagnetics Solver

Zhi Yao[1], Revathi Jambunathan[1], Yadong Zeng[2] and Andy Nonaka[1]

Abstract

We present a new, high-performance coupled electrodynamics-micromagnetics solver for full physical modeling of signals in microelectronic circuitry. The overall strategy couples a finite-difference time-domain (FDTD) approach for Maxwell's equations to a magnetization model described by the Landau-Lifshitz-Gilbert (LLG) equation. The algorithm is implemented in the Exascale Computing Project software framework, AMReX, which provides effective scalability on manycore and GPU-based supercomputing architectures. Furthermore, the code leverages ongoing developments of the Exascale Application Code, WarpX, primarily developed for plasma wakefield accelerator modeling. Our novel temporal coupling scheme provides second-order accuracy in space and time by combining the integration steps for the magnetic field and magnetization into an iterative sub-step that includes a trapezoidal discretization for the magnetization. The performance of the algorithm is demonstrated by the excellent scaling results on NERSC multicore and GPU systems, with a significant (59x) speedup on the GPU using a node-by-node comparison. We demonstrate the utility of our code by performing simulations of an electromagnetic waveguide and a magnetically tunable filter.

1 Introduction

As microelectronic devices become increasingly integrated and miniaturized, new materials and technologies are being adopted into conventional circuit designs. Spintronics is a promising technology and an important candidate for upcoming computing architectures due to their capability of reducing energy consumption and providing non-volatile memory. Chappert et al. (2007) explained that in this new technology, instead of moving charges, spin orientation of magnetic materials, also known as magnetization, is used as the information carrier. Chappert et al. (2007) and Garello et al. (2013) also reviewed current applications including magnetic sensors and magnetic random access memory (MRAM) devices for computer architectures. Ultimately, "spin currents" could even replace charge currents for the transfer and processing of information, allowing faster and lower-energy operations, which further motivates advancing the field of spin electronics. However, to propel this nascent technology into mainstream microelectronics ecosystem, pre-fabrication analysis is critical to determine and optimize the performance of new designs. Modeling and simulations provide an accurate lens to guide the development and therefore are ubiquitous in industrial, academic, and laboratory settings. But existing software cannot accurately capture the very different physical characteristics that span over disparate spatial and temporal scales. Therefore, a trial-and-error approach is often used to test the performance of spintronic devices, thereby increasing the cost and time for development. In addition to spintronic technologies, other areas of innovation in next-generation microelectronics, including 5G application, quantum circuitry, and multi-ferroic systems, are also in pressing need for simulation tools

that can adequately capture interactions of multiple physics and accurately predict device performance. Thus, there is an active demand for improved fidelity of electromagnetic (EM) simulations via higher spatiotemporal resolution and/or enhanced numerical strategies to seamlessly couple different physical models used to describe relevant physics in various regions of the circuit.

Existing commercially-available simulation packages are generally not suitable to perform such modeling due to two major drawbacks. First, the software and algorithms are often proprietary, offering users limited customization of algorithms or incorporation of new physics required to model these new devices. Additionally, existing black-box algorithmic implementations may not inform users of potential assumptions that may break the validity of the approach for realistic devices. Second, scalability is limited to smaller clusters, rather than multicore/GPU-based exascale systems. One of the most widely-used EM software, Ansys HFSS (High Frequency Structure Simulator) package, offers a platform to solve versatile EM applications, with the functionality to read in user-defined magnetic permeability tensors. However, it does not allow users to tailor their own mathematical models to recover the dynamical nonlinear material behavior such as nonlinear magnetic properties. COMSOL Multiphysics software is another commercial

[1]Lawrence Berkeley National Laboratory, Berkeley, CA, USA
[2]University of Minnesota Twin Cities, Minneapolis, MN, USA

Corresponding author:
Zhi Yao, 1 Cyclotron Road, Mailstop 50A3111, Berkeley, CA 94720
Email: jackie_zhiyao@lbl.gov

package, which allows for customized mathematical models, however, it may not be easily scalable on GPUs, limiting its applicability. Therefore, many researchers are actively developing their own software.

Most software model the spin behavior of micromagnetic materials in isolation. Object Oriented MicroMagnetic Framework (OOMMF), developed by Donahue and Porter (1999), has been developed to simulate spin dynamics in complex structures. Recently Fu et al. (2015) have extended OOMMF with GPU implementation and has demonstrated 35x speed-up compared to CPUs. Other solvers that model micromagnetic material include mumax developed by Vansteenkiste et al. (2014) and PC micromagnetic simulator (Simulmag) developed by Oti (1997). The recent burgeoning of magnetic material-based computing and communication components has necessitated a close examination of the behavior of magnetic spins that are strongly influenced by the EM signals from the circuit. This study requires a hybrid solver coupling the EM signals with magnetic spin behavior, i.e., Maxwell equations with the Landau-Lifshitz-Gilbert (LLG) equation describing the spin precession dynamics. Various approaches have been used to study this coupled physics. Couture et al. (2017) have integrated a finite-element micromagnetic model and an integral-equation EM model to simulate the effect of eddy currents on magnetic spins. Venugopal et al. (2019) have built a micromagnetic algorithm which predicts nonlinear ferromagnetic resonance frequency (FMR) shift phenomenon under the effect of time-harmonic high-power microwave EM fields. Note that the magnetostatic assumption ($\partial \mathbf{D}/\partial t = 0$ in Ampere's law) applied by both Couture et al. (2017) and Venugopal et al. (2019) limits such tools in describing wave properties in circuits such as impedance matching. Efforts have also been made to implement fully dynamical EM waves ($\partial \mathbf{D}/\partial t \neq 0$), which can accurately model realistic devices. Yao et al. (2018) have developed an alternating-direction-implicit (ADI) based time-domain micromagnetic algorithm. Additionally, Vacus and Vukadinovic (2005) have implemented a 2D explicit Maxwell-LLG coupled algorithm, followed by Aziz (2009) proposing a full three-dimensional implementation.

In this paper, we present a new coupled fully-dynamic electrodynamics-micromagnetics solver that is fully open-source and portable from laptops to manycore/GPU exascale systems. The core solver in our code is a finite-difference time-domain (FDTD) implementation for Maxwell's equations that has been adapted to conditions found in micro-electronic circuitry. Thus we have incorporated spatially-varying material properties, boundary conditions, and external sources to model our target problems. We have also developed a new numerical coupling scheme to incorporate the LLG model of magnetization to allow for regions in space containing magnetic materials. The overall scheme is second-order accurate in space and time. In order to achieve portability and performance on a range of platforms, our code leverages the developments of two DOE Exascale Computing Project (ECP) code frameworks. First, the AMReX software library developed by Zhang et al. (2019, 2020) is the product of the ECP co-design center for adaptive, structured grid calculations. AMReX provides complete data structure and parallel communication support for massively parallel manycore/GPU implementations of structured grid simulations such as FDTD. Second, the WarpX accelerator code developed by Vay et al. (2018) is an ECP application code for modeling plasma wakefield accelerators and contains many features that have been leveraged by our code. These features include core computational kernels for FDTD, an overall time stepping framework, and I/O. Looking forward, we will take advantage of adaptive mesh refinement (AMR) support, embedded boundaries for advanced geometry, and machine learning workflows that are continually being updated and improved by the WarpX team.

The rest of this paper is organized as follows. In Section 2 we describe the equations that govern the involved physics. In Section 3, the details for our numerical approach are elaborated. In Section 4, we provide more details on the software framework, scaling studies, and a comparison of host vs. GPU performance. In Section 5 we describe three test problems that validate our approach and implementation. The first is a numerical convergence test to demonstrate the spatial and temporal order of accuracy of our coupled EM-micromagnetic solver with spatially inhomogeneous material properties. The second is an X-band air-filled waveguide which has known analytical solutions that are used to validate the EM propagation in confined structures. The third is a magnetically tunable filter realized by the waveguide with a yttrium iron garnet (YIG) thin slab inserted longitudinally which validates the coupling between the EM module and the LLG micromagnetics module. Finally, in Section 6 we summarize this work and discuss future directions.

2 Mathematical Model

We begin with the full-form of the dynamic Maxwell's equations, Ampere and Faraday's laws,

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}, \tag{1}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{2}$$

where, $\mathbf{D} = \epsilon \mathbf{E}$ is the electric displacement, $\mathbf{E}$ is the electric field, ϵ is the permittivity of the medium, $\mathbf{B}$ is the magnetic flux density, $\mathbf{H}$ is the magnetic field, and $\mathbf{J} = \sigma \mathbf{E} + \mathbf{J}_{\mathrm{src}}$ is the electric current density with conductivity, σ, and external source, $\mathbf{J}_{\mathrm{src}}$. The permittivity, ϵ is the product of vacuum permittivity, ϵ_0, and unit-less relative permittivity, $\epsilon = \epsilon_0 \epsilon_r$. Similarly, the permeability, μ, is the product of vacuum permeability, μ_0 and unit-less relative permeability, $\mu = \mu_0 \mu_r$. In non-magnetic materials, the constitutive relation between $\mathbf{B}$ and $\mathbf{H}$ is simply $\mathbf{B} = \mu \mathbf{H}$. In magnetic materials, permeability could be described using the constitutive relation $\mathbf{B} = \mu \mathbf{H} = \mu_0 (\mathbf{H} + \mathbf{M})$, where $\mathbf{M}$ is the magnetization (total magnetic moments per unit volume). It is easily derived from the above relations that $\mu_r = 1 + \mathbf{M}/\mathbf{H}$. Typically, the value of μ_r is close to 1 in non-magnetic materials due to their very low intrinsic magnetization. In contrast, as explained by Lax et al. (1962), the value of μ_r in magnetic materials is usually much larger than 1, especially in paramagnetic and ferromagnetic materials, due to large magnetization $\mathbf{M}$ values. Moreover, in magnetic materials, μ_r typically varies with frequency

and the field intensity, i.e. μ_r has a dispersive frequency spectrum and nonlinear dependency on $\mathbf{H}$. The reason is that the magnetization, $\mathbf{M}$, required to determine μ_r evolves and oscillates as a function of $\mathbf{H}$, which in turn is solved by the Maxwell's equations. Therefore, the value of the μ_r is not known a priori for magnetic materials and instead of simply specifying an approximate μ_r value as the user-defined material property, it is more precise to predict the macroscopic material permeability for magnetic materials. As a result, solving the evolution of magnetization $\mathbf{M}$ in magnetic materials is essential in our model to describe the dynamic material behavior under the effect of EM fields.

One of the most widely used equation of motion to describe the spinning electrons is the LLG equation, conceived by Landau and Lifshitz (1992). The LLG equation describing the evolution of magnetization, $\mathbf{M}$, is

$$\frac{\partial \mathbf{M}}{\partial t} = \mu_0 \gamma (\mathbf{M} \times \mathbf{H}_{\text{eff}}) + \frac{\alpha}{|\mathbf{M}|} \mathbf{M} \times \frac{\partial \mathbf{M}}{\partial t} + \boldsymbol{\tau}, \quad (3)$$

where γ represents the gyromagnetic ratio, with the value of $\gamma = -1.759 \times 10^{11}$ C/kg (note the negative sign indicating the opposite directions between the magnetic moment and the electron spin) for electrons. The term α is the Gilbert damping constant, with typical values on the order of 10^{-5} to 10^{-3} (note the positive sign indicating the decreasing precession angle of $\mathbf{M}$). It is also worth noting that the Gilbert damping constant, α, encompasses all the physical mechanisms that lead to a damped $\mathbf{M}$, namely, lattice vibrations, spin-spin relaxation, spin-orbit coupling, and others. We chose to use the LLG form as the governing equation for magnetization evolution because it describes the overall damping process adequately and can be introduced in the magnetic susceptibility expression by a simple mathematical procedure, summarized by Lax et al. (1962). $\mathbf{H}_{\text{eff}}$ in (3) is the effective magnetic field that drives the magnetic spins and can be decomposed into several terms accounting for the contributions from the oscillating EM field $\mathbf{H}$ (denoted as $\mathbf{H}_{\text{EM}}$ from now on) as well as external static bias $\mathbf{H}_0$, anisotropy $\mathbf{H}_{\text{ani}}$, and exchange coupling $\mathbf{H}_{\text{exch}}$. As the first demonstration of the model capability, we considered only the external DC bias $\mathbf{H}_0$ and the oscillating EM field $\mathbf{H}_{\text{EM}}$. At this point, it is straightforward to categorize the first two terms in the LLG equation: the first term, $\mu_0 \gamma (\mathbf{M} \times \mathbf{H}_{\text{eff}})$, is the torque term which generates an angular momentum vector tangential to the spinning orbit of $\mathbf{M}$; the second term, $(\alpha/|\mathbf{M}|)\mathbf{M} \times (\partial \mathbf{M}/\partial t)$, introduces a damping vector perpendicular to $\mathbf{M}$ and points toward the central axis of the $\mathbf{M}$ precession trajectory. Note that neither the torque term nor the Gilbert damping term influences the magnitude of the magnetization vector, $|\mathbf{M}|$. The additional "torque" term, $\boldsymbol{\tau}$, expresses physics that is unique to spintronic devices, and takes various forms depending on the scenarios, such as the spin-transfer-torque(STT)-based structures. In this work, which is aimed to provide demonstration of the code essential capability, we focus only on the framework for coupling magnetization with EM signals and omit the $\boldsymbol{\tau}$ term to reduce complexity of our validation tests. In future works, we will implement these terms to include more physics, specifically, $\mathbf{H}_{\text{ani}}$, $\mathbf{H}_{\text{exch}}$, $\boldsymbol{\tau}$, and so on, as would be needed to recover physics such as magnetic domain movement,

spin waves, STT, and others. In fact, the more complex spatial variation of the magnetization in the ferromagnets comes mainly from the these terms. Under the assumption of including only EM coupling and DC bias as mentioned above, the magnitude of $\mathbf{M}$ stays constant. Since in both RF and memory applications, magnetization is mostly biased to saturation, it is reasonable to use $|\mathbf{M}| = M_s$, where M_s is a spatially-dependent material property representing the saturation level.

In our numerical approach, we choose $\mathbf{H}$ as the primary connection between the EM module and micromagnetics module, as $\mathbf{H}$ affects both electric currents and magnetization as discussed previously. Thus, we use a modified version of Faraday's law that has $\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M})$ substituted in (2) in magnetic regions to give

$$\frac{1}{\mu_0} \nabla \times \mathbf{E} = -\frac{\partial \mathbf{H}}{\partial t} - \frac{\partial \mathbf{M}}{\partial t}. \quad (4)$$

and in non-magnetic regions,

$$\frac{1}{\mu} \nabla \times \mathbf{E} = -\frac{\partial \mathbf{H}}{\partial t} \quad (5)$$

To summarize, we have built the coupled model to include spatial inhomogeneity of materials, containing both magnetic and non-magnetic regions. In magnetic regions, our coupled algorithm solves both Maxwell's equations (1), (4) and the LLG equation (3). In non-magnetic regions, the simple constitutive relation $\mathbf{B} = \mu \mathbf{H}$ is applied and only Maxwell's equations (1) and (5) are solved. This is an important distinction as our algorithm is designed to model devices containing both magnetic and non-magnetic regions.

3 Numerical Approach

3.1 Spatial Discretization

We use a uniform Cartesian mesh with cell-sizes $\Delta x, \Delta y$, and Δz. We employ the standard Yee (1966) grid configuration for electrodynamics where the normal components of $\mathbf{H}$ fields are defined on cell-faces, and the tangential components of $\mathbf{E}$ and $\mathbf{J}$ are defined on the cell-edges as illustrated in Fig. 1. The magnetization, $\mathbf{M}$, is also defined on cell-faces, but with an important distinction. At each face, all three components of $\mathbf{M}$ are stored, as opposed to $\mathbf{H}$ and $\mathbf{B}$, where only the normal component is stored.

Using this approach, the curl of $\mathbf{H}$ can be simply and naturally represented with second-order accuracy at the locations of $\mathbf{E}$, and vice versa. At various points in the algorithm, transverse components of $\mathbf{H}$ are required on faces (e.g., H_y on an x-face). In these cases, we compute these values using a second-order four-point average of the nearest values. The material properties ($\epsilon, \mu, M_s, \sigma, \gamma, \alpha$) are specified through user-defined functions of space and/or time supplied at run-time using the function parser capability of the code. To compute the material properties, we first evaluate them from the parsed function point-wise at cell centers and then later interpolate these values to other locations on the grid as required using second-order stencils. By making this choice we are using the interpretation that any given computational cell can be thought of as a finite volume cell consisting of a material with given properties;
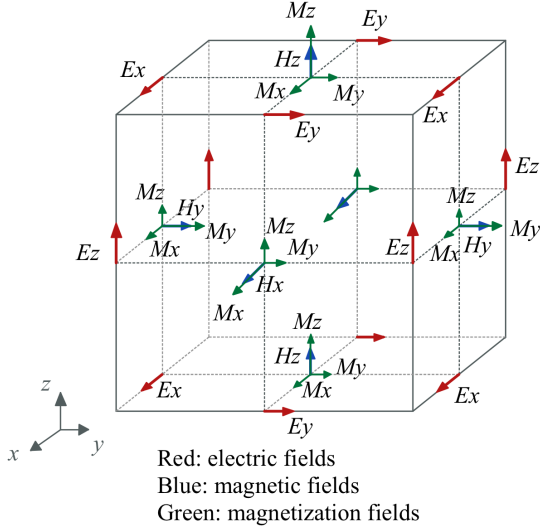
Figure 1. Yee's grid with normal magnetic field on cell faces, tangential electric field on cell edges, and all three components of the magnetization at each cell faces.

in the future as we consider more complex, non-grid-aligned geometries, we will consider the viability of using a volume fraction weighted parameterization of the material, and compare that approach to more advanced discretizations involving embedded boundaries. The algorithm as presented here requires σ and ϵ on edges where $\mathbf{E}$ and $\mathbf{J}$ are located, and γ, α, and M_s on faces where $\mathbf{M}$ and $\mathbf{H}$ are located. Note that we also parse the initial condition for $\mathbf{M}$ at cell centers and average these values to faces; in this way there is a consistent definition of the material property M_s and the initial field $\mathbf{M}$ that allows the initial condition $|\mathbf{M}| = M_s$ to hold discretely. The $\mathbf{E}$ and $\mathbf{H}$ fields are initialized with user-defined functions directly onto the locations of these fields on the yee-grid, as are any external sources corresponding to these fields.

3.2 Temporal Discretization

Explicit time-domain algorithm is implemented in this work as it leads to less numerical dispersion errors compared to implicit algorithms proposed by Zheng and Chen (2000); Namiki (1999); Yao et al. (2018), albeit its computational cost. For future developments, we may explore the option of implementing implicit algorithms to improve computational efficiency by employing larger time-steps compared to explicit methods, with the recognition of challenges in the GPU implementation and field boundary issues described by Yao et al. (2018). The overall scheme resembles the well-known second-order leapfrog scheme used to advance $\mathbf{H}$ and $\mathbf{E}$ in standard FDTD schemes. The temporal discretization adopted is similar to that described by Aziz (2009). In this scheme, at every time step, $\mathbf{H}$ is first advanced over a half time step ($\mathbf{H}^n \rightarrow \mathbf{H}^{n+1/2}$), followed by a full time step update of $\mathbf{E}$ ($\mathbf{E}^n \rightarrow \mathbf{E}^{n+1}$), concluding with an additional half time step update of $\mathbf{H}$ ($\mathbf{H}^{n+1/2} \rightarrow \mathbf{H}^{n+1}$). We again note that in magnetic regions, we choose to update both $\mathbf{H}$ and $\mathbf{M}$ simultaneously using (4) and a second-order accurate

iteration scheme which satisfies the LLG equation (3) to a user-specified tolerance.

As mentioned in Section 2, analytically, the magnitude of $\mathbf{M}$ at each point of space cannot change since the terms in the right-hand-side are orthogonal to the direction of $\mathbf{M}$. However, since we are using finite-sized time steps, numerically, the magnitude of $\mathbf{M}$ will slightly change with each time step. Thus, after each update of $\mathbf{M}$, we renormalize $\mathbf{M}$ using the function $\mathcal{N}(\mathbf{M}) \equiv \mathbf{M} \times (M_s/|\mathbf{M}|)$. The renormalization step is included in the algorithm descriptions below.

Additionally, spatially-varying, time-dependent soft-source and hard-source external excitations are supported for both the $\mathbf{E}$ and $\mathbf{H}$ fields. Taflove and Hagness (2005) defines soft sources in terms of the time-rate of field change, and are applied at the beginning of the time step. Hard sources are enforced directly for the corresponding fields after their updates at each time step.

3.2.1 First-Order Temporal Scheme Although the focus of this paper is the second-order scheme, and the majority of simulations are not performed using this first-order scheme, describing this method in full allows for an easier understanding of how we implemented the second-order method. In Section 3.2.2 we will discuss the modifications to the first-order scheme that are required for second-order temporal accuracy. An analytically equivalent form of the LLG equations described in (3) is,

$$\frac{\partial \mathbf{M}}{\partial t} = \mu_0 \gamma_{\mathrm{L}} (\mathbf{M} \times \mathbf{H}_{\mathrm{eff}}) + \frac{\alpha \mu_0 \gamma_{\mathrm{L}}}{|\mathbf{M}|} \mathbf{M} \times (\mathbf{M} \times \mathbf{H}_{\mathrm{eff}}),$$
(6)

where $\gamma_L = \gamma/(1 + \alpha^2)$ Lax et al. (1962), and $\boldsymbol{\tau}$ has been omitted. This formulation will be used in our first-order temporal scheme.

Step 1: We compute $\mathbf{M}^{n+\frac{1}{2}}$ by reformulating the scalar form of the LLG equation (6) using a forward-Euler discretization proposed by Vacus and Vukadinovic (2005),

$$\mathbf{M}^{n+\frac{1}{2}} = \mathbf{M}^n$$
$$+ \Delta t \left[\mu_0 \gamma_{\mathrm{L}} (\mathbf{M} \times \mathbf{H}_{\mathrm{eff}}) + \frac{\alpha \mu_0 \gamma_{\mathrm{L}}}{M_s} \mathbf{M} \times (\mathbf{M} \times \mathbf{H}_{\mathrm{eff}}) \right]^n$$
(7)

Note that the transverse components of $\mathbf{H}_{\mathrm{eff}}$ are spatially averaged to update the respective $\mathbf{M}$ components at the cell-faces. We normalize $\mathbf{M}^{n+\frac{1}{2}} = \mathcal{N}(\mathbf{M}^{n+\frac{1}{2}})$ and then compute $\mathbf{H}^{n+\frac{1}{2}}$ by solving Faraday's law (4) with a forward-Euler discretization of the electric field,

$$\frac{1}{\mu_0} \nabla \times \mathbf{E}^n = \frac{\mathbf{H}^n - \mathbf{H}^{n+\frac{1}{2}}}{\Delta t/2} + \frac{\mathbf{M}^n - \mathbf{M}^{n+\frac{1}{2}}}{\Delta t/2}, \quad (8)$$

which simplifies to

$$\mathbf{H}^{n+\frac{1}{2}} = \mathbf{H}^n + \mathbf{M}^n - \mathbf{M}^{n+\frac{1}{2}} - \frac{\Delta t}{2\mu_0} \nabla \times \mathbf{E}^n. \quad (9)$$

In nonmagnetic regions, i.e. in regions where $\mathbf{M} = 0$, considering Faraday's law (5) is sufficient to compute $\mathbf{H}$. Therefore, in regions where $M_s = 0$, we simply set $\mathbf{M} = 0$ and integrate $\mathbf{H}$ using (9) but with μ_0 replaced by the local

value of μ, which reduces to the form described in (5).

Step 2: We compute $\mathbf{E}^{n+1}$ by discretizing Ampere's law (1) using time-centered representations of the magnetic field and current,

$$\nabla \times \mathbf{H}^{n+\frac{1}{2}} = \frac{1}{2}\left(\sigma \mathbf{E}^n + \sigma \mathbf{E}^{n+1}\right) \\ + \mathbf{J}_{\mathrm{src}}^{n+\frac{1}{2}} + \epsilon \frac{\mathbf{E}^{n+1} - \mathbf{E}^n}{\Delta t}, \quad (10)$$

which simplifies to

$$\mathbf{E}^{n+1} = \left(\frac{\sigma}{2} + \frac{\epsilon}{\Delta t}\right)^{-1} \\ \times \left[\nabla \times \mathbf{H}^{n+\frac{1}{2}} - \left(\frac{\sigma}{2} - \frac{\epsilon}{\Delta t}\right)\mathbf{E}^n - \mathbf{J}_{\mathrm{src}}^{n+\frac{1}{2}}\right]. \quad (11)$$

Step 3: We compute $\mathbf{M}^{n+1}$ following a similar procedure described in **Step 1**, using a forward-Euler discretization of the LLG equation (6):

$$\mathbf{M}^{n+1} = \mathbf{M}^{n+\frac{1}{2}} \\ + \Delta t \left[\mu_0 \gamma_{\mathrm{L}}(\mathbf{M} \times \mathbf{H}_{\mathrm{eff}}) + \frac{\alpha \mu_0 \gamma_{\mathrm{L}}}{M_s}\mathbf{M} \times (\mathbf{M} \times \mathbf{H}_{\mathrm{eff}})\right]^{n+\frac{1}{2}} \\ (12)$$

We normalize $\mathbf{M}^{n+1} = \mathcal{N}(\mathbf{M}^{n+1})$ and then compute $\mathbf{H}^{n+1}$ by integrating Faraday's law (4) with a backward-Euler discretization of the electric field,

$$\frac{1}{\mu_0}\nabla \times \mathbf{E}^{n+1} = \frac{\mathbf{H}^{n+\frac{1}{2}} - \mathbf{H}^{n+1}}{\Delta t/2} + \frac{\mathbf{M}^{n+\frac{1}{2}} - \mathbf{M}^{n+1}}{\Delta t/2}, \\ (13)$$

which simplifies to

$$\mathbf{H}^{n+1} = \mathbf{H}^{n+\frac{1}{2}} + \mathbf{M}^{n+\frac{1}{2}} - \mathbf{M}^{n+1} - \frac{\Delta t}{2\mu_0}\nabla \times \mathbf{E}^{n+1}. \\ (14)$$

Again, in nonmagnetic regions, we simply set $\mathbf{M} = 0$ and integrate $\mathbf{H}$ using (14) but with μ_0 replaced by the local value of μ.

3.2.2 Second-Order Temporal Scheme We replace **Step 1** and **Step 3** described in the first-order method of Section 3.2.1 with an iterative scheme that utilizes a trapezoidal discretization of the vector form of the LLG equation (3). In this algorithm, the exposition of **Step 1** and **Step 3** are identical, except that in **Step 1** we advance $(\mathbf{H}, \mathbf{M})$ from t^n to $t^{n+\frac{1}{2}}$ using the curl of $\mathbf{E}^n$, and in **Step 3** we advance $(\mathbf{H}, \mathbf{M})$ from $t^{n+\frac{1}{2}}$ to t^{n+1} using the curl of $\mathbf{E}^{n+1}$. Thus, we only write out the details for **Step 1**.

Step 1: We begin by setting $(\mathbf{H}, \mathbf{M})^{n+\frac{1}{2}, r=0} = (\mathbf{H}, \mathbf{M})^n$, where r is the counter for the iterative procedure. Then we iteratively solve (18)-(21) given below, beginning with $r = 1$ until the change in $\mathbf{M}^{n+\frac{1}{2}, r}$ from one iteration to the next is sufficiently small. Specifically, the iterations stop when the maximum value of $(\mathbf{M}^{n+\frac{1}{2}, r} - \mathbf{M}^{n+\frac{1}{2}, r-1})/M_s$ over the entire domain is smaller than a user-defined tolerance that we choose to be 10^{-6}. Note M_s is the local value at each corresponding mesh. Consider a second-order, trapezoidal

discretization of (3),

$$\frac{\mathbf{M}^{n+\frac{1}{2}} - \mathbf{M}^n}{\Delta t/2} = \\ \frac{\mu_0 \gamma}{2}(\mathbf{M}^n \times \mathbf{H}_{\mathrm{eff}}^n) + \frac{\mu_0 \gamma}{2}(\mathbf{M}^{n+\frac{1}{2}} \times \mathbf{H}_{\mathrm{eff}}^{n+\frac{1}{2}}) \\ + \frac{\alpha}{M_s}\left(\frac{\mathbf{M}^n + \mathbf{M}^{n+\frac{1}{2}}}{2}\right) \times \frac{\mathbf{M}^{n+\frac{1}{2}} - \mathbf{M}^n}{\Delta t/2}, \quad (15)$$

Since $\mathbf{M}^n \times \mathbf{M}^n = 0$, (15) can be expressed as

$$\frac{\mathbf{M}^{n+\frac{1}{2}} - \mathbf{M}^n}{\Delta t/2} = \\ \frac{\mu_0 \gamma}{2}(\mathbf{M}^n \times \mathbf{H}_{\mathrm{eff}}^n) + \frac{\mu_0 \gamma}{2}(\mathbf{M}^{n+\frac{1}{2}} \times \mathbf{H}_{\mathrm{eff}}^{n+\frac{1}{2}}) \\ - \frac{\alpha}{M_s}\frac{(\mathbf{M}^{n+\frac{1}{2}} \times \mathbf{M}^n)}{\Delta t/2}. \quad (16)$$

Reordering terms gives

$$\mathbf{M}^{n+\frac{1}{2}} = \mathbf{M}^n + \frac{\mu_0 \gamma \Delta t}{4}(\mathbf{M}^n \times \mathbf{H}_{\mathrm{eff}}^n) \\ - \mathbf{M}^{n+\frac{1}{2}} \times \left(-\frac{\mu_0 \gamma \Delta t}{4}\mathbf{H}_{\mathrm{eff}}^{n+\frac{1}{2}} + \frac{\alpha}{M_s}\mathbf{M}^n\right). \quad (17)$$

If we replace $\mathbf{M}^{n+\frac{1}{2}}$ with $\mathbf{M}^{n+\frac{1}{2}, r}$ and $\mathbf{H}_{\mathrm{eff}}^{n+\frac{1}{2}}$ with $\mathbf{H}_{\mathrm{eff}}^{n+\frac{1}{2}, r-1}$, we are essentially lagging some terms in the iterative update, noting the accuracy of the procedure improves with iteration count. Also, we can now directly solve for $\mathbf{M}^{n+\frac{1}{2}, r}$ using the following analytical expression derived by Vacus and Vukadinovic (2005),

$$\mathbf{M}^{n+\frac{1}{2}, r} = \frac{\boldsymbol{b} + (\boldsymbol{a} \cdot \boldsymbol{b})\boldsymbol{a} - \boldsymbol{a} \times \boldsymbol{b}}{1 + |\boldsymbol{a}|^2}, \quad (18)$$

where

$$\boldsymbol{a} = -\left[\frac{\mu_0 |\gamma| \Delta t}{4}\mathbf{H}_{\mathrm{eff}}^{n+\frac{1}{2}, r-1} + \frac{\alpha}{M_s}\mathbf{M}^n\right], \quad (19)$$

$$\boldsymbol{b} = \mathbf{M}^n - \frac{\mu_0 |\gamma| \Delta t}{4}(\mathbf{M}^n \times \mathbf{H}_{\mathrm{eff}}^n). \quad (20)$$

We normalize $\mathbf{M}^{n+\frac{1}{2}, r} = \mathcal{N}(\mathbf{M}^{n+\frac{1}{2}, r})$ and then compute $\mathbf{H}^{n+\frac{1}{2}, r}$ by integrating Faraday's law as in the first-order algorithm, with an updated approximation for the time-derivative of $\mathbf{M}$,

$$\mathbf{H}^{n+\frac{1}{2}, r} = \mathbf{H}^n + \mathbf{M}^n - \mathbf{M}^{n+\frac{1}{2}, r} - \frac{\Delta t}{2\mu_0}\nabla \times \mathbf{E}^n. \quad (21)$$

We continue to iterate over r using (18) and (21). After the iterations converge, we set $(\mathbf{H}, \mathbf{M})^{n+\frac{1}{2}} = (\mathbf{H}, \mathbf{M})^{n+\frac{1}{2}, r}$ and proceed to **Step 2**. Note that in regions where $M_s = 0$, we set $\mathbf{M} = 0$, skip this iterative scheme, and instead integrate $\mathbf{H}$ using (21) but with μ replacing μ_0.

3.2.3 Boundary Conditions Our solver supports three types of boundary conditions that are required to ensure the correct treatment of EM waves at the computational domain boundary for modeling microelectronic devices. Periodic boundary is implemented through the AMReX and WarpX infrastructure, where the EM waves leaving one face of the

computational domain re-enter from the opposite face by exchanging guard-cell information handled by the AMReX code. To model a perfect conductor such that EM waves are reflected at the boundary, we have implemented the well-known perfect electric conductor (PEC) condition, by enforcing zero tangential electric field and normal magnetic field at the domain boundary. We have validated the PEC implementation by ensuring that the reflection coefficient is -1 for a simple test case with a sinusoid EM wave reflected from the infinite PEC boundary.

The third boundary condition is an open boundary to model continuity of the medium and ensure EM waves cleanly exit the domain without re-entering or getting reflected. We achieve this by using a perfectly matched layer (PML) first developed by Berenger et al. (1994); Berenger (1996) boundary condition which damps out EM waves using a non-physical conductivity in the guard cells outside the domain boundary. Our solver leverages the two-step time-centered PML implementation, called "PML-2SC", implemented in WarpX by Shapoval et al. (2019). However, the implementation only supported vacuum medium, and in our solver, we extended it to include macroscopic material properties such as σ, ϵ and μ. In the macroscopic PML-2SC implementation, field components $\mathbf{E}$ and $\mathbf{H}$ are updated using macroscopic Maxwell's equations that account for spatially varying material properties and then the updated field components are multiplied by the corresponding PML damping coefficients, similar to the approach described by Shapoval et al. (2019). We have verified that the $\mathbf{E}$ and $\mathbf{H}$ fields are dissipated with $< 0.1\%$ reflection for non-magnetic material aligned with the domain boundaries, wherein the relative permeability $\mu_r >= 1$. Extending this boundary condition for magnetic materials using relative permeability self-consistently computed from the local magnetization, $\mathbf{M}$ and magnetic field, $\mathbf{H}$ will be implemented as future work.

4 Software

We implement our code using AMReX developed by Zhang et al. (2019, 2020), which is a software framework developed and supported by the DOE Exascale Computing Project AMReX Co-Design Center. AMReX contains many features for solving partial differential equations on structured grids; here we discuss relevant features for our present implementation, as well as future plans that will incorporate additional features.

AMReX manages data and operations on structured grids in a manner that can efficiently use the full range of computer architectures from laptops to manycore/GPU supercomputing architectures. We divide the computational domain into non-overlapping grids, and each grid is assigned to an MPI rank. AMReX uses an MPI+X fine-grained parallelization strategy, where X can be OpenMP (for multicore architectures), or CUDA (for GPU-based architectures). Each of these strategies are implemented with the same front-end code using specialized looping structures within AMReX and the portability across various platforms is ensured by AMReX during compile-time. Each MPI process applies computational kernels only to the data in grids that they own in the form of triply-nested loops (over each spatial dimension). For pure MPI calculations, the loop is interpreted as a standard "i/j/k" loop. For MPI+OpenMP calculations, the bounds of the loop is further subdivided over logical tiles, and each OpenMP thread loops over a particular tile. For MPI+CUDA calculations, AMReX performs a kernel launch and offloads each data point to CUDA threads that in turn perform computations. AMReX manages data movement by keeping data on the GPU devices as much as possible, avoiding costly communication between the host and device. Thus, whenever possible, data movement to/from the host/GPU and also between MPI ranks is limited to ghost cell data exchanges, which occur a small number of times per time step. In Section 4.1 we demonstrate the efficiency and scalability of our code using pure MPI, MPI+OpenMP, and MPI+CUDA simulations on NERSC systems. This analysis is made realized built-in AMReX profiling tools; however more in-depth analysis is possible with an extensive variety of compatible profilers such as CrayPat, IPM, and Nsight. Data from the simulation can be efficiently written using a user-defined number of MPI ranks, to prevent overwhelming the system with simultaneous writes. Visualization can be performed with a number of publicly-available software packages, including Amrvis, VisIt, Paraview, and yt.

The code was originally created as a fork from the WarpX Maxwell solver in order to inherit the framework for the time-advancement, boundary conditions, and parsing of material properties and sources. Since then, new field-update kernels have been added to support the features described in this paper such as spatially-varying material properties, the evolution of the magnetization and magnetic field, boundary condition support for PML and PEC, the ability to parse spatially-varying material properties $(\epsilon, \mu, M_s, \sigma, \gamma, \alpha)$, and the ability to add external $\mathbf{E}$ and $\mathbf{H}$ waveforms (both hard and soft sources). All codes are publicly available on the open-source github platform. AMReX can be obtained at https://github.com/AMReX-Codes/amrex and the microelectronics code can be obtained at https://github.com/ECP-WarpX/artemis.

4.1 Code Performance

To assess the scaling and performance of our code on HPC systems we perform tests on the NERSC supercomputer. We consider both the Haswell and GPU partitions on the Cori system. The Haswell partition consists of 2,388 nodes; each node contains an Intel Xeon "Haswell" Processor with 32 physical cores and 128GB memory. The GPU partition is a test-bed representative of the forthcoming Perlmutter system consisting of 18 nodes; each node contains a 20-core Intel Xeon Gold "Skylake" processor with 372GB memory and 8 NVIDIA Tesla V100 "Volta" GPUs each with 16GB memory. We perform weak and strong scaling tests and then compare kernel computation and communication times across the different architectures using pure MPI, MPI+OpenMP, and MPI+CUDA.

To perform weak scaling studies we use the magnetically tunable filter case described in Sec. 5.2.2. We perform three sets of simulations to demonstrate performance portability across heterogeneous computer architectures on the NERSC Cori systems. The first set of simulations employ the pure MPI paradigm on the Haswell partition with a maximum

of 32 MPI processes per node. For the second set of simulations, we use a hybrid MPI+OpenMP approach on the Haswell partition, where we use up to 8 MPI processes per node, and 4 OpenMP threads per MPI process. The third set of simulations employ the MPI+CUDA parallelization strategy on the GPU partition of Cori, which has a maximum of 8 NVIDIA V100 GPUs, and we use one MPI processor per GPU. Note that a node on the GPU partition contains 20 physical cores on the host, yet we only use a maximum of 8 cores (one per GPU).

The baseline case for all weak scaling tests performed with different parallelization strategies has a grid size of 1024 × 4 × 512 cells (2,097,152 total). For this base case we use 1/8th of a node for each set of simulations, i.e. 4 MPI ranks on a Haswell partition node, 1 MPI rank × 4 OpenMP threads on a Haswell partition node, and 1 MPI rank + 1 GPU on a GPU partition node. We note that this simulation uses roughly 4GB memory, which is 25% of the memory available on 1/8th of a Haswell partition node and also 25% of the memory available on 1/8th of the Cori GPU node. The entire spatial domain is divided into grids (commonly called sub-grids in other EM simulation codes), and we assign 1 grid per MPI process; thus the 4 MPI-rank run uses four 1024 × 4 × 128 grids and the 1 MPI-rank runs use one grid spanning the domain. For the OpenMP runs we use a tile size of 1024 × 4 × 8 to subdivide each grid. For the weak-scaling study, we increase the domain size in the z direction consistent with the increase in the core count to maintain a constant amount of computational work per core.

Fig. 2 shows weak scaling results for the three sets of simulations, indicating the simulation time per time step as a function of the number of nodes. We omit the timing associated with initialization, including memory allocation and data transfer on the host and GPU, as these occur only once at the beginning of the simulation. We note that we performed tests on the Haswell partition up to 512 nodes (16,384 cores), which is roughly 20% of the entire system. We performed tests on the GPU partition using up to 8 nodes (64 GPUs), which is 50% of the testbed. For the 1/8th node baseline runs, the GPU simulations performed 32 times faster than the pure MPI simulations on the Haswell partition, and 100 times faster than the MPI+OpenMP simulations on the Haswell partition. At a glance, we see that for the 1-node through 8-node runs, the GPU simulations performed approximately 59x faster than the pure MPI simulations on the Haswell partition, and approximately 112x faster than the MPI+OpenMP simulations on the Haswell partition. We attribute the factor of two slowdown with MPI+OpenMP compared pure-MPI to the overhead required to spawn the OpenMP threads. The Haswell partition simulations achieve nearly perfect scaling up to 512 nodes beyond the 1-node threshold. The reason is that as we increase the number of processors within the node, the MPI communication-time keeps increasing until the node is saturated and then the computation-to-communication ratio remains the same across all nodes. The GPU partition simulations achieve nearly perfect scaling up to 8 nodes beyond the 1/4-node threshold, i.e., 2 MPI-GPU processors. The increase in simulation time from 1-GPU to 2-GPU simulation is mainly dominated by the communication time across their respective MPI hosts. In this paper we are not using dynamic load
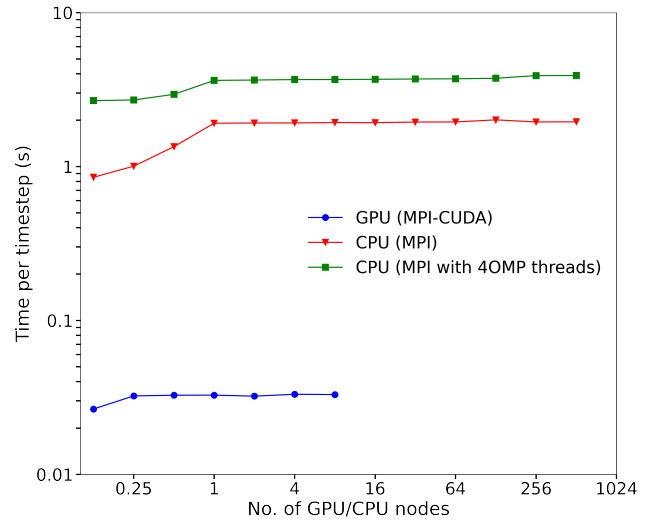

Figure 2. Weak scaling for pure MPI, MPI+OpenMP, and MPI+CUDA on the NERSC Cori Haswell and GPU partitions. Note that one Haswell partition node contains 32 physical cores, and one GPU partition node contains 8 GPUs. The weak scaling is nearly perfect for the Haswell partition simulations past the 1 node threshold up to 20% of the entire system. The weak scaling is also nearly perfect for the GPU partition simulations past the 1/4-node threshold up to 50% of the testbed. Also, using 1 node or more the GPU simulations run 59x and 112x faster than the pure MPI and MPI+OpenMP simulations on a node-by-node comparison.

balancing and assign an equal number of grid points to all processors using a fixed grid structure. In the future, as we move to adaptively refined simulations, dynamic load balancing will become more important. One particular code optimization we may explore is the use of grid structures where magnetic regions contain a larger number of smaller grids; the computational requirements for the evolution of **M** are much more demanding than **E** and **H** so we would be focusing computational resources in these regions. The communication exchange in our algorithm is relatively small, where ghost cell exchanges occur once after each variable advancement routine. There are no communication-intensive linear solvers in our implementation. Thus, we expect this scaling trend to continue on the forthcoming Perlmutter system since we have demonstrated nearly perfect scaling with 16,384 MPI processes on the Haswell partition.

For our strong scaling tests, we chose a problem with 1024 × 4 × 2048 cells for all the simulations, as it uses nearly all the available memory on 1/8th of the Haswell and GPU partition node. We performed two sets of simulations, pure MPI and MPI+CUDA. In both sets we hold a constant number of grid points in the domain, and divide the domain into a larger number of smaller grids such that each MPI process is assigned one grid. In Fig. 3, we show strong scaling and the corresponding strong scaling efficiency, the latter defined as the ratio of runtime to ideal scaling runtime from the 1/8th node cases. The departure from ideal strong scaling occurs simply because there is not enough work per MPI process compared to the non-compute communication overhead as a larger number of smaller grids are used, which is an effect seen most noticeably in the GPU runs. As the compute work per processor decreases, the computational
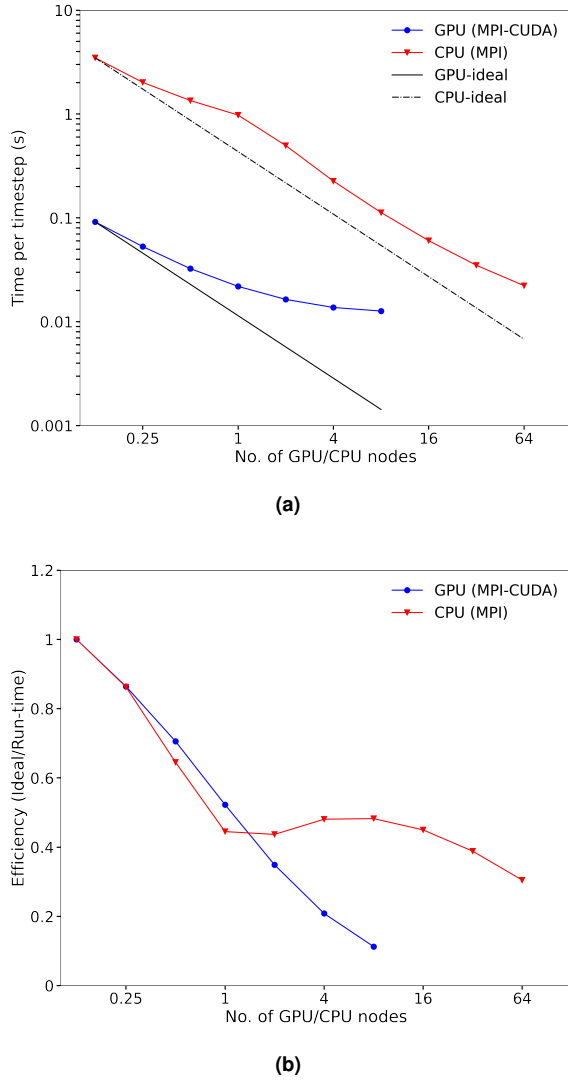
(a)

(b)

Figure 3. (a) Strong scaling performance and strong scaling efficiency on the NERSC Cori Haswell and GPU partitions. The departure from ideal scaling is more pronounced on the GPU partition runs, as there is not enough computational work per GPU to make the most effective use of the system.

overhead, notably communication, takes a larger percentage of runtime. As expected, the GPU runs outperform the MPI runs by an increasingly larger margin as the compute work per MPI process increases.

We detail the timing of various routines for the three sets of simulations (on one single node) in Tab. 1. We see that each individual kernel, as well as the communication overhead, are significantly faster on the GPU partition, such that the overall speedup is 59x and 122x compared to the pure MPI and MPI+OpenMP simulations. The speedup for the computational kernels "Evolve HM" and "Evolve E" is significant, ranging from 84x up to 256x. We note that unlike the CPU-simulations, communication dominates the GPU simulations requiring 60% of the total runtime. Nevertheless the time spent in communication in the GPU simulations is still 38x and 43x smaller compared to the MPI and MPI+OpenMP run.

5 Validation

5.1 Convergence Test

We performed a coupled LLG-Maxwell test-case to demonstrate the order of accuracy of the numerical scheme previously described in Sec. 3 with spatially varying material properties. In our test problem, we consider a fully periodic domain with $L = 0.1$ m (all units in this paper are MKS unless noted otherwise) on each side, centered about the origin. We initialize a three-dimensional EM signal, such that at $t = 0$,

$$
\begin{aligned}
E_x &= 10^6 \cos\left(20\pi z\right), \\
E_y &= 10^6 \cos\left(20\pi x\right), \\
E_z &= 10^6 \cos\left(20\pi y\right),
\end{aligned}
\tag{22}
$$

$$
\begin{aligned}
H_x &= \frac{10^6}{120\pi} \cos\left(20\pi y\right), \\
H_y &= \frac{10^6}{120\pi} \cos\left(20\pi z\right), \\
H_z &= \frac{10^6}{120\pi} \cos\left(20\pi x\right).
\end{aligned}
\tag{23}
$$

Note each of the electric and magnetic field components in (22) and (23) satisfy the relation $E_p/H_p = 120\pi = \sqrt{\mu_0/\epsilon_0}$, with the ratio being the intrinsic impedance of vacuum. Defining r as the distance from the origin (center of the domain), the material properties have the following spatially-varying profiles,

$$
\begin{aligned}
\sigma &= 0.1 e^{-1000 r^2}, & (24) \\
\epsilon &= \epsilon_0 (1 + e^{-1000 r^2}), & (25) \\
M_s &= 10^4 (1 + e^{-1000 r^2}), & (26) \\
\alpha &= 0.5 (1 + e^{-1000 r^2}), & (27) \\
\gamma &= -1.759 \times 10^{11} (1 + e^{-1000 r^2}). & (28)
\end{aligned}
$$

We also set $M_x = M_s$ and $M_y = M_z = 0$ at the beginning of the simulation. Note that since $M_s > 0$, only vacuum permeability is used to update the fields as discussed in Sec. 2. In this particular test, we do not include any additional bias so that $\mathbf{H}_{\text{eff}} = \mathbf{H}$. We perform three simulations with 32^3, 64^3, and 128^3 grid size, such that, we increasingly refine the cell-size of each dimension by a factor of two. We call these test cases as coarse (c), medium (m), and fine (f), respectively. To compare the results from these simulations at the same physical time, a CFL=0.9 is used to define the timestep based on the speed of light in vacuum and the simulations are run for 5, 10, and 20 timesteps, respectively, to reach the same physical time. The convergence rate is defined as the base-2 log of the ratio of errors between the coarse-medium, E_c^m, and medium-fine solutions, E_m^f.

$$
\text{Rate} = \log_2 \left(\frac{E_c^m}{E_m^f} \right)
\tag{29}
$$

E_c^m and E_m^f are obtained by computing the L^1 norm between the coarser and the finer solution of a field, ϕ, averaged/interpolated to the same grid points as the former. Note that, for face-centered data, the solutions from the finer simulation is re-constructed on the coarser grid by averaging

Table 1. Profiling output for single-node simulations for pure MPI, MPI+OpenMP, and MPI+CUDA over 10 time steps. We indicate percent of runtime for each routine in the first three columns, and the speedup of the GPU runs compared to the non-GPU runs in the last two columns.

Kernel name	pure MPI (s)	MPI+OpenMP (s)	MPI+CUDA (s)	Speedup pure MPI	Speedup MPI+OpenMP
Evolve HM	8.38 (54%)	21.05 (71%)	0.1 (38%)	84x	211x
Evolve E	1.13 (7.3%)	1.79 (6.1%)	0.007 (2.6%)	161x	256x
Communication	6.01 (39%)	6.73 (23%)	0.158 (60%)	38x	43x
Total Time	15.52	29.57	0.265	59x	112x

the 4 fine face-centered values overlying the coarser grid. Similarly, for the edge-centered data, the finer solutions are re-constructed on the coarser grid by averaging the two fine-edge values overlying the coarse-edge. Errors E_c^m and E_m^f are defined as,

$$E_c^m = \frac{1}{Nc_{\text{pts}}} \sum_{i,j,k} |\phi_m - \phi_c|; E_m^f = \frac{1}{Nm_{\text{pts}}} \sum_{i,j,k} |\phi_f - \phi_m| \tag{30}$$

where, ϕ_c, ϕ_m, and ϕ_f are the fields obtained from the coarse, medium, and fine solutions, and Nc_{pts} and Nm_{pts} are total number of grid points used for the coarse and medium simulations, respectively. As shown in Tab. 2, we obtain clean second-order convergence for all variables. Even though we only report convergence in L^1, we obtain nearly identical convergence rates in L^0 and L^2. We have also performed similar tests using the first-order magnetization coupling scheme described in Sec. 3.2.1. As expected, the convergence rates for the **M** components dropped to first order, and the rates for the **E** and **H** fields reduced to as low as 1.74 due to coupling with the first-order **M** field.

Table 2. Convergence rates in the L^1 norm for all field variables.

Variable	E_{32}^{64}	E_{64}^{128}	Rate
E_x	1819.4	455.23	2.00
E_y	1820.5	455.04	2.00
E_z	1882.8	470.47	2.00
H_x	9.6659	2.4359	1.99
H_y	9.4752	2.3788	1.99
H_z	9.4590	2.3607	2.00
M_x	5.1343	1.2870	2.00
M_y	4.7851	1.1989	2.00
M_z	4.9642	1.2429	2.00

We only report convergence in L^1, but we note that we obtain nearly identical convergence rates in L^0 and L^2. We also performed similar tests using the first-order magnetization coupling scheme described in Sec. 3.2.1. As expected, the convergence rates for the **M** components dropped to first order, and the rates for the **E** and **H** fields was more erratic, with convergence rates ranging from 1.74 to 2.35.

5.2 Practical Device Applications

We validate our code by simulating the characteristics of two different practical waveguide devices. We perform two validation tests, one to validate the core macroscopic

Maxwell module, and second to validate the coupled Maxwell-LLG implementation. In the first case, the standard X-band waveguide is filled with air (see Fig. 4a) to demonstrate EM wave propagation in the presence of known boundary conditions. In the second case shown in Fig. 4b, we model a magnetically tunable filter realized by a ferrite loaded X-band waveguide structure. The magnetic spins interact with the elliptically polarized EM wave in the waveguide, and thus the wave exhibits exotic behaviors such as non-reciprocity and high attenuation around the ferromagnetic resonance (FMR) frequency. Thus, this test-case serves as a good validation for the coupled Maxwell-LLG algorithm.

5.2.1 Air-filled Waveguide The air-filled rectangular waveguide described by Pozar (2012) is used to validate the core macroscopic EM module in our solver. As illustrated in Fig. 4a, the width of the waveguide is $W = 14.95$ mm, and the height is $h = 10.16$ mm. We apply PEC boundary condition at the four side walls (in x and y) of the waveguide. PEC is also applied at one end in the longitudinal direction (in z), reducing the simulation volume by a factor of 2. This type of boundary conditions is also referred to as anti-symmetric boundary conditions in other numerical frameworks. In this case, we apply this PEC boundary condition at $z = +L/2$, where L is the length of the domain and equal to 500 mm. At the same location, we superimpose an electric current source I in addition to the PEC boundary, which excites the EM wave propagating in the $-z$ direction. Following conventions, the waveguide operates at its fundamental mode, or in the other words, the TE_{10} mode, with a cutoff frequency of $f_c = c/2W = 10.03$ GHz, where c is the speed of light in vacuum. To capture the cutoff knee point, the current excitation I takes a modified Gaussian pulse form, as shown below:

$$I = \exp\left[\frac{-(t - 3T_p)^2}{2T_p^2}\right] \cos(\omega_0 t) \cos\left(\frac{2\pi x}{W/2}\right), \tag{31}$$

In (31), $\omega_0 = 2\pi f_0$ sets the central frequency of the pulse, with the value of f_0 being 10.5GHz. T_p is the time width of the pulse, which is set to be one period of the excitation frequency, i.e. $T_p = 0.095238$ ns. Note that we apply a shape function $\cos(\frac{2\pi x}{W/2})$ for the purpose of mode matching to the TE_{10} mode. The current excitation is directly added on top of the H field at the PEC, such that it satisfies the boundary condition, $\Delta H_{\text{tangential}} = J_S$ at $z = 250$mm, where J_s represents the surface current density in the unit of A/m. In this simulation, we use $512 \times 4 \times 512$ grid points. The CFL factor is 0.8, leading to a time step of $\Delta t =$
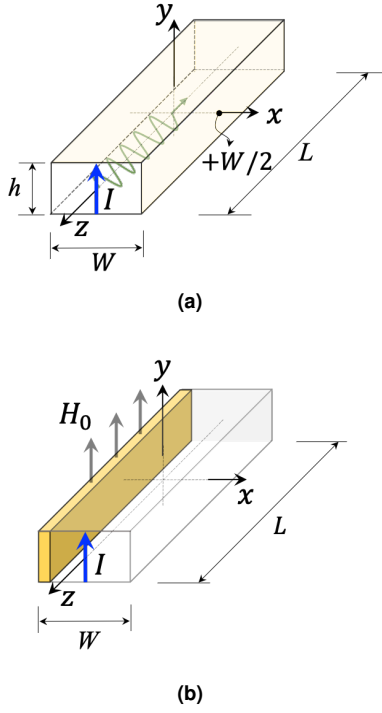
(a)

(b)

Figure 4. (a) Air-filled waveguide. (b) Magnetically tunable filter, realized by the same waveguide structure in Fig. 4a, with an inserted ferrite slab.

77.87961902 fs. We run the simulation to 80,000 time steps, which corresponds to a total of ∼65 EM wave cycles.

After the EM fields reach steady-state, we compute fast Fourier transform on the time-domain electric field at the central longitudinal line of the waveguide, i.e. ($x = 0, y = 0, -L/2 \leq z \leq +L/2$). The attenuation constant of the electric field is derived from the extracted damping speed of the field magnitude, with the signal-processing code, ESPRIT, developed by Roy and Kailath (1989); Wang and Ling (1998). Fig. 5 shows both the theoretical and simulated attenuation spectrum of the air-filled waveguide. The theoretical attenuation rate α is defined as

$$\alpha = \text{imag}\left[\sqrt{\omega^2 \mu_0 \epsilon_0 - (\pi/W)^2}\right] \quad (32)$$

The theoretical cutoff frequency, f_{c1}, of this waveguide structure is $f_{c1} = c/(2W) = 10.03\text{GHz}$. It can be observed that the α follows the hyperbolic shape before the cutoff frequency f_{c1}. Beyond the cutoff frequency, α becomes zero, indicating that the wave propagates along the waveguide with no attenuation. Both the curve shape and the cutoff frequency agree well the theoretical prediction.

5.2.2 Magnetically tunable filter To validate the interaction between the magnetic spins and the EM wave in the coupled Maxwell/LLG framework, a thin ferrite slab is inserted in the longitudinal direction of the waveguide, as shown in Fig. 4b. The ferrite material is yttrium iron garnet (YIG), which is widely used in RF front-ends and devices, due to its narrow magnetic linewidth ΔH. The dimensions and setup of this YIG-inserted waveguide are the same as that for the air-filled waveguide shown in Fig. 4a. In our simulation, the saturation magnetization is $4\pi M_S = 1750$ Gauss, or $1.3926e5$ A/m. A DC magnetic bias $H_0 = 2890$ Oersted =
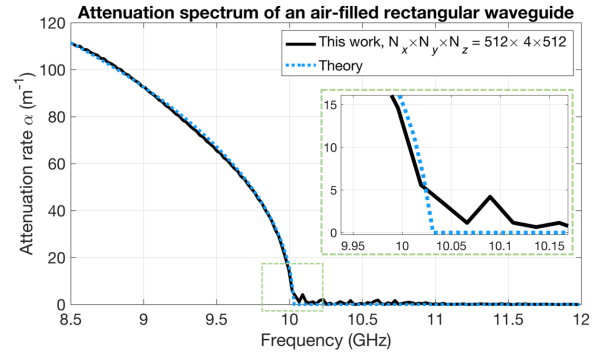

Figure 5. Comparison of attenuation spectrum of the air-filled waveguide obtained from computations with theory.

$2.2998e5$ A/m is applied along the in-plane direction of the ferrite slab and the tangential direction of the waveguide, therefore, H_0 is in the y direction. According to Kittel's equation, the FMR frequency of the thin slab is

$$f_{\text{FMR}} = 2.8 \text{ MHz} \times \sqrt{H_0(H_0 + M_S)}. \quad (33)$$

Note in (33), H_0 and M_S are both in units of Oersteds. Therefore, the theoretical FMR frequency is 10.25 GHz. The damping factor of YIG is $\alpha = 0.0051$, corresponding to an FMR linewidth of 35 Oersted at X band, and its relative permittivity is $\epsilon_r = 13$. The thickness of the YIG slab is 0.45 mm, with one of the faces touching the PEC wall of the waveguide.

We use the same post processing technique to extract the attenuation spectrum $\alpha(f)$ as that described in Sec. 5.2.1. Fig. 6 shows the comparison of the attenuation spectrum obtained from experimental tests reported by Gardiol and Vander Vorst (1971), theoretical analyses reported by Gardiol and Vander Vorst (1971); Uher et al. (1987), and the simulation of this work. The key features of the attenuation spectrum, such as waveguide cut-off frequency and FMR frequency, obtained from the simulations agree with experiments and theory. $\alpha(f)$ shows a hyperbolic shape below 9.2 GHz, which is the cutoff frequency f_{c2}. Note that f_{c2} is different from f_{c1} by 0.83 GHz, primarily due to the large relative permittivity $\epsilon_r = 13$ present in the YIG slab. Moreover, the largest attenuation rate of the EM wave is observed around 10.45 GHz. Tab. 3 shows a quantitative comparison of the cut-off frequency, FMR frequency, and peak attenuation obtained from the spectrum-curves shown in Fig. 6. The cut-off frequency from the simulation is within 1.2%, 0.35% and 0.25% discrepancies compared to experiment, theory 1 and theory 2, respectively. Similarly, the FMR frequency from the simulation is within 0.21%, 1.6% and 0.56% discrepancies compared to experiment, theory 1 and theory 2, respectively. However, the peak attenuation varies by 5.2%, 28% and 14% compared to experiment, theory 1 and theory 2, respectively. The discrepancy between the FMR frequencies predicted by the Kittel's equation and simulation/experiment can be attributed to the assumption of infinite planar dimension of the thin film used in the theories compared to finite-size thin films used in simulation and experiment. Due to similar reasons, our results do not recover as sharp of a peak at FMR as theory, and is in better agreement with experimental measurements.

Table 3. Key features of all attenuation spectra curves, $\alpha(f)$, in Fig. 6.

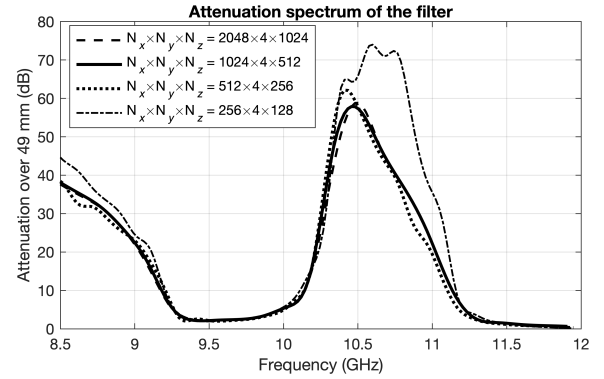| Curve name	Cutoff freq. f_{c2} (GHz)	FMR freq. f_{FMR} (GHz)	Peak attenuation α_{peak} (dB)
This work	9.2717	10.4568	57.8862
Experiment by Gardiol and Vander Vorst (1971)	9.1605	10.4354	55.0005
Theory 1 by Gardiol and Vander Vorst (1971)	9.2388	10.2873	80.5410
Theory 2 by Uher et al. (1987)	9.2485	10.3989	67.4374
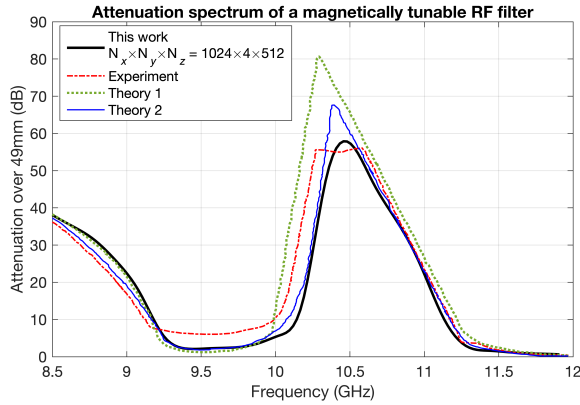
Figure 6. Comparison of computation result to experimental result and theoretical predictions for the attenuation spectrum of the magnetically tunable filter. The key features of the attenuation spectrum obtained from the simulations agree with experiments and theory.

Figure 7. Computational results for attenuation spectra of the magnetically tunable filter for varying resolutions. We observe that by the $1024 \times 4 \times 512$ case the solution does not significantly change with increasing resolution.

To verify grid-convergence, we performed simulations with different values of Δx and Δz (holding Δy constant as the TE10-mode EM field is constant in y direction), refining each by a factor of 2 with each simulation. The coarsest simulation has $256 \times 4 \times 128$ grid cells, and the finest simulation has $2048 \times 4 \times 1024$ cells. All other numerical parameters remain the same, including the CFL coefficient being 0.8 for all cases. The attenuation spectrum (analogous to Fig. 6) for these four simulations are shown in Fig. 7. We note that the coarsest simulation results for $\alpha(f)$ do not match the other results. The curves for the two finest simulations exhibit little visible difference. Therefore, we conclude that $1024 \times 4 \times 512$ grid cells used for comparison with theory and experiments is sufficient to capture the evolution of EM signals for this problem.

6 Conclusions and Future Work

We have developed an open-source, portable software for modeling a coupled LLG-Maxwell solver for microelectronic circuitry. The algorithm is second-order in space and time with support for spatially varying material properties. We demonstrated its scalability on the NERSC multicore and GPU partitions, including the GPU speedup compared to host CPU-only simulations on a node-by-node basis. The agreement between experiments, theories and our simulations of realistic air-filled waveguide and magnetically tunable filter has demonstrated the code capability of predicting the coupled physics of EM and micromagnetics.

There are a number of algorithmic improvements we are planning to implement to increase the applicability of our solver to larger classes of problems in EM and micromagnetics. These include adding spin exchange coupling, magnetic anisotropy, STT and spin-orbit-torque (SOT) in the LLG equation, to enable the software to model a wider range of spintronic devices such as MRAM, magnetic sensors, and magnetic RF front-ends proposed by Suhl (1956); Cui et al. (2019). We are also interested in modeling quantum hardware that includes regions of superconducting material for capturing the non-linear behavior of Josephson Junctions. We plan to explore various representations of superconductivity, ranging from the use of PEC conditions at the boundary to incorporating different constitutive models, such as the London equations. To explore more advanced geometrical features typically found in circuits, in particular curved and non-axis-aligned interfaces, we will consider increasing levels of geometrical fidelity. This can include a simple stair-step approach to define either one material type or another in each cell, to a blended volume weighted approach, to full inclusion of embedded boundaries developed by Cary et al. (2006). Embedded boundary, or "cut-cell" support is also available within the AMReX infrastructure, offering future possibilities for more advanced geometrical representation. Embedded boundaries are being actively developed by the WarpX team, allowing us to leverage the discretizations and kernels in collaboration with the WarpX project. We are also interested in incorporating spectral solvers such as PSATD to compare the possible tradeoffs between efficiency and accuracy. Also, we can leverage the linear solvers provided by AMReX to develop or re-implement implicit approaches such as ADI. AMReX also contains built-in geometric multigrid and Krylov-based linear solvers for elliptic and parabolic problems, as well as interfaces to external solvers available in the hypre and

PETSc packages. AMReX also provides support for adaptive mesh refinement by managing communication between grids at different levels of refinement. Incorporating AMR will offer increased spatial resolution in regions of interest. The AMR support being developed for WarpX can be used as a springboard for our Maxwell+LLG solver, as well as future multiphysics coupling efforts. We will consider the use of the AMReX particle class for other types of applications, such as phonon excitation proposed by Anufriev et al. (2017).

Acknowledgements

References

Anufriev R, Ramiere A, Maire J and Nomura M (2017) Heat guiding and focusing using ballistic phonon transport in phononic nanostructures. *Nature Communications* 8(1): 15505. DOI:10.1038/ncomms15505. URL https://doi.org/10.1038/ncomms15505.

Aziz MM (2009) Sub-nanosecond electromagnetic-micromagnetic dynamic simulations using the finite-difference time-domain method. *Progress In Electromagnetics Research* 15: 1–29.

Berenger JP (1996) Three-dimensional perfectly matched layer for the absorption of electromagnetic waves. *Journal of computational physics* 127(2): 363–379.

Berenger JP et al. (1994) A perfectly matched layer for the absorption of electromagnetic waves. *Journal of computational physics* 114(2): 185–200.

Cary JR, Abell D, Amundson J, Bruhwiler D, Busby R, Carlsson J, Dimitrov D, Kashdan E, Messmer P, Nieter C et al. (2006) Petascale self-consistent electromagnetic computations using scalable and accurate algorithms for complex structures. In: *Journal of Physics: Conference Series*, volume 46. IOP Publishing, p. 027.

Chappert C, Fert A and Van Dau F (2007) The emergence of spin electronics in data storage. *Nature Mater* 6: 813–823.

Couture S, Chang R, Volvach I, Goncharov A and Lomakin V (2017) Coupled finite-element micromagnetic—integral equation electromagnetic simulator for modeling magnetization—eddy currents dynamics. *IEEE Transactions on Magnetics* 53(12): 1–9.

Cui H, Yao Z and Wang Y (2019) Coupling Electromagnetic Waves to Spin Waves: A Physics-Based Nonlinear Circuit Model for Frequency-Selective Limiters. *IEEE Transactions on Microwave Theory and Techniques* 67(8). DOI:10.1109/TMTT.2019.2918517.

Donahue M and Porter D (1999) Oommf user's guide, version 1.0. *Interagency Report NISTIR 6376* .

Fu S, Cui W, Hu M, Chang R, Donahue MJ and Lomakin V (2015) Finite-difference micromagnetic solvers with the object-oriented micromagnetic framework on graphics processing units. *IEEE Transactions on Magnetics* 52(4): 1–9.

Gardiol FE and Vander Vorst AS (1971) Computer analysis of e-plane resonance isolators. *IEEE Transactions on Microwave Theory and Techniques* 19(3): 315–322. DOI:10.1109/TMTT.1971.1127505.

Garello K, Miron I, Avci C and et al (2013) Symmetry and magnitude of spin–orbit torques in ferromagnetic heterostructures. *Nature Nanotech* 8: 587–593.

Landau L and Lifshitz E (1992) On the theory of the dispersion of magnetic permeability in ferromagnetic bodies. In: *Perspectives in Theoretical Physics*. Elsevier, pp. 51–65.

Lax B, Button KJ and Hagger H (1962) *Microwave ferrites and ferrimagnetics*. MacGraw-Hill.

Namiki T (1999) A New FDTD algorithm based on alternating-Direction implicit method. *IEEE Transactions on Microwave Theory and Techniques* DOI:10.1109/22.795075.

Oti JO (1997) Simulmag version 1.0. *Electromagnetic Technology Division, NIST* .

Pozar DM (2012) *Microwave Engineering*. JohnWiley & Sons, Inc.

Roy R and Kailath T (1989) Esprit-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37(7): 984–995. DOI:10.1109/29.32276.

Shapoval O, Vay JL and Vincenti H (2019) Two-step perfectly matched layer for arbitrary-order pseudo-spectral analytical time-domain methods. *Computer Physics Communications* 235: 102–110.

Suhl H (1956) The nonlinear behavior of ferrites at high microwave signal levels. *Proceedings of IRE* 44(10): 1270–1284.

Taflove A and Hagness SC (2005) *Computational electromagnetics: the finite-difference time-domain method*. Artech House.

Uher J, Arndt F and Bornemann J (1987) Field theory design of ferrite-loaded waveguide nonreciprocal phase shifters with multisection ferrite or dielectric slab impedance transformers. *IEEE Transactions on Microwave Theory and Techniques* 35(6): 552–560. DOI:10.1109/TMTT.1987.1133703.

Vacus O and Vukadinovic N (2005) Dynamic susceptibility computations for thin magnetic films. *Journal of Computational and Applied Mathematics* 176(2): 263–281. DOI:http://dx.doi.org/10.1016/j.cam.2004.07.016. URL internal-pdf://140.18.172.127/DynamicSusceptibilitycomputationsforthinm.pdfhttp://www.sciencedirect.com/science/article/pii/S0377042704003322.

Vansteenkiste A, Leliaert J, Dvornik M, Helsen M, Garcia-Sanchez F and Van Waeyenberge B (2014) The design and verification of mumax3. *AIP advances* 4(10): 107133.

Vay JL, Almgren A, Bell J, Ge L, Grote D, Hogan M, Kononenko O, Lehe R, Myers A, Ng C et al. (2018) Warp-X: A new exascale computing platform for beam–plasma simulations. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 909: 476–479.

Venugopal A, Qu T and Victora R (2019) Nonlinear parallel-pumped fmr: Three and four magnon processes. *IEEE Transactions on Microwave Theory and Techniques* 68(2): 602–610.

Wang Y and Ling H (1998) Multimode parameter extraction for multiconductor transmission lines via single-pass fdtd and signal-processing techniques. *IEEE Transactions on Microwave Theory and Techniques* 46(1): 89–96. DOI:10.1109/22.654927.

Yao Z, Tok R, Itoh T and Wang Y (2018) A multiscale unconditionally stable time-domain (MUST) solver unifying electrodynamics and micromagnetics. *IEEE Transactions on Microwave Theory and Techniques* 66(6): 2683–2696.

Yee K (1966) Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *IEEE Transactions on antennas and propagation* 14(3): 302–307.

Zhang W, Almgren A, Beckner V, Bell J, Blaschke J, Chan C, Day M, Friesen B, Gott K, Graves D et al. (2019) AMReX: A framework for block-structured adaptive mesh refinement. *Journal of Open Source Software* 4(37): 1370–1370.

Zhang W, Almgren A, Beckner V, Bell J, Blaschke J, Chan C, Day M, Friesen B, Gott K, Graves D et al. (2020) AMReX: Block-structured adaptive mesh refinement for multiphysics applications. *submitted for publication* .

Zheng F and Chen Z (2000) Toward the development of a three-dimensional unconditionally stable finite-difference time-domain method. *IEEE Transactions on Microwave Theory and Techniques* DOI:10.1109/22.869007.