A FINITE-ELEMENT FRAMEWORK FOR A MIMETIC FINITE-DIFFERENCE DISCRETIZATION OF MAXWELL'S EQUATIONS[*]

JAMES H. ADLER[†], CASEY CAVANAUGH[†], XIAOZHE HU[†], AND
LUDMIL T. ZIKATANOV[‡]

Abstract. Maxwell's equations are a system of partial differential equations that govern the laws of electromagnetic induction. We study a mimetic finite-difference (MFD) discretization of the equations which preserves important underlying physical properties. We show that, after mass-lumping and appropriate scaling, the MFD discretization is equivalent to a structure-preserving finite-element (FE) scheme. This allows for a transparent analysis of the MFD method using the FE framework and provides an avenue for the construction of efficient and robust linear solvers for the discretized system. In particular, block preconditioners designed for FE formulations can be applied to the MFD system in a straightforward fashion. We present numerical tests which verify the accuracy of the MFD scheme and confirm the robustness of the preconditioners.

Key words. Maxwell's equations, finite-element method, mimetic finite-difference method, structure-preserving block preconditioners

AMS subject classifications. 35Q61, 65M60, 65M06, 65F08, 65Z05

DOI. 10.1137/20M1382568

1. Introduction. We consider the numerical solution of Maxwell's equations in a bounded connected domain $\Omega \in \mathbb{R}^3$:

$$(1.1) \qquad \frac{\partial \boldsymbol{B}}{\partial t} + \nabla \times \boldsymbol{E} = \boldsymbol{0} \quad \text{in } \Omega \times (0, T],$$

$$(1.2) \qquad \epsilon \frac{\partial \boldsymbol{E}}{\partial t} - \nabla \times \mu^{-1} \boldsymbol{B} = -\boldsymbol{j} \quad \text{in } \Omega \times (0, T],$$

$$(1.3) \qquad \nabla \cdot \epsilon \boldsymbol{E} = 0 \quad \text{in } \Omega \times (0, T],$$

$$(1.4) \qquad \nabla \cdot \boldsymbol{B} = 0 \quad \text{in } \Omega \times (0, T].$$

Here, $\boldsymbol{B}(\boldsymbol{x}, t)$ and $\boldsymbol{E}(\boldsymbol{x}, t)$ are the unknown magnetic and electric fields, $\epsilon(\boldsymbol{x})$ and $\mu(\boldsymbol{x})$ are the permittivity and permeability of the medium, respectively, and $\boldsymbol{j}(\boldsymbol{x}, t)$ is the current density satisfying $\nabla \cdot \boldsymbol{j} = 0$. For simplicity, we choose $\epsilon = \mu = 1$ and impose homogeneous essential (Dirichlet) boundary conditions which model a perfect conductor:

$$(1.5) \qquad \boldsymbol{B} \cdot \boldsymbol{n}\big|_{\partial\Omega} = 0, \quad \boldsymbol{n} \times \boldsymbol{E}\big|_{\partial\Omega} = \boldsymbol{0}.$$

More general cases can be handled with straightforward modifications. In particular, the analysis presented in this paper still holds with nonconstant $\epsilon(\boldsymbol{x})$ and $\mu(\boldsymbol{x})$, for example, by using a piecewise constant approximation for the parameters.

[†]Department of Mathematics, Tufts University, Medford, MA 02155 USA (james.adler@tufts.edu, casey.cavanaugh@tufts.edu, xiaozhe.hu@tufts.edu).

[‡]Department of Mathematics, Pennsylvania State University, University Park, PA 16802 USA (ludmil@psu.edu).

The coupled equations (1.1)–(1.2), Faraday's and Ampere's laws, model the interaction of the electric and magnetic fields, while the Gauss laws, (1.3)–(1.4), model the flux constraints of the individual fields. One of the major difficulties in numerically solving the Maxwell system is related to the constraints (1.3) and (1.4), as it is often necessary, by physical or other considerations, to have analogues of such identities on the discrete level.

To rectify this, structure-preserving discretizations are used to guarantee that conservation laws on the continuous level still hold at the discrete level. While there are a variety of such numerical methods, we focus in this paper on the relationship between a mimetic finite-difference (MFD) method and a structure-preserving mixed finite-element method (FEM) derived via finite-element exterior calculus (FEEC).

The MFD method is defined by operators designed to "mimic" the continuous level operators [8, 19, 22, 31]. This technique is straightforward to derive and, by drawing similarities to the continuous operators, it is quite easy to see that the discrete operators do in fact obey the continuous level properties. MFD is simple to implement, can be applied directly to the strong form of the PDE system, and has few mesh requirements (i.e., general polyhedral grids can be used). Like most finite-difference methods, however, the error estimates and convergence theory require high regularity; well-posedness is difficult to prove; and it is not always clear how to optimally solve the resulting linear system.

On the other hand, structure-preserving FEMs for Maxwell's equations have been widely studied (see, e.g., [2, 3, 26, 27]). Using FEEC [4, 16], one can show that de Rham exact sequences are guaranteed on the discrete level. Furthermore, in the FEEC setting, the convergence theory and a priori error estimates are well known, and showing well-posedness of the discretized weak form follows directly from Babuška–Brezzi theory [5, 12]. Moreover, the construction of efficient solvers is also well developed [9, 10, 23, 24, 25, 28, 27].

An important question that arises when using such discretizations for PDEs is how to efficiently solve the resulting linear system. One widely used method for solving Maxwell's system discretized by FEM is based on block preconditioners for Krylov methods, constructed using Schur complements. More generally, block preconditioners for saddle point systems, a class that the full Maxwell system falls into, are widely studied [9, 10, 23, 24, 25], and robustness and efficiency results are well established. In general, to ensure that the iterative solver does not destroy the properties of the discretization, preconditioners must be developed that also preserve the operator properties at each time step. This is essential for ensuring that the resulting numerical solution obeys the PDE constraints throughout both the spatial and the time domains. Such preconditioners have been developed for the full Maxwell system, (1.1)–(1.4), using mixed FEM discretizations [2, 27], as well as for the simplified time-harmonic form of Maxwell's equations [15, 32]. Another variety of electromagnetic applications comes in the form of magnetohydrodynamics, where a similar block-preconditioning approach can be used for a FEM divergence-free preserving discretization [28].

In this work, we adapt the preconditioners developed in [2] for the Maxwell system with impedance boundary conditions. As in [2, 3], we consider a variation of (1.1)–

(1.4), where an auxiliary pressure variable, $p(\boldsymbol{x}, t)$, is introduced:

$$
(1.6) \qquad \frac{\partial \boldsymbol{B}}{\partial t} + \nabla \times \boldsymbol{E} = \boldsymbol{0} \quad \text{in } \Omega \times (0, T],
$$

$$
(1.7) \qquad \frac{\partial \boldsymbol{E}}{\partial t} - \nabla \times \boldsymbol{B} + \nabla p = -\boldsymbol{j} \text{ in } \Omega \times (0, T],
$$

$$
(1.8) \qquad \frac{\partial p}{\partial t} + \nabla \cdot \boldsymbol{E} = 0 \quad \text{in } \Omega \times (0, T].
$$

It is straightforward to show that with suitable initial conditions, $p(\boldsymbol{x}, 0) = 0$ and $\nabla \cdot \boldsymbol{B}(\boldsymbol{x}, 0) = 0$, (1.1)–(1.4) is equivalent to (1.6)–(1.8). A structure-preserving discretization is essential to allowing this form to be solved in place of the full Maxwell system.

Ideally, we would like to have the ease and simplicity of the MFD method with all of the well-posedness and preconditioning theory that supports the FEM. Similar to [11, 29], in this paper, we apply the MFD method to the Maxwell system and then analyze it in a FE framework. By using mass-lumping schemes and scaled basis functions in the FEM, we show that the two methods yield equivalent linear systems. This equivalence is used to apply the FE theory to the MFD system to show well-posedness of the mimetic discretization. Additionally, since robust block preconditioners have been designed for the Maxwell FE system that preserve the constraints at all time iterations [2], we demonstrate how to slightly modify those results to obtain robust linear solvers for the MFD system of Maxwell's equations.

This paper is organized as follows. In section 2, we introduce the notation and discretization technique for MFD and apply it to the Maxwell system. Section 3 recalls the FE discretization and presents a mass-lumped alternative. Then section 4 draws connections between the two methods for Maxwell and shows their equivalence, and well-posedness of the MFD system is proven. Section 5 introduces and analyzes block preconditioners for the MFD system, proving their robustness. Finally, section 6 presents numerical results to demonstrate the theoretical results, and concluding remarks and future work are discussed in section 7.

2. The mimetic finite-difference method. Following [31], we construct a primal (Delaunay) tetrahedral mesh and a dual (Voronoi) polyhedral grid. Denote the vertices/nodes of the Delaunay triangulation by $\{\boldsymbol{x}_i^D\}_{i=1}^{N_D}$, and an edge on the Delaunay mesh connecting nodes $\boldsymbol{x}_i^D$ and $\boldsymbol{x}_j^D$ by $\boldsymbol{e}_{ij}^D$, with unit tangent vector $\boldsymbol{t}_{ij}^D$ pointing from vertex of lower index to vertex of higher index. The Delaunay tetrahedra are given by D_k, $k = 1, \ldots, N_V$. Each D_k has face set (boundary) ∂D_k. The neighbor set of tetrahedron D_k, given by $\mathcal{N}_k^D$, is defined as the set of indices of the tetrahedra that share common planes with D_k, i.e., $\mathcal{N}_k^D := \{m \,|\, \partial D_k \cap \partial D_m \neq \emptyset, \, m = 1, \ldots, N_V\}$. The common plane (face) between D_k and D_m is given by ∂D_{km}. For tetrahedron D_k with face ∂D_{km}, we define the unit outward normal vector $\boldsymbol{n}_{km}^D$ pointing outward from D_k. Analogous to the above definitions, we have for the dual Voronoi mesh, nodes, $\{\boldsymbol{x}_k^V\}_{k=1}^{N_V}$, edges, $\boldsymbol{e}_{km}^V$, with unit tangent vector $\boldsymbol{t}_{km}^V$, polyhedra, V_i, $i = 1, \ldots, N_D$, face set, ∂V_i, neighbor set, $\mathcal{N}_i^V$, common plane, ∂V_{ij}, and outward unit normal vector $\boldsymbol{n}_{ij}^V$.

This dual mesh configuration yields some useful properties that are exploited when defining the MFD operators. The Voronoi point $\boldsymbol{x}_k^V$ is the circumcenter of the Delaunay tetrahedron D_k. Additionally, the Delaunay nodes define the Voronoi polyhedra. We define V_i as the set of points in the domain that lie closer to Delaunay

node $\boldsymbol{x}_i^D$ than any other Delaunay node:

$$V_i := \{\boldsymbol{x} \in \Omega \,\big|\, |\boldsymbol{x} - \boldsymbol{x}_i^D| \le |\boldsymbol{x} - \boldsymbol{x}_j^D|, \, j = 1, \dots, N_D, \, j \ne i\}.$$

Furthermore, we have that each Delaunay edge $\boldsymbol{e}_{ij}^D$ is orthogonal to the Voronoi face ∂V_{ij}, and each Voronoi edge $\boldsymbol{e}_{km}^V$ is orthogonal to Delaunay face ∂D_{km}. This gives us a one-to-one correspondence between nodes on one mesh to polyhedra on the other, and edges on one mesh to faces on the other. Figure 1 illustrates an example mesh in two dimensions to further highlight the notation used. Note that this dual mesh configuration with these properties requires that the circumcenters of the Delaunay triangulation lie in the interior of the Delaunay tetrahedra. While this is not a strict requirement for the MFD method to work, it allows for simplicity in the analysis and implementation (see [13, 31]).

2.1. Grid functions and MFD operators. Following [19, 31], we define functions and operators on both the Delaunay and the Voronoi meshes. First, approximations of scalar functions in the domain are represented with scalar grid functions that are defined on the nodes of the meshes. Thus, scalar functions defined on the Delaunay nodes are constants on Voronoi polyhedra, and scalar functions on Voronoi nodes are constants on Delaunay tetrahedra. The corresponding function spaces are as follows:

$$(2.1) \qquad H_D := \{u(\boldsymbol{x}) \,\big|\, u(\boldsymbol{x}) = u(\boldsymbol{x}_i^D) = u_i^D \ \forall \boldsymbol{x} \in V_i, \, i = 1, \dots, N_D\},$$

$$(2.2) \qquad H_V := \{u(\boldsymbol{x}) \,\big|\, u(\boldsymbol{x}) = u(\boldsymbol{x}_k^V) = u_k^V \ \forall \boldsymbol{x} \in D_k, \, k = 1, \dots, N_V\}.$$

Vector functions are approximated on the Delaunay mesh with vector grid functions, where the function space is denoted by $\boldsymbol{H_D}$. For vector function $\boldsymbol{u}(\boldsymbol{x})$, we project it along the Delaunay edge and evaluate at the intersection of the Delaunay edges and Voronoi face. The space of vector grid functions on the Voronoi mesh, $\boldsymbol{H_V}$, is defined analogously on the Voronoi mesh:

$$(2.3) \qquad \boldsymbol{H_D} := \{\boldsymbol{u}(\boldsymbol{x}) \,\big|\, \boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u} \cdot \boldsymbol{t}_{ij}^D(\boldsymbol{x}_{ij}^D) = u_{ij}^D, \ \boldsymbol{x}_{ij}^D = \boldsymbol{e}_{ij}^D \cap \partial V_{ij}\},$$

$$(2.4) \qquad \boldsymbol{H_V} := \{\boldsymbol{u}(\boldsymbol{x}) \,\big|\, \boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u} \cdot \boldsymbol{t}_{km}^V(\boldsymbol{x}_{km}^V) = u_{km}^V, \ \boldsymbol{x}_{km}^V = \boldsymbol{e}_{km}^V \cap \partial D_{km}\}.$$

To build intuition, we first introduce the MFD operators componentwise and then later give the matrix definitions. In the continuous setting, the gradient maps scalar functions to vector functions. Analogously, the discrete gradient on the Delaunay mesh maps a scalar grid function defined on the nodes to a vector grid function defined on the edges, or $\mathrm{grad}_D u : H_D \to \boldsymbol{H_D}$. On edge $\boldsymbol{e}_{ij}^D$,

$$(\mathrm{grad}_D u)_{ij}^D = \frac{u_j^D - u_i^D}{|\boldsymbol{e}_{ij}^D|} \, \eta(i, j),$$

where η is an orientation constant,

$$\eta(i, j) = \begin{cases} 1, & j > i, \\ -1 & \text{otherwise}. \end{cases}$$

Similarly, the gradient on the Voronoi mesh, $\mathrm{grad}_V u : H_V \to \boldsymbol{H_V}$, is given by

$$(\mathrm{grad}_V u)_{km}^V = \frac{u_m^V - u_k^V}{|\boldsymbol{e}_{km}^V|} \, \eta(k, m).$$
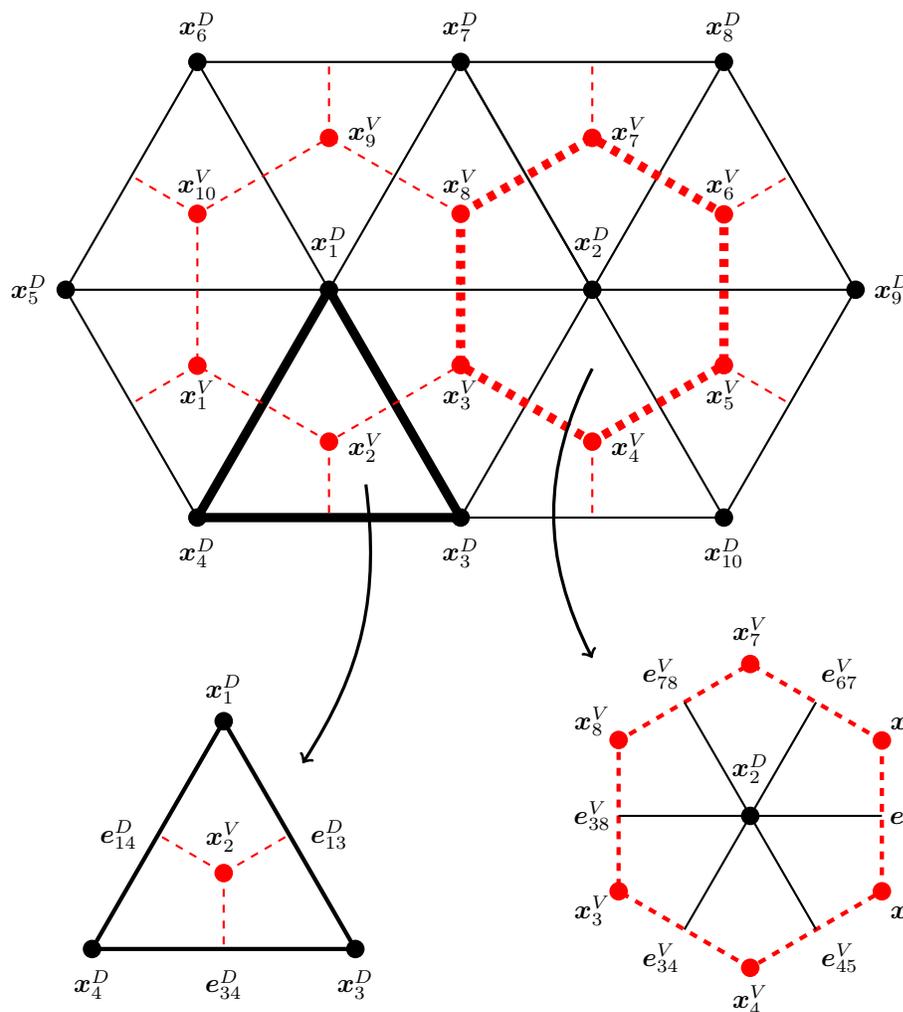
FIG. 1. *Top: Two-dimensional primal Delaunay mesh in black solid lines with corresponding dual Voronoi mesh in red dashed lines. Bottom left: Zoom in of Delaunay element D_2, corresponding to Voronoi node $\boldsymbol{x}_2^V$, with labeled Delaunay edges and nodes. Bottom right: Zoom in of Voronoi element V_2, corresponding to Delaunay node $\boldsymbol{x}_2^D$, with labeled Voronoi nodes and edges.*

To define the discrete divergence, first note that divergence maps vector functions to scalar functions. This differential operator on the Delaunay grid, $\mathrm{div}_D u : \boldsymbol{H_D} \to H_D$, corresponding to the outward flux of V_i is defined as

$$(\mathrm{div}_D u)_i^D = \frac{1}{|V_i|} \sum_{j \in \mathcal{N}_i^V} |\partial V_{ij}| u_{ij}^D (\boldsymbol{n}_{ij}^V \cdot \boldsymbol{t}_{ij}^D).$$

Similarly, on the Voronoi grid, the divergence $\mathrm{div}_V u : \boldsymbol{H_V} \to H_V$ is

$$(\mathrm{div}_V u)_k^V = \frac{1}{|D_k|} \sum_{m \in \mathcal{N}_k^D} |\partial D_{km}| u_{km}^V (\boldsymbol{n}_{km}^D \cdot \boldsymbol{t}_{km}^V).$$

The discrete curl operator maps from edges on one mesh (the circulation) to edges on the other mesh (the axis of rotation) by the geometric relationships between the dual meshes. Therefore, the Delaunay curl operator maps vector grid functions on the Delaunay mesh to a vector grid function on the Voronoi mesh, $\mathrm{curl}_D u : \boldsymbol{H_D} \to \boldsymbol{H_V}$, and is given by

$$(\mathrm{curl}_D u)_{km}^V = \frac{(\boldsymbol{t}_{km}^V \cdot \boldsymbol{n}_{km}^D)}{|\partial D_{km}|} \sum_{\boldsymbol{e}_{ij}^D \in \partial D_{km}} u_{ij}^D \, |\boldsymbol{e}_{ij}^D| \, \chi(\boldsymbol{n}_{km}^D, \boldsymbol{t}_{ij}^D),$$

where the constant χ essentially enforces the right-hand rule,

$$\chi(\boldsymbol{n}_{km}^D, \boldsymbol{t}_{ij}^D) = \begin{cases} 1, & \boldsymbol{t}_{ij}^D \text{ positively oriented,} \\ -1 & \text{otherwise.} \end{cases}$$

Similarly, we define the Voronoi curl operator $\mathrm{curl}_V u : \boldsymbol{H_V} \to \boldsymbol{H_D}$,

$$(\mathrm{curl}_V u)_{ij}^D = \frac{(\boldsymbol{t}_{ij}^D \cdot \boldsymbol{n}_{ij}^V)}{|\partial V_{ij}|} \sum_{\boldsymbol{e}_{km}^V \in \partial V_{ij}} u_{km}^V \, |\boldsymbol{e}_{km}^V| \, \chi(\boldsymbol{n}_{ij}^V, \boldsymbol{t}_{km}^V).$$

To define the MFD operators in matrix form, we introduce the edge-vertex signed incidence matrix, $\mathcal{G} \in \mathbb{R}^{M_D \times N_D}$, and the face-edge signed incidence matrix, $\mathcal{K} \in \mathbb{R}^{M_V \times M_D}$. Both are defined on the Delaunay triangulation, where N_D, M_V, and M_D denote the number of Delaunay nodes, Voronoi edges, and Delaunay edges, respectively. Similarly, on the Voronoi mesh, we have the signed incidence matrix $\mathcal{G}_V \in \mathbb{R}^{M_V \times N_V}$, where N_V denotes the number of nodes on the Voronoi mesh. The nonzero entries of $\mathcal{G}$, $\mathcal{G}_V$, and $\mathcal{K}$ are either 1 or -1, and the signs are consistent with the predetermined orientation of the edges and faces. Additionally, we introduce the following diagonal matrices encoding the mesh information pertaining to MFD:

$$\mathcal{D}_{\boldsymbol{e}^D} = \mathrm{diag}\left(|\boldsymbol{e}_{ij}^D|\right), \quad \mathcal{D}_{\partial D} = \mathrm{diag}\left(|\partial D_{km}|\right), \quad \mathcal{D}_D = \mathrm{diag}\left(|D_k|\right),$$

$$\mathcal{D}_{\boldsymbol{e}^V} = \mathrm{diag}\left(|\boldsymbol{e}_{km}^V|\right), \quad \mathcal{D}_{\partial V} = \mathrm{diag}\left(|\partial V_{ij}|\right), \quad \mathcal{D}_V = \mathrm{diag}\left(|V_i|\right).$$

The matrix representations are derived from the componentwise definitions using the incidence matrices as the actions of the operators and the diagonal matrices for the appropriate scaling. Thus,

$$(2.5) \qquad \mathrm{grad}_D := \mathcal{D}_{\boldsymbol{e}^D}^{-1} \mathcal{G}, \qquad \mathrm{div}_D := \mathcal{D}_V^{-1} \mathcal{G}^T \mathcal{D}_{\partial V}, \qquad \mathrm{curl}_D := \mathcal{D}_{\partial D}^{-1} \mathcal{K} \mathcal{D}_{\boldsymbol{e}^D},$$

$$(2.6) \qquad \mathrm{grad}_V := \mathcal{D}_{\boldsymbol{e}^V}^{-1} \mathcal{G}_V, \qquad \mathrm{div}_V := \mathcal{D}_D^{-1} \mathcal{G}_V^T \mathcal{D}_{\partial D}, \qquad \mathrm{curl}_V := \mathcal{D}_{\partial V}^{-1} \mathcal{K}^T \mathcal{D}_{\boldsymbol{e}^V}.$$

With this construction, it is known that the mimetic operators are structure-preserving, i.e., $\mathrm{curl}_D \mathrm{grad}_D = 0$, $\mathrm{curl}_V \mathrm{grad}_V = 0$, $\mathrm{div}_V \mathrm{curl}_D = 0$, and $\mathrm{div}_D \mathrm{curl}_V = 0$ [8, 31]. Using these relationships, two exact sequences exist for MFD,

$$(2.7) \qquad\qquad H_D \xrightarrow{\mathrm{grad}_D} \boldsymbol{H_D} \xrightarrow{\mathrm{curl}_D} \boldsymbol{H_V} \xrightarrow{\mathrm{div}_V} H_V,$$

$$(2.8) \qquad\qquad H_V \xrightarrow{\mathrm{grad}_V} \boldsymbol{H_V} \xrightarrow{\mathrm{curl}_V} \boldsymbol{H_D} \xrightarrow{\mathrm{div}_D} H_D.$$

Note again that the nature of the discretization method, when used for Maxwell's system, enforces the PDE constraints strongly at the discrete level.

2.2. MFD for Maxwell's equations. Since the energy conservation property is important in electromagnetic applications, we consider the Crank–Nicolson scheme (a symplectic time integrator) with time step τ and suitable initial conditions given by appropriate interpolation to the dual meshes, p_D^0, $\boldsymbol{E}_D^0$, and $\boldsymbol{B}_V^0$. The fully discretized system becomes the following: find $p_D^n \in H_D$, $\boldsymbol{E}_D^n \in \boldsymbol{H_D}$, and $\boldsymbol{B}_V^n \in \boldsymbol{H_V}$ such that

$$(2.9) \qquad \frac{2}{\tau}\boldsymbol{B}_V^n + \mathrm{curl}_D\,\boldsymbol{E}_D^n = \boldsymbol{g}_{\boldsymbol{B}}^V,$$

$$(2.10) \qquad \frac{2}{\tau}\boldsymbol{E}_D^n - \mathrm{curl}_V\,\boldsymbol{B}_V^n + \mathrm{grad}_D\,p_D^n = \boldsymbol{g}_{\boldsymbol{E}}^D,$$

$$(2.11) \qquad \frac{2}{\tau}p_D^n + \mathrm{div}_D\,\boldsymbol{E}_D^n = g_p^D,$$

where the functions on the right-hand sides are given by

$$\boldsymbol{g}_{\boldsymbol{B}}^V := \frac{2}{\tau}\boldsymbol{B}_V^{n-1} - \mathrm{curl}_D\,\boldsymbol{E}_D^{n-1},$$

$$\boldsymbol{g}_{\boldsymbol{E}}^D := \frac{2}{\tau}\boldsymbol{E}_D^{n-1} + \mathrm{curl}_V\,\boldsymbol{B}_V^{n-1} - \mathrm{grad}_D\,p_D^{n-1} - (\boldsymbol{j}_D^n + \boldsymbol{j}_D^{n-1}),$$

$$g_p^D := \frac{2}{\tau}p_D^{n-1} - \mathrm{div}_D\,\boldsymbol{E}_D^{n-1}.$$

The current density, $\boldsymbol{j}_D \in \boldsymbol{H_D}$, is $(\boldsymbol{j}_D)_{ij}^D = |\partial V_{ij}^D|^{-1} \int_{\partial V_{ij}} \boldsymbol{j} \cdot \boldsymbol{n}_{ij}^V\,d\boldsymbol{x}$.

Remark 2.1. Recall that $p(\boldsymbol{x}, t) = 0$ for all $\boldsymbol{x} \in \Omega$ and $t \geq 0$ with initial condition $p(\boldsymbol{x}, 0) = 0$. Therefore, (2.9)–(2.11) could be solved without including p, and the analysis that follows remains the same even without p. However, p is included to demonstrate the relationship between the grad, curl, and div spaces, which are chosen such that the equations obey the mappings given by the sequences (2.7)–(2.8). Also, note that we could have put $\boldsymbol{E}$ and p on the Voronoi mesh, and $\boldsymbol{B}$ on the Delaunay mesh instead of the choice above. However, putting the magnetic field on the Voronoi mesh, $\boldsymbol{B}_V \in \boldsymbol{H_V}$, guarantees $\mathrm{div}_V\,\boldsymbol{B}_V = 0$. Thus, the divergence of the magnetic field is constant zero on the Voronoi nodes, which gives us a divergence-free magnetic field on all Delaunay tetrahedra.

Using the definitions of the MFD operators, the linear system for the MFD scheme (2.9)–(2.11) is given by

$$(2.12) \qquad \underbrace{\begin{bmatrix} \frac{2}{\tau}\mathcal{I}_{\boldsymbol{e}^V} & \mathcal{D}_{\partial D}^{-1}\,\mathcal{K}\,\mathcal{D}_{\boldsymbol{e}^D} & \\ -\mathcal{D}_{\partial V}^{-1}\,\mathcal{K}^T\,\mathcal{D}_{\boldsymbol{e}^V} & \frac{2}{\tau}\mathcal{I}_{\boldsymbol{e}^D} & \mathcal{D}_{\boldsymbol{e}^D}^{-1}\,\mathcal{G} \\ & -\mathcal{D}_V^{-1}\,\mathcal{G}^T\,\mathcal{D}_{\partial V} & \frac{2}{\tau}\mathcal{I}_V \end{bmatrix}}_{=:\mathcal{A}_{\mathrm{MFD}}} \begin{bmatrix} \boldsymbol{B}_V^n \\ \boldsymbol{E}_D^n \\ p_D^n \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}_{\boldsymbol{B}}^V \\ \boldsymbol{g}_{\boldsymbol{E}}^D \\ g_p^D \end{bmatrix}.$$

3. Finite-element framework. Next, we consider a structure-preserving mixed FEM for the Maxwell system [4]. To approximate the inner-product terms on the computational domain, we implement mass lumping, which results in diagonal mass matrices. This gives us FE blocks in terms of MFD mesh information, which is useful when drawing connections in the next section.

Consider the differential operator, $\mathfrak{D}$, and Sobolev space,

$$H(\mathfrak{D}) := \{u \in L^2(\Omega), \mathfrak{D}u \in L^2(\Omega)\},$$

where $\mathfrak{D}$ is grad, curl, or div. Let $\|\cdot\|$ and $\langle\cdot,\cdot\rangle$ denote the L^2 norm and inner product, respectively. Define the finite-dimensional function spaces with appropriate boundary conditions, $H_{h,0}(\mathfrak{D})$. For the magnetic field, $\boldsymbol{B}_h \in \boldsymbol{H}_{h,0}(\mathrm{div})$, we use the Raviart–Thomas (RT) element, the Nédélec element for the electric field $\boldsymbol{E}_h \in \boldsymbol{H}_{h,0}(\mathrm{curl})$, and the Lagrange element for the auxiliary pressure $p_h \in H_{h,0}(\mathrm{grad})$. Defining $\boldsymbol{V}_h := \boldsymbol{H}_{h,0}(\mathrm{div}) \times \boldsymbol{H}_{h,0}(\mathrm{curl}) \times H_{h,0}(\mathrm{grad})$, the FEM discretization for Maxwell's equations becomes the following: find $(\boldsymbol{B}_h^n, \boldsymbol{E}_h^n, p_h^n) \in \boldsymbol{V}_h$ such that for all $(\boldsymbol{C}_h, \boldsymbol{F}_h, q_h) \in \boldsymbol{V}_h$,

$$(3.1) \qquad \frac{2}{\tau}\langle \boldsymbol{B}_h^n, \boldsymbol{C}_h\rangle + \langle \nabla \times \boldsymbol{E}_h^n, \boldsymbol{C}_h\rangle = (\boldsymbol{g_B}, \boldsymbol{C}_h),$$

$$(3.2) \qquad \frac{2}{\tau}\langle \boldsymbol{E}_h^n, \boldsymbol{F}_h\rangle - \langle \boldsymbol{B}_h^n, \nabla \times \boldsymbol{F}_h\rangle + \langle \nabla p_h^n, \boldsymbol{F}_h\rangle = (\boldsymbol{g_E}, \boldsymbol{F}_h),$$

$$(3.3) \qquad \frac{2}{\tau}\langle p_h^n, q_h\rangle - \langle \boldsymbol{E}_h^n, \nabla q_h\rangle = (g_p, q_h),$$

where the functionals on the right-hand side are given by

$$(\boldsymbol{g_B}, \boldsymbol{C}_h) = \frac{2}{\tau}\langle \boldsymbol{B}_h^{n-1}, \boldsymbol{C}_h\rangle - \langle \nabla \times \boldsymbol{E}_h^{n-1}, \boldsymbol{C}_h\rangle,$$

$$(\boldsymbol{g_E}, \boldsymbol{F}_h) = \frac{2}{\tau}\langle \boldsymbol{E}_h^{n-1}, \boldsymbol{F}_h\rangle - \langle \nabla p_h^{n-1}, \boldsymbol{F}_h\rangle + \langle \boldsymbol{B}_h^{n-1}, \nabla \times \boldsymbol{F}_h\rangle - \langle \boldsymbol{j}^n + \boldsymbol{j}^{n-1}, \boldsymbol{F}_h\rangle,$$

$$(g_p, q_h) = \frac{2}{\tau}\langle p_h^{n-1}, q_h\rangle + \langle \boldsymbol{E}_h^{n-1}, \nabla q_h\rangle.$$

To write this as a linear system, we introduce discrete gradient and curl operators. Let $\{\phi_i^{\mathrm{grad}}\}$, $\{\boldsymbol{\phi}_i^{\mathrm{curl}}\}$, and $\{\boldsymbol{\phi}_i^{\mathrm{div}}\}$ be the bases of $H_{h,0}(\mathrm{grad})$, $\boldsymbol{H}_{h,0}(\mathrm{curl})$, and $\boldsymbol{H}_{h,0}(\mathrm{div})$, respectively. Let $\{\boldsymbol{\eta}_i^{\mathrm{curl}}\}$ be the degrees of freedom for the Nédélec space and $\{\boldsymbol{\eta}_i^{\mathrm{div}}\}$ for the RT space. Then define the FE discrete gradient, $\mathcal{G}^{FE}$, and curl, $\mathcal{K}^{FE}$, in terms of the degrees of freedom,

$$(3.4) \qquad \mathcal{G}_{ij}^{FE} := \boldsymbol{\eta}_i^{\mathrm{curl}}\left(\nabla \phi_j^{\mathrm{grad}}\right) = \frac{1}{|e_i|}\int_{e_i} \nabla \phi_j^{\mathrm{grad}} \cdot \boldsymbol{t}_i \, ds,$$

$$(3.5) \qquad \mathcal{K}_{ij}^{FE} := \boldsymbol{\eta}_i^{\mathrm{div}}\left(\nabla \times \boldsymbol{\phi}_j^{\mathrm{curl}}\right) = \frac{1}{|f_i|}\int_{f_i} \nabla \times \boldsymbol{\phi}_j^{\mathrm{curl}} \cdot \boldsymbol{n}_i \, dS.$$

Note that the degrees of freedom are scaled by mesh data, the curl degrees of freedom are scaled by inverse edge lengths, and div degrees of freedom are scaled by inverse face areas. Thus, the FE linear system for (3.1)–(3.3) is given by

$$(3.6) \qquad \underbrace{\begin{bmatrix} \frac{2}{\tau}\mathcal{M}_{\boldsymbol{B}} & \mathcal{M}_{\boldsymbol{B}}\mathcal{K}^{FE} & \\ -\left(\mathcal{K}^{FE}\right)^T \mathcal{M}_{\boldsymbol{B}} & \frac{2}{\tau}\mathcal{M}_{\boldsymbol{E}} & \mathcal{M}_{\boldsymbol{E}}\mathcal{G}^{FE} \\ & -\left(\mathcal{G}^{FE}\right)^T \mathcal{M}_{\boldsymbol{E}} & \frac{2}{\tau}\mathcal{M}_p \end{bmatrix}}_{=:\mathcal{A}_{\mathrm{FE}}} \begin{bmatrix} \boldsymbol{B}_h^n \\ \boldsymbol{E}_h^n \\ p_h^n \end{bmatrix} = \begin{bmatrix} \boldsymbol{g_B} \\ \boldsymbol{g_E} \\ g_p \end{bmatrix},$$

with mass matrices given by $(\mathcal{M}_p)_{ij} = \langle \phi_j^{\mathrm{grad}}, \phi_i^{\mathrm{grad}}\rangle$, $(\mathcal{M}_{\boldsymbol{E}})_{ij} = \langle \boldsymbol{\phi}_j^{\mathrm{curl}}, \boldsymbol{\phi}_i^{\mathrm{curl}}\rangle$, and $(\mathcal{M}_{\boldsymbol{B}})_{ij} = \langle \boldsymbol{\phi}_j^{\mathrm{div}}, \boldsymbol{\phi}_i^{\mathrm{div}}\rangle$.

3.1. Mass lumping. Now that we have linear systems for two discretization methods, our goal is to draw similarities between (2.12) and (3.6). Notice that the discrete differential operators are already in the same blocks; however, the MFD

system has blocks with diagonal entries containing mesh information while the FEM system has mass matrices. Using ideas from [6, 7, 13, 30] to approximate the mass matrices with diagonal matrices, we implement mass-lumping schemes. The quadrature schemes associated with the lumping result in a quantity that has physical significance, namely a volume unit for three-dimensional meshes and an area on the mesh for the two-dimensional case. Therefore, we expect that mass lumping gives diagonal matrices where the diagonal represents some sort of volume (three-dimensional) or area (two-dimensional) in the dual mesh setup. We see in the next section that this is true for appropriate choices of quadrature weights and that these specific lumped matrices help draw more connections between the FE and MFD systems.

First, consider the Lagrange element mass matrix, $\mathcal{M}_p$. For simplicity, assume we have a regular primal (Delaunay) mesh for the FE grid and an associated dual (Voronoi) mesh. The entries of $\mathcal{M}_p$ are integrals of the Lagrange basis functions but can be rewritten in terms of a quadrature rule,

$$(\mathcal{M}_p)_{ij} = \langle \phi_i^{\mathrm{grad}}, \phi_j^{\mathrm{grad}} \rangle = \sum_{l=1}^{N_V} \int_{D_l} \phi_i^{\mathrm{grad}} \phi_j^{\mathrm{grad}} d\boldsymbol{x} \approx \sum_{l=1}^{N_V} \sum_{k=1}^{d+1} \omega_{lk}\ \phi_i^{\mathrm{grad}}(x_{lk}) \phi_j^{\mathrm{grad}}(x_{lk}),$$

where d is the dimension, the sum in l is over all of the elements, $d+1$ is the number of nodes per element, and ω_{lk} and x_{lk} are the quadrature weights and nodes, respectively. Typically, Gaussian quadrature is used, but we manufacture a new rule such that the choice of nodes and weights gives the approximation, $\mathcal{M}_p \approx \mathcal{D}_V$.
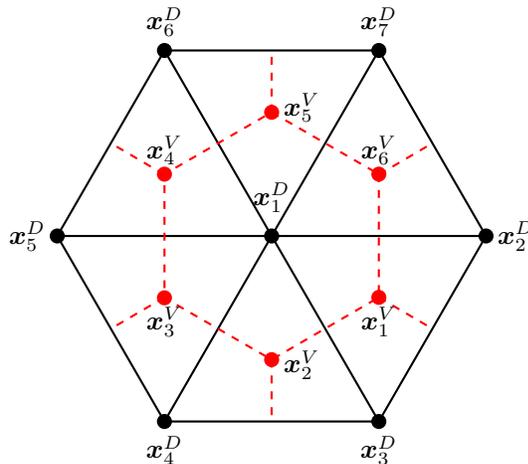

FIG. 2. *Two-dimensional subset of regular dual mesh (as in Figure* 1*). Delaunay mesh in black solid lines and Voronoi mesh in red dashed line.*

To illustrate this further, examine a subset of a regular dual mesh setup in two dimensions, shown in Figure 2. On the submesh, we choose the quadrature points x_{lk} to be the Delaunay nodes, $\boldsymbol{x}_m^D$, and then determine what the appropriate weights should be. Consider two cases: $i = j$ and $i \neq j$. When $i \neq j$, we have that all terms in the sum are zero, as $\phi_i^{\mathrm{grad}}(\boldsymbol{x}_n^D) = 0$ if $i \neq n$ and $\phi_j^{\mathrm{grad}}(\boldsymbol{x}_n^D) = 0$ if $j \neq n$ by the definition of the FE basis functions and degrees of freedom. The only way to get a nonzero term in the sum is if $i = j = n$. When $i = j$, note that each term in the sum is only nonzero when $x_{lk} = \boldsymbol{x}_i^D$. Consider the small mesh in Figure 2 and the case

when $i = j = 1$. Define the element enumeration using MFD notation; i.e., let D_1 be the Delaunay element defined by Voronoi node $\boldsymbol{x}_1^V$, and let D_l be defined by $\boldsymbol{x}_l^V$. Let k enumerate the nodes in an element in increasing order. For example, for $l = 1$, we have $x_{11} = \boldsymbol{x}_1^D$, $x_{12} = \boldsymbol{x}_2^D$, and $x_{13} = \boldsymbol{x}_3^D$ in Figure 2. The quadrature rule on this submesh becomes

$$\sum_{l=1}^{N_V} \sum_{k=1}^{d+1} \omega_{lk}\, \phi_1^{\text{grad}}(x_{lk})\phi_1^{\text{grad}}(x_{lk}) = \omega_{11}\phi_1^{\text{grad}}(x_{11})\phi_1^{\text{grad}}(x_{11}) + \omega_{21}\phi_1^{\text{grad}}(x_{21})\phi_1^{\text{grad}}(x_{21})$$

$$+ \omega_{31}\phi_1^{\text{grad}}(x_{31})\phi_1^{\text{grad}}(x_{31}) + \omega_{41}\phi_1^{\text{grad}}(x_{41})\phi_1^{\text{grad}}(x_{41})$$

$$+ \omega_{51}\phi_1^{\text{grad}}(x_{51})\phi_1^{\text{grad}}(x_{51}) + \omega_{61}\phi_1^{\text{grad}}(x_{61})\phi_1^{\text{grad}}(x_{61})$$

$$= \omega_{11} + \omega_{21} + \omega_{31} + \omega_{41} + \omega_{51} + \omega_{61}.$$

To get the correct approximation by mass lumping, this sum must equal $|V_1|$, the area of the Voronoi cell defined by the six Voronoi nodes. Therefore, we choose $\omega_{11} = |V_1 \cap D_1|$, $\omega_{21} = |V_1 \cap D_2|$, ..., $\omega_{61} = |V_1 \cap D_6|$. Substituting in the weights we see that

$$\sum_{l=1}^{N_V} \sum_{k=1}^{d+1} \omega_{lk}\, \phi_1^{\text{grad}}(x_{lk})\phi_1^{\text{grad}}(x_{lk}) = \omega_{11} + \omega_{21} + \omega_{31} + \omega_{41} + \omega_{51} + \omega_{61}$$

$$= |V_1 \cap D_1| + |V_1 \cap D_2| + \cdots + |V_1 \cap D_6|$$

$$= |V_1|.$$

The same argument can be applied more generally to the $i = j$ case, where we want $\omega_{lk} = |D_l \cap V_i|$ when $x_{lk} = \boldsymbol{x}_i^D$. This gives us the approximation $(\mathcal{M}_p)_{ii} \approx |V_i|$. The same idea applies to the three-dimensional case, except we have four nodes in each element and the weights represent a volume instead of an area.

In general, for the Lagrange elements with mass matrix $\mathcal{M}_p$, we use the following quadrature rule to lump the entries onto the diagonal. Summing over elements and nodes per element, we have for scalar functions $u, v \in H(\text{grad})$,

$$\langle u, v \rangle_{\widetilde{\mathcal{M}_p}} := \sum_{l=1}^{N_V} \sum_{k=1}^{d+1} \omega_{lk}^{\text{grad}}\, u(x_{lk})v(x_{lk}), \quad \|u\|_{\widetilde{\mathcal{M}_p}}^2 := \langle u, u \rangle_{\widetilde{\mathcal{M}_p}},$$

where the x_{lk} are given by the nodes of the Delaunay mesh, and $\omega_{lk}^{\text{grad}} = |D_l \cap V_i|$ when $x_{lk} = \boldsymbol{x}_i^D$. Thus, we have that $\mathcal{M}_p \approx \widetilde{\mathcal{M}_p} = \mathcal{D}_V$, which is the standard mass-lumping scheme for $H(\text{grad})$ [30].

Similarly for the RT elements, we introduce the following inner product, summing over elements and faces per element, for $\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{H}(\text{div})$:

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\widetilde{\mathcal{M}_B}} := \sum_{l=1}^{N_V} \sum_{k=1}^{d+1} \omega_{lk}^{\text{div}} \left(\frac{1}{f_{lk}} \int_{f_{lk}} \boldsymbol{u} \cdot \hat{\boldsymbol{n}}\, dS \right) \left(\frac{1}{f_{lk}} \int_{f_{lk}} \boldsymbol{v} \cdot \hat{\boldsymbol{n}}\, dS \right),$$

$$\|\boldsymbol{u}\|_{\widetilde{\mathcal{M}_B}}^2 := \langle \boldsymbol{u}, \boldsymbol{u} \rangle_{\widetilde{\mathcal{M}_B}}.$$

Using the same idea as the previous lumping scheme and following [7], we choose quadrature weights to be $\omega_{lk}^{\text{div}} = \left| \boldsymbol{e}_{lm}^V \cap D_l \right| |\partial D_{lm}|$ when $f_{lk} = \partial D_{lm}$, where ∂D_{lm} is the ith face in the Delaunay mesh enumeration. Then the mass matrix is approximated by $\mathcal{M}_B \approx \widetilde{\mathcal{M}_B} = \mathcal{D}_{\partial D}\mathcal{D}_{\boldsymbol{e}^V}$.

Finally, we examine the Nédélec element mass matrix, $\mathcal{M}_{\boldsymbol{E}}$, by following [6]. We have the entries computed with the following quadrature rule: for $\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{H}(\mathrm{curl})$,

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\widetilde{\mathcal{M}_{\boldsymbol{E}}}} := \sum_{l=1}^{N_D} \sum_{k=1}^{\frac{d(d+1)}{2}} \omega_{lk}^{\mathrm{curl}} \left(\frac{1}{e_{lk}} \int_{e_{lk}} \boldsymbol{u} \cdot \hat{\boldsymbol{t}} \, ds \right) \left(\frac{1}{e_{lk}} \int_{e_{lk}} \boldsymbol{v} \cdot \hat{\boldsymbol{t}} \, ds \right),$$

$$\|\boldsymbol{u}\|_{\widetilde{\mathcal{M}_{\boldsymbol{E}}}}^2 := \langle \boldsymbol{u}, \boldsymbol{u} \rangle_{\widetilde{\mathcal{M}_{\boldsymbol{E}}}},$$

where edge $e_{lk} = \boldsymbol{e}_{mn}^D$ is the ith edge in the Delaunay mesh enumeration, and $e_{lk} = \boldsymbol{e}_{mn}^D$ is contained in Delaunay tetrahedron D_l. Choosing weights to be $\omega_{lk}^{\mathrm{curl}} = |\partial V_{mn} \cap D_l| \, |\boldsymbol{e}_{mn}^D|$ yields $\mathcal{M}_{\boldsymbol{E}} \approx \widetilde{\mathcal{M}_{\boldsymbol{E}}} = \mathcal{D}_{\partial V} \mathcal{D}_{\boldsymbol{e}^D}$.

Remark 3.1. The mass-lumping schemes can be modified for nonconstant $\epsilon(\boldsymbol{x})$ and $\mu(\boldsymbol{x})$ by taking a constant approximation of the coefficients on each element. With the piecewise constant approximation, the quadrature weights and sparsity of the mass-lumping schemes are unchanged.

Putting this all together, we have the mass-lumped FE system as

$$(3.7) \qquad \underbrace{\begin{bmatrix} \frac{2}{\tau} \widetilde{\mathcal{M}_{\boldsymbol{B}}} & \widetilde{\mathcal{M}_{\boldsymbol{B}}} \mathcal{K}^{FE} & \\ -\left(\mathcal{K}^{FE}\right)^T \widetilde{\mathcal{M}_{\boldsymbol{B}}} & \frac{2}{\tau} \widetilde{\mathcal{M}_{\boldsymbol{E}}} & \widetilde{\mathcal{M}_{\boldsymbol{E}}} \mathcal{G}^{FE} \\ & -\left(\mathcal{G}^{FE}\right)^T \widetilde{\mathcal{M}_{\boldsymbol{E}}} & \frac{2}{\tau} \widetilde{\mathcal{M}_p} \end{bmatrix}}_{=:\widetilde{\mathcal{A}}_{\mathrm{FE}}} \begin{bmatrix} \boldsymbol{B}_h^n \\ \boldsymbol{E}_h^n \\ p_h^n \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}_B \\ \boldsymbol{g}_E \\ g_p \end{bmatrix}.$$

4. Connections between MFD and FEM. To study the well-posedness of the mimetic discretization (2.9)–(2.11) and design efficient solvers for the resulting linear system, we draw connections to the FE scheme, noting that with mass lumping, the two systems have the same block structure. First, we rewrite the FE gradient and curl operators in terms of the MFD incidence matrices as follows:

$$(4.1) \qquad\qquad \mathcal{G}^{FE} = \mathcal{D}_{\boldsymbol{e}^D}^{-1} \mathcal{G}, \qquad \mathcal{K}^{FE} = \mathcal{D}_{\partial D}^{-1} \mathcal{K} \mathcal{D}_{\boldsymbol{e}^D},$$

where we scale incident matrices on the mesh to be consistent with the definitions in (3.4)–(3.5). Applying a left scaling to the mass-lumping schemes, and substituting in (4.1) to (3.7), gives a new scaled FE linear system,

$$(4.2)$$
$$\underbrace{\begin{bmatrix} \mathcal{D}_{\partial D}^{-1} \mathcal{D}_{\boldsymbol{e}^V}^{-1} & & \\ & \mathcal{D}_{\partial V}^{-1} \mathcal{D}_{\boldsymbol{e}^D}^{-1} & \\ & & \mathcal{D}_V^{-1} \end{bmatrix}}_{=:\mathcal{A}_{\mathrm{SFE}}} \widetilde{\mathcal{A}}_{\mathrm{FE}} \begin{bmatrix} \boldsymbol{B}_h^n \\ \boldsymbol{E}_h^n \\ p_h^n \end{bmatrix} = \begin{bmatrix} \mathcal{D}_{\partial D}^{-1} \mathcal{D}_{\boldsymbol{e}^V}^{-1} & & \\ & \mathcal{D}_{\partial V}^{-1} \mathcal{D}_{\boldsymbol{e}^D}^{-1} & \\ & & \mathcal{D}_V^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{g}_B \\ \boldsymbol{g}_E \\ g_p \end{bmatrix}.$$

Substituting in $\widetilde{\mathcal{M}_p} = D_V$, $\widetilde{\mathcal{M}_{\boldsymbol{B}}} = \mathcal{D}_{\partial D} \mathcal{D}_{\boldsymbol{e}^V}$, and $\widetilde{\mathcal{M}_{\boldsymbol{E}}} = \mathcal{D}_{\partial V} \mathcal{D}_{\boldsymbol{e}^D}$, we get $\mathcal{A}_{\mathrm{SFE}} = \mathcal{A}_{\mathrm{MFD}}$ and recover *exactly* the MFD system in (2.12).

With the above equivalence, we apply an FE well-posedness proof to the mass-lumped FE system, thus obtaining the well-posedness of the mimetic system. For simplicity in dealing with the scaled system, we also introduce the function space, $\widetilde{\boldsymbol{V}_h} := \widetilde{\boldsymbol{H}_{h,0}}(\mathrm{div}) \times \widetilde{\boldsymbol{H}_{h,0}}(\mathrm{curl}) \times \widetilde{H_{h,0}}(\mathrm{grad})$, which is $\boldsymbol{V}_h$ with basis functions scaled to reflect the left scaling in (4.2):

$$\widetilde{\boldsymbol{H}}_{h,0}(\mathrm{div}) = \mathrm{span}\{\widetilde{\boldsymbol{\phi}}_i^{\mathrm{div}}\}, \quad \text{where} \quad \widetilde{\boldsymbol{\phi}}_i^{\mathrm{div}} = \left(\mathcal{D}_{\partial D}^{-1}\mathcal{D}_{eV}^{-1}\right)_{ii} \boldsymbol{\phi}_i^{\mathrm{div}};$$

$$\widetilde{\boldsymbol{H}}_{h,0}(\mathrm{curl}) = \mathrm{span}\{\widetilde{\boldsymbol{\phi}}_i^{\mathrm{curl}}\}, \quad \text{where} \quad \widetilde{\boldsymbol{\phi}}_i^{\mathrm{curl}} = \left(\mathcal{D}_{\partial V}^{-1}\mathcal{D}_{\boldsymbol{e}^D}^{-1}\right)_{ii} \boldsymbol{\phi}_i^{\mathrm{curl}};$$

$$\widetilde{H}_{h,0}(\mathrm{grad}) = \mathrm{span}\{\widetilde{\phi}_i^{\mathrm{grad}}\}, \quad \text{where} \quad \widetilde{\phi}_i^{\mathrm{grad}} = \left(\mathcal{D}_V^{-1}\right)_{ii} \phi_i^{\mathrm{grad}}.$$

Now (4.2), and thus the MFD Maxwell system (2.12), can be written in variational form (for simplicity, the subscripts indicating the time-step iteration and inclusion in the FE space are excluded) as follows: find $(\boldsymbol{B}, \boldsymbol{E}, p) \in \boldsymbol{V}_h$ such that for all $(\widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q}) \in \widetilde{\boldsymbol{V}}_h$,

$$(4.3) \qquad \frac{2}{\tau}\langle \boldsymbol{B}, \widetilde{\boldsymbol{C}}\rangle_{\widetilde{\mathcal{M}}_B} + \langle \nabla \times \boldsymbol{E}, \widetilde{\boldsymbol{C}}\rangle_{\widetilde{\mathcal{M}}_B} = (\boldsymbol{g}_B, \widetilde{\boldsymbol{C}}),$$

$$(4.4) \qquad -\langle \boldsymbol{B}, \nabla \times \widetilde{\boldsymbol{F}}\rangle_{\widetilde{\mathcal{M}}_B} + \frac{2}{\tau}\langle \boldsymbol{E}, \widetilde{\boldsymbol{F}}\rangle_{\widetilde{\mathcal{M}}_E} + \langle \nabla p, \widetilde{\boldsymbol{F}}\rangle_{\widetilde{\mathcal{M}}_E} = (\boldsymbol{g}_E, \widetilde{\boldsymbol{F}}),$$

$$(4.5) \qquad -\langle \boldsymbol{E}, \nabla\widetilde{q}\rangle_{\widetilde{\mathcal{M}}_E} + \frac{2}{\tau}\langle p, \widetilde{q}\rangle_{\widetilde{\mathcal{M}}_p} = (g_p, \widetilde{q}).$$

Note that MFD in this case can be considered as a Petrov–Galerkin FEM.

Following the notation in [2], we introduce the bilinear form

$$(4.6)$$
$$a(\boldsymbol{B}, \boldsymbol{E}, p; \widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q}) := \frac{2}{\tau}\langle \boldsymbol{B}, \widetilde{\boldsymbol{C}}\rangle_{\widetilde{\mathcal{M}}_B} + \langle \nabla \times \boldsymbol{E}, \widetilde{\boldsymbol{C}}\rangle_{\widetilde{\mathcal{M}}_B} - \langle \boldsymbol{B}, \nabla \times \widetilde{\boldsymbol{F}}\rangle_{\widetilde{\mathcal{M}}_B}$$
$$+ \frac{2}{\tau}\langle \boldsymbol{E}, \widetilde{\boldsymbol{F}}\rangle_{\widetilde{\mathcal{M}}_E} + \langle \nabla p, \widetilde{\boldsymbol{F}}\rangle_{\widetilde{\mathcal{M}}_B} - \langle \boldsymbol{E}, \nabla\widetilde{q}\rangle_{\widetilde{\mathcal{M}}_E} + \frac{2}{\tau}\langle p, \widetilde{q}\rangle_{\widetilde{\mathcal{M}}_p}$$

and the weighted norms

$$(4.7) \qquad \|\boldsymbol{B}\|_{\mathrm{div}}^2 := \frac{2}{\tau}\|\boldsymbol{B}\|_{\widetilde{\mathcal{M}}_B}^2 + \|\nabla \cdot \boldsymbol{B}\|^2,$$

$$(4.8) \qquad \|\boldsymbol{E}\|_{\mathrm{curl}}^2 := \frac{2}{\tau}\|\boldsymbol{E}\|_{\widetilde{\mathcal{M}}_E}^2 + \frac{\tau}{2}\|\nabla \times \boldsymbol{E}\|_{\widetilde{\mathcal{M}}_B}^2,$$

$$(4.9) \qquad \|p\|_{\mathrm{grad}}^2 := \frac{2}{\tau}\|p\|_{\widetilde{\mathcal{M}}_p}^2 + \frac{\tau}{2}\|\nabla p\|_{\widetilde{\mathcal{M}}_E}^2,$$

$$(4.10) \qquad \|\,(\boldsymbol{B}, \boldsymbol{E}, p)\,\|^2 := \|\boldsymbol{B}\|_{\mathrm{div}}^2 + \|\boldsymbol{E}\|_{\mathrm{curl}}^2 + \|p\|_{\mathrm{grad}}^2.$$

The following theorem shows that (4.3)–(4.5) is well-posed, and therefore (2.9)–(2.11) is well-posed.

THEOREM 4.1. *If $\boldsymbol{g}_B \in (\boldsymbol{H}_{h,0}(\mathrm{div}))'$, the MFD system* (4.3)–(4.5) *is well-posed; namely, it satisfies the inf-sup condition,*

$$(4.11) \qquad \sup_{\substack{(\widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q}) \in \widetilde{\boldsymbol{V}}_h \\ (\widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q}) \neq \boldsymbol{0}}} \frac{a(\boldsymbol{B}, \boldsymbol{E}, p; \widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q})}{\|(\widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q})\|} \geq \frac{1}{4}\|\,(\boldsymbol{B}, \boldsymbol{E}, p)\,\|,$$

and is bounded:

$$(4.12) \qquad a(\boldsymbol{B}, \boldsymbol{E}, p; \widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q}) \leq C\|\boldsymbol{B}, \boldsymbol{E}, p\|\,\|(\widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q})\|.$$

Proof. Consider a variation of (4.6), where the term $\langle \nabla \cdot \boldsymbol{B}, \nabla \cdot \widetilde{\boldsymbol{C}}\rangle$ is added:

$$(4.13) \qquad \hat{a}(\boldsymbol{B}, \boldsymbol{E}, p; \widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q}) := a(\boldsymbol{B}, \boldsymbol{E}, p; \widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q}) + \langle \nabla \cdot \boldsymbol{B}, \nabla \cdot \widetilde{\boldsymbol{C}}\rangle.$$

Since $\nabla \cdot \boldsymbol{B} = 0$ for all $t \geq 0$, (4.6) and (4.13) are equivalent. We proceed by showing that (4.13) satisfies the *inf-sup* condition,

$$(4.14) \qquad \sup_{\boldsymbol{0} \neq (\widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q}) \in \widetilde{\boldsymbol{V}_h}} \frac{\hat{a}(\boldsymbol{B}, \boldsymbol{E}, p; \widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q})}{\|\|(\widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q})\|\|} \geq \frac{1}{4} \|\|(\boldsymbol{B}, \boldsymbol{E}, p)\|\|,$$

and is bounded:

$$(4.15) \qquad \hat{a}(\boldsymbol{B}, \boldsymbol{E}, p; \widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q}) \leq C \|\|(\boldsymbol{B}, \boldsymbol{E}, p)\|\| \, \|\|(\widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q})\|\|.$$

Let $\widetilde{\boldsymbol{C}} = \boldsymbol{B} + \frac{\tau}{2} \nabla \times \boldsymbol{E}$, $\widetilde{\boldsymbol{F}} = \boldsymbol{E} + \frac{\tau}{2} \nabla p$, and $\widetilde{q} = p$. Substituting into the bilinear form (4.13), using the fact that $\nabla \cdot \nabla \times \boldsymbol{E} = 0$ and $\nabla \times \nabla p = \boldsymbol{0}$, and simplifying, we have

$$\hat{a}(\boldsymbol{B}, \boldsymbol{E}, p; \widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q}) = \frac{2}{\tau} \|\boldsymbol{B}\|^2_{\widetilde{\mathcal{M}}_B} + \|\nabla \cdot \boldsymbol{B}\|^2 + \frac{2}{\tau} \|\boldsymbol{E}\|^2_{\widetilde{\mathcal{M}}_E} + \frac{\tau}{2} \|\nabla \times \boldsymbol{E}\|^2_{\widetilde{\mathcal{M}}_B}$$
$$+ \frac{2}{\tau} \|p\|^2_{\widetilde{\mathcal{M}}_p} + \frac{\tau}{2} \|\nabla p\|^2_{\widetilde{\mathcal{M}}_E} + \langle \nabla \times \boldsymbol{E}, \boldsymbol{B} \rangle_{\widetilde{\mathcal{M}}_B} + \langle \nabla p, \boldsymbol{E} \rangle_{\widetilde{\mathcal{M}}_E}.$$

Using Young's inequality to bound the cross terms, it follows that

$$\hat{a}(\boldsymbol{B}, \boldsymbol{E}, p; \widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q}) \geq \frac{2}{\tau} \|\boldsymbol{B}\|^2_{\widetilde{\mathcal{M}}_B} + \|\nabla \cdot \boldsymbol{B}\|^2 + \frac{2}{\tau} \|\boldsymbol{E}\|^2_{\widetilde{\mathcal{M}}_E} + \frac{\tau}{2} \|\nabla \times \boldsymbol{E}\|^2_{\widetilde{\mathcal{M}}_B}$$
$$+ \frac{2}{\tau} \|p\|^2_{\widetilde{\mathcal{M}}_p} + \frac{\tau}{2} \|\nabla p\|^2_{\widetilde{\mathcal{M}}_E} - \frac{\tau}{4} \|\nabla \times \boldsymbol{E}\|^2_{\widetilde{\mathcal{M}}_B} - \frac{1}{\tau} \|\boldsymbol{B}\|^2_{\widetilde{\mathcal{M}}_B}$$
$$- \frac{1}{\tau} \|\boldsymbol{E}\|^2_{\widetilde{\mathcal{M}}_E} - \frac{\tau}{4} \|\nabla p\|^2_{\widetilde{\mathcal{M}}_E}$$
$$= \frac{1}{\tau} \|\boldsymbol{B}\|^2_{\widetilde{\mathcal{M}}_B} + \|\nabla \cdot \boldsymbol{B}\|^2 + \frac{1}{\tau} \|\boldsymbol{E}\|^2_{\widetilde{\mathcal{M}}_E} + \frac{\tau}{4} \|\nabla \times \boldsymbol{E}\|^2_{\widetilde{\mathcal{M}}_B}$$
$$+ \frac{2}{\tau} \|p\|^2_{\widetilde{\mathcal{M}}_p} + \frac{\tau}{4} \|\nabla p\|^2_{\widetilde{\mathcal{M}}_E}$$
$$\geq \frac{1}{2} \|\|(\boldsymbol{B}, \boldsymbol{E}, p)\|\|^2.$$

Next, bound the norms of the test functions using the triangle inequality and Young's inequality:

$$\|\|(\widetilde{\boldsymbol{C}}, \widetilde{\boldsymbol{F}}, \widetilde{q})\|\|^2 = \|\boldsymbol{B} + \frac{\tau}{2} \nabla \times \boldsymbol{E}\|^2_{\mathrm{div}} + \|\boldsymbol{E} + \frac{\tau}{2} \nabla p\|^2_{\mathrm{curl}} + \|p\|^2_{\mathrm{grad}}$$
$$\leq \|\boldsymbol{B}\|^2_{\mathrm{div}} + \frac{\tau^2}{4} \|\nabla \times \boldsymbol{E}\|^2_{\mathrm{div}} + \tau \|\boldsymbol{B}\|_{\mathrm{div}} \|\nabla \times \boldsymbol{E}\|_{\mathrm{div}}$$
$$+ \|\boldsymbol{E}\|^2_{\mathrm{curl}} + \frac{\tau^2}{4} \|\nabla p\|^2_{\mathrm{curl}} + \tau \|\boldsymbol{E}\|_{\mathrm{curl}} \|\nabla p\|_{\mathrm{curl}} + \|p\|^2_{\mathrm{grad}}$$
$$\leq \|\boldsymbol{B}\|^2_{\mathrm{div}} + \frac{\tau^2}{4} \|\nabla \times \boldsymbol{E}\|^2_{\mathrm{div}} + \|\boldsymbol{B}\|^2_{\mathrm{div}} + \frac{\tau^2}{4} \|\nabla \times \boldsymbol{E}\|^2_{\mathrm{div}}$$
$$+ \|\boldsymbol{E}\|^2_{\mathrm{curl}} + \frac{\tau^2}{4} \|\nabla p\|^2_{\mathrm{curl}} + \|\boldsymbol{E}\|^2_{\mathrm{curl}} + \frac{\tau^2}{4} \|\nabla p\|^2_{\mathrm{curl}} + \|p\|^2_{\mathrm{grad}}$$
$$\leq 4 \|\|(\boldsymbol{B}, \boldsymbol{E}, p)\|\|.$$

Weak coercivity, (4.14), follows directly. To show boundedness, (4.15), apply Cauchy–Schwarz twice to (4.13). The well-posedness of bilinear form $\hat{a}$ defined in (4.13) follows directly from Babuška–Brezzi theory. Since $\hat{a}$, (4.13), and the original bilinear form a, (4.6), are equivalent, the scaled, mass-lumped FE system, which is equivalent to the MFD system, (4.3)–(4.5), is well-posed (arguments similar to those in Lemma 1 and Theorem 8 of [18] give the result). This implies that (4.6) satisfies (4.11)–(4.12), which completes the proof. □

5. Block preconditioners based on exact block factorization. One of the benefits of drawing connections between MFD and FEM, in addition to the ability to show well-posedness of the MFD discretization, is that robust linear solvers developed for FEM [9, 10, 23, 24, 25] can now be applied to the MFD system. In this work, we extend the ideas from [2, 24] to the MFD system and develop robust preconditioners based on block factorization, exploiting the structure-preserving nature of the discretization.

Good block preconditioners are often based on Schur complements and their approximations, and the accuracy of the approximations greatly influences the performance of the preconditioner. However, the structure-preserving discretization allows for Schur complements to be computed exactly, and the exact sequence of the discrete spaces yields sparse Schur complements that are used directly without approximation.

To more clearly exploit the structure-preserving nature of the MFD discretization, we rewrite the blocks of (2.12) back in the MFD operator notation given by (2.5)–(2.6),

$$(5.1) \qquad \mathcal{A}_{\mathrm{MFD}}\, \boldsymbol{x} = \boldsymbol{b} \iff \begin{bmatrix} \frac{2}{\tau}\mathcal{I}_{e^V} & \mathrm{curl}_D & \\ -\mathrm{curl}_V & \frac{2}{\tau}\mathcal{I}_{e^D} & \mathrm{grad}_D \\ & -\mathrm{div}_D & \frac{2}{\tau}\mathcal{I}_V \end{bmatrix} \begin{bmatrix} \boldsymbol{B}_V^n \\ \boldsymbol{E}_D^n \\ p_D^n \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}_{\boldsymbol{B}}^V \\ \boldsymbol{g}_{\boldsymbol{E}}^D \\ g_p^D \end{bmatrix}.$$

Recall that the structure-preserving discretization enforces the properties of the gradient, curl, and divergence (curl grad $= 0$ and div curl $= 0$) on the discrete level as $\mathrm{curl}_D\mathrm{grad}_D = 0$, $\mathrm{curl}_V\mathrm{grad}_V = 0$, $\mathrm{div}_V\mathrm{curl}_D = 0$, and $\mathrm{div}_D\mathrm{curl}_V = 0$. Exploiting these properties gives the exact block factorization of (2.12),
(5.2)

$$\mathcal{A}_{\mathrm{MFD}} = \underbrace{\begin{bmatrix} \mathcal{I}_{e^V} & & \\ -\frac{\tau}{2}\mathrm{curl}_V & \mathcal{I}_{e^D} & \\ & -\frac{\tau}{2}\mathrm{div}_D & \mathcal{I}_V \end{bmatrix}}_{\mathcal{L}} \underbrace{\begin{bmatrix} \frac{2}{\tau}\mathcal{I}_{e^V} & & \\ & \mathcal{S}_{\boldsymbol{E}} & \\ & & \mathcal{S}_p \end{bmatrix}}_{\mathcal{S}} \underbrace{\begin{bmatrix} \mathcal{I}_{e^V} & \frac{\tau}{2}\mathrm{curl}_D & \\ & \mathcal{I}_{e^D} & \frac{\tau}{2}\mathrm{grad}_D \\ & & \mathcal{I}_V \end{bmatrix}}_{\mathcal{U}},$$

with the Schur complements computed exactly as

$$\mathcal{S}_{\boldsymbol{E}} = \frac{\tau}{2}\mathrm{curl}_V\mathrm{curl}_D + \frac{2}{\tau}\mathcal{I}_{e^D}, \qquad\qquad \mathcal{S}_p = \frac{\tau}{2}\mathrm{div}_D\mathrm{grad}_D + \frac{2}{\tau}\mathcal{I}_V.$$

For the remainder of the paper, we drop the subscript notation and just represent $\mathcal{A}_{\mathrm{MFD}}$ by $\mathcal{A}$.

Several block preconditioners can be designed from the exact factorization, (5.2). A natural choice of preconditioner is $\mathcal{S}^{-1}$. However, this involves computing the inverse of the Schur complements, making this choice impractical. To rectify this, replace the Schur complements, $\mathcal{S}_{\boldsymbol{E}}$ and $\mathcal{S}_p$, with good preconditioners, $\mathcal{Q}_{\boldsymbol{E}}$ and $\mathcal{Q}_p$. For example, an HX-preconditioner [17] can be used for $\mathcal{Q}_{\boldsymbol{E}}$ and a standard multigrid preconditioner for $\mathcal{Q}_p$. Note that the top left entry in $\mathcal{S}$ is a scaled identity matrix, so no spectrally equivalent approximation is needed. For the remaining two blocks, we assume that

$$(5.3) \qquad\qquad\qquad c_{1,\boldsymbol{E}} \le \lambda(\mathcal{Q}_{\boldsymbol{E}}\mathcal{S}_{\boldsymbol{E}}) \le c_{2,\boldsymbol{E}},$$
$$(5.4) \qquad\qquad\qquad c_{1,p} \le \lambda(\mathcal{Q}_p\mathcal{S}_p) \le c_{2,p}.$$

This implies that for $\mathcal{Q} = \mathrm{diag}\big(\big(\frac{2}{\tau}\mathcal{I}_{e^V}\big)^{-1}, \mathcal{Q}_{\boldsymbol{E}}, \mathcal{Q}_p\big)$, we have

$$c_1 \le \lambda(\mathcal{Q}\mathcal{S}) \le c_2,$$

where $c_1 = \min(c_{1,\boldsymbol{E}}, c_{1,p})$ and $c_2 = \max(c_{2,\boldsymbol{E}}, c_{2,p})$.

From this factorization, we consider three different block preconditioners,

$$(5.5) \qquad \mathcal{X}_{\mathcal{LS}} := \mathcal{Q}\mathcal{L}^{-1}, \quad \mathcal{X}_{\mathcal{SU}} := \mathcal{U}^{-1}\mathcal{Q}, \quad \mathcal{X}_{\mathcal{LSU}} := \mathcal{U}^{-1}\mathcal{Q}\mathcal{L}^{-1},$$

where $\mathcal{L}^{-1}$ and $\mathcal{U}^{-1}$ can be computed directly:

$$\mathcal{L}^{-1} = \begin{bmatrix} \mathcal{I}_{\boldsymbol{e}^V} & & \\ \frac{\tau}{2}\mathrm{curl}_V & \mathcal{I}_{\boldsymbol{e}^D} & \\ & \frac{\tau}{2}\mathrm{div}_D & \mathcal{I}_V \end{bmatrix}, \qquad \mathcal{U}^{-1} = \begin{bmatrix} \mathcal{I}_{\boldsymbol{e}^V} & -\frac{\tau}{2}\mathrm{curl}_D & \\ & \mathcal{I}_{\boldsymbol{e}^D} & -\frac{\tau}{2}\mathrm{grad}_D \\ & & \mathcal{I}_V \end{bmatrix}.$$

In the following theorem, we prove that these preconditioners are robust with respect to the discretization parameters. By bounding the eigenvalues for the preconditioned system, we guarantee good performance of GMRES. Note that the following proof can be done to show that the constants c_1 and c_2 are also independent of the PDE parameters, ϵ and μ, but for simplicity we only consider the case where $\epsilon = \mu = 1$. Otherwise, the identity matrices in the diagonal matrix of the decomposition would be scaled by the PDE parameter values.

THEOREM 5.1. *Let* $\mathcal{X}_{\mathcal{LS}}$, $\mathcal{X}_{\mathcal{SU}}$, *and* $\mathcal{X}_{\mathcal{LSU}}$ *be defined by* (5.5), *and assume the spectral equivalent properties* (5.3)–(5.4) *hold. Then*

$$(5.6) \qquad \lambda\left(\mathcal{X}_{\mathcal{LS}}\mathcal{A}\right) \in [c_1, c_2], \quad \lambda\left(\mathcal{X}_{\mathcal{SU}}\mathcal{A}\right) \in [c_1, c_2], \quad \lambda\left(\mathcal{X}_{\mathcal{LSU}}\mathcal{A}\right) \in [c_1, c_2],$$

where $c_1 = \min\left(c_{1,\boldsymbol{E}}, c_{1,p}\right)$ *and* $c_2 = \max\left(c_{2,\boldsymbol{E}}, c_{2,p}\right)$ *are constants that do not depend on discretization parameters,* h *and* τ.

Proof. First, consider $\mathcal{X}_{\mathcal{LS}}\mathcal{A}$,

$$\mathcal{X}_{\mathcal{LS}}\mathcal{A} = \mathcal{Q}\mathcal{L}^{-1}\mathcal{L}\mathcal{S}\mathcal{U} = \mathcal{Q}\mathcal{S}\mathcal{U} = \begin{bmatrix} \mathcal{I}_{\boldsymbol{e}^V} & \frac{\tau}{2}\mathrm{curl}_D & \\ & \mathcal{Q}_{\boldsymbol{E}}\mathcal{S}_{\boldsymbol{E}} & \mathcal{Q}_{\boldsymbol{E}}\mathrm{grad}_D \\ & & \mathcal{Q}_p\mathcal{S}_p \end{bmatrix}.$$

The eigenvalues $\lambda\left(\mathcal{X}_{\mathcal{LS}}\mathcal{A}\right)$ are determined by the eigenvalues of the diagonal blocks since $\mathcal{X}_{\mathcal{LS}}\mathcal{A}$ is block upper triangular. The first block is an identity matrix, whose eigenvalues are all ones. For the other two blocks, we use (5.3)–(5.4) to bound the eigenvalues and, overall, have $\lambda\left(\mathcal{X}_{\mathcal{LS}}\mathcal{A}\right) \in [c_1, c_2]$.

To bound the eigenvalues of $\mathcal{X}_{\mathcal{SU}}\mathcal{A}$, consider the eigenvalue problem,

$$\mathcal{X}_{\mathcal{SU}}\mathcal{A}\boldsymbol{x} = \lambda\boldsymbol{x} \iff \mathcal{A}\boldsymbol{x} = \lambda\mathcal{X}_{\mathcal{SU}}^{-1}\boldsymbol{x}.$$

By substituting in the decompositions of $\mathcal{A}$ and $\mathcal{X}_{\mathcal{SU}}$, it follows that

$$\mathcal{L}\mathcal{S}\mathcal{U}\boldsymbol{x} = \lambda\mathcal{Q}^{-1}\mathcal{U}\boldsymbol{x}.$$

Let $\boldsymbol{y} = \mathcal{U}\boldsymbol{x}$, and left multiply by $\mathcal{Q}$. Then $\mathcal{Q}\mathcal{L}\mathcal{S}\boldsymbol{y} = \lambda\boldsymbol{y}$, where

$$\mathcal{Q}\mathcal{L}\mathcal{S} = \begin{bmatrix} \mathcal{I}_{\boldsymbol{e}^V} & & \\ -\mathcal{Q}_{\boldsymbol{E}}\mathrm{curl}_V & \mathcal{Q}_{\boldsymbol{E}}\mathcal{S}_{\boldsymbol{E}} & \\ & -\mathcal{Q}_p\mathrm{div}_D & \mathcal{Q}_p\mathcal{S}_p \end{bmatrix}.$$

Since $\mathcal{Q}\mathcal{L}\mathcal{S}$ is block lower triangular, again the eigenvalues only depend on the diagonal blocks. Thus, by (5.3)–(5.4), $\lambda\left(\mathcal{X}_{\mathcal{SU}}\mathcal{A}\right) \in [c_1, c_2]$.

Finally, consider the preconditioned system

$$\mathcal{X}_{\mathcal{LSU}}\mathcal{A}\boldsymbol{x} = \mathcal{U}^{-1}Q\mathcal{L}^{-1}\mathcal{LSU}\boldsymbol{x} = \mathcal{U}^{-1}Q\mathcal{SU}\boldsymbol{x} = \lambda\boldsymbol{x}.$$

Left multiplying by $\mathcal{U}$ and letting $\boldsymbol{y} = \mathcal{U}\boldsymbol{x}$ yields

$$Q\mathcal{S}\boldsymbol{y} = \lambda\boldsymbol{y},$$

where

$$Q\mathcal{S} = \begin{bmatrix} \mathcal{I}_{\boldsymbol{e}^V} & & \\ & \mathcal{Q}_{\boldsymbol{E}}\mathcal{S}_{\boldsymbol{E}} & \\ & & \mathcal{Q}_p\mathcal{S}_p \end{bmatrix}.$$

By the same reasoning as for the other two cases, we conclude that $\lambda\left(\mathcal{X}_{\mathcal{LSU}}\mathcal{A}\right) \in [c_1, c_2]$. □

5.1. Preservation of the divergence-free magnetic field. The goal of using a structure-preserving discretization for the Maxwell system is to enforce the PDE constraints on the discrete level. However, even if the discretization holds, the approximate solve for each time step of a linear solver can destroy these properties. The following theorem ensures that at each iteration of the linear solver, the divergence-free condition for $\boldsymbol{B}$ is preserved.

THEOREM 5.2. *Let $\boldsymbol{x}^0 = \left(\boldsymbol{B}_V^0, \boldsymbol{E}_D^0, p_D^0\right)^T$ be the initial guess for the MFD system satisfying $\mathrm{div}_V\,\boldsymbol{B}_V^0 = 0$, and let $\boldsymbol{b} = \left(\boldsymbol{g}_{\boldsymbol{B}}^V, \boldsymbol{g}_{\boldsymbol{E}}^D, g_p^D\right)^T$ be the MFD right-hand side satisfying $\mathrm{div}_V\,\boldsymbol{g}_{\boldsymbol{B}}^V = 0$. Then all iterations, $\boldsymbol{x}^k = \left(\boldsymbol{B}_V^k, \boldsymbol{E}_D^k, p_D^k\right)^T$, of the preconditioned GMRES method satisfy $\mathrm{div}_V\,\boldsymbol{B}_V^k = 0$, where $\mathcal{X}$ is any of the preconditioners defined in* (5.5).

Proof. Define the Krylov subspace,

$$K^k(\mathcal{X}\mathcal{A}, \boldsymbol{r}^0) = \mathrm{span}\{\boldsymbol{r}^0, \mathcal{X}\mathcal{A}\boldsymbol{r}^0, (\mathcal{X}\mathcal{A})^2\,\boldsymbol{r}^0, \ldots, (\mathcal{X}\mathcal{A})^k\,\boldsymbol{r}^0\},$$

with $\boldsymbol{r}^0 = \left(\boldsymbol{r}_{\boldsymbol{B}}^0, \boldsymbol{r}_{\boldsymbol{E}}^0, r_p^0\right)^T := \mathcal{X}\left(\boldsymbol{b} - \mathcal{A}\boldsymbol{x}^0\right)$ such that for each iteration,

$$(5.7) \qquad\qquad \boldsymbol{x}^k \in \boldsymbol{x}^0 + K^k\left(\mathcal{X}\mathcal{A}, \boldsymbol{r}^0\right).$$

By the assumption of the divergence-free initial data, it follows that $\mathrm{div}_V\boldsymbol{r}_{\boldsymbol{B}}^0 = 0$. Next, let $\boldsymbol{v}^m = \left(\boldsymbol{v}_{\boldsymbol{B}}^m, \boldsymbol{v}_{\boldsymbol{E}}^m, \boldsymbol{v}_p^m\right)^T := (\mathcal{X}\mathcal{A})^m\,\boldsymbol{r}^0$ for $m = 0, 1, \ldots, k-1$. Since $\boldsymbol{v}^m = \mathcal{X}\mathcal{A}\boldsymbol{v}^{m-1}$, we compute $\boldsymbol{v}_{\boldsymbol{B}}^m$ for each of the preconditioners in (5.5). Setting $\mathcal{X} = \mathcal{X}_{\mathcal{LS}}$,

$$\boldsymbol{v}_{\boldsymbol{B}}^m = \left(\frac{2}{\tau}\mathcal{I}_{\boldsymbol{e}^V}\right)^{-1}\left(\frac{2}{\tau}\boldsymbol{v}_{\boldsymbol{B}}^{m-1} + \mathrm{curl}_D\boldsymbol{v}_{\boldsymbol{E}}^{m-1}\right) = \boldsymbol{v}_{\boldsymbol{B}}^{m-1} + \frac{\tau}{2}\mathrm{curl}_D\boldsymbol{v}_{\boldsymbol{E}}^{m-1}.$$

For $\mathcal{X} = \mathcal{X}_{\mathcal{SU}}$,

$$\boldsymbol{v}_{\boldsymbol{B}}^m = \boldsymbol{v}_{\boldsymbol{B}}^{m-1} + \frac{\tau}{2}\mathrm{curl}_D\left(\boldsymbol{v}_{\boldsymbol{E}}^{m-1} + \mathcal{Q}_{\boldsymbol{E}}\mathrm{curl}_V\boldsymbol{v}_{\boldsymbol{B}}^{m-1} - \frac{2}{\tau}\mathcal{Q}_{\boldsymbol{E}}\boldsymbol{v}_{\boldsymbol{E}}^{m-1} - \mathcal{Q}_{\boldsymbol{E}}\mathrm{grad}_D\boldsymbol{v}_p^{m-1}\right),$$

and for $\mathcal{X} = \mathcal{X}_{\mathcal{LSU}}$,

$$\boldsymbol{v}_{\boldsymbol{B}}^m = \boldsymbol{v}_{\boldsymbol{B}}^{m-1}$$
$$+ \mathrm{curl}_D\left(\frac{\tau}{2}\boldsymbol{v}_{\boldsymbol{E}}^{m-1} - \frac{\tau^2}{4}\mathcal{Q}_{\boldsymbol{E}}\mathrm{curl}_V\,\mathrm{curl}_D\boldsymbol{v}_{\boldsymbol{E}}^{m-1} - \mathcal{Q}_{\boldsymbol{E}}\boldsymbol{v}_{\boldsymbol{E}}^{m-1} - \frac{\tau}{2}\mathcal{Q}_{\boldsymbol{E}}\mathrm{grad}_D\boldsymbol{v}_p^{m-1}\right).$$

Applying div_V to $\boldsymbol{v}_{\boldsymbol{B}}^m$ for each preconditioner above, all terms with curl_D in front are zero since $\operatorname{div}_V \operatorname{curl}_D = 0$. Then $\operatorname{div}_V \boldsymbol{v}_{\boldsymbol{B}}^m = 0$ if $\operatorname{div}_V \boldsymbol{v}_{\boldsymbol{B}}^{m-1} = 0$. By an inductive argument, since $\operatorname{div}_V \boldsymbol{r}_{\boldsymbol{B}}^0 = 0$, we have that $\operatorname{div}_V \boldsymbol{v}_{\boldsymbol{B}}^m = 0$. By (5.7) and the definition of $\boldsymbol{v}^m$, $\boldsymbol{x}^k$ is a linear combination of $\boldsymbol{v}^m$, $m = 0, 1, \ldots, k-1$. This implies that $\boldsymbol{B}_V^k$ is a linear combination of $\boldsymbol{v}_{\boldsymbol{B}}^m$. Since $\operatorname{div}_V \boldsymbol{v}_{\boldsymbol{B}}^m = 0$, then $\operatorname{div}_V \boldsymbol{B}_V^k = 0$ for all k. $\square$

6. Numerical results. To demonstrate the theoretical results presented in the previous sections, consider the following test problem with essential Dirichlet boundary conditions:

$$(6.1) \qquad \boldsymbol{E}(\boldsymbol{x}, t) = \frac{1}{\pi} e^{-t} \begin{bmatrix} -\cos(\pi x_1)\sin(\pi x_2)\sin(\pi x_3) \\ \sin(\pi x_1)\cos(\pi x_2)\sin(\pi x_3) \\ 0 \end{bmatrix},$$

$$(6.2) \qquad \boldsymbol{B}(\boldsymbol{x}, t) = e^{-t} \begin{bmatrix} -\sin(\pi x_1)\cos(\pi x_2)\cos(\pi x_3) \\ -\cos(\pi x_1)\sin(\pi x_2)\cos(\pi x_3) \\ 2\cos(\pi x_1)\cos(\pi x_2)\sin(\pi x_3) \end{bmatrix},$$

$$(6.3) \qquad p(\boldsymbol{x}, t) = 0,$$

$$(6.4) \qquad \boldsymbol{j}(\boldsymbol{x}, t) = -e^{-t}\left(\frac{1}{\pi} + 3\pi\right) \begin{bmatrix} \cos(\pi x_1)\sin(\pi x_2)\sin(\pi x_3) \\ -\sin(\pi x_1)\cos(\pi x_2)\sin(\pi x_3) \\ 0 \end{bmatrix}.$$

While the analysis presented in this paper holds for a general Voronoi mesh, for simplicity, we consider a nondegenerate mesh, where the Voronoi points do not lie on the boundary or outside of the corresponding Delaunay tetrahedra. To get a nondegenerate Voronoi mesh, we design a Delaunay triangulation for the FE domain that consists of a cube with a rectangular pyramid on each face (see Figure 3). The pyramids are defined by the faces of the unit cube and the points $\left(-\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$, $\left(\frac{3}{2}, \frac{1}{2}, \frac{1}{2}\right)$, $\left(\frac{1}{2}, -\frac{1}{2}, \frac{1}{2}\right)$, $\left(\frac{1}{2}, \frac{3}{2}, \frac{1}{2}\right)$, $\left(\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}\right)$, $\left(\frac{1}{2}, \frac{1}{2}, \frac{3}{2}\right)$. Uniform refinement is used to get more resolved meshes, and Table 1 lists the geometric information for the different resolutions considered. Numerical experiments are implemented in the HAZmath package [1] written by the authors. All timed numerical results are done using a workstation with an 8-core 3-GHz Intel Xeon Sandy Bridge CPU and 256 GB of RAM.
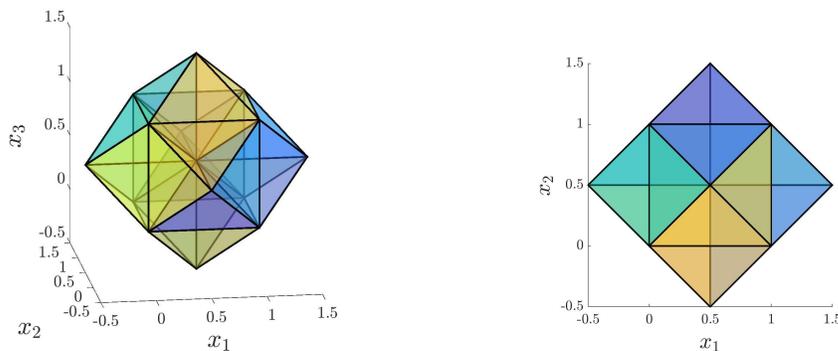

FIG. 3. *Left: Delaunay mesh with h = 1; Right: cross section at z = 1/2.*

TABLE 1
Geometric information for the Delaunay meshes.

	h	Vertices	Edges	Faces	DoF
Mesh 1	1/4	369	2,096	3,264	5,729
Mesh 2	1/8	2,465	15,520	25,344	43,329
Mesh 3	1/16	17,985	119,360	199,680	337,025
Mesh 4	1/32	137,345	936,064	1,585,152	2,658,561
Mesh 5	1/64	1,073,409	7,414,016	12,632,064	21,119,489

To demonstrate the equivalence of the two discretizations, MFD and FEM are implemented and solved with $\mathcal{X}_{\mathcal{LSU}}$-preconditioned FGMRES with restart after 100 iterations to a relative residual tolerance of 10^{-8}. We expect FE convergence rates with respect to mesh size for both. More precisely, with L^2 norm in space and L^∞ norm in time, $\boldsymbol{E}$ and $\boldsymbol{B}$ modeled with lowest order Nédélec and RT elements, respectively, are expected to converge with $\mathcal{O}(h+\tau^2)$, where h is the mesh partitioning and τ is the time-step size. Results for p are excluded, as it is just an auxiliary variable with no physical relevance to the problem. By choosing a small time step and few time iterations ($\tau = .0125$ for 8 time steps, i.e., final time $t = 0.1$), the spatial error dominates. This is confirmed in Figure 4, where the spatial convergence of the MFD scheme is identical to the FEM one.
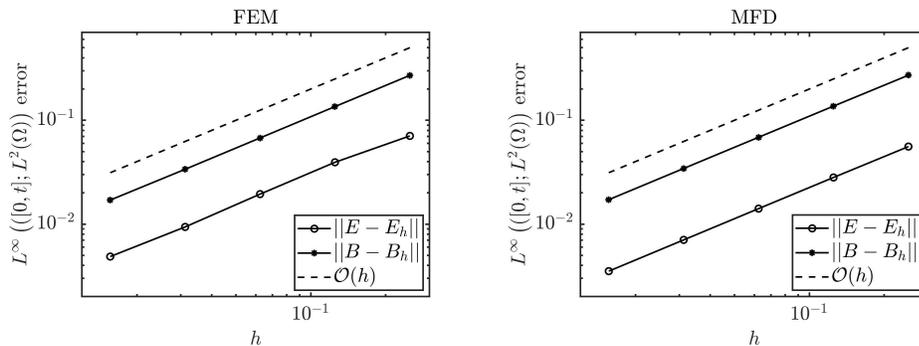

FIG. 4. $L^\infty\left([0,t]; L^2(\Omega)\right)$ error for FEM (left) and MFD (right) for $\tau = 0.0125$ after 8 time steps ($t = 0.1$) solving with the $\mathcal{X}_{\mathcal{LSU}}$ preconditioner and FGMRES.

Next, performance comparisons are made between the MFD and FE methods using the preconditioners developed in section 5 for MFD and in [2] for FEM. Again using test problem (6.1)–(6.4) and meshes given by Table 1, we test the preconditioners defined in (5.5) for robustness with respect to time step, τ, and mesh size, h. The diagonal blocks of the preconditioners are solved inexactly by preconditioned GMRES with a relative residual reduction set at 10^{-2} for the outer iteration. Flexible GMRES is used as the outer iteration with a relative residual stopping criterion of 10^{-8}, with restart after 100 iterations. Iteration counts for both MFD and FEM with preconditioners $\mathcal{X}_{\mathcal{LS}}$, $\mathcal{X}_{\mathcal{SU}}$, and $\mathcal{X}_{\mathcal{LSU}}$ are reported in Tables 2 and 3. In all experiments, iteration counts are averaged over the number of time steps needed to reach $t = 1$. The spectral equivalent approximations used in $\mathcal{Q}$ are an HX preconditioner [17, 20] for $\mathcal{S}_{\boldsymbol{E}}$ and a standard algebraic multigrid method for $\mathcal{S}_p$.

The results show little variation in iteration count with varying parameters, indicating that the block preconditioners based on exact block factorization are effective

TABLE 2

Iteration counts for the block preconditioners based on block factorization for MFD. Left: block lower triangular, $\mathcal{X}_{\mathcal{LS}}$; Center: block upper triangular, $\mathcal{X}_{\mathcal{SU}}$; Right: symmetric, $\mathcal{X}_{\mathcal{LSU}}$.

τ \ h	$\mathcal{X}_{\mathcal{LS}}$					$\mathcal{X}_{\mathcal{SU}}$					$\mathcal{X}_{\mathcal{LSU}}$				
	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$
0.2	5	5	5	6	8	5	5	5	6	6	4	3	4	4	4
0.1	3	4	5	5	6	4	5	5	5	6	2	3	3	4	4
0.05	4	3	4	4	5	4	4	5	5	5	3	2	3	3	3
0.025	3	4	3	4	4	3	4	4	5	5	2	3	2	3	3
0.0125	2	3	4	3	3	3	3	4	4	4	2	2	3	2	3

TABLE 3

Iteration counts for the block preconditioners based on block factorization for FEM. Left: block lower triangular, $\mathcal{X}_{\mathcal{LS}}$; Center: block upper triangular, $\mathcal{X}_{\mathcal{SU}}$; Right: symmetric, $\mathcal{X}_{\mathcal{LSU}}$.

τ \ h	$\mathcal{X}_{\mathcal{LS}}$					$\mathcal{X}_{\mathcal{SU}}$					$\mathcal{X}_{\mathcal{LSU}}$				
	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$
0.2	4	4	4	4	4	5	5	5	5	4	3	3	3	3	3
0.1	5	4	4	4	4	5	5	5	5	4	3	3	3	3	3
0.05	4	4	4	4	3	5	4	4	4	4	3	2	3	3	3
0.025	3	3	3	3	3	4	4	3	3	2	3	2	2	2	2
0.0125	3	3	3	3	3	3	3	4	3	3	3	2	2	2	3

and robust with respect to mesh size, h, and time step size, τ, for both MFD and FEM as expected. Furthermore, MFD has iteration counts comparable to FE.

To further compare the two methods, CPU solve time per time iteration is examined for fixed time-step size, as well as the time scaling of the solve time per time step for the three preconditioned FGMRES solvers, both with respect to mesh refinement. In Table 4, the average solve time over 10 iterations with fixed τ is reported. We see that as the number of degrees of freedom increases, the FE method beats MFD, especially for $\mathcal{X}_{\mathcal{LS}}$ and $\mathcal{X}_{\mathcal{SU}}$. This is further demonstrated in Figure 5, where a side-by-side comparison of the three preconditioners for MFD and FEM is given. This time disparity is likely given by the fact that MFD takes slightly more GMRES iterations per solve, which could be due to the fact that the MFD system is not symmetric while FEM is.

TABLE 4

Average CPU solve time per time iteration over 10 time steps with $\tau = 0.1$ for MFD and FEM with all preconditioners.

	h	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$
MFD	$\mathcal{X}_{\mathcal{LS}}$	1.32×10^{-2}	1.06×10^{-1}	9.95×10^{-1}	1.31×10^{1}	1.68×10^{2}
	$\mathcal{X}_{\mathcal{SU}}$	1.48×10^{-2}	1.12×10^{-1}	1.10×10^{0}	1.41×10^{1}	1.96×10^{2}
	$\mathcal{X}_{\mathcal{LSU}}$	1.06×10^{-2}	8.32×10^{-2}	8.15×10^{-1}	1.12×10^{1}	1.41×10^{2}
FEM	$\mathcal{X}_{\mathcal{LS}}$	2.05×10^{-2}	1.08×10^{-1}	8.84×10^{-1}	9.62×10^{0}	1.09×10^{2}
	$\mathcal{X}_{\mathcal{SU}}$	2.45×10^{-2}	1.24×10^{-1}	1.02×10^{0}	1.19×10^{1}	1.17×10^{2}
	$\mathcal{X}_{\mathcal{LSU}}$	1.79×10^{-2}	9.17×10^{-2}	7.57×10^{-1}	8.04×10^{0}	1.06×10^{2}

Finally, it is expected that the solve time scales $\mathcal{O}\left(N \log(N)\right)$, where N is the

number of degrees of freedom. Figure 6 verifies this, where both FE and MFD results follow the trend of the reference line for all three preconditioners.

Based on the results presented, we conclude that all three preconditioners are robust and effective for solving the MFD system and give results comparable to the FE method. In terms of both solve time and iteration counts, $\mathcal{X}_{\mathcal{LSU}}$ is the best preconditioner for both methods, which was also concluded in [2] for the FE method.
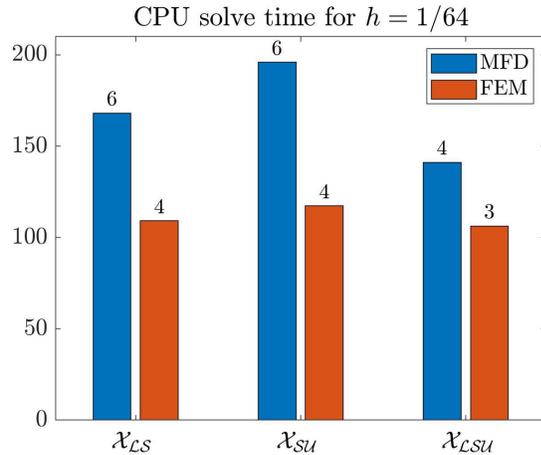

FIG. 5. *Comparison of solve time averaged over* 10 *time steps of* $\tau = 0.1$ *for MFD and FEM for fixed mesh size* $h = \frac{1}{64}$. *The number of FGMRES iterations is given above each bar.*
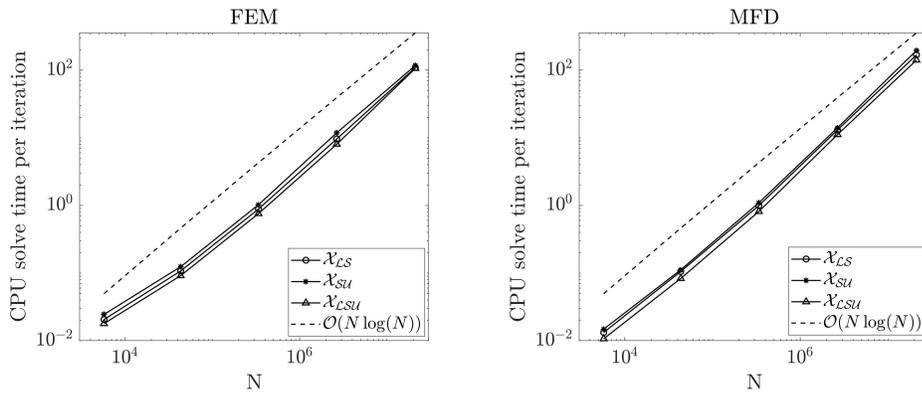

FIG. 6. *Time scaling of solve time averaged over* 10 *time steps of* $\tau = 0.1$ *for FEM (left) and MFD (right), where* N *is the number of degrees of freedom.*

7. Conclusions. By examining MFD in an FE framework, we are able to exploit the FEM tools and theory to strengthen the MFD method for Maxwell. From this equivalence of the two methods, well-posedness of the MFD system is proven with Babuška–Brezzi theory. Numerical results demonstrate that the FE convergence theory is recoverable in the MFD implementation. Furthermore, in showing well-posedness of the Maxwell MFD discretization and the connection to the FEM, robust block preconditioners developed for the FEM in [2] are adopted for solving the MFD

linear system efficiently. When using structure-preserving discretizations, the goal is to enforce the PDE constraints, particularly a divergence-free magnetic field for Maxwell's equations, at all time steps and all solve iterations. The block preconditioners for GMRES developed here are shown to be robust and guarantee all properties of the discretization at each time step.

All of the results in this paper apply to lowest-order FE and MFD methods. While higher-order MFD methods exist [14, 21, 22], connections to higher-order FEM are unclear and require further investigation. If such relationships are found, the analysis presented here should be valid. Additionally, a priori error estimates for MFD for Maxwell could be derived in a fashion similar to the techniques used for Darcy flow in [13]. Furthermore, now that an FE framework for MFD has been developed for the full Maxwell system, other electromagnetic applications can be explored, such as MHD, time-harmonic Maxwell, or $\boldsymbol{H}(\text{curl})$ and $\boldsymbol{H}(\text{div})$ problems, in general. The structure-preserving nature of the MFD discretization opens the door for many more physical applications with PDE constraints to be explored.

REFERENCES

[1] J. H. ADLER, X. HU, AND L. T. ZIKATANOV, *HAZmath: A Simple Finite Element, Graph, and Solver Library*, https://bitbucket.org/hazmath/hazmath/wiki/Home, 2014–2020.

[2] J. H. ADLER, X. HU, AND L. T. ZIKATANOV, *Robust solvers for Maxwell's equations with dissipative boundary conditions*, SIAM J. Sci. Comput., 39 (2017), pp. S3–S23, https://doi.org/10.1137/16M1073339.

[3] J. H. ADLER, V. PETKOV, AND L. T. ZIKATANOV, *Numerical approximations of asymptotically disappearing solutions of Maxwell's equations*, SIAM J. Sci. Comput., 35 (2013), pp. S386–S401, https://doi.org/10.1137/120879385.

[4] D. N. ARNOLD, R. S. FALK, AND R. WINTHER, *Finite element exterior calculus, homological techniques, and applications*, Acta Numer., 6 (2006), pp. 1–155.

[5] I. BABUŠKA, *The finite element method with Lagrangian multipliers*, Numer. Math., 20 (1972/73), pp. 179–192, https://doi.org/10.1007/BF01436561.

[6] J. BARANGER, J. F. MAITRE, AND F. OUDIN, *Application de la théorie des éléments finis mixtes à l'étude d'une classe de schémas aux volumes différences finis pour les problèmes elliptiques*, C. R. Acad. Sci. Paris Sér. I Math., 319 (1994), pp. 401–404.

[7] J. BARANGER, J. F. MAITRE, AND F. OUDIN, *Connection between finite volume and mixed finite element methods*, RAIRO Modél. Math. Anal. Numér., 30 (1996), pp. 445–465.

[8] L. BEIRÃO DA VEIGA, K. LIPNIKOV, AND G. MANZINI, *The Mimetic Finite Difference Method for Elliptic Problems*, Springer, Cham, 2014.

[9] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.

[10] M. BENZI AND G. H. GOLUB, *A preconditioner for generalized saddle point problems*, SIAM J. Maxtrix Anal. Appl., 26 (2004), pp. 20–41, https://doi.org/10.1137/S0895479802417106.

[11] V. A. BOKIL, N. GIBSON, V. GYRYA, AND D. MCGREGOR, *Dispersion reducing methods for edge discretizations of the electric vector wave equation*, J. Comput. Phys., 287 (2015), pp. 88–109.

[12] F. BREZZI, *On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers*, Rev. Française Automat. Informat. Recherche Opérationnelle Sér. Rouge, 8 (1974), pp. 129–151.

[13] F. BREZZI, M. FORTIN, AND L. D. MARINI, *Error analysis of piecewise constant pressure approximations of Darcy's law*, Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 1547–1559.

[14] J. CASTILLO, J. HYMAN, M. SHASHKOV, AND S. STEINBERG, *High-order mimetic finite difference methods on nonuniform grids*, in Proceedings of the Third International Conference on Spectral and High Order Methods (COSAHOM'95), 1995, pp. 347–361.

[15] C. GREIF AND D. SCHOTZAU, *Preconditioners for the discretized time-harmonic Maxwell equations in mixed form*, Numer. Linear Algebra Appl., 14 (2007), pp. 281–297.

[16] R. HIPTMAIR, *Finite elements in computational electromagnetism*, Acta Numer., 11 (2002), pp. 237–339.

[17] R. HIPTMAIR AND J. XU, *Nodal auxiliary space preconditioning in H(curl) and H(div) spaces*, SIAM J. Numer. Anal., 45 (2007), pp. 2483–2509, https://doi.org/10.1137/060660588.

[18] K. Hu, Y. Ma, and J. Xu, *Stable finite element methods preserving* $\nabla \cdot B = 0$ *exactly for MHD models*, Numer. Math., 135 (2017), pp. 371–396, https://doi.org/10.1007/s00211-016-0803-4.

[19] J. M. Hyman and M. Shashkov, *Mimetic discretizations for Maxwell's equations*, J. Comput. Phys., 151 (1999), pp. 881–909.

[20] T. V. Kolev and P. S. Vassilevski, *Parallel auxiliary space AMG for H(curl) problems*, J. Comput. Math., 27 (2009), pp. 604–623, https://doi.org/10.4208/jcm.2009.27.5.013.

[21] K. Lipnikov and V. Gyrya, *High-order mimetic finite difference method for diffusion problems on polygonal meshes*, J. Comput. Phys., 227 (2008), pp. 8841–8854.

[22] K. Lipnikov, G. Manzini, and M. Shashkov, *Mimetic finite difference method*, J. Comput. Phys., 257 (2014), pp. 1163–1227.

[23] D. Loghin and A. J. Wathen, *Analysis of preconditioners for saddle point problems*, SIAM J. Sci. Comput., 25 (2004), pp. 2029–2049, https://doi.org/10.1137/S1064827502418203.

[24] Y. Ma, K. Hu, X. Hu, and J. Hu, *Robust preconditioners for incompressible MHD models*, J. Comput. Phys., 316 (2016), pp. 721–746.

[25] K. A. Mardal and R. Winther, *Preconditioning discretizations of systems of partial differential equations*, Numer. Linear Algebra Appl., 18 (2011), pp. 1–40.

[26] P. Monk, *Finite Element Methods for Maxwell's Equations*, Numer. Math. Sci. Comput., Oxford University Press, New York, 2003.

[27] E. G. Phillips, J. N. Shadid, and E. C. Cyr, *Scalable preconditioners for structure preserving discretizations of Maxwell equations in first order form*, SIAM J. Sci. Comput., 40 (2018), pp. B723–B742, https://doi.org/10.1137/17M1135827.

[28] E. G. Phillips, J. N. Shadid, E. C. Cyr, H. C. Elman, and R. P. Pawlowski, *Block preconditioners for stable mixed nodal and edge finite element representations of incompressible resistive MHD*, SIAM J. Sci. Comput., 38 (2016), pp. B1009–B1031, https://doi.org/10.1137/16M1074084.

[29] C. Rodrigo, F. J. Gaspar, X. Hu, and L. T. Zikatanov, *Finite element framework for some mimetic finite difference discretizations*, Comput. Math. Appl., 70 (2015), pp. 2661–2673.

[30] V. Thomée, *Galerkin Finite Element Methods for Parabolic Problems*, Springer, Berlin, 1997.

[31] P. N. Vabishchevich, *Finite-difference approximation of mathematical physics problems on irregular grids*, Comput. Methods Appl. Math., 5 (2005), pp. 294–330.

[32] S.-L. Wu, T.-Z. Huang, and C.-X. Li, *Modified block preconditioners for the discretized time-harmonic Maxwell equations in mixed form*, J. Comput. Appl. Math., 237 (2013), pp. 419–431.