MONOLITHIC MULTIGRID FOR A B-FIELD INCOMPRESSIBLE, RESISTIVE MAGNETOHYDRODYNAMICS MODEL

THOMAS R. BENSON, JAMES H. ADLER, ERIC C. CYR, SCOTT P. MACLACHLAN, AND RAYMOND S. TUMINARO

Abstract. Magnetohydrodynamics (MHD) models describe a wide range of plasma physics applications, from thermonuclear fusion in tokamak reactors to astrophysical models. These models are characterized by a nonlinear system of partial differential equations in which the flow of the fluid strongly couples to the evolution of electromagnetic fields. As a result, the discrete linearized systems that arise in the numerical solution of these equations are generally difficult to solve, and require effective preconditioners to be developed. This paper investigates monolithic multigrid preconditioners for a one-fluid, viscoresistive MHD model in two dimensions that utilizes a second Lagrange multiplier added to Faraday's law to enforce the divergence-free constraint on the magnetic field. We consider the extension of a well-known relaxation scheme from the fluid dynamics literature, Vanka relaxation, to this formulation. To isolate the relaxation scheme from the rest of the multigrid method, we utilize structured grids, geometric interpolation operators, Galerkin coarse grid operators, and inf-sup stable elements for both constraints in the system. Numerical results are shown for the Hartmann flow problem, a standard test problem in MHD.

1. Introduction

Magnetohydrodynamics (MHD) are models of plasma physics that are used to describe the flow of electrically conductive fluids in the presence of electromagnetic fields. In this paper, we consider the single-fluid model in which the behavior of the ions and that of the electrons is averaged together. The resulting equations are a combination of the Navier-Stokes equations of fluid dynamics and Maxwell's Equations of electromagnetism, which form a strongly coupled system of nonlinear partial differential equations (PDEs). There are a wide variety of physical assumptions at play in the model, regarding coupling to the electric field, the current density, and Ohm's law, among other things, that lead to different formulations of the MHD equations. This paper focuses on time-independent solutions of a viscoresistive, incompressible model of MHD, and, in particular, on preconditioning the linear systems that arise from the linearization and discretization of the resulting equations. Previous work [2] utilized a vector-potential formulation to enforce the solenoidal constraint, and to generally simplify the problem. Here, we use a formulation that maintains the primitive variables, including the magnetic field, $\mathbf{B}$, and enforces the solenoidal constraint weakly by using a Lagrange multiplier that is added to Faraday's law [15, 29, 31]. Note that the Lagrange multiplier approach is only one such technique for including the solenoidal condition; others include exact-penalty methods [25] and vector-potential formulations [2, 12, 32].

As the equations of MHD are nonlinear, we employ Newton's method to linearize the system. In order to solve the linearized systems numerically, we use a mixed finite-element method that results in linear problems of saddle-point type. Saddle-point problems arise in various contexts, ranging from economics to fluid dynamics; an extensive general overview can be found in [8]. Numerous solution and preconditioning strategies have been proposed, and often the solver or preconditioner is very closely tied to the problem being solved. Block-factorization approaches manipulate the constituent blocks of the Jacobian operator in order to resolve the coupling in the system using workhorse algorithms such as algebraic multigrid (AMG) [27] on simpler problems [13, 33, 35]. These techniques have been extensively studied and, by utilizing this paradigm, the known parallel and mesh resolution scalability properties can be exploited to create scalable preconditioners for more complex problems, such as MHD [12]. Additionally, physics-based preconditioners have been used to factor the matrix into blocks that are amenable to multigrid-type methods [19, 24].

This work was partially supported by NSF grant DMS-1216972. The work of SM was supported by an NSERC discovery grant. Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52-07NA27344. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94-AL85000. Part of this material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program.

In this paper, we consider monolithic multigrid preconditioners for linearizations of this MHD formulation, which are less common in the field of MHD than the above block preconditioners due to the often greater complexity of implementation. Monolithic multigrid for coupled systems is, in fact, one of the earliest ideas in the multigrid literature [9,10]. Monolithic schemes for incompressible fluid flow problems are well-studied, with treatments of the Stokes equations in [14,23,30] and of the Navier-Stokes equations in [17,18,20,34]. In the MHD literature, a monolithic nonlinear multigrid method is shown in the context of finite differences in [1]. A fully-coupled AMG approach for a vector-potential formulation of resistive MHD is shown in [32]. However, this approach relies upon an equal-order discretization in which unknowns for each variable are collocated at mesh nodes; thus, it cannot be used for the mixed discretization shown here. Similarly, fully-coupled AMG methods have been used in the context of FOSLS discretizations of MHD [4]. Finally, in [2], two families of relaxation methods for the vector-potential formulation are described and used within a monolithic geometric multigrid preconditioner. In this paper, the Vanka-type relaxation method is extended to the Lagrange multiplier formulation described below.

2. Background

Following the the formulation of the MHD equations in [29,31], we introduce a Lagrange multiplier, r, to enforce the solenoidal constraint, $\nabla \cdot \mathbf{B} = 0$. This Lagrange multiplier appears as a nonphysical term, ∇r, in Faraday's law. The set of steady-state MHD equations in a domain $\Omega \subset \mathbb{R}^2$ that we consider in this paper is then written

$$-\nabla \cdot \left[\frac{2}{Re}\boldsymbol{\varepsilon}(\mathbf{u})\right] + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p - (\nabla \times \mathbf{B}) \times \mathbf{B} = \mathbf{f}, \tag{1}$$

$$\frac{1}{Re_m}\nabla \times \nabla \times \mathbf{B} - \nabla \times (\mathbf{u} \times \mathbf{B}) - \nabla r = \mathbf{g}, \tag{2}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{3}$$

$$\nabla \cdot \mathbf{B} = 0. \tag{4}$$

Here, $\mathbf{u}$ is the fluid velocity, p is the hydrodynamic pressure, and $\mathbf{B}$ is the magnetic field, and $\mathbf{f}$ and $\mathbf{g}$ are appropriate source data. The strain-rate tensor is $\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)$. The nondimensional parameters are Re, the hydrodynamic Reynolds number, and Re_m, the magnetic Reynolds number. The system is closed with some appropriate set of boundary conditions.

In two dimensions, we define the curl operator, $\nabla\times$, applied to a vector, $\mathbf{v} = (v_1, v_2)^T$, to be $\nabla \times \mathbf{v} = \frac{\partial v_2}{\partial x} - \frac{\partial v_1}{\partial y}$, and applied to a scalar, s, to be $\nabla \times s = (\frac{\partial s}{\partial y}, -\frac{\partial s}{\partial x})^T$. Likewise, the cross product of two vectors is $\mathbf{u} \times \mathbf{B} = (u_1, u_2)^T \times (b_1, b_2)^T = u_1 b_2 - u_2 b_1$, and the cross product of a scalar with a vector is $s \times \mathbf{v} = (-sv_2, sv_1)^T$. Note that these relationships arise naturally from the embedding of two-dimensional vector fields in three dimensions.

The system (1)–(4) is nonlinear, and thus we employ Newton's method as a nonlinear solver. Accordingly, we must compute the linearizations:

$$J(\mathbf{x}_n)\delta\mathbf{x} = -R(\mathbf{x}_n),$$

where $\mathbf{x}_n = (\mathbf{u}_n, \mathbf{B}_n, p_n, r_n)^T$ is the current approximation to the solution, $\delta\mathbf{x} = \mathbf{x}_{n+1} - \mathbf{x}_n = (\delta\mathbf{u}, \delta\mathbf{B}, \delta p, \delta r)^T$ is the Newton step in the update $\mathbf{x}_{n+1} = \mathbf{x}_n + \delta\mathbf{x}_n$, $J(\mathbf{x}_n)$ is the Jacobian operator around $\mathbf{x}_n$, and $R(\mathbf{x}_n)$ is the nonlinear residual with respect to $\mathbf{x}_n$. In this paper, we consider linearizing the system first, and then discretizing the resulting linearized system via the finite-element method. To define the variational form, we construct the (continuous) solution spaces for the Newton updates:

$$\begin{aligned}
\mathbf{V} &= \mathbf{H}_0^1(\Omega) = \{\mathbf{v} \in \mathbf{H}^1(\Omega) \,:\, \mathbf{v} = 0 \text{ on } \partial\Omega_D\},\\
\mathbf{C} &= \mathbf{H}_0(\mathrm{curl}, \Omega) = \{\mathbf{c} \in \mathbf{L}^2(\Omega) \,:\, \nabla \times \mathbf{c} \in \mathbf{L}^2(\Omega),\ \mathbf{n} \times \mathbf{c} = 0 \text{ on } \partial\Omega\},\\
Q &= L_0^2(\Omega) = \{q \in L^2(\Omega) \,:\, \int_\Omega p \,\mathrm{d}\mathbf{x} = 0\},\\
S &= H_0^1(\Omega) = \{s \in H^1(\Omega) \,:\, s = 0 \text{ on } \partial\Omega\}.
\end{aligned}$$

We denote by $\|\cdot\|_0$ the usual norm on $L^2(\Omega)$ or the vector version $\mathbf{L}^2(\Omega)$. Similarly, we denote by $\|\cdot\|_1$ the usual norm in $H^1(\Omega)$ or $\mathbf{H}^1(\Omega)$. Finally, for a vector $\mathbf{c} \in \mathbf{H}(\mathrm{curl}, \Omega)$, we define $\|\mathbf{c}\|_{\mathrm{curl}}^2 = \|\mathbf{c}\|_0^2 + \|\nabla \times \mathbf{c}\|_0^2$ to be the norm in $\mathbf{H}(\mathrm{curl}, \Omega)$.

We start Newton's method with an initial guess $\mathbf{x}_0 = (\mathbf{u}_0, \mathbf{B}_0, p_0, r_0)^T$ that satisfies the given boundary conditions and, at each linearization, we look for updates to the solution $(\delta\mathbf{u}, \delta\mathbf{B}, \delta p, \delta r) \in \mathbf{V} \times \mathbf{C} \times Q \times S$. The sequence of linear variational problems arising from Newton's method takes the form:

Weak Form (Linearized, continuous). *Given* $\mathbf{x}_n = (\mathbf{u}_n, \mathbf{B}_n, p, r)$, *find* $(\delta\mathbf{u}, \delta\mathbf{B}, \delta p, \delta r) \in \mathbf{V} \times \mathbf{C} \times Q \times S$ *such that*

$$\begin{aligned}(5)\qquad &\int_\Omega \frac{2}{Re}\boldsymbol{\varepsilon}(\delta\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) - \delta p \nabla\cdot\mathbf{v} + [(\mathbf{u}_n\cdot\nabla)\delta\mathbf{u} + (\delta\mathbf{u}\cdot\nabla)\mathbf{u}_n]\cdot\mathbf{v}\,\mathrm{d}\mathbf{x} \\ &\qquad - \int_\Omega [(\nabla\times\mathbf{B}_n)\times\delta\mathbf{B} + (\nabla\times\delta\mathbf{B})\times\mathbf{B}_n]\cdot\mathbf{v}\,\mathrm{d}\mathbf{x} \\ &= \int_\Omega \mathbf{f}\cdot\mathbf{v} - \frac{1}{Re}\boldsymbol{\varepsilon}(\mathbf{u}_n) : \boldsymbol{\varepsilon}(\mathbf{v}) + p_n\nabla\cdot\mathbf{v} - (\mathbf{u}_n\cdot\nabla)\mathbf{u}_n\cdot\mathbf{v} + [(\nabla\times\mathbf{B}_n)\times\mathbf{B}_n]\cdot\mathbf{v}\,\mathrm{d}\mathbf{x} \\ &\qquad + \int_{\partial\Omega_N} p_N\mathbf{n}\cdot\mathbf{v}\,\mathrm{d}S\end{aligned}$$

$$\begin{aligned}(6)\qquad &\int_\Omega \left[\frac{1}{Re_m}(\nabla\times\delta\mathbf{B}) - (\mathbf{u}_n\times\delta\mathbf{B}) - (\delta\mathbf{u}\times\mathbf{B}_n)\right]\cdot(\nabla\times\mathbf{c}) - \nabla\delta r\cdot\mathbf{c}\,\mathrm{d}\mathbf{x} \\ &= \int_\Omega \mathbf{g}\cdot\mathbf{c} - \left[\frac{1}{Re_m}(\nabla\times\mathbf{B}_n) - (\mathbf{u}_n\times\mathbf{B}_n)\right]\cdot(\nabla\times\mathbf{c}) + \nabla r_n\cdot\mathbf{c}\,\mathrm{d}\mathbf{x}\end{aligned}$$

$$(7)\qquad \int_\Omega q\nabla\cdot\delta\mathbf{u}\,\mathrm{d}\mathbf{x} = -\int_\Omega q\nabla\cdot\mathbf{u}_n\,\mathrm{d}\mathbf{x}$$

$$(8)\qquad \int_\Omega \nabla s\cdot\delta\mathbf{B}\,\mathrm{d}\mathbf{x} = -\int_\Omega \nabla s\cdot\mathbf{B}_n\,\mathrm{d}\mathbf{x}$$

for all $(\mathbf{v}, \mathbf{c}, q, s) \in \mathbf{V}\times\mathbf{C}\times Q\times S$.

The boundary terms in (6) vanish because we strongly enforce that $\mathbf{c}\in\mathbf{C} = \mathbf{H}_0(\mathrm{curl},\Omega)$, so that $\mathbf{n}\times\mathbf{c} = 0$. Likewise, the boundary integral in (8) has been eliminated by enforcing $s\in S = H_0^1(\Omega)$.

We now discretize (5)–(8) using a mixed finite-element method. We approximate the solution of this system using finite-element functions $(\delta\mathbf{u}^h, \delta\mathbf{B}^h, \delta p^h, \delta r^h) \in \mathbf{V}^h\times\mathbf{C}^h\times Q^h\times S^h$, where $\mathbf{V}^h\times Q^h$ is a standard inf-sup stable pair for the incompressible Navier-Stokes problem, $\mathbf{C}^h$ is the first family of Nédélec elements [22], and S^h is an H^1-conforming space [29, 31]. For this paper, we consider a discretization of Ω into triangular elements, and thus we choose $\mathbf{P}_2 - P_1$ (Taylor-Hood) elements for $\mathbf{V}^h\times Q^h$. We choose the lowest-order Nédélec space for Q^h and P_1 for S^h. The well-posedness of both the continuous and discrete formulations is shown in [29, 31].

After the finite-element discretization, we must solve a linear system of the following block form for each Newton step:

$$(9)\qquad \mathcal{A}x = \begin{bmatrix} F & Z & B^T & 0 \\ Y & D & 0 & C^T \\ B & 0 & 0 & 0 \\ 0 & C & 0 & 0 \end{bmatrix}\begin{bmatrix} x_{\mathbf{u}} \\ x_{\mathbf{B}} \\ x_p \\ x_r \end{bmatrix} = \begin{bmatrix} f_{\mathbf{u}} \\ f_{\mathbf{B}} \\ f_p \\ f_r \end{bmatrix} = b.$$

Here, $x_{\mathbf{u}}$, $x_{\mathbf{B}}$, x_p, x_r are the discretized Newton corrections for $\mathbf{u}$, $\mathbf{B}$, p, and r, respectively, and $f_{\mathbf{u}}$, $f_{\mathbf{B}}$, f_p, and f_r are the corresponding blocks of the nonlinear residual.

3. Monolithic Multigrid Method

To solve System (9), we will use preconditioned GMRES, as the problem is not symmetric. As (9) is a saddle-point problem, we must use a specialized preconditioner, and we specifically consider the development of an effective monolithic multigrid preconditioner. The choice of relaxation or smoothing operator is known to be critically important to the performance of any multigrid method. In this paper, we fix geometric interpolation operators for each of the finite-element spaces described above and extend the well-known Vanka relaxation scheme to (9).

The components of the method that we fix are as follows. First, we geometrically coarsen elements by a factor of two in each direction. Specifically, to form the coarse grid, we exploit the structured nature of the grid to agglomerate four triangular fine-grid elements into one triangular coarse-grid element. Second,

the interpolation operators are chosen to be block-structured operators in which the diagonal blocks are the standard finite-element interpolation operators corresponding to the test space for each variable. That is,

$$P = \begin{bmatrix} P_{\mathbf{u}} & & & \\ & P_{\mathbf{B}} & & \\ & & P_p & \\ & & & P_r \end{bmatrix},$$

where $P_{\mathbf{u}}$ is the vector-quadratic ($\mathbf{P}_2$) interpolation operator; $P_{\mathbf{B}}$ is the lowest-order first-family Nédélec interpolation operator; and P_p and P_r are both the linear (P_1) interpolation operator. Next, Galerkin coarse-grid operators are used. That is $\mathcal{A}_c = P^T \mathcal{A} P$. Finally, we use $V(1,1)$-cycles, with one pre- and one post-relaxation step.

In this paper, we emphasize Vanka-type relaxation methods [34]. In [2], these methods are used in the context of a finite-element discretization of a vector-potential formulation of the MHD system. This work extends the methods further to the case of the finite-element discretization the Lagrange-multiplier **B**-field formulation (1)–(4). A key difference here is the presence of two Lagrange multipliers, as well as use of the full **B** field, which is now discretized using curl-conforming vector elements.

3.1. **Vanka Relaxation.** We define the sets of degrees of freedom (DOFs) to be $\mathcal{S}_{\mathbf{u}} = \{u_1, \ldots, u_{n_u}\}$, $\mathcal{S}_{\mathbf{B}} = \{b_1, \ldots, b_{n_B}\}$, $\mathcal{S}_p = \{p_1, \ldots, p_{n_p}\}$, and $\mathcal{S}_r = \{r_1, \ldots, r_{n_r}\}$, where n_u, n_B, n_p, and n_r are the numbers of **u**, **B**, p, and r DOFs, respectively, in the system. Note that for this discretization, we have $n_p = n_r$. Let $\mathcal{S} = \mathcal{S}_{\mathbf{u}} \cup \mathcal{S}_{\mathbf{B}} \cup \mathcal{S}_p \cup \mathcal{S}_r$ be the set of all DOFs in the system. The N Vanka blocks, $\mathcal{S}_\ell \subset \mathcal{S}$, $\ell = 1, \ldots, N$, are chosen such that each block contains some elements of each of $\mathcal{S}_{\mathbf{u}}$, $\mathcal{S}_{\mathbf{B}}$, $\mathcal{S}_p$, and $\mathcal{S}_r$. Moreover, the Vanka scheme is an *overlapping* block-Gauss-Seidel method, and, thus, DOFs are allowed to appear in multiple blocks so long as $\cup_\ell \mathcal{S}_\ell = \mathcal{S}$ (i.e., each DOF appears in at least one block).

In fluid dynamics applications, the standard approach to decomposing $\mathcal{S}$ into the subsets, $\mathcal{S}_\ell$, is to "seed" the choice of the Vanka blocks by the incompressibility constraint, or (equivalently) by the pressure degrees of freedom. In this case, however, we have two such constraints, the incompressibility constraint and the solenoidal constraint. In the finite-element discretization, we have chosen the respective finite-element test spaces, Q^h and S^h, to both be represented by P_1 basis functions, which provides a convenient way to form the fully-coupled Vanka blocks. In particular, we pair each DOF in the incompressibility constraint with the corresponding DOF in the solenoidal constraint and we use these pairs to seed the Vanka blocks, which results in $n_p = n_r$ blocks.

Algebraically, this means that we consider row ℓ in each of the matrices B and C in (9) simultaneously, and we define $\mathcal{S}_\ell$ to be the degrees of freedom corresponding to nonzero entries in either of those rows, as well as the seed DOFs, $p_\ell \in \mathcal{S}_p$ and $r_\ell \in \mathcal{S}_r$. Notice that the incompressibility constraint couples a pressure DOF to velocity DOFs and the solenoidal constraint couples a Lagrange-multiplier DOF to magnetics DOFs, resulting in Vanka blocks $\mathcal{S}_\ell$ that include DOFs from all variables in the system. It is important to contrast this to the method shown in [2], in which the solenoidal constraint was eliminated by using the vector-potential formulation. In that case, there was no algebraic method of this type that yielded blocks with contributions from all variables, and geometric information was required to incorporate the vector potential DOFs. However, the use of the Lagrange multiplier, r, to incorporate the solenoidal constraint in this paper means that the algebraic construction is possible.

Geometrically, this can be seen as isolating collocated pressure/Lagrange-multiplier DOFs (located at nodes in the mesh since we use a P_1 discretization) and including all of the velocity and magnetics DOFs in the stencil surrounding them, as well as those seed DOFs. Thus, the incompressibility and solenoidal constraints are both enforced on the DOFs in the Vanka block at each step of the Vanka iteration.

The collection $\{\mathcal{S}_\ell\}$ defines the Vanka blocks upon which the remainder of the method is constructed. For each Vanka block $\mathcal{S}_\ell$, the global solution is updated according to

$$x \leftarrow x + V_\ell \left(\omega M_{\ell\ell}^{-1}\right) V_\ell^T \left(b - \mathcal{A}x\right), \tag{10}$$

and the updates are computed in a Gauss-Seidel fashion. Here, V_ℓ^T is a restriction operator that takes global vectors to local vectors containing only the entries corresponding to DOFs in block ℓ, and V_ℓ is a prolongation operator that takes the entries in a local vector and inserts them appropriately into a global vector [28]. Additionally, we utilize an underrelaxation parameter, ω, to ensure effective relaxation. Finally, $M_{\ell\ell} = V_\ell^T M V_\ell$ is the Vanka submatrix of dimension $|\mathcal{S}_\ell| \times |\mathcal{S}_\ell|$ that is used to compute the update. The

matrix M is chosen to be some suitable preconditioner for the system matrix $\mathcal{A}$; different choices of M differentiate the Vanka methods that we use here.

Remark 1. For the numerical experiments in this paper, we consider ω in Equation (10) to be a scalar. It is possible, however, to consider a diagonal matrix scaling. As we show in Section 4, scaling each component independently was not necessary to achieve good performance in the case of the test problems that we consider.

Remark 2. Note that the definition of $M_{\ell\ell} = V_\ell^T M V_\ell$ follows a linear algebra definition of block Gauss-Seidel. However, one could also imagine lumping techniques that exploit other information, such as finite-element data, to lump off-diagonal entries onto a desired nonzero pattern in the submatrices. These approaches are not considered here.

As the final step to defining the Vanka-type relaxation methods used here, we delineate several methods by the choice of M in (10). In particular, and similarly to [2], we define the "Full" Vanka, "Diagonal" Vanka, and "Economy" Vanka methods. To facilitate the exposition, we define, relative to (9):

$$\begin{aligned} \hat{F} &= \begin{bmatrix} F & Z \\ Y & D \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} B & 0 \\ 0 & C \end{bmatrix}, \\ x_{\hat{\mathbf{u}}} &= \begin{bmatrix} x_{\mathbf{u}} \\ x_{\mathbf{B}} \end{bmatrix}, \qquad x_{\hat{p}} = \begin{bmatrix} x_p \\ x_r \end{bmatrix}, \\ f_{\hat{\mathbf{u}}} &= \begin{bmatrix} f_{\mathbf{u}} \\ f_{\mathbf{B}} \end{bmatrix}, \qquad f_{\hat{p}} = \begin{bmatrix} f_p \\ f_r \end{bmatrix}. \end{aligned} \tag{11}$$

Now, we can rewrite (9) as

$$\mathcal{A}x = \begin{bmatrix} \hat{F} & \hat{B}^T \\ \hat{B} & 0 \end{bmatrix} \begin{bmatrix} x_{\hat{\mathbf{u}}} \\ x_{\hat{p}} \end{bmatrix} = \begin{bmatrix} f_{\hat{\mathbf{u}}} \\ f_{\hat{p}} \end{bmatrix}. \tag{12}$$

We begin by defining the "Full" Vanka method, in which we choose $M = M^{\text{full}} = \mathcal{A}$:

$$M^{\text{full}} = \begin{bmatrix} \hat{F} & \hat{B}^T \\ \hat{B} & 0 \end{bmatrix}.$$

This choice results in submatrices, $M_{\ell\ell}^{\text{full}} = V_\ell^T M^{\text{full}} V_\ell$, that are essentially dense matrices with a maximal size of 52×52. While these are certainly not excessively large, we must solve one such system per Vanka block, and this is quite costly, especially as the number of such blocks grows large. In order to reduce the computational effort per block, the other two Vanka methods replace the dense submatrix with a structured, sparser submatrix that can be inverted more quickly.

The first sparse Vanka method is the "Diagonal" Vanka method, described for the Navier-Stokes equations in [18, 20]. This method uses

$$M = M^{\text{diag}} = \begin{bmatrix} \text{diag}(\hat{F}) & \hat{B}^T \\ \hat{B} & 0 \end{bmatrix},$$

where

$$\text{diag}(\hat{F}) = \begin{bmatrix} \text{diag}(F) & 0 \\ 0 & \text{diag}(D) \end{bmatrix},$$

The Vanka submatrices, $M_{\ell\ell}^{\text{diag}}$, arising from this choice are indeed much more sparse than the Full Vanka submatrices above. As noted in [2], the diagonal approach leads to decoupling of the fluid physics from the magnetic physics. Note that we can expand and permute M^{diag}, with permutation matrix Q, to be

$$Q^T \begin{bmatrix} \text{diag}(\hat{F}) & \hat{B}^T \\ \hat{B} & 0 \end{bmatrix} Q = \left[\begin{array}{cc|cc} \text{diag}(F) & B^T & 0 & 0 \\ B & 0 & 0 & 0 \\ \hline 0 & 0 & \text{diag}(D) & C^T \\ 0 & 0 & C & 0 \end{array}\right], \tag{13}$$

which is a fully-decoupled block-diagonal system in which one block corresponds to the fluid variables, $\mathbf{u}$ and p, and the other to the magnetics variables, $\mathbf{B}$ and r. That is, all coupling between the fluid velocity field and the magnetic field, represented by the matrices Y and Z, has been eliminated. We expect that ignoring this

coupling altogether will lead to diminishing performance in physical regimes in which the coupling between these variables is increased.

The other sparse Vanka method is a modification of the "Economy" Vanka method introduced in [2] for the vector-potential formulation such that it is more compatible with the **B**-field formulation described above. We now consider a preconditioner for $\hat{F}$ defined by

$$\hat{F}^{\text{econ}} = \begin{bmatrix} \text{blkDiag}(F) & \tilde{Z} \\ \tilde{Y} & D \end{bmatrix},$$

where blkDiag(F) is the block-diagonal matrix in which each row has only the two entries corresponding to the collocated x- and y-components of velocity, $\tilde{Z}$ has one entry in each row corresponding to a velocity DOF on an edge that corresponds to the magnetics DOF on that same edge and zero entries in the rows corresponding to nodal velocity DOFs, $\tilde{Y}$ has two entries in each row corresponding to the x- and y-component of velocity DOFs on the edge shared by the magnetics DOF to which the row corresponds, and D is not modified at all. This approach maintains the structure of the Arnold-Falk-Winther relaxation scheme [5,6] for the magnetics block of the system. This gives

$$M = M^{\text{econ}} = \begin{bmatrix} \hat{F}^{\text{econ}} & \hat{B}^T \\ \hat{B} & 0 \end{bmatrix}.$$

Since this approach maintains some entries in Y and Z, the velocity-magnetics coupling is still present, though obviously weaker than in the case of Full Vanka, which results in higher average iteration counts for the test problems shown below.

Computationally, we note that the costs of Full Vanka are highest. The submatrices are dense, and thus require the most storage and the most computational effort to factor. Diagonal Vanka, on the other hand, requires storage equal to three vectors, the diagonal of $\hat{F}_{\ell\ell}$ as well as $B_{\ell\ell}$, $C_{\ell\ell}$ (and possibly $B_{\ell\ell}^T$ and $C_{\ell\ell}^T$ if boundary conditions are not symmetrically eliminated). Thus, the storage is very sparse and the factorization of each submatrix is much simpler than in the Full Vanka case. The Economy Vanka method is a middle ground. It requires substantially less storage than the Full Vanka method, as the submatrices here are quite sparse (only the $D_{\ell\ell}$ block is dense). The submatrix factorization does require that the dense $D_{\ell\ell}$ block be factored, but the size of this block is small enough that the cost is not substantial when compared to the factorization of the Full Vanka submatrices.

4. Numerical Results

In order to test the monolithic multigrid method described above, we consider the Hartmann channel flow problem, which has a known analytical solution. This allows us to verify that we find the correct solution and that the finite-element method is converging as expected. We use Newton's method to solve the nonlinear problem, and we use monolithic multigrid-preconditioned GMRES to solve each linearization. Within the multigrid method, the coarsest grid corresponds to an 8×8 grid of quadrilaterals, each divided into two triangular elements. Table 1 shows the number of elements, unknowns, and nonzero Jacobian entries for each grid size considered below. Additionally, we use only geometric interpolation operators and Galerkin coarse grid operators in $V(1,1)$-cycles. The choice of relaxation operator varies among Full Vanka, Diagonal Vanka, and Economy Vanka.

Remark 3. Note that when grid sizes are quoted as "$N \times N$", N is the number of *quadrilaterals* in the mesh, each of which is divided into two triangles, giving a mesh of $2N^2$ triangular elements.

The discretization for each Newton step is done using the FEniCS library [21]. All of the linear solver components have been implemented using the Trilinos package [16]. The multigrid hierarchy and components are all managed by the MueLu library [26], the Braess-Sarazin relaxation methods are in Teko [11], and the Vanka methods are in Ifpack2 [16]. The Krylov solver is managed by the Belos package [7]. The results shown below have been run in serial on a machine with two Intel Xeon E5-2637 v3 CPUs at 3.50GHz with 256GB DDR3 RAM at 2133MHz.

4.1. Hartmann Problem.

We consider the Hartmann problem on $\Omega = [-\frac{1}{2}, \frac{1}{2}]^2$. This problem models the flow of a fluid through a duct with a transverse applied magnetic field, $\mathbf{B}_D = (0,1)^T$. The flow is driven in the x-direction by an externally applied pressure gradient $\frac{\partial p}{\partial x} = -G$. Furthermore, we assume a channel with insulating top and bottom walls, and there are no external forces applied ($\mathbf{f} = \mathbf{g} = \mathbf{0}$). In this problem, a

TABLE 1. The number of elements in the mesh and the number of unknowns and nonzero entries in the linear systems for the relevant mesh sizes considered below. The "Levels" column indicates the number of levels in a multigrid hierarchy in which the corresponding mesh is the fine grid.

Grid Size	Levels	Elements	Unknowns	NNZ
8×8	1	128	948	35,936
16×16	2	512	3,556	140,336
32×32	3	2,048	13,764	554,576
64×64	4	8,192	54,148	2,204,816
128×128	5	32,768	214,788	8,792,336
256×256	6	131,072	855,556	35,115,536
512×512	7	524,288	3,415,044	140,354,576
1024×1024	8	2,097,152	13,645,828	561,203,216

key nondimensional parameter is the Hartmann number, $Ha = \sqrt{ReRe_m}$. The boundary conditions for this problem are:

$$\begin{aligned} \mathbf{u} &= \mathbf{0} && \text{on } y = \pm\tfrac{1}{2} \\ \mathbf{u} &= (u(y), 0)^T && \text{on } x = \pm\tfrac{1}{2} \\ \mathbf{n} \times \mathbf{B} &= \mathbf{n} \times \mathbf{B}_D && \text{on } \partial\Omega \\ r &= 0 && \text{on } \partial\Omega, \end{aligned}$$

where $u(y)$ is the x-component of the analytical solution described below.

The analytical solution to this problem is $\mathbf{u}^* = (u(y), 0)^T$, $\mathbf{B}^* = (b(y), 1)^T$, $p^* = p(x, y)$ and $r^* = 0$, where

$$\begin{aligned} u(y) &= \frac{GRe}{2Ha\tanh(Ha/2)}\left(1 - \frac{\cosh(yHa)}{\cosh(Ha/2)}\right) \\ b(y) &= \frac{G}{2}\left(\frac{\sinh(yHa)}{\sinh(Ha/2)} - 2y\right) \\ p(x, y) &= -Gx - \frac{1}{2}b(y)^2. \end{aligned}$$

In this study, we vary the Reynolds and magnetic Reynolds numbers across a range of values. The pressure gradient, G, is chosen such that the maximum magnitude of the velocity field is 1.

4.1.1. *Parameter Study.* The Vanka relaxation scheme requires that the damping parameter, ω, be chosen. In order to gain intuition on the appropriate parameter choice, we consider the test problem for $Re = Re_m = 1$. In these experiments, the linear systems were required to be solved to a relative tolerance of 10^{-5} at each Newton step and Newton's method was run until an absolute tolerance of 10^{-8} had been achieved on the norm of the nonlinear residual. We vary the parameter ω from 0.1 to 1.0 in steps of 0.1. For each of the relaxation schemes, we report the average number of GMRES iterations required per Newton step.

The results for the Full Vanka-based method are shown in the left plot of Figure 1. We see that the optimal parameter is $\omega = 0.7$ and that this is independent of the grid size. Here, we observe a sharp upper limit on acceptable ω values that appears to decrease with each grid refinement. As a result of this sharp upper limit, we choose to make a choice of $\omega = 0.6$ out of a feeling of caution when running later experiments with Full Vanka. These results are consistent with the parameter studies shown in [2], from which similar conclusions are drawn (cf. the top row of Figure 2 in that paper).

The results for the Diagonal Vanka-based preconditioner are shown in the center plot of Figure 1. The optimal parameter, regardless of grid size, is shown to be $\omega = 0.5$. While the parameter choice is the same for each grid size, the average number of iterations required increase as the mesh is refined. Thus, the Diagonal Vanka-based method does not lead to grid-independent performance of the preconditioner.

Finally, the results for the Economy Vanka-based preconditioner are shown in the center plot of Figure 1. In this case, the optimal parameter choice is $\omega = 0.6$ for all grid sizes. Unlike the Diagonal Vanka-based method, the Economy Vanka-based method retains the grid-independent performance shown by the Full Vanka-based method while requiring only an increase of 2-3 average iterations.

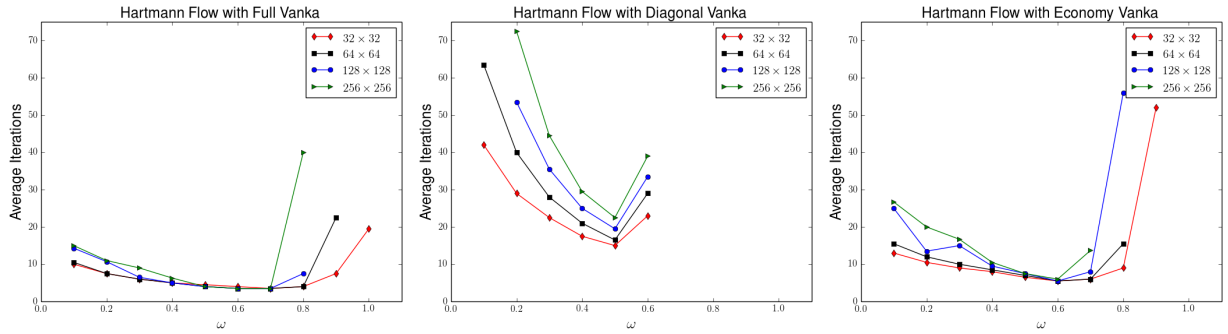

FIGURE 1. The average number of preconditioned GMRES iterations required to reduce the linear residual norm below a relative tolerance of 10^{-5} for the Hartmann flow test problem with $Re = Re_m = 1$ as the underrelaxation parameter, ω, varies from 0.1 to 1.0 in steps of 0.1. Results for the Full Vanka relaxation scheme are shown on the left, for the Diagonal Vanka scheme appear in the center, and for the Economy Vanka scheme appear on the right. Data points not shown indicate that a linear solve failed to converge within 50 iterations and the simulation terminated.

4.1.2. *Robustness to physical parameters and mesh refinement.* Next we consider varying the physical parameters of the problem, namely Re and Re_m. We first consider the Full Vanka-based preconditioner, the results for which are summarized in the top row of Table 2. We see in these tables that the performance of the algorithm is very robust to this range of parameter values when $Re_m < 64$, requiring between 3.5 and 4.5 iterations of GMRES on average to solve each linear system. We do note that for $Re_m = 64$ the performance of both Newton's method and the linear solver suffers. In this case, the initial linearization is solved quickly (6-8 linear iterations), with later linearizations requiring substantially more iterations. For the case in which $Re = 16$ and $Re_m = 64$ on the 128×128 grid, GMRES fails to solve the second linearized system. Finally, we note that the performance of the linear solver is largely unaffected by mesh refinement. In fact, we generally see a slight improvement for $Re < 64$.

Next we discuss results for the Diagonal Vanka-based preconditioner, shown in the middle row of Table 2. We see that the average iteration counts do not scale well with mesh refinement, nearly doubling across the shown grid sizes. As with Full Vanka, this method struggles as Re_m grows large, requiring about twice as many linear iterations as the other parameter values.

Due to the lack of grid-independence shown by the Diagonal Vanka-based preconditioner, we now consider the Economy Vanka-based preconditioner, which has been designed to offer more robustness to mesh refinement as well as lower average iteration counts than the Diagonal Vanka-based method. The results are shown in the bottom row of Table 2. We have used $\omega = 0.5$ for these experiments, to be consistent with the parameter choice for the Diagonal Vanka-based method. Whereas the average iteration count for the Diagonal Vanka-based preconditioner nearly doubled on the 1024×1024, we see a much more mild increase with the Economy Vanka-based method, of about only two iterations in the worst case. We also observe good robustness to the physical parameters when $Re_m < 64$, with average iteration counts varying by about two across the entire range of parameters for both grid sizes shown. In this case, however, the method fails when $Re_m = 64$ and $Re \geq 16$ on the coarser grid and $Re \geq 4$ on the finer grid. The exact cause of this is being investigated and the discovery thereof is included in the future work on this project.

5. Conclusions and Future Work

We have presented the extension of a monolithic multigrid preconditioning techniques to this Lagrange-multiplier formulation. We have shown that using a monolithic multigrid preconditioner with Full or Economy Vanka relaxation leads to an effective preconditioner for the Lagrange-multiplier formulation of the resistive MHD model described here. In particular, we have shown results for a Hartmann flow test problem in two dimensions that demonstrates that this method leads to convergence of the linear systems in a small number of iterations and is robust to both changes in physical parameters as well as mesh refinement.

In [2], we discussed the extension of a Braess-Sarazin-type relaxation scheme to the vector-potential formulation. There is on-going work to extend this method to this Lagrange-multiplier formulation. Moreover

TABLE 2. The average number of preconditioned GMRES iterations required to reduce the linear residual norm below a relative threshold of 10^{-4} for the Hartmann flow test problem for different values of Re and Re_m. The top row shows Full Vanka relaxation with $\omega = 0.6$; the middle row shows Diagonal Vanka with $\omega = 0.5$; and the bottom row shows Economy Vanka with $\omega = 0.5$. The number in parentheses indicates the number of Newton steps taken to reduce the norm of the nonlinear residual below an absolute tolerance of 10^{-8}. Results for the 128×128 problem are shown on the left and for the 1024×1024 problem on the right. A value of "x(x)" indicates that Newton's method broke down.

128×128 — Full Vanka

Re \ Re_m	1	4	16	64
1	4.2(4)	4.0(4)	4.0(4)	9.4(5)
4	4.0(4)	3.8(4)	4.2(5)	13.0(6)
16	3.8(4)	3.5(4)	4.2(5)	x(x)
64	3.8(4)	4.4(5)	4.6(5)	13.0(7)

1024×1024 — Full Vanka

Re \ Re_m	1	4	16	64
1	4.0(4)	3.8(4)	3.5(4)	9.2(5)
4	4.0(4)	3.8(4)	3.8(5)	13.7(6)
16	4.0(4)	3.5(4)	3.6(5)	15.0(7)
64	6.5(4)	6.4(5)	5.0(5)	18.4(7)

128×128 — Diagonal Vanka

Re \ Re_m	1	4	16	64
1	16.8(4)	17.5(4)	17.8(4)	30.6(5)
4	17.2(4)	17.8(4)	18.0(5)	31.7(6)
16	17.2(4)	17.2(4)	17.6(5)	29.1(7)
64	18.0(4)	17.4(5)	17.8(5)	x(x)

1024×1024 — Diagonal Vanka

Re \ Re_m	1	4	16	64
1	28.5(4)	31.2(4)	33.5(4)	66.8(5)
4	29.5(4)	32.5(4)	34.4(5)	60.3(6)
16	30.5(4)	32.2(4)	34.2(5)	70.9(7)
64	32.5(4)	31.2(5)	34.4(5)	x(x)

128×128 — Economy Vanka

Re \ Re_m	1	4	16	64
1	8.8(4)	8.2(4)	7.8(4)	15.2(5)
4	8.2(4)	7.8(4)	8.2(5)	29.0(6)
16	7.5(4)	7.2(4)	7.8(5)	x(x)
64	9.0(5)	8.0(5)	7.8(5)	x(x)

1024×1024 — Economy Vanka

Re \ Re_m	1	4	16	64
1	9.2(4)	9.5(4)	10.2(5)	20.6(5)
4	9.8(5)	8.5(4)	9.2(5)	x(x)
16	9.8(5)	9.6(5)	8.8(5)	x(x)
64	10.0(5)	10.7(6)	10.0(6)	x(x)

we are testing both Vanka-type and Braess-Sarazin-type relaxation methods in a distributed parallel computing environment. Once this is complete, we will consider time-dependent problems in both two and three dimensions, such as the Kelvin-Helmholtz and the Island Coalescence problems [3, 12, 19, 32]. We will then be able to compare the techniques presented here to the state-of-the-art block-factorization preconditioners that are widely used [12, 19, 24, 25].

REFERENCES

[1] M. F. ADAMS, R. SAMTANEY, AND A. BRANDT, *Toward textbook multigrid efficiency for fully implicit resistive magnetohydrodynamics*, Journal of Computational Physics, 229 (2010), pp. 6208 – 6219.

[2] J. ADLER, T. R. BENSON, E. CYR, S. P. MACLACHLAN, AND R. S. TUMINARO, *Monolithic multigrid methods for two-dimensional resistive magnetohydrodynamics*, SIAM Journal on Scientific Computing, 38 (2016), pp. B1–B24.

[3] J. H. ADLER, M. BREZINA, T. A. MANTEUFFEL, S. F. MCCORMICK, J. W. RUGE, AND L. TANG, *Island coalescence using parallel first-order system least squares on incompressible resistive magnetohydrodynamics*, SIAM Journal on Scientific Computing, 35 (2013), pp. S171–S191.

[4] J. H. ADLER, T. A. MANTEUFFEL, S. F. MCCORMICK, J. W. RUGE, AND G. D. SANDERS, *Nested iteration and first-order system least squares for incompressible, resistive magnetohydrodynamics*, SIAM Journal on Scientific Computing, 32 (2010), pp. 1506–1526.

[5] D. ARNOLD, R. FALK, AND R. WINTHER, *Preconditioning in H(div) and applications*, Math. Comp., 66 (1997), pp. 957–984.

[6] ———, *Multigrid in H(div) and H(curl)*, Numer. Math., 85 (2000), pp. 197–217.

[7] E. BAVIER, M. HOEMMEN, S. RAJAMANICKAM, AND H. THORNQUIST, *Amesos2 and Belos: Direct and iterative solvers for large sparse linear systems*, Scientific Programming, 20 (2012), pp. 241 – 255.

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

[9] A. BRANDT, *Multigrid techniques: 1984 guide with applications to fluid dynamics*, GMD–Studien Nr. 85, Gesellschaft für Mathematik und Datenverarbeitung, St. Augustin, 1984.

[10] A. BRANDT AND N. DINAR, *Multigrid solutions to elliptic flow problems*, in Numerical Methods for Partial Differential Equations, S. V. Parter, ed., Academic Press, New York, 1979, pp. 53–147.

[11] E. Cyr, J. Shadid, and R. Tuminaro, *Teko: A block preconditioning capability with concrete example applications in Navier-Stokes and MHD*, Accepted, SISC, (2016).
[12] E. Cyr, J. Shadid, R. Tuminaro, R. Pawlowski, and L. Chacón, *A new approximate block factorization preconditioner for two-dimensional incompressible (reduced) resistive MHD*, SIAM Journal on Scientific Computing, 35 (2013), pp. B701–B730.
[13] H. Elman, V. E. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro, *A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible Navier-Stokes equations*, J. Comput. Phys., 227 (2008), pp. 1790–1808.
[14] F. Gaspar, Y. Notay, C. Oosterlee, and C. Rodrigo, *A simple and efficient segregated smoother for the discrete Stokes equations*, SIAM Journal on Scientific Computing, 36 (2014), pp. A1187–A1206.
[15] C. Greif, D. Li, D. Schötzau, and X. Wei, *A mixed finite element method with exactly divergence-free velocities for incompressible magnetohydrodynamics*, Computer Methods in Applied Mechanics and Engineering, 199 (2010), pp. 2840 – 2855.
[16] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley, *An overview of the Trilinos project*, ACM Trans. Math. Softw., 31 (2005), pp. 397–423.
[17] V. John and G. Matthies, *Higher-order finite element discretizations in a benchmark problem for incompressible flows*, International Journal for Numerical Methods in Fluids, 37 (2001), pp. 885–903.
[18] V. John and L. Tobiska, *Numerical performance of smoothers in coupled multigrid methods for the parallel solution of the incompressible Navier-Stokes equations*, International Journal for Numerical Methods in Fluids, 33 (2000), pp. 453–473.
[19] D. A. Knoll and L. Chacón, *Coalescence of magnetic islands, sloshing, and the pressure problem*, Physics of Plasmas, 13 (2006).
[20] M. Larin and A. Reusken, *A comparative study of efficient iterative solvers for generalized Stokes equations*, Numerical Linear Algebra with Applications, 15 (2008), pp. 13 – 34.
[21] A. Logg, K.-A. Mardal, G. N. Wells, et al., *Automated Solution of Differential Equations by the Finite Element Method*, Springer, 2012.
[22] J.-C. Nédélec, *Mixed finite elements in* $\mathbf{R}^3$, Numer. Math., 35 (1980), pp. 315–341.
[23] C. Oosterlee and F. Gaspar, *Multigrid relaxation methods for systems of saddle point type*, Applied Numerical Mathematics, 58 (2008), pp. 1933 – 1950.
[24] B. Philip, L. Chacón, and M. Pernice, *Implicit adaptive mesh refinement for 2D reduced resistive magnetohydrodynamics*, Journal of Computational Physics, 227 (2008), pp. 8855 – 8874.
[25] E. G. Phillips, H. C. Elman, E. C. Cyr, J. N. Shadid, and R. P. Pawlowski, *A block preconditioner for an exact penalty formulation for stationary MHD*, SIAM Journal on Scientific Computing, 36 (2014), pp. B930–B951.
[26] A. Prokopenko, J. J. Hu, T. A. Wiesner, C. M. Siefert, and R. S. Tuminaro, *MueLu users guide 1.0*, Tech. Rep. SAND2014-18874, Sandia National Labs, 2014.
[27] J. W. Ruge and K. Stüben, *Algebraic multigrid (AMG)*, in Multigrid Methods, S. F. McCormick, ed., vol. 3 of Frontiers in Applied Mathematics, SIAM, Philadelphia, PA, 1987, pp. 73–130.
[28] Y. Saad, *Iterative methods for sparse linear systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, second ed., 2003.
[29] A. Schneebeli and D. Schötzau, *Mixed finite elements for incompressible magneto-hydrodynamics*, Comptes Rendus Mathematique, 337 (2003), pp. 71 – 74.
[30] J. Schöberl and W. Zulehner, *On Schwarz-type smoothers for saddle point problems*, Numer. Math., 95 (2003), pp. 377–399.
[31] D. Schötzau, *Mixed finite element methods for stationary incompressible magnetohydrodynamics*, Numerische Mathematik, 96 (2004), pp. 771–800.
[32] J. Shadid, R. Pawlowski, J. Banks, L. Chacón, P. Lin, and R. Tuminaro, *Towards a scalable fully-implicit fully-coupled resistive MHD formulation with stabilized FE methods*, Journal of Computational Physics, 229 (2010), pp. 7649 – 7671.
[33] M. ur Rehman, T. Geenen, C. Vuik, G. Segal, and S. P. MacLachlan, *On iterative methods for the incompressible Stokes problem*, International Journal for Numerical Methods in Fluids, 65 (2011), pp. 1180–1200.
[34] S. P. Vanka, *Block-implicit multigrid calculation of two-dimensional recirculating flows*, Computer Methods in Applied Mechanics and Engineering, 59 (1986), pp. 29 – 48.
[35] R. Verfürth, *A combined conjugate gradient - multi-grid algorithm for the numerical solution of the Stokes problem*, IMA Journal of Numerical Analysis, 4 (1984), pp. 441–455.

Lawrence Livermore National Laboratory, Livermore, CA
E-mail address: benson31@llnl.gov

Department of Mathematics, Tufts University, Medford, MA
E-mail address: james.adler@tufts.edu

Sandia National Laboratories, Albuquerque, NM
E-mail address: eccyr@sandia.gov

Department of Mathematics and Statistics, Memorial University of Newfoundland, St. John's, NL, Canada
E-mail address: smaclachlan@mun.ca

Sandia National Laboratories, Livermore, CA
E-mail address: rstumin@sandia.gov