UNIVERSITY OF MARYLAND COMPUTER SCIENCE CS-TR-5032
UNIVERSITY OF MARYLAND INSTITUTE FOR ADVANCED COMPUTER STUDIES TR-2014-02

A BLOCK PRECONDITIONER FOR AN EXACT PENALTY FORMULATION FOR STATIONARY MHD[*]

EDWARD G. PHILLIPS[†], HOWARD C. ELMAN[‡], ERIC C. CYR[§], JOHN N. SHADID[§], AND ROGER P. PAWLOWSKI[§]

Abstract. The magnetohydrodynamics (MHD) equations are used to model the flow of electrically conducting fluids in such applications as liquid metals and plasmas. This system of non-self adjoint, nonlinear PDEs couples the Navier-Stokes equations for fluids and Maxwell's equations for electromagnetics. There has been recent interest in fully coupled solvers for the MHD system because they allow for fast steady-state solutions that do not require pseudo-time stepping. When the fully coupled system is discretized, the strong coupling can make the resulting algebraic systems difficult to solve, requiring effective preconditioning of iterative methods for efficiency. In this work, we consider a finite element discretization of an exact penalty formulation for the stationary MHD equations. This formulation has the benefit of implicitly enforcing the divergence free condition on the magnetic field without requiring a Lagrange multiplier. We consider extending block preconditioning techniques developed for the Navier-Stokes equations to the full MHD system. We analyze operators arising in block decompositions from a continuous perspective and apply arguments based on the existence of approximate commutators to develop new preconditioners that account for the physical coupling. This results in a family of parameterized block preconditioners for both Picard and Newton linearizations. We develop an automated method for choosing the relevant parameters and demonstrate the robustness of these preconditioners for a range of the physical non-dimensional parameters and with respect to mesh refinement.

Key words. magnetohydrodynamics, iterative methods, preconditioners

AMS subject classifications. 76W05, 65F08, 65M22

1. Introduction.

The magnetohydrodynamics (MHD) model describes the flow of electrically conducting fluids in the presence of magnetic fields. A principal application of MHD is the modeling of plasma physics, ranging from plasma confinement for thermonuclear fusion to astrophysical plasma dynamics [13]. MHD is also used to model the flow of liquid metals, for instance in magnetic pumps, liquid metal blankets in fusion reactor concepts, and aluminum electrolysis [19]. The model consists of a non-self-adjoint, nonlinear system of partial differential equations (PDEs) that couple the Navier-Stokes equations for fluid flow to a reduced set of Maxwell's equations for electromagnetics. Because multiple physical processes are represented in the model, the PDEs can span over a range of length- and time-scales, making the equations difficult to solve and requiring a robust, accurate means of approximating the solution. Decoupled solution methods which solve the fluid and magnetic systems separately and possibly couple the systems by an outer iteration have been commonly employed as solvers for the transient and steady MHD systems. In the context of transient systems these methods are commonly used in operator splitting techniques, for steady state solves a fixed point iteration serves to couple the system (see e.g. [2], and the references in [23]).

[*] This work was supported in part by the U. S. Department of Energy under grant DEFG0204ER25619 and contract DE-AC04-94AL85000, and by the U. S. National Science Foundation under grant DMS1115317.

[†] Applied Mathematics & Statistics, and Scientific Computation Program, University of Maryland, College Park, MD (egphillips@math.umd.edu)

[‡] Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD (elman@cs.umd.edu)

[§] Sandia National Laboratories, Albuquerque, NM (eccyr@sandia.gov, jnshadi@sandia.gov, rppawlo@sandia.gov)

Instead, we focus on developing solvers for the fully coupled MHD system, accounting for the fluid and magnetic systems and the coupling between them simultaneously, enabling us to obtain steady-state solutions quickly. When the fully coupled system is discretized, the strong coupling can make the resulting algebraic systems difficult to solve. For realistic applications where the systems are very large, effective preconditioning of iterative methods is required for efficiency. Some recently developed solvers for fully coupled MHD formulations include a coupled AMG preconditioned Newton-Krylov method for a vector potential formulation [23], a multigrid preconditioned Newton-Krylov method for a parabolic reformulation of the MHD equations [1], and a block preconditioned Newton-Krylov method for a vector potential formulation [4].

In this work, we consider an exact penalty formulation of the MHD equations [14]. This formulation, although limited to convex domains, leads to a discrete three-variable block structure which we will show to be amenable to block preconditioning strategies of a type used successfully for the Navier-Stokes equations. We consider linear systems arising from both Newton and Picard linearizations. Different linearizations lead to variations in local operators within the block structure of the linear systems, and we will investigate the sensitivity of solvers in relation to these structural differences, allowing for robust preconditioning for a range of approximation techniques. Because of the MHD equations' inherent relationship to the Navier-Stokes equations, we consider extensions of solution strategies based on block preconditioners developed for the discretized Navier-Stokes equations [7, 8]. These preconditioning strategies are based on the assumption that certain differential operators corresponding to blocks of the discrete system approximately commute with one another. Using deiscrete analogues of continuous commuting relationships leads to discrete operators that can be used as block preconditioners.

The remainder of the paper is structured as follows. In Section 2 we develop the exact penalty weak formulation for the MHD equations and consider issues of discretization and linearization. In Section 3, we consider preconditioners for the linear systems arising from Picard linearization in two dimensions. We look specifically at structural issues, such as the importance of ordering of variables for the block preconditioning strategy, and we modify preconditioning strategies for the Navier-Stokes equations for MHD. In Section 4, we extend the strategies developed for the Picard iteration to linear systems arising from Newton's method for MHD problems. Finally, in Section 5 we demonstrate the effectiveness of our preconditioners on a set of example problems, experimenting with relevant parameters, and in Section 6 we draw conclusions.

2. An Exact Penalty Weak Formulation and Finite Element Discretization.

We consider the steady-state equations of MHD for a single incompressible, homogeneous fluid with the electromagnetic properties of a perfect dielectric medium. In a non-dimensionalized form, these equations can be written as

$$\vec{u}\cdot\nabla\vec{u} - R^{-1}\Delta\vec{u} + \nabla p + S\vec{B}\times(\nabla\times\vec{B}) = \vec{f}, \tag{2.1a}$$

$$\nabla\cdot\vec{u} = 0, \tag{2.1b}$$

$$R_m^{-1}\nabla\times(\nabla\times\vec{B}) - \nabla\times(\vec{u}\times\vec{B}) = \vec{0}, \tag{2.1c}$$

$$\nabla\cdot\vec{B} = 0, \tag{2.1d}$$

where the unknowns are the fluid velocity $\vec{u}$, the pressure p, and the magnetic induction $\vec{B}$. The parameters are the (fluid) Reynolds number R, the magnetic Reynolds number R_m, and the coupling coefficient S, and $\vec{f}$ is a forcing term. In this study, we complete the system by prescribing the boundary conditions

$$\vec{u} = \vec{g}, \tag{2.2a}$$

$$\vec{B} \times \vec{n} = \vec{q}, \tag{2.2b}$$

on $\partial\Omega$. In this form, we can clearly see the incompressible Navier-Stokes equations in (2.1a) and (2.1b) and Maxwell's equations in (2.1c) and (2.1d). The two equation sets are coupled by the Lorentz force $S\vec{B} \times (\nabla \times \vec{B})$ on the momentum equation and the magnetic convection term $-\nabla \times (\vec{u} \times \vec{B})$ on the induction equation.

It is important to observe that (2.1) is overdetermined in that it is a system of $2d + 2$ equations in only $2d + 1$ unknowns. Many strategies exist for incorporating the solenoidal condition (2.1d) into the other three equations to ensure the solvability of the system. These methods include exact penalty [11, 14], Lagrange multiplier [2, 22], vector potential [17, 23], and divergence cleaning [5] formulations. For this work, we will use an exact penalty formulation in which the solenoidal condition is weakly enforced in the weak form. Although it only holds for convex domains, this exact penalty formulation has the benefit of not requiring the addition of a Lagrange multiplier and allowing for the use of nodal elements. We will also see that the addition of the exact penalty term eliminates the occurence of the singular curl-curl operator, replacing it with something similar to a weak Laplacian. Furthermore, it can be shown that for this weak formulation, there is only an inf-sup stability condition on $\vec{u}$ and p; that is, there is no stability constraint on the magnetic induction $\vec{B}$. The structural simplicity implied by these benefits then makes the exact penalty formulation a prime avenue for studying the algebraic complexities of MHD systems.

In order to use the exact penalty weak formulation of the MHD equations, we restrict our study to the case where Ω is a bounded convex polyhedron. In this setting, note that $H(curl, \Omega) \cap H(div, \Omega)$ is embedded in $(H^1(\Omega))^d$ [3]. Furthermore, to simplify the statement of the weak formulation, we consider homogeneous Dirichlet boundary conditions, i.e. $\vec{g} = \vec{q} = \vec{0}$. Then, we consider weak solutions $\vec{u} \in \mathbf{H}_0^1(\Omega), p \in L^2(\Omega), \vec{B} \in \mathbf{H}_\tau^1(\Omega)$ to the MHD equations (2.1), where $\mathbf{H}_0^1(\Omega) = (H_0^1(\Omega))^d$ and $\mathbf{H}_\tau^1(\Omega) = \{\vec{v} \in (H^1(\Omega))^d | \vec{v} \times \vec{n} = 0\}$. Defining the space $W = \mathbf{H}_0^1(\Omega) \times L^2(\Omega) \times \mathbf{H}_\tau^1(\Omega)$ and representing the solution as $U = (\vec{u}, p, \vec{B})$ and the test function as $V = (\vec{v}, q, \vec{C})$, we state the exact penalty nonlinear weak formulation as: Find $U \in W$ such that

$$\mathcal{N}(U, V) = \langle \vec{f}, \vec{v} \rangle, \quad \forall V \in W, \tag{2.3}$$

where the nonlinear form $\mathcal{N}$ is

$$\begin{aligned} \mathcal{N}(U, V) = {} & (\vec{u} \cdot \nabla\vec{u}, \vec{v}) + R^{-1}(\nabla\vec{u}, \nabla\vec{v}) - (p, \nabla \cdot \vec{v}) + (q, \nabla \cdot \vec{u}) \\ & + S(\vec{v} \times \vec{B}, \nabla \times \vec{B}) - S(\vec{u} \times \vec{B}, \nabla \times \vec{C}) \\ & + SR_m^{-1}(\nabla \times \vec{B}, \nabla \times \vec{C}) + SR_m^{-1}(\nabla \cdot \vec{B}, \nabla \cdot \vec{C}). \end{aligned} \tag{2.4}$$

Every term in this form except the final one is obtained from multiplying (2.1a) by $\vec{v}$, (2.1b) by q, and (2.1c) by $S\vec{C}$, integrating by parts, and summing. The term $SR_m^{-1}(\nabla \cdot \vec{B}, \nabla \cdot \vec{C})$ is the exact penalty term, included to weakly enforce the solenoidal constraint (2.1d). This follows from the fact that if $\vec{B} \in \mathbf{H}^1(\Omega)$ and Ω is a bounded convex polyhedron, then there exists a scalar $c \in H^2(\Omega)$ such that

$$\nabla \cdot \nabla c = \nabla \cdot \vec{B}, \tag{2.5}$$

with $c = 0$ on $\partial\Omega$ and $\nabla c \in \mathbf{H}^1_\tau(\Omega)$ [14]. Letting $V = (\vec{0}, 0, \nabla c)$, we obtain from (2.3) that $(\nabla \cdot \vec{B}, \nabla \cdot \vec{B}) = 0$, and hence that (2.1d) is enforced almost everywhere in Ω.

Linearizing the nonlinear weak formulation (2.3) leads to a set of systems of the form

$$\mathcal{B}^{(n)}(\delta U, V) = R^{(n)}, \forall V \in W, \tag{2.6a}$$
$$U^{(n+1)} = U^{(n)} + \delta U, \tag{2.6b}$$

where $R^{(n)}$ is the nonlinear residual and $\mathcal{B}^{(n)}$ is a bilinear form defined by the linearization method. Two common linearization techniques are Picard iteration and Newton's method. A version of Picard iteration that leads to a coercive weak form gives the bilinear form

$$\begin{aligned} \mathcal{B}^{(n)}_P(U, V) = (\vec{a} \cdot \nabla \vec{u}, \vec{v}) + R^{-1}(\nabla \vec{u}, \nabla \vec{v}) - (p, \nabla \cdot \vec{v}) + (q, \nabla \cdot \vec{u}) \\ + S(\vec{v} \times \vec{b}, \nabla \times \vec{B}) - S(\vec{u} \times \vec{b}, \nabla \times \vec{C}) \\ + SR_m^{-1}(\nabla \times \vec{B}, \nabla \times \vec{C}) + SR_m^{-1}(\nabla \cdot \vec{B}, \nabla \cdot \vec{C}), \end{aligned} \tag{2.7}$$

where

$$\vec{a} = \vec{u}^{(n)}, \quad \vec{b} = \vec{B}^{(n)} \tag{2.8}$$

are the previous velocity and magnetic field in the Picard iteration. This results in linear forms of the fluid convection $(\vec{a} \cdot \nabla \vec{u}, \vec{v})$, the Lorentz force $S(\vec{v} \times \vec{b}, \nabla \times \vec{B})$ and the magnetic convection $-S(\vec{u} \times \vec{b}, \nabla \times \vec{C})$. Newton's method takes as its bilinear form

$$\begin{aligned} \mathcal{B}^{(n)}_N(U, V) = (\vec{a} \cdot \nabla \vec{u}, \vec{v}) + (\vec{u} \cdot \nabla \vec{a}, \vec{v}) + R^{-1}(\nabla \vec{u}, \nabla \vec{v}) - (p, \nabla \cdot \vec{v}) \\ + (q, \nabla \cdot \vec{u}) + S(\vec{v} \times \vec{b}, \nabla \times \vec{B}) + S(\vec{v} \times \vec{B}, \nabla \times \vec{b}) \\ - S(\vec{u} \times \vec{b}, \nabla \times \vec{C}) - S(\vec{a} \times \vec{B}, \nabla \times \vec{C}) \\ + SR_m^{-1}(\nabla \times \vec{B}, \nabla \times \vec{C}) + SR_m^{-1}(\nabla \cdot \vec{B}, \nabla \cdot \vec{C}). \end{aligned} \tag{2.9}$$

It is known that both bilinear forms (2.7) and (2.9) are continuous and coercive and satisfy an inf-sup condition on W [14]. Thus, when a unique solution to (2.3) exists, there exists a unique solution to each linear problem in the above Picard iteration. Furthermore, the nonlinear iteration converges to the unique solution to (2.3) from any initial iterate. Newton's method also converges when a unique solution exists provided that the initial iterate is close enough to the solution [14]. In this case, Newton's method convergences quadratically.

Analogous results are proven when the finite element method is applied and W is replaced with a finite dimensional subspace W_h. In this case, the major difference is that the inf-sup condition must be satisfied on W_h, i.e. we require there to exist a constant $\beta > 0$ such that

$$\inf_{q \in W_{\mathbf{p}}} \sup_{\vec{v} \in W_{\mathbf{u}}} \frac{(\nabla \cdot \vec{v}, q)}{||\vec{v}||_1 ||q||_0} \geq \beta, \tag{2.10}$$

where $W_{\mathbf{p}}$ is the discrete space for p and $W_{\mathbf{u}}$ is the discrete space for $\vec{u}$. Note that $\vec{B}$ does not appear in this condition, and it is exactly the same condition required for

Discrete	Continuous	Interpretation	Approximate Norm
$A\mathbf{B}$	$-SR_m^{-1}\Delta\vec{B}$	Magnetic diffusion	$SR_m^{-1}h^{-2}$
$\tilde{A}\mathbf{B}$	$-S\nabla\times(\vec{a}\times\vec{B})$	Magnetic convection	$S\Vert\vec{a}\Vert h^{-1}$
$-Z^t\mathbf{u}$	$-S\nabla\times(\vec{u}\times\vec{b})$	Magnetic convection	$S\Vert\vec{b}\Vert h^{-1}$
$Z\mathbf{B}$	$S\vec{b}\times(\nabla\times\vec{B})$	Lorentz force	$S\Vert\vec{b}\Vert h^{-1}$
$\tilde{Z}\mathbf{B}$	$S\vec{B}\times(\nabla\times\vec{b})$	Lorentz force	$S\Vert\nabla\times\vec{b}\Vert$
$F\mathbf{u}$	$\vec{a}\cdot\nabla\vec{u}-R^{-1}\Delta\vec{u}$	Fluid convection-diffusion	$\Vert\vec{a}\Vert h^{-1}+R^{-1}h^{-2}$
$\tilde{F}\mathbf{u}$	$\vec{u}\cdot\nabla\vec{a}$	Fluid convection	$\Vert\nabla\vec{a}\Vert$
$B^t\mathbf{p}$	∇p	Pressure gradient	h^{-1}
$-B\mathbf{u}$	$\nabla\cdot\vec{u}$	Divergence	h^{-1}

Table 2.1

Definitions of discrete operators as they correspond to continuous operators.

stability of the discrete Navier-Stokes equations. Thus, there is no restriction on the discrete space chosen to approximate $\vec{B}$, and any stable element pair for the Navier-Stokes equations (e.g. Q_2-Q_1) can be used for $\vec{u}$ and p. For ease of implementation, we will discretize $\vec{B}$ using the same finite-dimensional space used for $\vec{u}$.

Applying the finite element method, with a stable $\vec{u}$-p element pair, to the linearized formulation (2.6), we obtain a sequence of linear systems of the form

$$\mathcal{A}\mathbf{x}=\mathbf{f}, \tag{2.11}$$

where $\mathbf{x}=(\mathbf{u},\mathbf{p},\mathbf{B})$ contains the coefficients of the discrete solution δU, $\mathbf{f}$ is the discrete nonlinear residual, and $\mathcal{A}$ is the discretization of the weak form. (We will represent vector coefficients in boldface.) The structure of $\mathcal{A}$ depends on the bilinear form associated with the linearization. The matrices resulting from the Picard and Newton linearizations can be written in block form as

$$\mathcal{A}_P=\begin{pmatrix} F & B^t & Z \\ B & 0 & 0 \\ -Z^t & 0 & A \end{pmatrix},\quad \mathcal{A}_N=\begin{pmatrix} F+\tilde{F} & B^t & Z+\tilde{Z} \\ B & 0 & 0 \\ -Z^t & 0 & A+\tilde{A} \end{pmatrix}, \tag{2.12}$$

where the component matrices derive from continuous operators as in Table 2.1. Note that the extra terms $\tilde{F}$, $\tilde{Z}$, and $\tilde{A}$ are are due to the additional coupling in the Newton weak form, corresponding to the terms in the right-hand side of the identity

$$\mathcal{B}_N^{(n)}-\mathcal{B}_P^{(n)}=(\vec{u}\cdot\nabla\vec{a},\vec{v})+S(\vec{v}\times\vec{B},\nabla\times\vec{b})-S(\vec{a}\times\vec{B},\nabla\times\vec{C}). \tag{2.13}$$

For either linearization, $\mathcal{A}$ is a large, sparse matrix. Thus, for efficiency, a preconditioned iterative method should be considered for solving the systems (2.11). Because $\mathcal{A}$ is nonsymmetric and indefinite, we use preconditioned GMRES for these solves.

3. A Block Preconditioner For the 2D Picard System. We consider preconditioning $\mathcal{A}_P$ using a strategy based on approximating Schur complements that generalizes techniques commonly employed for discretizations of the Navier-Stokes equations. To motivate our preconditioning strategy, we consider the block LU decomposition of $\mathcal{A}_P$,

$$\mathcal{A}_P=\begin{pmatrix} I & 0 & 0 \\ BF^{-1} & I & 0 \\ -Z^tF^{-1} & Z^tF^{-1}B^tX^{-1} & I \end{pmatrix}\begin{pmatrix} F & B^t & Z \\ 0 & X & -BF^{-1}Z \\ 0 & 0 & Y \end{pmatrix}, \tag{3.1}$$

where

$$X = -BF^{-1}B^t, \tag{3.2}$$

$$Y = A + Z^tF^{-1}Z + Z^tF^{-1}B^tX^{-1}BF^{-1}Z. \tag{3.3}$$

It is easy to show that the minimum polynomial for the block lower triangular factor $L = \mathcal{A}_PU^{-1}$ is $(I-L)^3$. The minimum polynomial is cubic, which implies that if we could use U as a right preconditioner for $\mathcal{A}_P$, then preconditioned GMRES would converge in at most three iterations [20]. In practice, it is infeasible to apply the action of U^{-1} exactly. Hence, we construct preconditioners by developing techniques for approximating the actions of the inverses of the matrices on the block diagonal of the upper triangular factor. Thus, we consider the preconditioner

$$\mathcal{P}_P = \begin{pmatrix} \hat{F} & B^t & Z \\ 0 & \hat{X} & -B\hat{F}^{-1}Z \\ 0 & 0 & \hat{Y} \end{pmatrix}, \tag{3.4}$$

where “hatted” operators indicate approximations. The convection-diffusion operator F can be handled well by multigrid, and many effective approximations exist for the pressure Schur complement X arising in discretizations of the Navier-Stokes equations [7]. The new difficulty is the Schur complement Y associated with the magnetic field. The nesting of multiple inverse operators as well as the summing of several terms within Y presents an additional challenge in developing expressions for $\hat{Y}$.

Note that the structure of $\mathcal{P}_P$ in (3.4) derives from the ordering used for the components of $\mathbf{x}$, $(\mathbf{u}, \mathbf{p}, \mathbf{B})$. If the components are reordered, $\mathcal{A}_P$ has a different block structure, as do the resulting block LU decompositions. We consider only reorderings in which the rows and columns have the same ordering, so that square blocks remain on the diagonal. The only two other orderings that permit block LU decompositions are $(\mathbf{u}, \mathbf{B}, \mathbf{p})$ and $(\mathbf{B}, \mathbf{u}, \mathbf{p})$. Like $(\mathbf{u}, \mathbf{p}, \mathbf{B})$, the ordering $(\mathbf{u}, \mathbf{B}, \mathbf{p})$ gives rise to Schur complements that are nested, multi-term, and for this reason, we will not pursue this ordering further. We note that preconditioning a system similar in structure to that obtained from the $(\mathbf{u}, \mathbf{p}, \mathbf{B})$ ordering has been studied from another perspective in [4].

The ordering $(\mathbf{B}, \mathbf{u}, \mathbf{p})$ gives the expressions below. (We use the same notation for the complete matrices as above despite the change in ordering. Thus, for the remainder of this paper, $\mathcal{A}_P, \mathcal{A}_N, X$, and Y refer to operators arising from the $(\mathbf{B}, \mathbf{u}, \mathbf{p})$ ordering.)

$$\mathcal{A}_P = \begin{pmatrix} A & -Z^t & 0 \\ Z & F & B^t \\ 0 & B & 0 \end{pmatrix}, \quad \mathcal{A}_N = \begin{pmatrix} A+\tilde{A} & -Z^t & 0 \\ Z+\tilde{Z} & F+\tilde{F} & B^t \\ 0 & B & 0 \end{pmatrix}. \tag{3.5}$$

We then have the block LU decomposition

$$\mathcal{A}_P = \begin{pmatrix} I & 0 & 0 \\ ZA^{-1} & I & 0 \\ 0 & BX^{-1} & I \end{pmatrix} \begin{pmatrix} A & -Z^t & 0 \\ 0 & X & B^t \\ 0 & 0 & Y \end{pmatrix}, \tag{3.6}$$

where

$$X = F + ZA^{-1}Z^t, \tag{3.7}$$

$$Y = -BX^{-1}B^t. \tag{3.8}$$

The performance of a preconditioner based on this factorization is now contingent upon developing effective approximations for the operators X and Y. Observe that X here can be viewed as a perturbed convection-diffusion operator. In Section 3.1, we will expand on the nature of the perturbation $ZA^{-1}Z^t$. In this light, we argue that strategies developed for the Navier-Stokes Schur complement $-BF^{-1}B^t$ can be employed to approximate Y. This will be demonstrated in Section 3.2.

3.1. An approximation for X. Consider the analogue of (3.7) in the continuous space, i.e., where the discrete operators are replaced with their corresponding continuous operators from table 2.1. Then X can be viewed as an approximation to the continuous operator

$$\mathcal{F}+\mathcal{K}, \tag{3.9}$$

where

$$\mathcal{F}\vec{u} := \vec{a}\cdot\nabla\vec{u} - R^{-1}\Delta\vec{u} \tag{3.10}$$

is the convection-diffusion operator and

$$\begin{aligned}\mathcal{K}\vec{u} &:= S\vec{b}\times\{\nabla\times(-SR_m^{-1}\Delta)^{-1}[S\nabla\times(\vec{u}\times\vec{b})]\}\\ &= -SR_m\vec{b}\times\nabla\times\Delta^{-1}\nabla\times(\vec{u}\times\vec{b})\end{aligned} \tag{3.11}$$

is an operator resulting from the coupling between $\vec{u}$ and $\vec{B}$. For two-dimensional problems, we have the identity

$$-\Delta_p c = \nabla\times\nabla\times c \tag{3.12}$$

for any scalar functions c, where we use the subscript p to indicate the scalar Laplacian as opposed to the vector Laplacian Δ. Furthermore, the Laplacian and the curl commute in 2D such that

$$\Delta\nabla\times c = \nabla\times\Delta_p c. \tag{3.13}$$

Replacing c with $\Delta_p^{-1}c$, this yields

$$\nabla\times\Delta_p^{-1}c = \Delta^{-1}\nabla\times c. \tag{3.14}$$

Together with (3.12), this relation implies that

$$\nabla\times\Delta^{-1}\nabla\times c = -c. \tag{3.15}$$

Applying this identity to the expression (3.11) where $c=\vec{u}\times\vec{b}$, we obtain

$$\mathcal{K}\vec{u} = SR_m\vec{b}\times(\vec{u}\times\vec{b}). \tag{3.16}$$

A discretization of this operator can now easily be constructed and we obtain an approximation for X of the form

$$X \approx \hat{X} := F + K, \tag{3.17}$$

where K is a discretization of $\mathcal{K}$. K can be written as

$$K = SR_m Q_{\mathbf{u}} \begin{pmatrix} diag(b_y^2) & diag(-b_x b_y) \\ diag(-b_x b_y) & diag(b_x^2) \end{pmatrix}, \tag{3.18}$$

where $Q_{\mathbf{u}}$ is the velocity mass matrix, b_x and b_y are the x and y components of $\vec{b}$, and $diag(c)$ is a diagonal matrix containing the values of the function c at each degree of freedom in the discrete domain. These diagonal matrices can be easily constructed by taking the discrete values of b_x and b_y from the previous Picard iterate. The approximation to X can thus be regarded as a convection-diffusion operator perturbed by a scaled mass matrix. This is similar in structure to a time-dependent convection-diffusion operator, and preconditioning methods for F, such as multigrid, will extend well to approximate the action of X^{-1}.

3.2. An Approximation for Y. Now consider the discrete operator Y. We will proceed by considering relationships among commutators for the continuous differential operators corresponding to our discrete operators. This strategy was presented for the Navier-Stokes equations [7], where an approximation to the Schur complement $BF^{-1}B^t$ is needed. It was observed that the divergence and convection-diffusion operators approximately commute, i.e.

$$\nabla \cdot \mathcal{F} \approx \mathcal{F}_p \nabla \cdot, \tag{3.19}$$

where $\mathcal{F}_p$ is a convection-diffusion operator defined on the pressure space, referred to as the pressure convection-diffusion operator. This approximation holds when the convection direction $\vec{a}$ is smooth. A discrete version of (3.19) is given by

$$Q_{\mathbf{p}}^{-1} B Q_{\mathbf{u}}^{-1} F \approx Q_{\mathbf{p}}^{-1} F_{\mathbf{p}} Q_{\mathbf{p}}^{-1} B, \tag{3.20}$$

where $Q_{\mathbf{p}}$ is the pressure mass matrix and $F_{\mathbf{p}}$ is the discrete analogue of $\mathcal{F}_p$. We will discuss how to construct $F_{\mathbf{p}}$ later. Through algebraic manipulation of the approximation (3.20), the Schur complement approximation

$$BF^{-1}B^t \approx Q_{\mathbf{p}} F_{\mathbf{p}}^{-1} (B Q_{\mathbf{u}}^{-1} B^t) \tag{3.21}$$

is obtained, where the operator $BQ_{\mathbf{u}}^{-1}B^t$ can be treated as a discrete Laplacian operator [8]. This approximation is desirable because it circumvents the nested nature of the exact Schur complement so that the action of its inverse entails only matrix multiplications and solves with simple operators.

We could make a similar assumption about a commutator on the operator $\mathcal{K}$,

$$\nabla \cdot \mathcal{K} \approx \mathcal{K}_p \nabla \cdot \tag{3.22}$$

where $\mathcal{K}_p$ is an analogue to $\mathcal{K}$ on the pressure space. Unlike $\mathcal{F}$, which is a diagonal operator, $\mathcal{K}$ is a coupled 2D operator. For this reason, a direct analogue to $\mathcal{K}$ on the pressure space does not exist in general. In fact an operator $\mathcal{K}_p$ exists such that (3.22) holds only when $SR_m \vec{b} \approx \vec{0}$. Thus, however $\mathcal{K}_p$ is defined, the error associated with (3.22) must be at least proportional to $SR_m ||\vec{b}||$. Based on this observation, we propose a continuous Schur complement operator of the form

$$\nabla \cdot (\mathcal{F} + \mathcal{K}) \approx (\mathcal{F}_p + \alpha \mathcal{K}_p) \nabla \cdot, \tag{3.23}$$

where the scalar parameter α can be regarded as a relaxation parameter; that is, when (3.22) does not hold, α can be taken to be small to "relax" the error associated with this approximation. Discretizing this relation, we obtain

$$Q_{\mathbf{p}}^{-1} B Q_{\mathbf{u}}^{-1} X \approx Q_{\mathbf{p}}^{-1} (F_{\mathbf{p}} + \alpha K_{\mathbf{p}}) Q_{\mathbf{p}}^{-1} B, \tag{3.24}$$

where $K_{\mathbf{p}}$ is the discrete analogue to $\mathcal{K}_p$. Note that if $F_{\mathbf{p}} + \alpha K_{\mathbf{p}}$ is dominated by $F_{\mathbf{p}}$ then this operator commutes with the divergence operator with little error. Assuming the norms of $F_{\mathbf{p}}$ and $K_{\mathbf{p}}$ to be approximately the same as those of F and K, this condition holds when

$$\alpha H^2 h^2 ||\vec{b}||^2 \ll 1 + Rh||\vec{a}||, \tag{3.25}$$

where $H = \sqrt{SRR_m}$ is the Hartmann number. From this, we can see that α must be small when $Hh||\vec{b}||$ is relatively large and α can be large if $Rh||\vec{a}||$ is relatively large. Furthermore, for any problem parameters, α can be large if h is made small enough. Hence, adequate mesh refinement can allow us to take $\alpha = 1$.

Through straightforward algebraic manipulation of (3.24), we obtain

$$Y = -BX^{-1}B^t \approx -Q_{\mathbf{p}}(F_{\mathbf{p}} + \alpha K_{\mathbf{p}})^{-1} B Q_{\mathbf{u}}^{-1} B^t. \tag{3.26}$$

We define $F_{\mathbf{p}}$ using a strategy introduced to develop the Least-Squares Commutator (LSC) Schur approximation to the Schur complement for the Navier-Stokes equations [9]. That is, $F_{\mathbf{p}} Q_{\mathbf{p}}^{-1}$ is computed one row at a time, where each row is obtained by solving the least-squares problem

$$\min ||[BQ_{\mathbf{u}}^{-1}F]_{j*} - [F_{\mathbf{p}} Q_{\mathbf{p}}^{-1}]_{j*} B||_{Q_{\mathbf{u}}^{-1}}, \tag{3.27}$$

so that (3.20) is enforced in a least-squares sense row by row. This results in the definition

$$F_{\mathbf{p}} := BQ_{\mathbf{u}}^{-1} F Q_{\mathbf{u}}^{-1} B^t (BQ_{\mathbf{u}}^{-1} B^t)^{-1} Q_{\mathbf{p}}. \tag{3.28}$$

Applying the same strategy for $K_{\mathbf{p}}$ to enforce the discrete version of (3.22), we obtain

$$K_{\mathbf{p}} := BQ_{\mathbf{u}}^{-1} K Q_{\mathbf{u}}^{-1} B^t (BQ_{\mathbf{u}}^{-1} B^t)^{-1} Q_{\mathbf{p}}. \tag{3.29}$$

Substituting these definitions into (3.26) gives the approximation

$$Y \approx \hat{Y}_\alpha := -BQ_{\mathbf{u}}^{-1} B^t [BQ_{\mathbf{u}}^{-1}(F + \alpha K) Q_{\mathbf{u}}^{-1} B^t]^{-1} BQ_{\mathbf{u}}^{-1} B^t. \tag{3.30}$$

In practice, $Q_{\mathbf{u}}$ can be replaced by $diag(Q_{\mathbf{u}})$ for ease of computation [25]. Then the action of $\hat{Y}^{-1}$ requires two solves with $B(diag(Q_{\mathbf{u}}))^{-1}B^t$ which is a sparse (scaled) Laplacian and is easy to handle with multigrid. Observe that when $\alpha = 1$, this approximation corresponds to applying the LSC methodology directly to the operator $X = F + K$. Thus, the choice of $\alpha = 1$ is an intuitive choice but does not correct for the commutation error on K. We will return to the preconditioner resulting from the choice $\alpha = 1$ in Section 5.

In an effort to develop intuition for a good choice of α, we consider the impact of the approximation when it is applied to a single component of a Fourier series, i.e. where p is of the form

$$p = e^{i\vec{\omega}\cdot\vec{x}}, \tag{3.31}$$

corresponding to the vector frequency $\vec{\omega}$. We would like to choose α so that the effect of the exact Schur complement Y on p is comparable to the effect of the Schur complement approximation $\hat{Y}_\alpha$. Translating this condition to the continuous setting, we require

$$\nabla \cdot (\mathcal{F} + \mathcal{K})^{-1} \nabla p \approx \Delta_p [\nabla \cdot (\mathcal{F} + \alpha\mathcal{K})\nabla]^{-1} \Delta_p p. \tag{3.32}$$

This gives

$$\begin{aligned}\alpha \approx &||\vec{\omega}||^4 \frac{||\vec{\omega}||^4 + H^2(\vec{b}\cdot\vec{\omega})^2 + R^2(\vec{a}\cdot\vec{\omega})^2}{[||\vec{\omega}||^4 + H^2(\vec{b}\cdot\vec{\omega})^2]^2 + R^2(\vec{a}\cdot\vec{\omega})^2||\vec{\omega}||^4} \\ &+ i\frac{RH^2(\vec{a}\cdot\vec{\omega})(\vec{b}\cdot\vec{\omega})^2||\vec{\omega}||^2}{[||\vec{\omega}||^4 + H^2(\vec{b}\cdot\vec{\omega})^2]^2 + R^2(\vec{a}\cdot\vec{\omega})^2||\vec{\omega}||^4}.\end{aligned} \tag{3.33}$$

If we restrict α to be a real constant, we should choose $\alpha \approx \Re(\alpha(\vec{\omega}))$, i.e

$$\alpha = ||\vec{\omega}||^4 \frac{||\vec{\omega}||^4 + H^2(\vec{b}\cdot\vec{\omega})^2 + R^2(\vec{a}\cdot\vec{\omega})^2}{[||\vec{\omega}||^4 + H^2(\vec{b}\cdot\vec{\omega})^2]^2 + R^2(\vec{a}\cdot\vec{\omega})^2||\vec{\omega}||^4} \tag{3.34}$$

for a particular $\vec{\omega}$.

In general, our experiments have shown that the value of α is insensitive to the direction chosen for the Fourier mode $\vec{\omega}$ as long as the magnitude of $\vec{\omega}$ is chosen properly. We take $\vec{\omega}$ to be in the direction of $\vec{a}$, as $\vec{a}$ can be considered an approximation to $\vec{u}$, the direction of fluid flow. With this choice, the expression (3.34) reduces to

$$\alpha \approx \frac{||\vec{\omega}||^4 + H^2||\vec{b}||^2||\vec{\omega}||^2\cos^2(\theta) + R^2||\vec{a}||^2||\vec{\omega}||^2}{[||\vec{\omega}||^2 + H^2||\vec{b}||^2\cos^2(\theta)]^2 + R^2||\vec{a}||^2||\vec{\omega}||^2} \tag{3.35}$$

where θ is the angle between $\vec{a}$ and $\vec{b}$, and this approximation is entirely determined by the problem parameters and $||\vec{\omega}||$. The discretization of the problem should set a bound on the magnitude of any Fourier mode resolved by the mesh. That is, the most oscillatory Fourier mode should oscillate with period proportional to the mesh size on the pressure space h_p. Thus, we propose setting $||\vec{\omega}|| = h_p^{-1}$, yielding

$$\alpha \approx \alpha_* := \frac{1 + H^2h_p^2||\vec{b}||^2\cos^2(\theta) + R^2h_p^2||\vec{a}||^2}{[1 + H^2h_p^2||\vec{b}||^2\cos^2(\theta)]^2 + R^2h_p^2||\vec{a}||^2}. \tag{3.36}$$

The value of α_* can be computed from the mesh Hartmann number Hh and the mesh Reynolds number Rh, both of which are readily available, and the quantities $||\vec{a}||$, $||\vec{b}||$, and $\cos(\theta)$. In this study, we use the average values of the latter quanities over the domain from the previous iterate in the nonlinear iteration. As defined, α_* follows the trends implied by (3.25); that is, α_* is small if $H^2h^2||\vec{b}||^2$ dominates $R^2h^2||\vec{a}||^2$ and α^* is close to 1 if $R^2h^2||\vec{a}||^2$ dominates $H^2h^2||\vec{b}||^2$. Furthermore, as h is refined, α_* approaches 1.

With approximations to both X and Y, we can now write the form of our preconditioner for the linear systems arising from a Picard iteration

$$\mathcal{P}_{P,\alpha} = \begin{pmatrix} \hat{A} & -Z^t & 0 \\ 0 & \hat{X} & B^t \\ 0 & 0 & \hat{Y}_\alpha \end{pmatrix}, \tag{3.37}$$

where $\hat{X}$ is an approximation to X as defined in (3.17) and $\hat{Y}_\alpha$ is an approximation to Y as defined in (3.30).

4. A Block Preconditioner for the Newton System. Now consider the block LU factorization for the (reordered) matrix arising from Newton's method,

$$\mathcal{A}_N = \begin{pmatrix} I & 0 & 0 \\ (Z+\tilde{Z})(A+\tilde{A})^{-1} & I & 0 \\ 0 & B\tilde{X}^{-1} & I \end{pmatrix} \begin{pmatrix} A+\tilde{A} & -Z^t & 0 \\ 0 & \tilde{X} & B^t \\ 0 & 0 & \tilde{Y} \end{pmatrix}, \tag{4.1}$$

where

$$\tilde{X} = F + \tilde{F} + (Z+\tilde{Z})(A+\tilde{A})^{-1}Z^t, \tag{4.2}$$

$$\tilde{Y} = -B\tilde{X}^{-1}B^t. \tag{4.3}$$

Although $\tilde{X}$ does not simplify in its continuous form to an operator that can be explicitly constructed, because the Picard matrix and its associated block LU decomposition can be regarded as an approximation of the Newton matrix, we regard the **u-B** coupling term $(Z+\tilde{Z})(A+\tilde{A})^{-1}Z^t$ as a modification of the analogous term for the Picard iteration $ZA^{-1}Z^t \approx K$. To illustrate this, consider the identity

$$(Z+\tilde{Z})(A+\tilde{A})^{-1}Z^t = ZA^{-1}Z^t + E, \tag{4.4}$$

where E is the perturbation defined to be

$$E := (\tilde{Z} - ZA^{-1}\tilde{A})(A+\tilde{A})^{-1}Z^t. \tag{4.5}$$

Considering the approximate norms of the components of E, we find that

$$||E|| = O\left(SR_m h||\vec{b}||\frac{||\nabla\times\vec{b}|| + R_m||\vec{a}||\ ||\vec{b}||}{1+R_m h||\vec{a}||}\right). \tag{4.6}$$

Because the magnitude of E is proportional to h, E can be effectively neglected if the mesh is refined enough. We develop an approximation to E for the cases when the mesh is not refined enough to neglect it. This approximation is important when R_m and $R_m h$ are both large.

Translating each discrete operator in (4.5) and simplifying using the relation (3.12) produces a discrete analogue $\mathcal{E}$ of E defined as

$$\mathcal{E}\vec{u} := S\left[\cdot\times(\nabla\times\vec{b}) - R_m\vec{b}\times(\vec{a}\times\cdot)\right]\left[R_m^{-1}\Delta + \nabla\times(\vec{a}\times\cdot)\right]^{-1}\nabla\times(\vec{u}\times\vec{b}). \tag{4.7}$$

Assuming that the vectors $\vec{a}$ and $\vec{b}$ are smooth allows us to approximate (4.7) by

$$\mathcal{E}\vec{u} \approx -SR_m\vec{b}\times(\vec{a}\cdot\nabla)_p(-R_m^{-1}\Delta + \vec{a}\cdot\nabla)_p^{-1}(\vec{u}\times\vec{b}), \tag{4.8}$$

where we use the subscript p here to denote that the operator functions on the scalar pressure space. To use (4.8) to construct an approximation to E we make the approximation

$$(\vec{a}\cdot\nabla)_p \approx (-R_m^{-1}\Delta + \vec{a}\cdot\nabla)_p \tag{4.9}$$

when R_m is large. This is the important case to consider because the **u-B** coupling is strongest when R_m is large, hence necessitating a good approximation to E. Based on this observation, we make the approximation

$$(\vec{a}\cdot\nabla)_p \approx (1-\gamma)(-R_m^{-1}\Delta + \vec{a}\cdot\nabla)_p, \tag{4.10}$$

where the new parameter γ is included to correct for any error resulting from assumption (4.9) and is assumed to be close to 0 when R_m is large. Given (4.10), we obtain the approximation

$$\mathcal{E} \approx (\gamma - 1) S R_m \vec{b} \times (\vec{u} \times \vec{b}) = (\gamma - 1)\mathcal{K}, \tag{4.11}$$

and E can be approximated by

$$E \approx (1 - \gamma) K. \tag{4.12}$$

Notice that when E is negligible, we should set $\gamma = 1$, but another value of γ may be needed if R_m is large. Using the approximation (4.12) and recalling that $-ZA^{-1}Z^t \approx K$, we have the approximation

$$\tilde{X} \approx \hat{X}_\gamma := F + \tilde{F} + \gamma K. \tag{4.13}$$

As we did for the Picard iteration, we can use $p = e^{i\vec{\omega}\cdot x}$ to gain some intuition about the choice of γ. Evaluating (4.10) with this choice of p, we find that

$$\gamma(\vec{\omega}) \approx \frac{||\vec{\omega}||^4}{||\vec{\omega}||^4 + R_m^2(\vec{a}\cdot\vec{\omega})^2} - i\frac{R_m(\vec{a}\cdot\vec{\omega})}{||\vec{\omega}||^4 + R_m^2(\vec{a}\cdot\vec{\omega})^2}. \tag{4.14}$$

Restricting γ to be real gives

$$\gamma \approx \frac{||\vec{\omega}||^4}{||\vec{\omega}||^4 + R_m^2(\vec{a}\cdot\vec{\omega})^2} \tag{4.15}$$

for some $\vec{\omega}$.

We can learn more about a good choice of $\vec{\omega}$ by considering the discretization of (4.9)

$$N_{\mathbf{p}} \approx (1 - \gamma)(A_{\mathbf{p}} + N_{\mathbf{p}}), \tag{4.16}$$

where $N_{\mathbf{p}}$ is a discretization of $(\vec{a}\cdot\nabla)_p$ and $A_{\mathbf{p}}$ is a discretization of $-R_m^{-1}\Delta_p$. This can be rewritten as

$$\gamma I \approx A_{\mathbf{p}}(A_{\mathbf{p}} + N_{\mathbf{p}})^{-1}. \tag{4.17}$$

From this we can obtain the order of magnitude of γ to be

$$\gamma = O\left(\frac{1}{1 + R_m h_p ||\vec{a}||}\right). \tag{4.18}$$

If we let

$$\vec{\omega} = \vec{a}\sqrt{\frac{R_m}{h_p||\vec{a}||}}, \tag{4.19}$$

then (4.15) satisfies (4.18). This choice of Fourier mode is reasonable from a physical perspective. It is chosen in the direction of flow $\vec{a}$ and for a given h_p, its magnitude is proportional to $\sqrt{R_m||\vec{a}||}$, the width of characteristic layers appearing for the convection-diffusion operator $(\vec{a}\cdot\nabla - R_m^{-1}\Delta)_p$. Furthermore, as the mesh is refined, higher frequency modes can be resolved, and $||\vec{\omega}||$ increases to reflect this. Hence, we use the approximation

$$\gamma \approx \gamma_* := \frac{1}{1 + R_m h_p ||\vec{a}||}, \tag{4.20}$$

which depends only on the mesh magnetic Reynolds number $R_m h_p$ and the quantity $||\vec{a}||$ which can be taken as the average over the domain.

Using $\hat{X}_\gamma$ as defined in (4.13) to approximate $\tilde{X}$ and $\hat{Y}_\alpha$ as defined in (3.30) to approximate $\tilde{Y}$, we have the preconditioner for the Newton system

$$\mathcal{P}_{N,\alpha,\gamma} := \begin{pmatrix} \widehat{A + \tilde{A}} & -Z^t & 0 \\ 0 & \hat{X}_\gamma & B^t \\ 0 & 0 & \hat{Y}_\alpha \end{pmatrix}. \tag{4.21}$$

We note that the choice of α should incorporate γ. It is easy to show that, according to the same arguments from Section 3.2, α should be defined as

$$\alpha \approx \alpha_*(\gamma) := \frac{1 + \gamma H^2 h_p^2 ||\vec{b}||^2 \cos^2(\theta) + R^2 h_p^2 ||\vec{a}||^2}{[1 + \gamma H^2 h_p^2 ||\vec{b}||^2 \cos^2(\theta)]^2 + R^2 h_p^2 ||\vec{a}||^2}. \tag{4.22}$$

5. Computational Results. In this section, we evaluate the performance of the preconditioners for solving a set of benchmark problems. Our implementation is in the Trilinos framework [16] using the Teko package to construct the block preconditioners and a GMRES Krylov solver from AztecOO [15]. For component solves, we use algebraic multigrid from the ML package [10], with incomplete factorization smoothers coming from IFPACK. Specifically, for solves on the velocity and magnetic spaces we use one V-cycle of AMG with two pre- and post-sweeps of a Schwarz domain decomposition smoother with ILU(0) on each subdomain and one level of overlap between subdomains. For solves on the pressure space we use one V-cycle of AMG with five sweeps of a Gauss-Seidel smoother. This AMG technology has been demonstrated to be algorithmically scalable for both an equal order stabilized finite element formulation of the full MHD system and as a component solve in physics based preconditioners [4, 23]. All problems were run on the Red Sky computer at Sandia National Laboratories. For all problems we use a stable Q_2-Q_1 finite element pair for the velocity and pressure and Q_2 elements for the magnetic field to match the velocity. We use uniform grids for every problem. The relative residual tolerance is 10^{-5} for the nonlinear iteration and 10^{-6} for the GMRES inner iteration. We apply GMRES without any restarting. When considering the performance of the preconditioners, reported linear iteration counts are averaged over all nonlinear iterations up to a maximum of twenty nonlinear iterations. We consider two two-dimensional test problems, a lid driven cavity in the presence of a magnetic field (adapted from the fluid problem in [8]) and the Hartmann flow problem (as detailed in [12]). To explore the robustness of our preconditioning strategies, we test their performance over a range of R and R_m on these problems. We set $S = 1$ and let the variation in R_m account for different degrees of fluid-magnetic coupling.

5.1. MHD Lid Driven Cavity. The lid driven cavity problem models the flow of a conducting fluid in the square cavity $[0, 1] \times [0, 1]$, driven by the motion of its lid from left to right with the magnetic field $(-1, 0)$ imposed parallel to the lid. No-flow conditions are imposed on the bottom, left, and right sides of the cavity, and the

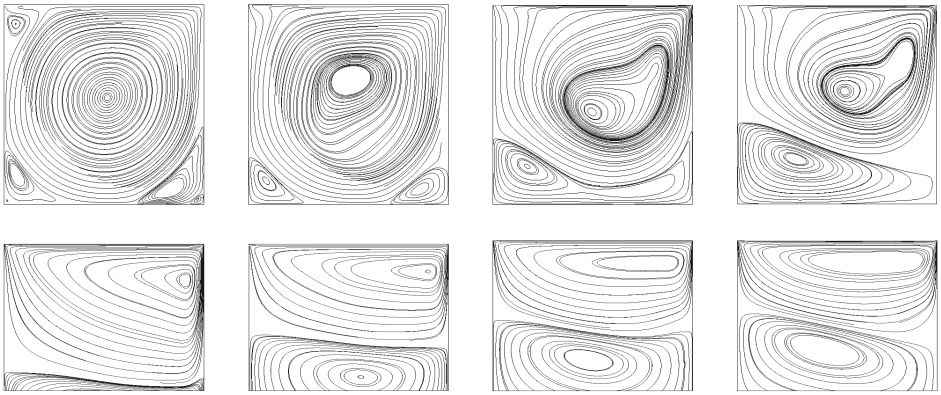

FIG. 5.1. *Streamlines for the MHD lid driven cavity problem with* $R = 5000$ *and* $R_m = 0, 0.1, 0.2, 0.3, 0.4, 5, 10, 20, 30$. *The four latter cases are zoomed in to* $[0,1] \times [0.8,1]$.

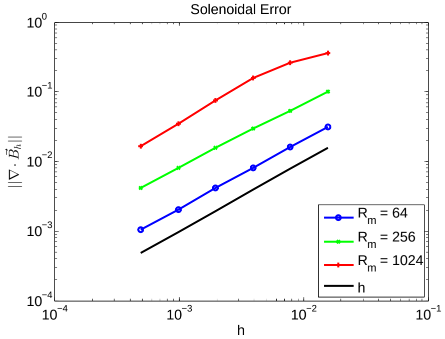

FIG. 5.2. *Magnitude of the divergence of* $\vec{B}_h$ *versus* h *for* $R = 64$ *and* $R_m = 64, 256, 1024$.

horizontal velocity of the lid is prescribed to be 1. The tangential component of the magnetic field is specified on the boundary $\partial\Omega$ as

$$\vec{B} \times \vec{n} = (-1, 0) \times \vec{n}. \tag{5.1}$$

For $R < R_c \approx 7800$, the cavity flow problem with no magnetic field has a stable solution dominated by one large eddy in the center of the cavity with smaller secondary eddies in the corners [24]. Imposing the magnetic field leads to weakening of the flow and, for stronger fields, merging of the secondary eddies. As the magnetic field is strengthened (that is, as R_m increases), the height of the primary eddy decreases as an effect of the increasing magnitude of the Lorentz force. Furthermore, other horizontal eddies of decreasing height develop, stretching from the left wall to the right wall and stacking on top of each other, with the number of horizontal eddies increasing with R_m. As a result of the braking effect of the Lorentz force, the flow in the lower part of the domain is almost stagnant for large R_m. Using solutions obtained from our code, this behavior is shown in Figure 5.1. Streamlines are plotted for the case where $R = 5000$ with increasing R_m. This set of problems is equivalent to those considered in [24], and our results are qualitatively very similar, with the same number and height of eddies appearing.

To demonstrate that the exact penalty formulation weakly enforces the solenoidal

	Picard				Newton			
R_m R	1	64	256	1024	1	64	256	1024
1	2	×	×	×	2	3	3	4
64	2	7	×	×	2	4	3	3
256	2	4	7^*	×	2	4	5	3
1024	2	3	4	11^*	2	3	6	16^*
4096	2	3	4	5	2	2	7	7^*

TABLE 5.1

Iterations required for convergence of the nonlinear iteration. '×' indicates no convergence within twenty iterations. '' indicates convergence required backtracking.*

	$\alpha = 1$				$\alpha = \alpha_*$			
R_m R	1	64	256	1024	1	64	256	1024
1	36	45	44	78	36	45	44	78
64	35	42	47	84	35	42	47	77
256	36	42	60	109	36	42	57	90
1024	44	44	89	193	44	43	77	142
4096	68	60	123	291	68	58	91	174

TABLE 5.2

Average GMRES iterations required for convergence with $\mathcal{P}_{P,\alpha}$ on the Picard linearization of the MHD lid driven cavity problem with $\alpha = 1$ and $\alpha = \alpha_*$.

condition, we have plotted the magnitude of the divergence of the computed magnetic field in Figure 5.2. Because we have found that $||\nabla \cdot \vec{B}_h||_{L^2(\Omega)}$ is independent of the fluid Reynolds number, we only show results for $R = 64$. It can be seen from the figure that this quantity is approximately proportional to $R_m h$. This is consistent with results obtained in the context of Marder cleaning for Maxwell's equations [18]. In this context, the penalty term $\frac{S}{R_m}(\nabla \cdot \vec{B}, \nabla \cdot \vec{C})$ is analogous to a pseudo-current term added to Maxwell's equations scaled by $\frac{S}{R_m}$. For time-dependent problems, the scaling of the pseudo-current term determines the rate at which the error in the solenoidal condition vanishes.

Table 5.1 shows the number of steps required by each nonlinear scheme on a uniform 512×512 element mesh, starting from a zero initial guess. As R_m increases, the fluid-magnetic coupling strengthens and the nonlinear problems become somewhat more difficult to solve. For entries in the table with asterisks, convergence was achieved using a backtracking strategy; see [21]. Newton's method was slightly more robust than Picard iteration. Although backtracking alone was not sufficient to make the Picard iteration fully robust for these examples, since our emphasis is on the linear solvers, we did not pursue this further.[1]

5.1.1. Picard Iteration. In this section, we consider the performance of the preconditioner $\mathcal{P}_{P,\alpha}$ on the MHD lid driven cavity problem for a range of parameters,

[1] This type of behavior is often sensitive to the initial guess and also to the properties of the globalization methods used; see [21] for other ways to enhance robustness. We also note that poor performance of the nonlinear solver is related to under-resolution of the mesh. For the problematic cases in Table 5.1, both the Picard iteration and Newton's method converge on finer meshes without use of globalization methods.

R_m R	1	64	256	1024
1	1.000	1.000	0.998	0.992
64	1.000	0.970	0.889	0.667
256	0.998	0.895	0.677	0.339
1024	0.996	0.777	0.407	0.123
4096	0.998	0.777	0.272	0.046

Table 5.3
Computed values of α_* for the second Picard iteration.

	$\alpha = 1, \gamma = 1$				$\alpha = \alpha_*, \gamma = 1$				$\alpha = \alpha_*, \gamma = \gamma_*$			
R_m R	1	64	256	1024	1	64	256	1024	1	64	256	1024
1	36	42	47	68	36	42	47	68	36	42	47	66
64	34	46	69	171	34	47	69	163	34	47	66	117
256	37	50	99	299	37	50	94	228	37	49	88	164
1024	44	50	169	584	44	50	133	288	44	50	121	249
4096	68	56	208	603	68	54	142	419	68	54	137	364

Table 5.4
Average GMRES iterations required for convergence with $\mathcal{P}_{P,\alpha,\gamma}$ on the Newton linearization of the MHD lid driven cavity problem.

using a 512 × 512 element mesh. To study the impact of the parameter α on the preconditioner, we compare the performance of $\mathcal{P}_{P,\alpha}$ with $\alpha = \alpha_*$ as defined in (3.36) and $\alpha = 1$, corresponding to the LSC preconditioner applied directly to X. GMRES iteration counts with these two preconditioners for various choices of R and R_m are reported in Table 5.2. The values of α_* used at the second nonlinear step are reported in Table 5.3. (Note that for the first step, $\vec{a} = \vec{0}$, and therefore $\alpha_* = 1$ independent of R and R_m.) From Table 5.2, it is clear that the automatically computed $\alpha = \alpha_*$ leads to improved performance compared to LSC preconditioning. The differences are minimal for small R and R_m (the iteration counts are identical for $R = 1$ and $R_m = 1$), but they become substantial for larger R and R_m. For example, for $R = 4096$, $R_m = 1024$, the counts for $\alpha = 1$ and $\alpha = \alpha_*$ are 291 and 174, respectively, a 40% reduction. Table 5.3 also shows that α_* is close to 1 for small R and R_m. These results are consistent with the observation made in Section 3 that when the fluid-magnetic coupling is strong, the validity of the approximation for the commutator (3.22) is weaker; the "relaxed" variants (3.23)–(3.24) improve performance. We also found that an optimal choice of α determined by a brute-force search produced iteration counts essentially the same as for $\alpha = \alpha_*$. Thus, the choice of parameter given by (3.36) determines an essentially automated version of the preconditioner (3.24).

5.1.2. Newton's Method. Here, we consider the performance of the preconditioner $\mathcal{P}_{N,\alpha,\gamma}$ on the linear systems arising from Newton's method applied to the MHD lid driven cavity problem. We consider a 512 × 512 mesh with R between 1 and 4096 and R_m between 1 and 1024. We compare three preconditioners: the fully parameterized preconditioner $\mathcal{P}_{N,\alpha_*,\gamma_*}$ with γ_* from (4.20) and α_* from (4.22); the preconditioner $\mathcal{P}_{N,\alpha_*,1}$ parameterized only by α with α_* from (3.36); and the unparameterized preconditioner $\mathcal{P}_{N,1,1}$. $\mathcal{P}_{N,\alpha_*,1}$ corresponds to applying the strategy derived for the Picard iteration to the Jacobian system, as though the fluid-magnetic

R \ R_m	1	64	256	1024
1	0.999	0.939	0.794	0.491
64	0.999	0.940	0.796	0.493
256	0.999	0.940	0.796	0.495
1024	0.999	0.940	0.796	0.493
4096	0.999	0.940	0.796	0.493

Table 5.5
Computed values of γ_* at the second Newton step.

coupling E is negligible. GMRES iteration counts are reported in Table 5.4 and the values of γ_* are reported in Table 5.5.

These results show that the (doubly) parameterized preconditioner for the Jacobian systems nearly always exhibit enhanced performance, with the most significant improvements occurring in regimes where the spatial resolution is weakest (large R or R_m). The impact of the new parameter is comparable to that of α, and we emphasize that, as above, the enhanced performance is obtained using an automated strategy for choosing parameters.

5.1.3. Robustness with Respect to Mesh Refinement. In Sections 5.1.1 and 5.1.2, we explored the robustness of our preconditioning strategies with respect to the physical parameters R and R_m, on a fixed mesh. We are also interested in their performance as the mesh is refined. We investigate this on the two problems corresponding to $R = 256$, $R_m = 256$ and $R = 256$, $R_m = 1024$. Because the multigrid component solves depend on the parallel architecture used, we also use this as an opportunity to study the parallel scalability of the preconditioners. That is, the Schwarz-ILU smoother for the multigrid solves decomposes the domain into as many subdomains as there are processors. Hence, we want to make sure that increasing the number of processors does not cause the performance of the full preconditioners to degrade. Toward both of these ends, we increase the number of processors as we refine the mesh so that the number of unknowns per processor remains approximately constant. That is, we perform a weak parallel scaling study by considering 64×64, 128×128, 256×256, 512×512, 1024×1024, and 2048×2048 element discretizations on 1, 4, 16, 64, 256, and 1024 processors respectively, keeping the number of unknowns per processor at approximately $70,000$.

We report average iteration counts and computation times per nonlinear step for these experiments in Figure 5.3 for the Picard iteration and Figure 5.4 for Newton's method. Newton's method converges on all grids considered for $R = 256, R_m = 256$ but only on the three most refined grids for $R = 256, R_m = 1024$. For the Picard iteration, we compare the preconditioner $\mathcal{P}_{P,\alpha_*}$ with α_* from (3.36) with the unparameterized preconditioner $\mathcal{P}_{P,1}$. As a frame of reference for parallel scalability, we also compare these preconditioners to a pure domain decomposition preconditioner using a SuperLU [6] ILUTP factorization with a drop tolerance of 0.0001 and a zero pivot threshold of 0.01 (labeled DD in the scaling plots). Thresholding and partial pivoting are necessary here because of the zero block on the diagonal for the pressure space. From the plots, it can be seen that both $\mathcal{P}_{P,\alpha_*}$ and $\mathcal{P}_{P,1}$ are robust with respect to mesh refinement, with iteration counts that are nearly constant or decreasing as the number of unknowns grows. The parameterized preconditioner also performs uniformly as well or better than the unparameterized preconditioner, the

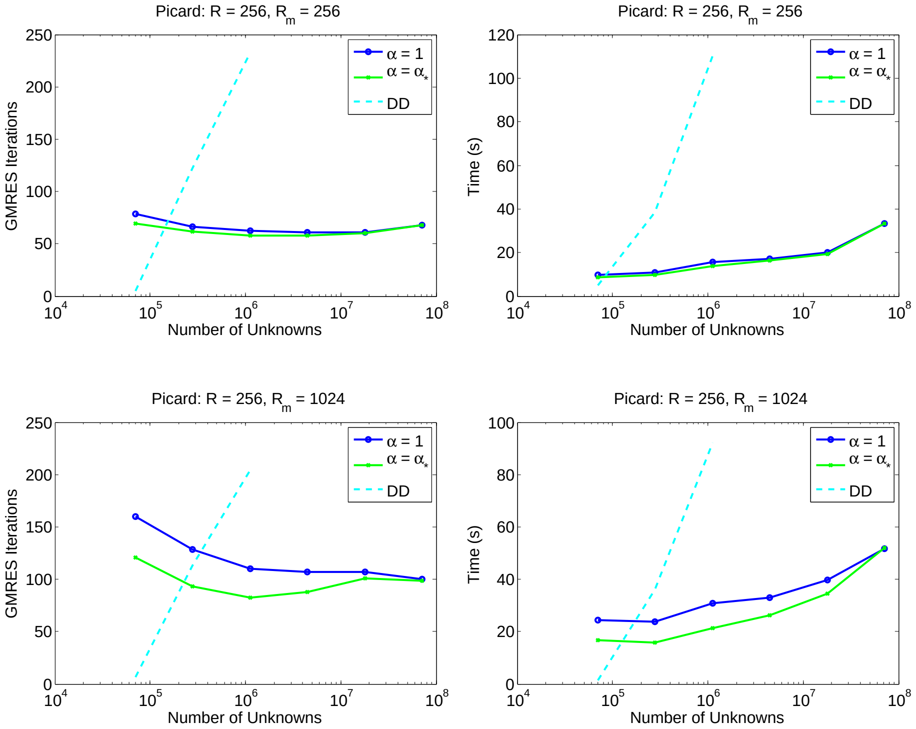

FIG. 5.3. *Mesh refinement results for the MHD lid driven cavity problem for the Picard iteration. Parameters are $R = 256, R_m = 256$ on the top and $R = 256, R_m = 1024$ on the bottom. Average GMRES iterations on the left and average linear solve time per nonlinear iteration on the right.*

benefit being more pronounced for problems with fewer unknowns or larger R_m. The iteration counts for the two preconditioners become similar as h is refined because α_* approaches 1 as h decreases. The domain decomposition preconditioner is not competitive and does not scale well. In fact, for all the tests run on more than 16 processors, GMRES runs out of memory before convergence with this preconditioner. While iteration counts tend to decrease, we see a slight increase in computation time as the mesh is refined. This appears to be due to increased communication costs in the component AMG solves as more processors are added, and we expect that performance can be enhanced by fine-tuning these solves.

For Newton's method, we compare $\mathcal{P}_{N,\alpha_*,\gamma_*}$, the fully parameterized preconditioner, with $\mathcal{P}_{N,\alpha_*,1}$, the preconditioner parameterized only for the matrix arising from Picard iteration and with $\mathcal{P}_{N,1,1}$, the unparameterized preconditioner. The plots for $R = 256, R_m = 1024$ contain only three data points because Newton's method diverges for this problem on the three coarsest meshes. We use γ_* from defined in (4.20). Iteration counts generally decrease for all three preconditioners as h is refined. Since both α_* and γ_* converge to 1 as h decreases, the performances of the three preconditioners become essentially the same as the problem size increases. In all but the least refined cases, the fully parameterized preconditioner outperforms the others, especially for problems with large R_m. The results then show the importance of the parameter γ in keeping iteration counts low.

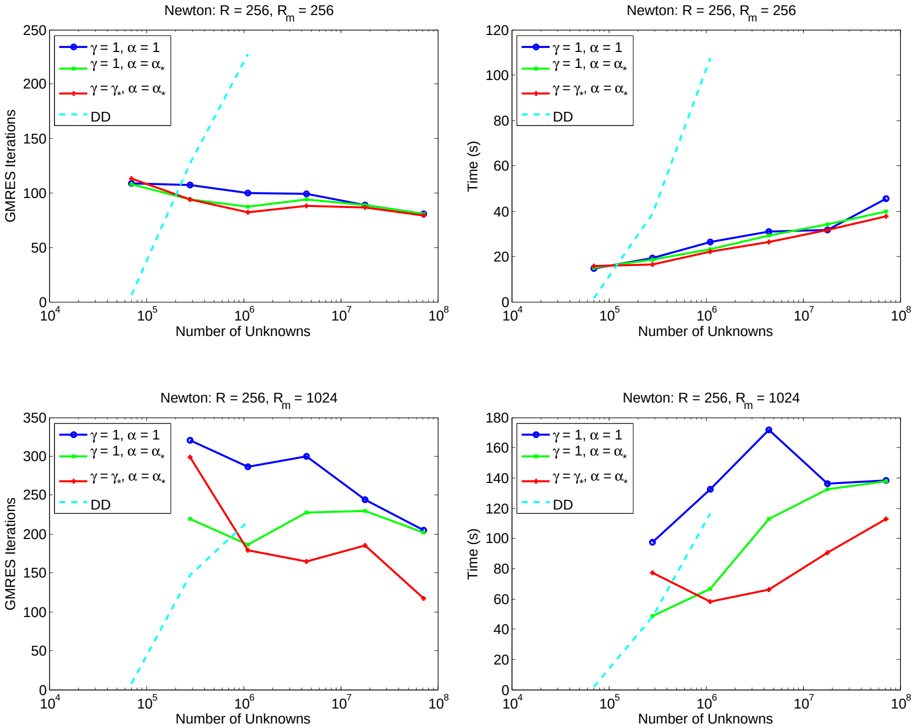

FIG. 5.4. *Mesh refinement results for the MHD lid driven cavity problem for Newton's method. Parameters are $R = 256, R_m = 256$ on the top and $R = 256, R_m = 1024$ on the bottom. Average GMRES iterations on the left and average linear solve time per nonlinear iteration on the right.*

5.2. Hartmann Flow. The Hartmann flow problem describes the flow of a conducting fluid through a channel in the presence of a transverse magnetic field. We consider the channel $[-1/2, 1/2]^2$ and the transverse field $(0, 1)$. With appropriate boundary conditions, this problem has the explicit analytic solution $\vec{u} = (u_x, 0), \vec{B} = (B_x, 1)$ with

$$u_x = \frac{GR(\cosh(H/2) - \cosh(Hy))}{2H\sinh(H/2)}, \tag{5.2a}$$

$$B_x = \frac{G(\sinh(Hy) - 2\sinh(H/2)y)}{2S\sinh(H/2)}, \tag{5.2b}$$

$$p = -Gx - SB_x^2/2, \tag{5.2c}$$

where G can be any scalar. Given R and R_m, we choose G so the maximum magnitude of $\vec{u}$ is normalized to 1. In our implementation, we prescribe the analytic value of $\vec{u}$ and $\vec{B} \times \vec{n} = (0, 1) \times \vec{n}$ on the boundary.

Because an analytic solution exists for this problem, we can compute the errors $||\vec{u}_{exact} - \vec{u}_h||_{L^2(\Omega)}$ and $||\vec{B}_{exact} - \vec{B}_h||_{L^2(\Omega)}$. These errors and the error in the solenoidal condition $||\nabla \cdot \vec{B}_h||_{L^2(\Omega)}$ are plotted in Figure 5.5. From this figure, it is clear that all three quantities are proportional to h^3. As with the MHD lid driven cavity problem, $||\nabla \cdot \vec{B}_h||_{L^2(\Omega)}$ is also related to R_m, depending to a lesser degree on R. Consequently, the errors in $\vec{u}$ and $\vec{B}$ depend similarly on R_m and R.

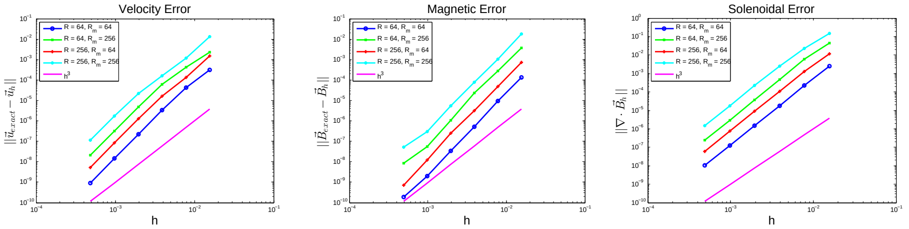

FIG. 5.5. *Magnitude of the error in $\vec{u}_h$ and $\vec{B}_h$ and the divergence of $\vec{B}_h$ versus h.*

	$\alpha = 1$			$\alpha = \alpha_*$		
R \ R_m	1	64	256	1	64	256
1	38	41	40	38	41	40
64	31	37	40	31	37	40
256	29	35	50	29	35	49
1024	36	37	61	36	37	61
4096	32	79	81	32	79	81

TABLE 5.6
Average GMRES iterations required for convergence with $\mathcal{P}_{P,\alpha}$ on the Picard linearization of the Hartmann flow problem.

We have tested the preconditioner $\mathcal{P}_{P,\alpha}$ on the linear systems resulting from the Picard linearization of the Hartmann flow problem on a fixed 512×512 element mesh. We found that nonlinear iterations did not converge for $R_m > 256$ on this mesh, so we do not consider $R_m = 1024$ for this problem. In Table 5.6, preconditioned GMRES iteration counts are compared for the choices $\alpha = 1$ and $\alpha = \alpha_*$ as defined in (3.36). In this case, the difference between the two preconditioners is marginal for all parameters.

We have also considered the performance of the preconditioner $\mathcal{P}_{N,\alpha,\gamma}$ on the linear systems resulting from Newton's method on a 512×512 element mesh. We compare $\mathcal{P}_{N,1,1}, \mathcal{P}_{N,\alpha_*,1}$, and $\mathcal{P}_{N,\alpha_*,\gamma_*}$ in Table 5.7. Here, we see the parameter γ having a much more profound effect on iteration counts than the parameter α. The choice of $\gamma = \gamma_*$ over $\gamma = 1$ leads to large improvements for moderate values of R_m. For instance, we can see a 42% improvement in iteration count for $R = 4099, R_m = 256$.

We note here that the performance of the preconditioners with respect to mesh refinement for the Hartmann flow problem follows the same trends as for the MHD lid driven cavity problem. For this reason, we do not include a figure analogous to Figures 5.3 and 5.4.

6. Conclusion.

We have presented a family of block preconditioners for the linear systems arising from both Picard and Newton linearizations of the exact penalty formulation of the stationary MHD equations. For the Picard linearization, we chose an ordering $(\mathbf{B}, \mathbf{u}, \mathbf{p})$ that leads to a simple Schur complement on the velocity space that can be approximated by simplifying the continuous operator corresponding to it. This results in approximating the velocity Schur complement by a perturbed convection-diffusion operator. By modifying the Least-Squares Commutator preconditioner for the Navier-Stokes equations, we have developed an approximation to the nested Schur complement on the pressure space. The preconditioner resulting from

	$\alpha = 1, \gamma = 1$			$\alpha = \alpha_*, \gamma = 1$			$\alpha = \alpha_*, \gamma = \gamma_*$		
R_m R	1	64	256	1	64	256	1	64	256
1	39	43	137	39	43	137	39	43	104
64	31	50	144	31	50	144	31	48	99
256	29	49	124	29	49	123	29	46	81
1024	36	55	141	36	55	141	36	52	89
4096	32	158	206	32	158	205	32	113	118

Table 5.7

Average GMRES iterations required for convergence with $\mathcal{P}_{P,\alpha,\gamma}$ on the Newton linearization of the Hartmann flow problem.

direct application of the LSC methodology to the perturbed convection-diffusion operator can be improved by "relaxing" assumptions on the existence of small commutators. Using the parameterized assumption (4.10), similar strategies can be applied to Newton's method.

We have presented an automated method for choosing the two parameters α and γ based on arguments from Fourier analysis. This method relies only on data which is readily available in a nonlinear iteration. Our numerical studies on the Picard linearization have demonstrated that the automated choice of α leads to significant improvement over $\alpha = 1$. For Newton's method, we have shown that both α and γ are important for decreasing iteration counts. The preconditioners are robust with respect to mesh refinement, their performances improving as the mesh is refined. In terms of parallel scalability, these preconditioners compare very well against a preconditioner based on domain decomposition.

REFERENCES

[1] L. Chacón, *An optimal, parallel, fully implicit Newton-Krylov solver for three-dimensional viscoresistive magnetohydrodynamics*, Physics of Plasmas, 15 (2008), p. 056103.

[2] R. Codina and N. Hernández, *Stabilized finite element approximation of the stationary magneto-hydrodynamics equations*, Computational Mechanics, 38 (2006), pp. 344–355.

[3] M. Costabel and M. Dauge, *Weighted regularization of Maxwell equations in polyhedral domains*, Numerische Mathematik, 93 (2002), pp. 239–277.

[4] E. Cyr, J. Shadid, R. Tuminaro, R. Pawlowski, and L. Chacón, *A new approximate block factorization preconditioner for 2D incompressible (reduced) resistive MHD*, SIAM Journal on Scientific Computing, 35 (2013), pp. B701–B730.

[5] A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T. Schnitzer, and M. Wesenberg, *Hyperbolic divergence cleaning for the MHD equations*, Journal of Computational Physics, 175 (2002), pp. 645–673.

[6] J. Demmel, S. Eisenstat, J. Gilbert, X. Li, and J. Liu, *A supernodal approach to sparse pivoting*, SIAM Journal on Matrix Analysis and Applications, 2 (1999), pp. 720–755.

[7] H. Elman, V. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro, *A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible Navier-Stokes equations*, Journal of Computational Physics, 227 (2008), pp. 1790–1808.

[8] H. Elman, D. Silvester, and A. Wathen, *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*, Oxford University Press, 2005.

[9] H. Elman and R. Tuminaro, *Boundary conditions in approximate commutator preconditioners for the Navier-Stokes equations*, Electronic Transactions on Numerical Analysis, 35 (2009), pp. 1068–9613.

[10] M. Gee, C. Siefert, J. Hu, R. Tuminaro, and M. Sala, *ML 5.0 smoother aggregration user's guide*, Tech. Rep. SAND2006-2649, Sandia National Laboratories, 2006.

[11] J.-F. Gerbeau, *A stabilized finite element method for the incompressible magnetohydrodynamics equations*, Numerische Mathematik, 87 (2000), pp. 83–111.

[12] J.-F. Gerbeau, C. L. Bris, and T. Lelièvre, *Mathematical Methods for the Magnetohydrodynamics of Liquid Metals*, Oxford University Press, 2006.

[13] J. Goedbloed and S. Poedts, *Principles of Magnetohydrodynamics with Applications to Laboratory and Astrophysical Plasmas*, Cambridge University Press, 2004.

[14] M. Gunzburger, A. Meir, and J. Peterson, *On the existence, uniqueness, and finite element approximation of solutions of the equations of stationary, incompressible magnetohydrodynamics*, Mathematics of Computation, 56 (1991), pp. 523–563.

[15] M. Heroux, *AztecOO user guide*, Tech. Rep. SAND2004-3796, Sandia National Laboratories, 2004.

[16] M. Heroux, R. Bartlett, V. Howle, R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, A. Williams, and K. Stanley, *An overview of the Trilinos project*, ACM Transactions on Mathematical Software, 31 (2005), pp. 397–423.

[17] P. Lin, J. Shadid, R. Tuminaro, M. Sala, G. Hennigan, and R. Pawlowski, *A parallel fully-coupled algebraic multilevel preconditioner applied to multiphysics PDE applications: drift-diffusion, flow/transport/reaction, resistive MHD*, International Journal for Numerical Methods in Fluids, 64 (2010), pp. 1148–1179.

[18] B. Marder, *A method for incorporating Gauss' law into electromagnetic PIC codes*, Journal of Computational Physics, 68 (1987), pp. 48–55.

[19] R. Moreau, *Magnetohydrodynamics*, Kluwer Academic Publishers, 1990.

[20] M. Murphy, G. Golub, and A. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM Journal on Scientific Computing, 21 (2000), pp. 1969–1972.

[21] R. Pawlowski, J. Shadid, J. Simonis, and H. Walker, *Globalization techniques for Newton-Krylov methods and applications to the fully-coupled solution of the Navier-Stokes equations*, SIAM Review, 48 (2006), pp. 700–721.

[22] D. Schötzau, *Mixed finite element methods for stationary incompressible magnetohydrodynamics*, Numerische Mathematik, 96 (2004), pp. 771–800.

[23] 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.

[24] V. Shatrov, G. Mutschke, and G. Gerberth, *Three-dimensional linear stability analysis of lid-driven magnetohydrodynamic cavity flow*, Physics of Fluids, 15 (2003), pp. 2141–2151.

[25] A. Wathen, *Realistic eigenvalue bounds for the Galerkin mass matrix*, IMA Journal on Numerical Analysis, 7 (1987), pp. 449–457.