SIAM J. Sci. Comput.
Vol. 42, No. 1, pp. B57–B79

A SCALABLE APPROXIMATE INVERSE BLOCK PRECONDITIONER FOR AN INCOMPRESSIBLE MAGNETOHYDRODYNAMICS MODEL PROBLEM[*]

MICHAEL WATHEN[†] AND CHEN GREIF[‡]

Abstract. We introduce a new approximate inverse preconditioner for a mixed finite element discretization of an incompressible magnetohydrodynamics model problem. The derivation exploits the nullity of the discrete curl-curl operator in the Maxwell subproblem. We show that the inverse of the coefficient matrix contains zero blocks and use discretization considerations to obtain a practical preconditioner based on further sparsification. We demonstrate the viability of our approach with a set of numerical experiments.

Key words. incompressible magnetohydrodynamics, saddle-point linear systems, null space, preconditioners, approximate inverse, Krylov subspace methods

AMS subject classifications. 65F08, 65F10, 65F15, 65F50, 65N22

DOI. 10.1137/19M1255409

1. Introduction.

Incompressible magnetohydrodynamics (MHD) describes the flow of an electrically conductive fluid in the presence of a magnetic field [2, 7, 11, 21]. Given a sufficiently smooth domain Ω, consider the steady-state incompressible MHD model [11, Chapter 2]:

$$-\nu\,\Delta \boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p - \kappa\,(\nabla\times\boldsymbol{b})\times\boldsymbol{b} = \boldsymbol{f} \qquad \text{in } \Omega, \tag{1.1a}$$

$$\nabla\cdot\boldsymbol{u} = 0 \qquad \text{in } \Omega, \tag{1.1b}$$

$$\kappa\nu_m\,\nabla\times(\nabla\times\boldsymbol{b}) + \nabla r - \kappa\,\nabla\times(\boldsymbol{u}\times\boldsymbol{b}) = \boldsymbol{g} \qquad \text{in } \Omega, \tag{1.1c}$$

$$\nabla\cdot\boldsymbol{b} = 0 \qquad \text{in } \Omega. \tag{1.1d}$$

Here $\boldsymbol{u}$ is the velocity, p is the hydrodynamic pressure, and $\boldsymbol{b}$ is a magnetic field and the Lagrange multiplier associated with the divergence constraint on the magnetic field is denoted by r. The vector functions $\boldsymbol{f}$ and $\boldsymbol{g}$ represent external forcing terms. The three dimensionless parameters that characterize this model are the hydrodynamic viscosity ν, the magnetic viscosity ν_m, and the coupling number κ.

To complete the model, we consider the inhomogeneous Dirichlet boundary conditions:

$$\boldsymbol{u} = \boldsymbol{u}_0 \qquad \text{on } \partial\Omega, \tag{1.2a}$$

$$\boldsymbol{n}\times\boldsymbol{b} = \boldsymbol{n}\times\boldsymbol{b}_0 \qquad \text{on } \partial\Omega, \tag{1.2b}$$

$$r = r_0 \qquad \text{on } \partial\Omega \tag{1.2c}$$

[*] Submitted to the journal's Computational Methods in Science and Engineering section April 10, 2019; accepted for publication (in revised form) October 16, 2019; published electronically January 7, 2020.

https://doi.org/10.1137/19M1255409

Funding: The work of the second author was partially supported by an NSERC Discovery Grant.

[†] STFC Rutherford Appleton Laboratory, Didcot, Oxfordshire, UK, OX11 0QX (michael.wathen@stfc.ac.uk).

[‡] Department of Computer Science, The University of British Columbia, Vancouver, B.C., V6T 1Z4, Canada (greif@cs.ubc.ca).

with $\boldsymbol{n}$ being the unit outward normal on $\partial\Omega$ and $\boldsymbol{u}_0$, $\boldsymbol{b}_0$, and r_0 being the functions defined on the boundary.

We will consider two nonlinear iteration schemes—Picard iteration and Newton's method. Using the notation

$$\begin{aligned}\boldsymbol{u}^{k+1} &= \boldsymbol{u}^k + \delta\boldsymbol{u}, \quad & p^{k+1} &= p^k + \delta p,\\ \boldsymbol{b}^{k+1} &= \boldsymbol{b}^k + \delta\boldsymbol{b}, \quad & r^{k+1} &= r^k + \delta r,\end{aligned}$$

where $*^{k+1}$ is the $k+1^{\mathrm{st}}$ iteration of either the Picard or Newton's iteration, the linearized system can be written as

$$\begin{aligned}&-\nu\Delta\delta\boldsymbol{u} + (\boldsymbol{u}^k\cdot\nabla)\delta\boldsymbol{u} + \alpha(\delta\boldsymbol{u}\cdot\nabla)\boldsymbol{u}^k\\ &\qquad + \nabla\delta p - \kappa(\nabla\times\delta\boldsymbol{b})\times\boldsymbol{b}^k - \alpha\kappa(\nabla\times\boldsymbol{b}^k)\times\delta\boldsymbol{b} = \boldsymbol{r}_u,\\ &\qquad\qquad \nabla\cdot\delta\boldsymbol{u} = r_p,\\ &\kappa\nu_m\nabla\times(\nabla\times\delta\boldsymbol{b}) + \nabla r - \kappa\nabla\times(\delta\boldsymbol{u}\times\boldsymbol{b}^k) - \alpha\kappa\nabla\times(\boldsymbol{u}^k\times\delta\boldsymbol{b}) = \boldsymbol{r}_b,\\ &\qquad\qquad \nabla\cdot\delta\boldsymbol{b} = r_r,\end{aligned}\tag{1.3}$$

where

$$\begin{aligned}\boldsymbol{r}_u &= \boldsymbol{f} - \left[-\nu\,\Delta\boldsymbol{u}^k + (\boldsymbol{u}^k\cdot\nabla)\boldsymbol{u}^k + \nabla p^k - \kappa\,(\nabla\times\boldsymbol{b}^k)\times\boldsymbol{b}^k\right],\\ r_p &= -\nabla\cdot\boldsymbol{u}^k,\\ \boldsymbol{r}_b &= \boldsymbol{g} - \left[\kappa\nu_m\,\nabla\times(\nabla\times\boldsymbol{b}^k) + \nabla r - \kappa\,\nabla\times(\boldsymbol{u}^k\times\boldsymbol{b}^k)\right],\\ r_r &= -\nabla\cdot\boldsymbol{b}^k,\end{aligned}$$

and

$$\alpha = \begin{cases} 0 & \text{for Picard,}\\ 1 & \text{for Newton.}\end{cases}\tag{1.4}$$

In subsequent sections, additional details on the schemes and the associated linear systems will be provided. Those systems have an interesting block structure, and the properties of their corresponding discrete operators may in some cases be exploited to obtain effective sparse approximations, based on Schur complements or null spaces.

In recent years, interest in the development of block preconditioning methods for the MHD model has increased; see [1, 6, 16, 24, 25, 29, 30, 31]. One of the main questions explored in those papers is how to approximate the Schur complements that arise in the course of forming the block matrices associated with the iterative solution procedure. While good scalable iterations with respect to mesh refinement are obtained, so far fully scalable iterations for large three-dimensional (3D) problems with high coupling numbers have not been fully developed. In this paper, we develop a new formula for the inverse of the coefficient matrix and use it to introduce a scalable block approximate inverse preconditioner. In the course of our derivation, we build upon and utilize the approximations and techniques used in [24, 25, 31] for block triangular preconditioners.

The remainder of the paper is structured as follows. In section 2 we introduce the finite element discretization that we use, discuss a few properties of the discrete operators, and present the linear system that arises from the discretization. In section 3, we derive a new formula for the inverse and show that the (exact) inverse has a few zero blocks. In section 4, we approximate Schur complements that appear in the formula by sparse operators and derive a new approximate inverse formula. In section 5, we use the approximate Schur complement to form a block triangular preconditioner. Section 6 presents numerical experiments that demonstrate the viability and effectiveness of this preconditioning approach. Finally, we offer some brief concluding remarks in section 7.

2. Discretization.

We consider a finite element discretization of the MHD model (1.1)–(1.2), where the hydrodynamic unknowns ($\boldsymbol{u}$ and p) are discretized with any stable mixed finite elements and the magnetic and multiplier unknowns are discretized through a mixed edge and nodal element pair. Using the same formulation as [31], we use Taylor–Hood elements [28] for $(\boldsymbol{u}, p)$ and the lowest order Nédélec [23] pair for $(\boldsymbol{b}, r)$. This choice of mixed finite elements avoids the need to stabilize the fluid/pressure variables and provides conforming $H(\text{curl})$ elements for the magnetic variables.

2.1. Finite element spaces.

The mixed finite element approximation used for (1.1)–(1.2) was introduced and analyzed in [27]. Thus, we find the weak solution $(\boldsymbol{u}, p, \boldsymbol{b}, r)$ in the standard Sobolev spaces

$$\begin{aligned}
\boldsymbol{u} \in \boldsymbol{V} &= \left\{\boldsymbol{v} \in H^1(\Omega)^d \,:\, \boldsymbol{v} = \boldsymbol{u}_0 \text{ on } \partial\Omega\right\},\\
p \in Q &= \{\, q \in L^2(\Omega) \,:\, (q,1)_\Omega = 0\},\\
\boldsymbol{b} \in \boldsymbol{C} &= \left\{\boldsymbol{c} \in L^2(\Omega)^d \,:\, \nabla \times \boldsymbol{c} \in L^2(\Omega)^{\bar{d}},\ \boldsymbol{n} \times \boldsymbol{c} = \boldsymbol{n} \times \boldsymbol{b}_0 \text{ on } \partial\Omega\right\},\\
s \in S &= \{r \in H^1(\Omega) \,:\, r = r_0 \text{ on } \partial\Omega\},
\end{aligned}$$

where we use $(\cdot,\cdot)_\Omega$ for all L^2-inner products and use $\bar{d} = 2d - 3$ to define the curl operator for both 2D and 3D vector fields [12].

Note that appropriate boundary conditions are incorporated as part of the definition of the spaces. In general, the issue of boundary conditions is rather involved and is strongly connected to the problem at hand and the domain. It requires taking into account geometric and physical conservation considerations, as the extensive discussion in [11, Chapter 4] demonstrates. For our model problem, Dirichlet boundary conditions are incorporated. For a discussion of Neumann and other boundary conditions we refer the reader to [11, 18, 25] and the references therein.

Let the domain Ω be divided into regular meshes $\mathcal{T}_h = \{K\}$ made up of triangles ($d = 2$) or tetrahedra ($d = 3$) with mesh size h. We will seek the weak solution $(\boldsymbol{u}_h, p_h, \boldsymbol{b}_h, r_h)$ in the finite element spaces

$$\begin{aligned}
\boldsymbol{u}_h \in \boldsymbol{V}_h &= \{\, \boldsymbol{u} \in \boldsymbol{V} \,:\, \boldsymbol{u}|_K \in \mathcal{P}_2(K)^d,\ K \in \mathcal{T}_h \,\},\\
p_h \in Q_h &= \{\, p \in Q \cap H^1(\Omega) \,:\, p|_K \in \mathcal{P}_1(K),\ K \in \mathcal{T}_h \,\},\\
\boldsymbol{b}_h \in \boldsymbol{C}_h &= \{\, \boldsymbol{b} \in \boldsymbol{C} \,:\, \boldsymbol{b}|_K \in \boldsymbol{R}_1(K),\ K \in \mathcal{T}_h \,\},\\
r_h \in S_h &= \{\, r \in S \,:\, r|_K \in \mathcal{P}_1(K),\ K \in \mathcal{T}_h \,\},
\end{aligned} \tag{2.1}$$

where we denote $\mathcal{P}_k(K)$ and $\boldsymbol{R}_1(K)$ as the space of polynomials of degree at most k and the space of Nédélec vector elements of the first kind [20, 23], respectively. From the linearization in (1.3), the weak formulation of the problem is given by finding

$(\boldsymbol{\delta} u_h, \delta p_h, \boldsymbol{\delta} b_h, \delta r_h) \in \boldsymbol{V}_h \times Q_h \times \boldsymbol{C}_h \times S_h$ such that

$$\begin{aligned} \widetilde{F}(\boldsymbol{u}_h^k; \boldsymbol{\delta} u_h, \boldsymbol{v}) + B(\boldsymbol{v}, p_h^k) + \widetilde{C}(\boldsymbol{b}_h^k; \boldsymbol{v}, \boldsymbol{\delta} b_h) &= R_u(\boldsymbol{u}_h^k, \boldsymbol{b}_h^k, p_h^k; \boldsymbol{v}), \\ B(\boldsymbol{\delta} u_h, q) &= R_p(\boldsymbol{u}_h^k; q), \\ \widetilde{M}(\boldsymbol{u}_h^k; \boldsymbol{\delta} b_h, \boldsymbol{c}) - C(\boldsymbol{b}_h^k; \boldsymbol{\delta} u_h, \boldsymbol{c}) + D(\boldsymbol{c}, \delta r_h) &= R_b(\boldsymbol{u}_h^k, \boldsymbol{b}_h^k, r_h^k; \boldsymbol{c}), \\ D(\boldsymbol{\delta} b_h, s) &= R_r(\boldsymbol{b}_h^k; s) \end{aligned} \tag{2.2}$$

for all $(\boldsymbol{v}, q, \boldsymbol{c}, s) \in \boldsymbol{V}_h \times Q_h \times \boldsymbol{C}_h \times S_h$, where

$$\begin{aligned} \widetilde{F}(\boldsymbol{w}; \boldsymbol{u}, \boldsymbol{v}) &= F(\boldsymbol{w}; \boldsymbol{u}, \boldsymbol{v}) + \alpha F_{\mathrm{NT}}(\boldsymbol{w}; \boldsymbol{u}, \boldsymbol{v}), \\ \widetilde{C}(\boldsymbol{d}; \boldsymbol{v}, \boldsymbol{b}) &= C(\boldsymbol{d}; \boldsymbol{v}, \boldsymbol{b}) + \alpha C_{\mathrm{NT}}(\boldsymbol{d}; \boldsymbol{v}, \boldsymbol{b}), \\ \widetilde{M}(\boldsymbol{w}; \boldsymbol{b}, \boldsymbol{c}) &= M(\boldsymbol{b}, \boldsymbol{c}) + \alpha M_{\mathrm{NT}}(\boldsymbol{w}; \boldsymbol{b}, \boldsymbol{c}). \end{aligned}$$

We use the subscript NT to denote Newton linearization. The linearization is around $(\boldsymbol{u}_h^k, \boldsymbol{b}_h^k)$, and thus the right-hand side represents the residual at the current iteration $(\boldsymbol{u}_h, p_h, \boldsymbol{b}_h, r_h)$:

$$\begin{aligned} R_u(\boldsymbol{u}_h^k, \boldsymbol{b}_h^k, p^k; \boldsymbol{v}) &= (\boldsymbol{f}, \boldsymbol{v})_\Omega - F(\boldsymbol{u}_h^k; \boldsymbol{u}_h^k, \boldsymbol{v}) - C(\boldsymbol{b}_h^k; \boldsymbol{v}, \boldsymbol{b}_h^k) - B(\boldsymbol{v}, p_h^k), \\ R_p(\boldsymbol{u}_h^k; q) &= -B(\boldsymbol{u}_h^k, q), \\ R_b(\boldsymbol{u}_h^k, \boldsymbol{b}_h^k, r^k; \boldsymbol{c}) &= (\boldsymbol{g}, c)_\Omega - M(\boldsymbol{b}_h^k, \boldsymbol{c}) + C(\boldsymbol{b}_h^k; \boldsymbol{u}_h^k, \boldsymbol{c}) - D(\boldsymbol{c}, r_h^k), \\ R_r(\boldsymbol{b}_h^k; s) &= -D(\boldsymbol{b}_h^k, s), \end{aligned} \quad . \tag{2.3}$$

The Picard forms that define (2.2) are given as

$$\begin{aligned} F(\boldsymbol{w}; \boldsymbol{u}, \boldsymbol{v}) &= \nu(\nabla \boldsymbol{u}, \nabla \boldsymbol{v})_\Omega + \big((\boldsymbol{w} \cdot \nabla)\boldsymbol{u}, \boldsymbol{v}\big)_\Omega, & B(\boldsymbol{u}, q) &= -(\nabla \cdot \boldsymbol{u},\, q)_\Omega, \\ M(\boldsymbol{b}, \boldsymbol{c}) &= \kappa \nu_m (\nabla \times \boldsymbol{b}, \nabla \times \boldsymbol{c})_\Omega, & D(\boldsymbol{b}, s) &= (\boldsymbol{b}, \nabla s)_\Omega, \\ C(\boldsymbol{d}; \boldsymbol{v}, \boldsymbol{b}) &= \kappa \left(\boldsymbol{v} \times \boldsymbol{d}, \nabla \times \boldsymbol{b}\right)_\Omega, \end{aligned} \tag{2.4}$$

and the Newton forms are given by

$$\begin{aligned} F_{\mathrm{NT}}(\boldsymbol{w}; \boldsymbol{u}, \boldsymbol{v}) &= \big((\boldsymbol{u} \cdot \nabla)\boldsymbol{w}, \boldsymbol{v}\big)_\Omega, \\ C_{\mathrm{NT}}(\boldsymbol{d}; \boldsymbol{v}, \boldsymbol{b}) &= \kappa \left(\boldsymbol{v} \times \boldsymbol{b}, \nabla \times \boldsymbol{d}\right)_\Omega, \\ M_{\mathrm{NT}}(\boldsymbol{w}; \boldsymbol{b}, \boldsymbol{c}) &= -\kappa \left(\boldsymbol{w} \times \boldsymbol{b}, \nabla \times \boldsymbol{c}\right)_\Omega. \end{aligned} \tag{2.5}$$

2.2. The linear system.

Upon discretization of the weak formulation in (2.2)–(2.3) with the corresponding forms given by (2.4), we obtain the following linear system:

$$\begin{pmatrix} F + \alpha F_{\mathrm{NT}} & B^T & C^T + \alpha C_{\mathrm{NT}}^T & 0 \\ B & 0 & 0 & 0 \\ -C & 0 & M + \alpha M_{\mathrm{NT}} & D^T \\ 0 & 0 & D & 0 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta p \\ \delta b \\ \delta r \end{pmatrix} = \begin{pmatrix} r_u \\ r_p \\ r_b \\ r_r \end{pmatrix} \tag{2.6}$$

with

$$\begin{aligned} r_u &= f - F u_k - C^T b_k - B^T p^k, \\ r_p &= -B u_k, \\ r_b &= g - M u_k + C b_k - D^T r^k, \\ r_r &= -D b_k, \end{aligned}$$

where F is a discrete convection-diffusion operator, B is a fluid divergence operator, M is the curl-curl operator, D is the magnetic divergence operator, and C represents the coupling terms. The dimensions are denoted as follows:

$$\dim(\delta u) = n_u, \quad \dim(\delta p) = m_u, \quad \dim(\delta b) = n_b, \quad \text{and} \quad \dim(\delta r) = m_b. \tag{2.7}$$

System (2.6) needs to be solved repeatedly, with changing right-hand sides, throughout the nonlinear iteration.

2.3. Null spaces. To complete the introduction of mathematical objects that we will be using for solving the problem, we establish notation and a necessary property of the null spaces involved. The function spaces that we are using allow for a seamless transition between the continuous and the discrete forms. Similarly to [13, section 2.2], we introduce the Helmholtz decomposition. To aid our discussion, let us denote the basis functions for the finite element spaces $\boldsymbol{C}_h$ and S_h in (2.1) as

$$\boldsymbol{C}_h = \operatorname{span}\langle \boldsymbol{\phi}_j \rangle_{j=1}^{n_b} \quad \text{and} \quad S_h = \operatorname{span}\langle \beta_i \rangle_{i=1}^{m_b}.$$

Thus, $\nabla S_h \subset \boldsymbol{C}_h$ such that there is a matrix $G \in \mathbb{R}^{n_b \times m_b}$:

$$\nabla \beta_j = \sum_{i=1}^{n_b} G_{ij} \boldsymbol{\phi}_j \quad \text{for} \quad j = 1, \dots, m_b.$$

Given a function $r_h \in S_h$ defined by $r_h = \sum_{j=1}^{m_b} r_j \beta_j$,

$$\nabla r_h = \sum_{i=1}^{n_b} \sum_{j=1}^{m_b} G_{ij} r_j \boldsymbol{\phi}_j,$$

such that for $q \in \mathbb{R}^{m_b}$, we have

$$b = Gr, \tag{2.8}$$

describing the coefficient vectors of

$$\boldsymbol{b}_h = \nabla r_h \tag{2.9}$$

in terms of the basis functions. For the lowest order elements the entries of G are

$$G_{ij} = \begin{cases} 1 & \text{if node } j \text{ is the head of edge } i, \\ -1 & \text{if node } j \text{ is the tail of edge } i \\ 0 & \text{otherwise.} \end{cases} \tag{2.10}$$

In a similar fashion to [13, Proposition 2.2], we now establish two useful matrix equalities.

PROPOSITION 2.1. *The following relationships with respect to the discrete gradient operator, G, hold:*

1. $MG = 0$,
2. $C^T G = 0$.

Proof. By (2.4), the bilinear form for C^T is given by

$$C(\boldsymbol{d}_h; \boldsymbol{v}_h, \boldsymbol{b}_h) = \kappa \left(\boldsymbol{v}_h \times \boldsymbol{d}_h, \nabla \times \boldsymbol{b}_h \right)_\Omega.$$

The bilinear form for M is given in the same equation, too. Incorporating (2.9) into the above given bilinear form for C and using subsequently the tensor identity $\nabla \times \nabla r_h \equiv 0$, we see that on the continuous level $C(\boldsymbol{d}_h; \boldsymbol{v}_h, \nabla r_h) = 0$. Incorporating the relation for the discrete relationship (2.8), the result for C^T follows. Trivially, for M the same relation holds, given that it has the same null space. ∎

We note that the matrix properties in Proposition 2.1 do not hold in a more general setting, with nonconstant coefficients or an irregular mesh.

Next, we consider the null spaces for the discrete operators associated with the Newton iteration. In Proposition 2.1 we define the null space for C^T. Now, consider the continuous representation of the coupling terms from (1.3):

$$\begin{aligned} C^T &\leftrightarrow (\nabla \times \delta b) \times b^k, & C_{\mathrm{NT}}^T &\leftrightarrow (\nabla \times b^k) \times \delta b, \\ C &\leftrightarrow \nabla \times (\delta u \times b^k), & M_{\mathrm{NT}} &\leftrightarrow \nabla \times (u^k \times \delta b), \end{aligned}$$

where $\leftrightarrow$ denotes an association between the matrix and its corresponding continuous form. We recall that $(\delta u, \delta b)$ are the unknowns and (u^k, b^k) is the solution at the previous step. It is straightforward to find an expression for δb and the null space of C^T. However, the null space of C_{NT}^T is significantly harder to compute, since the solution at the current nonlinear iteration would have to be in the same direction as the curl of the solution at the previous nonlinear iteration. This is also the case for the null spaces of C and M_{NT}.

Despite the challenge in explicitly forming null spaces for the operators associated with the Newton iteration, in practice we observe that we can successfully apply the same solution techniques we use for the Picard iteration. We illustrate this in section 6.

3. A new formula for the inverse of the MHD coefficient matrix. Examining (2.6), we observe that for the purpose of designing a block preconditioner, it may be useful to understand the properties of $M + \alpha M_{\mathrm{NT}}$ and $C^T + \alpha C_{\mathrm{NT}}^T$. The case $\alpha = 0$ corresponds to the Picard iteration, and those two matrices reduce to M and C^T, respectively. We have established in Proposition 2.1 that the null space of those matrices is identical, given by G. In this section we show that this property can be exploited to derive a new formula for the inverse.

Let us denote by $\mathcal{K}$ the coefficient matrix in the MHD model (2.6) and write it as

$$\mathcal{K} = \begin{pmatrix} \mathcal{K}_{\mathrm{NS}} & \mathcal{K}_{\mathrm{C}}^T \\ -\mathcal{K}_{\mathrm{C}} & \mathcal{K}_{\mathrm{M}} \end{pmatrix},$$

where $\mathcal{K}_{\mathrm{NS}}$ is the Navier–Stokes matrix, $\mathcal{K}_{\mathrm{C}}$ is the block for the coupling, and $\mathcal{K}_{\mathrm{M}}$ is the Maxwell matrix:

$$\begin{aligned} \mathcal{K}_{\mathrm{NS}} &= \begin{pmatrix} F & B^T \\ B & 0 \end{pmatrix}, & \mathcal{K}_{\mathrm{M}} &= \begin{pmatrix} M & D^T \\ D & 0 \end{pmatrix}, \\ \mathcal{K}_{\mathrm{C}}^T &= \begin{pmatrix} C^T & 0 \\ 0 & 0 \end{pmatrix}, & \mathcal{K}_{\mathrm{C}} &= \begin{pmatrix} C & 0 \\ 0 & 0 \end{pmatrix}. \end{aligned} \tag{3.1}$$

Then, by [5, equation (3.4)], the inverse is given by

$$\mathcal{K}^{-1} = \begin{pmatrix} \mathcal{K}_{\mathrm{NS}}^{-1} + \mathcal{K}_{\mathrm{NS}}^{-1}\mathcal{K}_{\mathrm{C}}^T\mathcal{S}^{-1}\mathcal{K}_{\mathrm{C}}{\mathcal{K}_{\mathrm{NS}}}^{-1} & -\mathcal{K}_{\mathrm{NS}}^{-1}\mathcal{K}_{\mathrm{C}}^T\mathcal{S}^{-1} \\ \mathcal{S}^{-1}\mathcal{K}_{\mathrm{C}}{\mathcal{K}_{\mathrm{NS}}}^{-1} & \mathcal{S}^{-1} \end{pmatrix}, \tag{3.2}$$

where $\mathcal{S}$ denotes the Schur complement,

$$\mathcal{S} = \mathcal{K}_{\mathrm{M}} + \mathcal{K}_{\mathrm{C}}\mathcal{K}_{\mathrm{NS}}^{-1}\mathcal{K}_{\mathrm{C}}^T. \tag{3.3}$$

The inverses $\mathcal{K}_{\mathrm{NS}}^{-1}$ and $\mathcal{S}^{-1}$ appear multiple times in (3.2), and we now derive explicit formulas that further reveal their block structure. Notably, using results that have appeared in [9], we show that $\mathcal{S}^{-1}$ has a zero (2,2) block.

Let us denote the inverse of the Navier–Stokes coefficient matrix as

$$\mathcal{K}_{\mathrm{NS}}^{-1} = \begin{pmatrix} K_1 & K_2 \\ K_3 & K_4 \end{pmatrix},$$

and let

$$L = DG \quad \text{and} \quad M_F = M + CK_1C^T + D^TW^{-1}D. \tag{3.4}$$

We then have the following useful result.

Lemma 3.1. *The inverse of the Schur complement, $\mathcal{S}$, can be written as*

$$\mathcal{S}^{-1} = \begin{pmatrix} M_F^{-1}(I - D^TL^{-1}G^T) & GL^{-1} \\ L^{-1}G^T & 0 \end{pmatrix}, \tag{3.5}$$

where G is the null space of M and M_F and L are defined in (3.4).

Proof. Writing out all the matrices involved in formula (3.3) for $\mathcal{S}$, we have

$$\mathcal{S} = \begin{pmatrix} M + CK_1C^T & D^T \\ D & 0 \end{pmatrix}.$$

By Proposition 2.1, the discrete gradient operator is the null space of M and C^T, and therefore

$$\mathrm{null}(M + CK_1C^T) = \mathrm{null}(M) \quad \text{and} \quad \dim(\mathrm{null}(M)) = m_b. \tag{3.6}$$

Thus, the (1,1) block of the Schur complement has the maximum nullity which still allows a nonsingular saddle-point system. Therefore, using [9, equation (3.6)], the inverse of the Schur complement is given by (3.5). □

The lemma and the null space properties of the discrete operators involved can now be used to generate a new formula for $\mathcal{K}^{-1}$, which shows the remarkable property that this inverse of the MHD coefficient matrix has five zero blocks.

Theorem 3.2. *The inverse of the MHD coefficient matrix, namely, $\mathcal{K}^{-1}$, is given by*

$$\mathcal{K}^{-1} = \begin{pmatrix} K_1 - K_1ZK_1 & K_2 - K_1ZK_2 & -K_1C^TM_F^{-1}H & 0 \\ K_3 - K_3ZK_1 & K_4 - K_3ZK_2 & -K_3C^TM_F^{-1}H & 0 \\ M_F^{-1}CK_1 & M_F^{-1}CK_2 & M_F^{-1}H & GL^{-1} \\ 0 & 0 & L^{-1}G^T & 0 \end{pmatrix}, \tag{3.7}$$

where L is defined in (3.4),

$$Z = C^TM_F^{-1}C, \quad \text{and} \quad H = I - D^TL^{-1}G^T.$$

Proof. The proof is rather straightforward and is based on using the result of Lemma 3.1, along with the following observation. From the inverse formula in (3.2) we see that there are block operations of the form $\mathcal{S}^{-1}\mathcal{K}_{\rm C}$ in the $(1,1)$ and $(2,1)$ blocks. Multiplying these out gives

$$\mathcal{S}^{-1}\mathcal{K}_{\rm C} = \begin{pmatrix} M_F^{-1}(I - DL^{-1}G^T)C & 0 \\ L^{-1}G^T C & 0 \end{pmatrix} = \begin{pmatrix} M_F^{-1}C & 0 \\ 0 & 0 \end{pmatrix}. \tag{3.8}$$

Using now (3.2) and Lemma 3.1, the block inverse formula in (3.2) is explicitly given by (3.7), as required. ∎

4. A new approximate inverse-based preconditioner.

Having developed a useful formula for the exact inverse of the MHD coefficient matrix, we now aim to find a preconditioner based on sparsifying the formula in (3.7), by exploiting null space properties and the magnitudes of coefficient matrix entries.

The inverse of the Navier–Stokes coefficient matrix is a useful quantity in the derivation that follows. Using [5, equation (3.4)], it is given by

$$\mathcal{K}_{\rm NS}^{-1} = \begin{pmatrix} K_1 & K_2 \\ K_3 & K_4 \end{pmatrix} = \begin{pmatrix} F^{-1} - F^{-1}B^T S_{\rm NS}^{-1} B F^{-1} & -F^{-1}B^T S_{\rm NS}^{-1} \\ -S_{\rm NS}^{-1} B F^{-1} & S_{\rm NS}^{-1} \end{pmatrix}, \tag{4.1}$$

where

$$S_{\rm NS} = BF^{-1}B^T$$

is the fluid Schur complement.

Our strategy in general will be to use our new formula for the inverse of the coefficient matrix and identify places where we may be able to drop elements of the inverse while maintaining an effective approximation to it from a preconditioning point of view.

4.1. Sparsification based on projection.

The formula for S^{-1} is given in (3.5). We now show that from a preconditioning point of view, it may be possible to drop the dense matrix term $D^T L^{-1} G^T$ while preserving approximately the same spectral properties of the preconditioned matrix.

To that end, observe that from Proposition 2.1 we have

$$HM = (I - D^T L^{-1} G^T)M = M.$$

From (3.4), L is the product of the matrices D and G, which is equivalent to selecting it to be a scalar Laplacian [13, Proposition 2.2], which is particularly appealing. Using this definition of L, we obtain the useful relation

$$G = M_L^{-1} D^T, \quad \text{where} \quad M_L = M + D^T L^{-1} D,$$

and from [9, Theorem 3.5], which states that

$$DM_L^{-1}D^T = L, \tag{4.2}$$

we obtain

$$HD^T = D^T - D^T L^{-1} D M_L^{-1} D^T = 0.$$

By (4.2), we have

$$\begin{aligned} H^2 &= (I - D^T L^{-1} D M_L^{-1})(I - D^T L^{-1} D M_L^{-1}) \\ &= I - 2D^T L^{-1} D M_L^{-1} + D^T L^{-1} D M_L^{-1} D^T L^{-1} D M_L^{-1} \\ &= I - 2D^T L^{-1} D M_L^{-1} + D^T L^{-1} D M_L^{-1} = H. \end{aligned}$$

TABLE 1

Relevant matrices and their approximate orders, entrywise, based on assuming that the mesh size is sufficiently small and the PDE parameters are moderate in size.

Matrix	Approximate order
F	$\mathcal{O}(h^{-2})$
B	$\mathcal{O}(h^{-1})$
C	$\mathcal{O}(h^{-1})$
M_L	$\mathcal{O}(h^{-2})$
$BF^{-1}B^T$	$\mathcal{O}(1)$

Thus, H is an orthogonal projector onto the range space of M and the null space of D^T. This property is related to the discrete Helmholtz decomposition, which holds in this case:

$$\ker(M) \oplus \ker(D) = \mathbb{R}^n.$$

Based on the above derivation, we conclude that turning H into the identity may give us an effective approximation to S^{-1}:

$$\hat{\mathcal{S}}^{-1} = \begin{pmatrix} M_F^{-1} & GL^{-1} \\ L^{-1}G^T & 0 \end{pmatrix}. \tag{4.3}$$

4.2. Sparsification based on discretization. Next, we further sparsify (4.3) by considering the inverse of M_F. To that end, we look at magnitudes of entries in the matrices that make up M_F. In the discussion that follows, we will assume that the mesh size h is small and the parameters involved in the model are moderate, such that we can perform an asymptotic analysis that allows us to drop terms merely based on the order with respect to h.

The main bottleneck in forming and approximating M_F is CK_1C^T, where we recall that K_1 is the $(1,1)$ block matrix of the inverse of the Navier–Stokes subproblem.

First, we rewrite M_F as

$$M_F = M_L + CK_1C^T \quad \text{with} \quad M_L = M + D^TL^{-1}D.$$

Then, by the Sherman–Morrison–Woodbury formula,

$$M_F^{-1} = M_L^{-1} - M_L^{-1}CK_1(K_1 - K_1C^TM_L^{-1}CK_1)^{-1}K_1C^TM_L^{-1}. \tag{4.4}$$

From approximate matrix orders given in Table 1, we can approximate the matrices in (4.4). We have numerically confirmed that

$$K_1 \approx \mathcal{O}(h^2) \quad \text{and} \quad K_1 - K_1C^TM_L^{-1}CK_1 \approx \mathcal{O}(h^2). \tag{4.5}$$

Taking the leading order approximation, we see that

$$M_L^{-1} \approx \mathcal{O}(h^2) \quad \text{and} \quad M_L^{-1}CK_1(K_1 - K_1C^TM_L^{-1}CK_1)^{-1}K_1C^TM_L^{-1} \approx \mathcal{O}(h^4).$$

Thus, we can use the approximation

$$M_F^{-1} \approx M_L^{-1}. \tag{4.6}$$

Finally, using the definition of L in (3.4), which we recall is equivalent to L being a scalar Laplacian, in [13, Corollary 3.2] the authors showed that $M + X$ is spectrally equivalent to $M + D^T L^{-1} D$. Thus, we replace M_L by $M + X$ in the final approximation to the inverse of the Schur complement:

$$\mathcal{S}_{\text{approx}}^{-1} = \begin{pmatrix} M_X^{-1} & GL^{-1} \\ L^{-1}G^T & 0 \end{pmatrix}, \tag{4.7}$$

where $M_X = M + X$.

In a similar fashion to the approximation of M_F, we consider the magnitude of the matrix entries to further simplify the inverse formula in (3.7). From the definition of the inverse of the Navier–Stokes block in (4.1), we take the leading order approximations of K_i for $i = 1, 2, 3, 4$, such that

$$K_1 \approx \mathcal{O}(h^2), \quad K_2 \approx \mathcal{O}(h), \quad K_3 \approx \mathcal{O}(h), \quad \text{and} \quad K_4 \approx \mathcal{O}(1).$$

Using these approximations we observe that

$$Z = C^T M_F^{-1} C \approx \mathcal{O}(1),$$

and thus we take the leading order terms of the upper block 2-by-2 matrix of (3.7) to obtain the approximation

$$\begin{pmatrix} K_1 - K_1 Z K_1 & K_2 - K_1 Z K_2 \\ K_3 - K_3 Z K_1 & K_4 - K_3 Z K_2 \end{pmatrix} \approx \begin{pmatrix} K_1 & K_2 \\ K_3 & K_4 \end{pmatrix}. \tag{4.8}$$

Hence, using (4.7)–(4.8) yields the following approximation of (3.7):

$$\widetilde{\mathcal{P}}_1^{-1} = \begin{pmatrix} K_1 & K_2 & -K_1 C^T M_X^{-1} & 0 \\ K_3 & K_4 & -K_3 C^T M_X^{-1} & 0 \\ M_X^{-1} C K_1 & M_X^{-1} C K_2 & M_X^{-1} & GL^{-1} \\ 0 & 0 & L^{-1} G^T & 0 \end{pmatrix}. \tag{4.9}$$

We recall that K_i, $i = 1, 2, 3, 4$, are defined in (4.1).

Finally, we consider the approximate orders of the individual blocks of (4.9). Removing the $\mathcal{O}(h^3)$ terms in the (1,3) and (3,1) blocks of (4.9) yields the approximation:

$$\mathcal{P}_1^{-1} = \begin{pmatrix} F^{-1} - F^{-1} B^T S_{\text{NS}}^{-1} B F^{-1} & F^{-1} B^T S_{\text{NS}}^{-1} & 0 & 0 \\ S_{\text{NS}}^{-1} B F^{-1} & -S_{\text{NS}}^{-1} & -S_{\text{NS}}^{-1} B F^{-1} C^T M_X^{-1} & 0 \\ 0 & M_X^{-1} C F^{-1} B^T S_{\text{NS}}^{-1} & M_X^{-1} & GL^{-1} \\ 0 & 0 & L^{-1} G^T & 0 \end{pmatrix}.$$

Table 2 shows the number of solves or matrix-vector multiplies per linear iteration for an efficient implementation of the approximate inverse preconditioner.

4.3. Spectral analysis. The effectiveness of our preconditioner is largely determined by the spectral structure of the preconditioned matrix. For the eigenvalue analysis below, we use $\widetilde{\mathcal{P}}_1^{-1}$ in (4.9) for practical reasons; the analysis is significantly easier in that case. The eigenvalue analysis for $\widetilde{\mathcal{P}}_1^{-1}\mathcal{K}$ provides a theoretical justification for the use of the more practical preconditioner $\mathcal{P}_1^{-1}$, and we use the latter for the numerical results in section 6.

TABLE 2
Number of solves and matrix-vector multiplies in applying $\mathcal{P}_1^{-1}$ to an arbitrary vector.

Solves/Multiplies	Number of operations
F^{-1}	4
A_p^{-1}	3
Q_p^{-1}	3
M_X^{-1}	2
L^{-1}	2
C^T or C	2
B^T or B	4
G^T or G	2

The $\widetilde{\mathcal{P}}_1^{-1}\mathcal{K}$ is given by

$$\widetilde{\mathcal{P}}_1^{-1}\mathcal{K} = \begin{pmatrix} I_{\mathrm{u}} + K_1 C^T M_L^{-1} C & 0 & K_1 C^T (I - M_L^{-1} M) & 0 \\ S_{\mathrm{NS}}^{-1} B F^{-1} C^T M_L^{-1} C & I_{\mathrm{p}} & S_{\mathrm{NS}}^{-1} B F^{-1} C^T (I - M_L^{-1} M) & 0 \\ 0 & 0 & M_L^{-1}(M + C K_1 C^T) & G \\ 0 & 0 & 0 & I_{\mathrm{r}} \end{pmatrix}. \tag{4.10}$$

Let us introduce a few identities utilizing the null space properties of C^T and M_L. Proposition 4.1 is particularly useful to simplify (4.10).

PROPOSITION 4.1. *The following relations hold:*

$$\begin{aligned} &\text{(i)} \qquad C^T(I - M_L^{-1}M) = 0, \\ &\text{(ii)} \qquad C^T(I + M_L^{-1}M) = 2C^T. \end{aligned}$$

Proof. Since C^T and M have the same null space, by Proposition 2.1 and the Helmholtz decomposition, we obtain

$$C^T(I - M_L^{-1}M)b = C^T(I - M_L^{-1}M)d,$$

where $d \notin \mathrm{Null}(M)$. Since $M_L^{-1}M$ is zero on the null space of M and the identity on the range space of M,

$$(I - M_L^{-1}M) \in \mathrm{Null}(M).$$

Therefore, identity (i) holds due to the fact that the null spaces of C^T and M are the same. Using similar arguments, identity (ii) can also be shown to be true. □

Using Proposition 4.1 simplifies (4.10) to

$$\widetilde{\mathcal{P}}_1^{-1}\mathcal{K} = \left(\begin{array}{cc|cc} I_{\mathrm{u}} + K_1 C^T M_L^{-1} C & 0 & 0 & 0 \\ S_{\mathrm{NS}}^{-1} B F^{-1} C^T M_L^{-1} C & I_{\mathrm{p}} & 0 & 0 \\ \hline 0 & 0 & M_L^{-1}(M + C K_1 C^T) & G \\ 0 & 0 & 0 & I_{\mathrm{r}} \end{array}\right). \tag{4.11}$$

In Theorem 4.2, (u, p, b, r) denote the incremental quantities from the nonlinear iteration and the dimensions are defined in (2.7).

THEOREM 4.2. *The matrix $\widetilde{\mathcal{P}}_1^{-1}\mathcal{K}$ has an eigenvalue $\lambda = 1$ of algebraic multiplicity at least $n_u - n_b + 3m_b + m_u$. The corresponding eigenvectors $\{v_i\}_{i=1}^{n_u - n_b + 3m_b + m_u}$ are given by*

$$v_i = (u_i,\, p_i,\, b_i,\, r_i),$$

where $u_i \in Null(C)$, $b_i \in Null(M)$, and p_i and r_i are free.

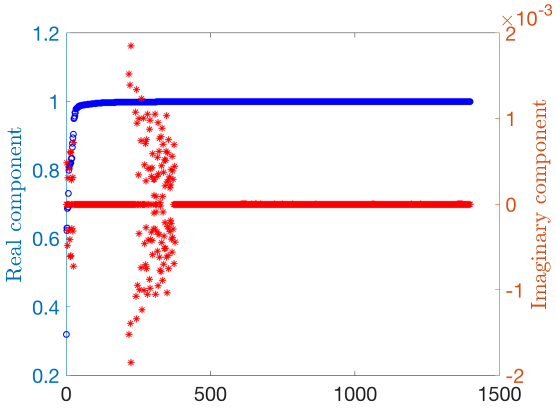


FIG. 1. *Real (blue) and imaginary (red) parts of eigenvalues of preconditioned matrix* $\mathcal{P}_1^{-1}\mathcal{K}$ *using the smooth solution given in* (6.2) *of section* 6.1. *The dimensions of these matrices are* 1399×1399.

Proof. Since (4.11) is block diagonal where the two 2-by-2 blocks are also triangular, generalized eigenvalue problem

$$\widetilde{\mathcal{P}}_1^{-1}\mathcal{K}v = \lambda v,$$

where $v = (u,\, p,\, b,\, r)$, can be written as

$$\lambda u = (I_{\mathrm{u}} + K_1 C^T M_L^{-1} C)u, \tag{4.12a}$$

$$\lambda p = p, \tag{4.12b}$$

$$\lambda b = M_L^{-1}(M + CK_1C^T)b, \tag{4.12c}$$

$$\lambda r = r. \tag{4.12d}$$

Consider $\lambda = 1$, and then (4.12b) and (4.12d) are automatically satisfied. Taking $u \in \mathrm{Null}(C)$ and $b \in \mathrm{Null}(M + CK_1C^T)$, (4.12a) and (4.12c) also hold. Since from Proposition 2.1, the null space of M and C^T are the same, we choose $b \in \mathrm{Null}(M)$. ◻

In practice, we use the approximate inverse preconditioner in $\mathcal{P}_1^{-1}$ with M_X as the approximation to the magnetic primal Schur complement and the pressure-convection diffusion (PCD) approximation developed in [8] for the fluid Schur complement, S_{NS}. The PCD approximation is based on

$$S_{\mathrm{NS}} = BF^{-1}B^T \approx A_p\, F_p^{-1} Q_p, \tag{4.13}$$

where the matrix A_p is the pressure Laplacian, F_p is the pressure convection-diffusion operator, and Q_p is the pressure mass matrix.

Figure 1 shows the eigenvalues of $\mathcal{P}_1^{-1}\mathcal{K}$ for the first numerical experiment we present in section 6. We see that the clustering around the eigenvalue $\lambda = 1$ is very strong.

5. Block triangular preconditioning. A common preconditioning approach in the literature for block systems is based on forming block triangular preconditioners. Often this is done by forming block decompositions and approximating the Schur complements arising within them. This approach is rather effective for the MHD model, as shown in [24, 25, 31], in particular for 2D problems. When it comes to 3D problems, scalability is often lost as the mesh is refined or the coupling and/or viscosity parameters vary.

In this section we return to this approach and show that the approximate block Schur complement inverse we established in (4.7) gives us an opportunity to derive a new preconditioner, which falls within the general category of block triangular preconditioners. Let us define

$$\tilde{\mathcal{P}}_2 = \begin{pmatrix} \mathcal{K}_{\rm NS} & \mathcal{K}_{\rm C}^T \\ 0 & -\mathcal{S} \end{pmatrix}, \tag{5.1}$$

where $\mathcal{K}_{\rm NS}$ and $\mathcal{K}_{\rm C}^T$ are defined in (3.1) and $\mathcal{S}$ in (3.3). From [15, 22], under mild invertibility conditions, the preconditioned matrix, $\tilde{\mathcal{P}}_2^{-1}\mathcal{K}$, has precisely two eigenvalues ±1 and is diagonalizable. We would therefore expect an appropriate Krylov subspace solver to converge within two iterations in exact arithmetic.

The direct solve for the Navier–Stokes system is computationally costly, so we approximate $\mathcal{K}_{\rm NS}$ with the Schur complement system:

$$\mathcal{P}_{\rm NS} = \begin{pmatrix} F & B^T \\ 0 & -S_{\rm NS} \end{pmatrix}, \tag{5.2}$$

where $S_{\rm NS}$ is the fluid Schur complement defined in (4.1). Using (5.2), we obtain the practical preconditioner

$$\mathcal{P}_2 = \begin{pmatrix} \mathcal{P}_{\rm NS} & \mathcal{K}_{\rm C}^T \\ 0 & -\mathcal{S} \end{pmatrix}, \tag{5.3}$$

where $\mathcal{P}_{\rm NS}$, $\mathcal{K}_{\rm C}^T$ and $\mathcal{S}$ are defined in (5.2), (3.1), and (3.3), respectively.

The preconditioner is interesting for this problem in terms of its spectral properties. In particular, as we did in section 4, we are able to take advantage of the common null spaces of the matrices M and C to obtain strong clustering of the eigenvalues of the preconditioned matrix. This is shown in Theorem 5.1, below.

THEOREM 5.1. *The matrix $\mathcal{P}_2^{-1}\mathcal{K}$ has an eigenvalue $\lambda = 1$ of algebraic multiplicity at least n_u and an eigenvalue $\lambda = -1$ of algebraic multiplicity at least n_b. The corresponding (known) eigenvectors are given as*

$\lambda = 1:$ *with eigenvectors $\{v_i\}_{i=1}^{n_b-m_b}$ and $\{v_j\}_{j=n_b-m_b+1}^{n_u}$, given as follows:*

$$v_i = (u_i, -S^{-1}Bu_i, b_i, 0) \quad \textit{and} \quad v_j = (u_j, -S^{-1}Bu_j, 0, 0),$$

where $b_i \in null(D) \neq 0$, $Cu_i = (2M + CK_1C^T)b_i$ and $u_j \in null(C)$.

$\lambda = -1:$ *with eigenvectors $\{v_i\}_{i=1}^{n_b-m_b}$ and $\{v_j\}_{j=n_b-m_b+1}^{n_b}$, given as follows:*

$$v_i = (u_i, 0, b_i, r_i) \quad \textit{and} \quad v_j = (0, 0, b_j, r_j), \tag{5.4}$$

where $u_i \in null(B) \neq 0$, $Fu_i + C^Tb_i = 0$, $b_j \in null(M)$, and r_i and r_j are free.

Proof. The corresponding eigenvalue problem is

$$\begin{pmatrix} F & B^T & C^T & 0 \\ B & 0 & 0 & 0 \\ -C & 0 & M & D^T \\ 0 & 0 & D & 0 \end{pmatrix} \begin{pmatrix} u \\ p \\ b \\ r \end{pmatrix} = \lambda \begin{pmatrix} F & B^T & C^T & 0 \\ 0 & -S_{\rm NS} & 0 & 0 \\ 0 & 0 & -(M+K_C) & -D^T \\ 0 & 0 & -D & 0 \end{pmatrix} \begin{pmatrix} u \\ p \\ b \\ r \end{pmatrix},$$

where $K_C = CK_1C^T$. The four block rows of the generalized eigenvalue problem can be written as

$$(1-\lambda)(Fu + B^Tp + C^Tb) = 0, \tag{5.5a}$$

$$Bu = -\lambda S_{\mathrm{NS}}\, p, \tag{5.5b}$$

$$(1+\lambda)(Mb + D^Tr) + \lambda CK_1C^Tb - Cu = 0, \tag{5.5c}$$

$$(1+\lambda)Db = 0. \tag{5.5d}$$

We split the eigenvalue analysis into two parts: $\lambda = 1$ and $\lambda = -1$.

$\lambda = 1$:

Equation (5.5a) is automatically satisfied. Equation (5.5b) simplifies to

$$p = -S_{\mathrm{NS}}^{-1}Bu.$$

From (5.5d) we have $Db = 0$, and hence $b \in \mathrm{null}(D)$. Let us take $r = 0$, and then (5.5c) yields

$$Cu = (2M + CK_1C^T)b. \tag{5.6}$$

Case 1. Consider $b = 0$, and then we have that $Cu = 0$. Hence, u must be in the null space of C. Since

$$\dim(\mathrm{null}(C)) \;=\; n_u - n_b + m_b,$$

this accounts for $n_u - n_b + m_b$ such eigenvectors.

Case 2. Consider $0 \neq b \in \mathrm{null}(D)$, and then $Cu = (2M + CK_1C^T)b$. Since the rank of C and $(2M + CK_1C^T)$ is $n_b - m_b$, then the condition (5.6) has at least $n_b - m_b$ linearly independent eigenvectors.

Therefore $\lambda = 1$ is an eigenvalue with algebraic multiplicity at least n_u.

$\lambda = -1$:

Equation (5.5d) is satisfied, and hence r is free. Simplifying (5.5c) gives

$$CK_1C^Tb + Cu = 0. \tag{5.7}$$

Let us take $u \in \mathrm{null}(B)$, and then $p = 0$ and the condition for b is

$$Fu + C^Tb = 0. \tag{5.8}$$

Under the condition that $u \in \mathrm{null}(B)$, (5.8) satisfies the equality (5.7).

Case 1. Consider $u = 0$, and then to satisfy (5.8) we require $C^Tb = 0$. Therefore, we take $b \in \mathrm{null}(C^T)$. Since the null space of C^T is made up of discrete gradients, then

$$\dim(\mathrm{null}(C^T)) \;=\; m_b.$$

Case 2. Consider $u \in \mathrm{null}(B)$ and $u \neq 0$, and then from (5.8) we have $u = -F^{-1}C^Tb$. Since the rank of C^T is $n_b - m_b$ and F is full rank, then there are only $n_b - m_b$ such linearly independent b's that determine u. Hence, for this case we obtain at least $n_b - m_b$ such eigenvectors.

Therefore $\lambda = -1$ is an eigenvalue with algebraic multiplicity at least n_b. □

$\mathcal{P}_2$ is defined in (5.3), but in practice we use the approximation for the inverse of the Schur complement (4.7). The eigenvalues of the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{K}$ are represented in Figure 2. As with the eigenvalues for the approximate inverse preconditioned matrix we see a small degradation of the eigenvalue clusters. However, we still see strong clustering around 1 and -1.

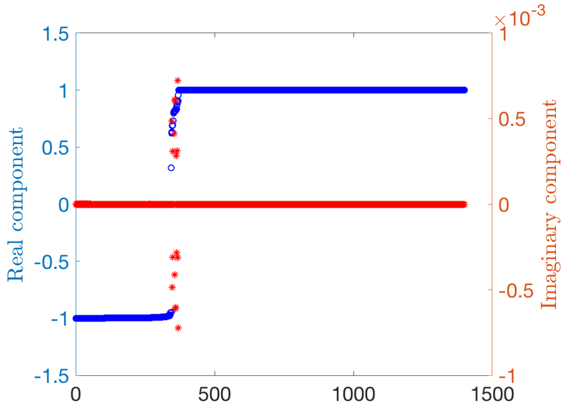


FIG. 2. *Real (blue) and imaginary (red) parts of eigenvalues of preconditioned matrix* $\mathcal{P}_2^{-1}\mathcal{K}$ *using the smooth solution given* (6.2)*. The dimensions of these matrices are* 1399×1399*.*

TABLE 3

Solution method for block systems associated with the preconditioners

Matrix	Implementation method
Q_p	single AMG V-cycle
A_p	single AMG V-cycle
$\hat{F}$ or F	Preconditioned AMG GMRES with tolerance 1e-2
$M+X$	AMG method developed in [14] with tolerance 1e-2
W	single AMG V-cycle

6. Numerical experiments. In this section we present several 3D numerical results to illustrate the performance and scalability of our preconditioning techniques.

Experimental setup. We use `FEniCS` [19], a finite element software package, to create the matrix system, and `PETSc` [3, 4] and `HYPRE` [10] to solve the resulting system of equations.

We set the nonlinear stopping tolerance to 1e-4 and the linear solve tolerance as 1e-3. For all experiments we use FGMRES(30) [26] as the linear solver and Picard or Newton for the nonlinear iteration schemes. The tolerances are not very strict, but they are fairly realistic for this challenging problem.

In section 4 we described the preconditioning approach, with a focus on the Picard iteration. For the Newton scheme, every solve associated with F and multiplication associated with C^T are replaced with $\hat{F} = F + F_{\mathrm{NT}}$ and $\hat{C}^T = C^T + C_{\mathrm{NT}}^T$, respectively, as per (2.6). Table 3 details the methods that we use to solve the systems associated with the block preconditioner. We note that G is constructed using (2.10).

We use the following notation:

- ℓ amesh level, *DoFs* is the total degrees of freedom, *time* is the average solve time;
- $\mathrm{it}_{\mathrm{NL}}$ is the number of nonlinear/Newton iterations to solve;
- it_{O} is the average number of linear/FGMRES iterations;
- $\mathrm{it}_{\mathrm{MX}}$ is the average of CG/auxiliary space iterations to solve $M+X$;
- it_{F} is the average of FGMRES iterations to solve $\hat{F}$.

We have inserted $\mathcal{P}_1$ or $\mathcal{P}_2$ superscripts to *time* or it_* to denote whether we use the approximate inverse or block preconditioner, respectively.

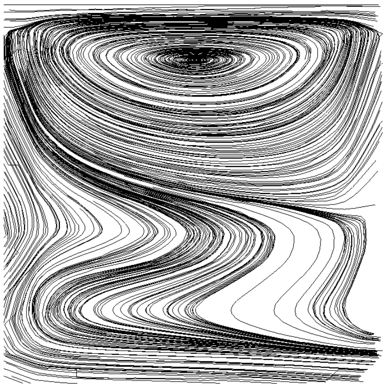


FIG. 3. *Stream lines of the velocity solution of cavity driven problem with* $\kappa = 1e1$, $\nu = 1e-1$, $\nu_m = 1e-1$, *and* $Ha = \sqrt{1000}$.

6.1. 3D cavity driven flow. The first example we consider is the classic lid driven cavity problem [8]. It is driven by the following Dirichlet boundary conditions:

$$\begin{aligned} \mathbf{u} &= (1,0,0) && \text{on} \quad z = 1, \\ \mathbf{u} &= (0,0,0) && \text{on} \quad x = \pm 1,\ y = \pm 1,\ z = -1, \\ \mathbf{n} \times \mathbf{b} &= \mathbf{n} \times \mathbf{b}_N && \text{on} \quad \partial\Omega, \\ r &= 0 && \text{on} \quad \partial\Omega, \end{aligned} \tag{6.1}$$

where $\mathbf{b}_N = (-1,0,0)$. Figure 3 depicts the stream lines for the velocity solution.

Scalability results. Along with the nondimensional parameters (ν, ν_m, and κ) described in section 1, we introduce the Hartmann number, which is equal to the ratio of electromagnetic and viscous forces. It is defined as

$$\text{Ha} = \sqrt{\frac{\kappa}{\nu\nu_m}}.$$

The larger the Hartmann number is, the stronger the coupling between the electromagnetics and hydrodynamics variables, and thus the more challenging the setting is for the numerical solution method.

Tables 4 and 5 show computational time and iteration counts using the approximate inverse and block triangular preconditioners, applying both the Newton and Picard nonlinear iteration schemes. Figures 4 and 5 show the timing results for a single solve of the Newton and Picard system, referring to $\textit{time}^{\mathcal{P}_1}$ and $\textit{time}^{\mathcal{P}_2}$ in the tables.

We can see that the approximate inverse preconditioner exhibits near-perfect scaling with respect to the FGMRES iterations for both the Newton and Picard iterations. On the other hand, the FGMRES iterations for the block triangular preconditioner increase each mesh level for the harder problem, $\text{Ha} = \sqrt{1000}$. For the easier parameter setup, Table 4, we can see that both preconditioners yield scalable iterations. The linear iterations for the Picard nonlinear scheme appear to be about half that for the Newton system. We speculate that this is because the preconditioners we derive are designed for the Picard iteration, for which it is easier to exploit the null space properties of the operators involved, as explained in section 2.3. However, we

TABLE 4

3D cavity driven using both the approximate inverse and block triangular preconditioner with parameters $\kappa = 1$, $\nu = 1$, $\nu_m = 1$, and $Ha = 1$.

ℓ	*DoFs*	$time^{\mathcal{P}_1}$	$it_{\rm NL}^{\mathcal{P}_1}$	$it_{\rm O}^{\mathcal{P}_1}$	$it_{\rm MX}^{\mathcal{P}_1}$	$it_{\rm F}^{\mathcal{P}_1}$	$time^{\mathcal{P}_2}$	$it_{\rm NL}^{\mathcal{P}_2}$	$it_{\rm O}^{\mathcal{P}_2}$	$it_{\rm MX}^{\mathcal{P}_2}$	$it_{\rm F}^{\mathcal{P}_2}$
1	14,012	1.2	3	10.0	1.7	1.6	0.8	4	21.5	2.0	2.0
2	28,436	2.8	3	10.0	1.8	1.6	1.9	4	21.5	2.0	2.0
3	64,697	11.4	3	9.7	1.9	1.8	6.9	4	21.0	2.0	2.0
4	245,276	34.6	4	11.0	2.1	1.9	13.7	3	18.3	2.0	2.0
5	937,715	255.2	4	10.5	2.2	1.9	135.9	3	19.0	2.5	2.0
6	5,057,636	1979.2	3	9.7	2.7	2.1	2273.5	3	22.3	3.0	2.5

(a) Newton iteration, $\alpha = 1$.

ℓ	*DoFs*	$time^{\mathcal{P}_1}$	$it_{\rm NL}^{\mathcal{P}_1}$	$it_{\rm O}^{\mathcal{P}_1}$	$it_{\rm MX}^{\mathcal{P}_1}$	$it_{\rm F}^{\mathcal{P}_1}$	$time^{\mathcal{P}_2}$	$it_{\rm NL}^{\mathcal{P}_2}$	$it_{\rm O}^{\mathcal{P}_2}$	$it_{\rm MX}^{\mathcal{P}_2}$	$it_{\rm F}^{\mathcal{P}_2}$
1	14,012	0.9	4	6.8	1.7	1.6	0.5	4	13.0	2.0	2.0
2	28,436	1.9	4	6.5	1.78	1.7	1.2	4	12.8	2.0	2.0
3	64,697	7.7	4	6.2	1.8	1.8	4.6	4	12.8	2.0	2.0
4	245,276	21.7	4	5.2	2.0	1.8	11.9	4	11.5	2.0	2.0
5	937,715	144.4	5	5.8	2.1	1.9	80.8	5	11.2	2.7	2.0
6	5,057,636	984.5	4	5.5	2.7	2.0	1074.7	4	11.8	3.1	2.5

(b) Picard iteration, $\alpha = 0$.

TABLE 5

3D cavity driven using both the approximate inverse and block triangular preconditioner with parameters $\kappa = 1e1, \nu = 1e-1, \nu_m = 1e-1$, and $Ha = \sqrt{1000}$.

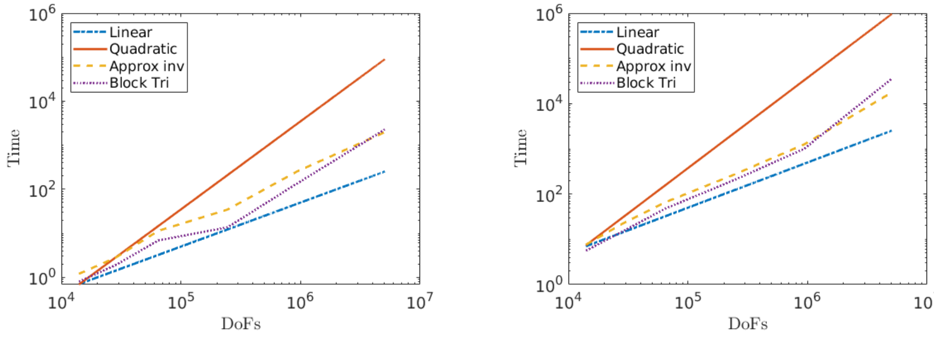
ℓ	*DoFs*	$time^{\mathcal{P}_1}$	$it_{\rm NL}^{\mathcal{P}_1}$	$it_{\rm O}^{\mathcal{P}_1}$	$it_{\rm MX}^{\mathcal{P}_1}$	$it_{\rm F}^{\mathcal{P}_1}$	$time^{\mathcal{P}_2}$	$it_{\rm NL}^{\mathcal{P}_2}$	$it_{\rm O}^{\mathcal{P}_2}$	$it_{\rm MX}^{\mathcal{P}_2}$	$it_{\rm F}^{\mathcal{P}_2}$
1	14,012	7.6	4	57.0	2.0	2.0	5.6	4	146.2	2.0	2.0
2	28,436	22.2	4	56.2	2.0	2.0	14.9	4	147.2	2.0	2.0
3	64,697	66.0	4	56.0	2.0	2.0	47.8	4	154.2	2.0	2.0
4	245,276	271.5	4	56.0	2.1	2.0	205.6	4	160.5	2.2	2.0
5	937,715	1255.2	4	55.5	2.8	2.0	1003.9	4	168.8	2.9	2.0
6	5,057,636	17656.4	4	58.5	3.0	2.1	35563.2	4	217.5	3.0	2.0

(a) Newton iteration, $\alpha = 1$.

ℓ	*DoFs*	$time^{\mathcal{P}_1}$	$it_{\rm NL}^{\mathcal{P}_1}$	$it_{\rm O}^{\mathcal{P}_1}$	$it_{\rm MX}^{\mathcal{P}_1}$	$it_{\rm F}^{\mathcal{P}_1}$	$time^{\mathcal{P}_2}$	$it_{\rm NL}^{\mathcal{P}_2}$	$it_{\rm O}^{\mathcal{P}_2}$	$it_{\rm MX}^{\mathcal{P}_2}$	$it_{\rm F}^{\mathcal{P}_2}$
1	14,012	6.2	5	43.0	1.9	2.0	3.5	6	93.0	2.0	2.0
2	28,436	15.4	6	42.5	1.9	2.0	10.3	6	98.3	2.0	2.0
3	64,697	48.0	6	41.8	2.0	2.0	35.4	6	113.2	2.0	2.0
4	245,276	212.9	6	43.0	2.3	2.0	153.3	7	114.6	2.2	2.0
5	937,715	1308.0	6	41.7	2.8	2.0	1306.8	7	116.0	2.9	2.0
6	5,057,636	19860.7	6	44.8	3.0	2.1	26196.3	7	141.6	3.1	2.1

(b) Picard iteration, $\alpha = 0$.

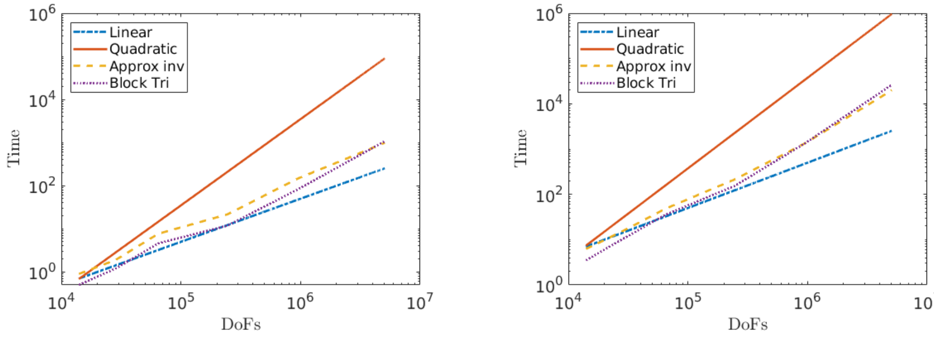
still obtain good results using the Newton iteration. We note that for the highest mesh level ($\ell = 6$) for both Tables 4 and 5 the approximate inverse preconditioner yields lower iteration counts. In fact, for the more difficult problem (Ha $= \sqrt{1000}$) the approximate inverse preconditioner is approximately four times quicker in terms of iteration counts and almost exactly two times quicker in terms of computational time than the block triangular preconditioner for $\ell = 6$. From the timing results in Figures 4 and 5, we see that the timings are approximately linear for most meshes. We do see some mild deterioration for the larger meshes. Given the good scaling in iterations and our observations about the computational cost per iteration, we would expect that a fully optimized code would yield linear a scaling in timings.



(a) Parameters: $\kappa = 1$, $\nu = 1$, $\nu_m = 1$, and Ha = 1.

(b) Parameters: $\kappa =$ 1e1, $\nu =$ 1e-1, $\nu_m =$ 1e-1, and Ha $= \sqrt{1000}$.

Fig. 4. *Timing results for cavity driven flow using the Newton iteration, $\alpha = 1$.*



(a) Parameters: $\kappa = 1$, $\nu = 1$, $\nu_m = 1$, and Ha = 1.

(b) Parameters: $\kappa =$ 1e1, $\nu =$ 1e-1, $\nu_m =$ 1e-1, and Ha $= \sqrt{1000}$.

Fig. 5. *Timing results for cavity driven flow using the Picard iteration, $\alpha = 0$.*

Computational cost of preconditioners. From the definition of the approximate inverse preconditioner it is obvious that each iteration is more expensive than the block triangular preconditioner. However, as seen in Table 4 and particularly in Table 5, the scalability of the results for the approximate inverse preconditioner produce a significant speed up in solution time for the larger problems. Table 6 shows the total number of solves and matrix-vector products for an application of the approximate inverse and the block triangular preconditioners per linear iteration for mesh level $\ell = 6$ in Table 5. In the table, "Total FGMRES operations" denotes the total number of solves or matrix-vector multiplies for one linear iteration (for the entire linear system). Thus, the column is calculated by multiplying the number of applications of each individual matrix (i.e., the inverse of a matrix or matrix-vector products) by the number of linear iterations, $it^*_{\rm O}$. For example, there are four applications of the inverse $\tilde{F}$ using the approximate inverse preconditioner (see Table 2) at each linear iteration and from mesh level $\ell = 6$ in Table 5 the number of average FGMRES is 58.5, and thus the value for "Total FGMRES operations" would be $4 \times 58.5 = 234.0$.

From the table, we see that the approximate inverse preconditioner has a smaller number of solves for both systems associated with vector-valued ($\hat{F}$ and M_X) and scalar-valued (A_p, Q_p and W) matrices. Also, the total number of matrix-vector

TABLE 6

Computational cost using $\mathcal{P}_1$ and $\mathcal{P}_2$ for mesh level $\ell = 6$ in Table 5 for one solve of the full linear system. The "Total FGMRES operations" column is calculated by multiplying the number of individual matrix operations (see Table 2) by the number of linear iterations, it_{O}^{*}. In this particular instance, the iteration times for $\mathcal{P}_1$ and $\mathcal{P}_2$ can be extracted from the row corresponding to $\ell = 6$ in Table 5 and are 58.5 and 217.5, respectively.

Linear operations	$\mathcal{P}_1 : it_{\mathrm{O}}^{\mathcal{P}_1} = 58.5$ Total FGMRES operations	$\mathcal{P}_2 : it_{\mathrm{O}}^{\mathcal{P}_2} = 217.5$ Total FGMRES operations
$\hat{F}^{-1}$	234.0	217.5
A_p^{-1}	175.5	217.5
Q_p^{-1}	175.5	217.5
M_X^{-1}	117.0	217.5
L^{-1}	117.0	217.5
$\hat{C}^T$ or C	117.0	217.5
B^T or B	234.0	217.5
G^T or G	117.0	435.0
Total execution time	17656.4	35563.2

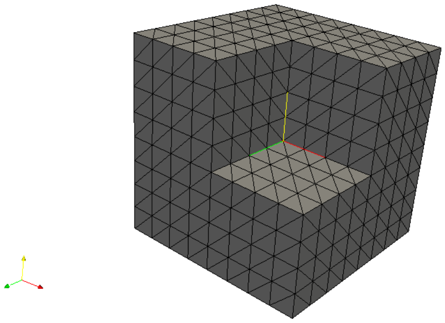


FIG. 6. *Example Fichera corner domain for mesh level $\ell = 3$.*

products is smaller for the approximate inverse preconditioner. This is reflected in the time it takes to solve this system, where the approximate inverse preconditioner is faster than the block triangular one. We also note that the iterations in Table 5 remain constant for the approximate inverse preconditioner compared to the block triangular one. Therefore, for harder problems on larger meshes it seems that the approximate inverse preconditioner is more efficient with respect to time and iterations.

6.2. Fichera corner. The second problem we consider is a smooth solution on a nonconvex domain. Specifically, the domain is a cube missing a corner, and it is known as the Fichera corner [17, section 4.3]. It is defined as

$$\Omega = (0,1)^3/[0.5,1) \times [0.5,1) \times [0.5,1).$$

An illustration of such a domain is given in Figure 6. We construct a problem whose exact solution is

$$\begin{aligned} \mathbf{u} &= \nabla \times (u_1, u_1, u_1) \quad \text{on} \quad \Omega, \\ p &= xyz(x-1)(y-1)(z-1)\exp(x) \quad \text{on} \quad \Omega, \\ \mathbf{b} &= \nabla \times (b_1, b_1, b_1) \quad \text{on} \quad \Omega, \\ r &= xyz(x-1)(y-1)(z-1)\exp(x+y+z) \quad \text{on} \quad \Omega, \end{aligned} \tag{6.2}$$

TABLE 7

Fichera corner using the approximate inverse preconditioner, $\alpha = 1$. Setup 1: $\kappa = 1e1$, $\nu = 1e-2$, $\nu_m = 1e-2$, and Ha $= \sqrt{1e5}$. Setup 2: $\kappa = 1e1$, $\nu = 1e-2$, $\nu_m = 1e-3$, and Ha $= 1000$.

ℓ	*DoFs*	Setup 1					Setup 2				
		$time^{\mathcal{P}_1}$	$it_{\rm NL}^{\mathcal{P}_1}$	$it_{\rm O}^{\mathcal{P}_1}$	$it_{\rm MX}^{\mathcal{P}_1}$	$it_{\rm F}^{\mathcal{P}_1}$	$time^{\mathcal{P}_1}$	$it_{\rm NL}^{\mathcal{P}_1}$	$it_{\rm O}^{\mathcal{P}_1}$	$it_{\rm MX}^{\mathcal{P}_1}$	$it_{\rm F}^{\mathcal{P}_1}$
1	34,250	15.6	4	29.2	2.6	2.0	102.3	7	211.7	2.0	2.0
2	57,569	30.4	4	29.2	2.7	2.0	242.3	7	237.0	2.1	2.0
3	89,612	52.9	4	28.8	2.9	2.0	440.7	7	252.1	2.1	2.0
4	332,744	232.2	4	27.8	3.0	2.0	2361.4	7	294.3	2.4	2.0
5	999,269	1026.3	4	27.8	3.0	3.0	8657.9	7	303.9	2.8	2.1
6	5,232,365	11593.5	5	28.6	3.0	3.0	111675.3	7	321.4	2.9	2.5

where

$$u_1 = x^2y^2z^2(x-1)^2(y-1)^2(z-1)^2\cos(x),$$
$$b_1 = x^2y^2z^2(x-1)^2(y-1)^2(z-1)^2\sin(y),$$

which defines the inhomogeneous Dirichlet boundary conditions and forcing terms $\boldsymbol{f}$ and $\boldsymbol{g}$.

Table 7 shows the timing and iteration results for the following two setups:

- Setup 1: $\kappa = 1e1, \nu = 1e-2, \nu_m = 1e-2$, and Ha $= \sqrt{1e5}$,
- Setup 2: $\kappa = 1e1, \nu = 1e-2, \nu_m = 1e-3$, and Ha $= 1000$.

We can see that for Setup 1 (Ha $= \sqrt{1e5}$) the outer FGMRES iterations remain constant. However, when considering Setup 2 (Ha $= 1000$) we start to see a large degradation in terms of the iteration counts. As mentioned in section 6.1, as the Hartmann number increases the numerical solution procedure becomes more challenging.

6.3. MHD generator.

The final test example considered is the more physically relevant MHD generator problem, similar to [25, section 5.2]. It describes unidirectional flow in a duct which induces an electromagnetic field. We consider the channel $[0,5]\times[0,1]\times[0,1]$. On the left and right boundaries we enforce the boundary condition $\mathbf{u} = (1,0,0)$ and on the other walls a no slip boundary condition is applied. Defining $\delta = 0.1$, $b_0 = 1$, $x_{\rm on} = 2$, and $x_{\rm off} = 2.5$, then the boundary condition associated with the magnetic unknowns is $\mathbf{n}\times\mathbf{b} = \mathbf{n}\times(0,\mathbf{b}_y,0)$, where

$$\mathbf{b}_y = \frac{b_0}{2}\left[\tanh\left(\frac{x-x_{\rm on}}{\delta}\right) - \tanh\left(\frac{x-x_{\rm off}}{\delta}\right)\right].$$

The timing and iteration results for the approximate inverse preconditioner are presented in Table 8. From the table we can see that the iteration counts decrease as the problem gets larger. Here we note that for the previous examples we were discretizating on unit cube domains, but for this example we have a uniformly triangulated mesh which is a box five times longer than it is wide or tall. We speculate that this decrease in the iterations is due to the fact the mesh size h is becoming small enough for mesh level $\ell \geq 3$, so that the fluid and magnetic viscosities are correctly captured on these finer meshes.

7. Conclusions and outlook.

Our numerical experiments demonstrate the viability and effectiveness of the approximate inverse preconditioner. We see strong scalability with respect to mesh and/or large Hartmann numbers. We also observe that our solver can handle nonconvex domains, such as the Fichera corner.

Table 8
MHD generator using the approximate inverse preconditioner with parameters $\alpha = 1, \kappa = 1, \nu = 1e-1, \nu_m = 1e-1$ *and* $Ha = 10$.

ℓ	*DoFs*	$time^{\mathcal{P}_1}$	$it_{\mathrm{NL}}^{\mathcal{P}_1}$	$it_{\mathrm{O}}^{\mathcal{P}_1}$	$it_{\mathrm{MX}}^{\mathcal{P}_1}$	$it_{\mathrm{F}}^{\mathcal{P}_1}$
1	2,199	1.5	3	172.7	1.5	1.9
2	13,809	13.3	3	108.0	1.5	2.0
3	96,957	260.8	4	105.2	1.9	2.0
4	724,725	1693.3	3	70.7	2.0	2.8
5	5,600,229	8515.7	3	68.0	2.1	2.6

Developing robust solvers for this problem is a challenging task, and we believe that our approach shows promise for tackling large-scale 3D problems with high Hartmann numbers.

Further developments will include handling variable coefficients and other real-world settings. There is a wealth of such problems; see [11]. Parallelization of the code may also be an important venue to pursue. The availability of the inverse formula and the construction of the approximate inverse present an opportunity to solve for each block in parallel without the latency that may occur for other approaches. Examining the structure of $\mathcal{P}_1$, we observe that each block column has some repetition for the sequence of systems which are solved. Therefore, one could apply each block column in parallel. This may reduce the overall computational time for the application of the approximate inverse preconditioner to one or two vector solves per block column.

Our dropping strategy focuses on mesh size considerations only. Future work may involve a different block matrix dropping strategy, which may take into account the nondimensional parameter setup (ν, ν_m or κ) of the problem. This may improve convergence for high Hartmann numbers or other challenging settings. That said, it would likely be difficult to obtain full scalability with respect to the Hartman number.

Finally, a detailed comparison between existing preconditioners for the MHD model would be highly desirable and is a high priority for us in future work. In [24, 25] the authors use quad elements, which are not supported in `FEniCS`. In [1] the authors use similar elements to the elements we use, but the solution approach is significantly different.

Acknowledgment. We are grateful to the two referees for their helpful comments and suggestions, which have greatly improved this paper.

REFERENCES

[1] J. H. Adler, T. R. Benson, E. C. Cyr, S. P. MacLachlan, and R. S. Tuminaro, *Monolithic multigrid methods for two-dimensional resistive magnetohydrodynamics*, SIAM J. Sci. Comput., 38 (2016), pp. B1–B24, https://doi.org/10.1137/151006135.

[2] F. Armero and J. C. Simo, *Long-term dissipativity of time-stepping algorithms for an abstract evolution equation with applications to the incompressible MHD and Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg., 131 (1996), pp. 41–90, https://doi.org/10.1016/0045-7825(95)00931-0.

[3] S. Balay, M. F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, and H. Zhang, *PETSc User's Manual*, Technical report ANL-95/11—Revision 3.4, Argonne National Laboratory, 2013; also available online from http://www.mcs.anl.gov/petsc.

[4] S. M. Balay, S. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, and H. Zhang, *PETSc*, 2014, http://www.mcs.anl.gov/petsc.

[5] M. Benzi, G. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137, https://doi.org/10.1017/S0962492904000212.

[6] E. C. Cyr, J. N. Shadid, R. S. Tuminaro, R. P. Pawlowski, and L. Chacón, *A new approximate block factorization preconditioner for two-dimensional incompressible (reduced) resistive MHD*, SIAM J. Sci. Comput., 35 (2013), pp. B701–B730, https://doi.org/10.1137/12088879X.

[7] P. A. Davidson, *An Introduction to Magnetohydrodynamics*, Cambridge Texts Appl. Math., Cambridge University Press, Cambridge, 2001, http://doi.org/10.1017/CBO9780511626333.

[8] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, 2nd ed., Oxford University Press, Oxford, 2014.

[9] R. Estrin and C. Greif, *On nonsingular saddle-point systems with a maximally rank deficient leading block*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 367–384, https://doi.org/10.1137/140989996.

[10] R. D. Falgout and U. Yang, hypre*: A library of high performance preconditioners*, in Proceedings of Computational Science—ICCS 2002, Lecture Notes in Comput. Sci. 2331, Springer, Berlin, 2002, pp. 632–641, http://doi.org/10.1007/3-540-47789-6_66.

[11] J.-F. Gerbeau, C. L. Bris, and T. Lelièvre, *Mathematical Methods for the Magnetohydrodynamics of Liquid Metals*, Oxford University Press, Oxford, 2006, http://www.oxfordscholarship.com/view/10.1093/acprof:oso/9780198566656.001.0001/acprof-9780198566656.

[12] C. Greif, D. Li, D. Schötzau, and X. Wei, *A mixed finite element method with exactly divergence-free velocities for incompressible magnetohydrodynamics*, Comput. Methods Appl. Mech. Engrg., 199 (2010), pp. 2840–2855, https://doi.org/10.1016/j.cma.2010.05.007.

[13] C. Greif and D. Schötzau, *Preconditioners for the discretized time-harmonic Maxwell equations in mixed form*, Numer. Linear Algebra Appl., 14 (2007), pp. 281–297, https://doi.org/10.1002/nla.515.

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

[15] I. Ipsen, *A note on preconditioning nonsymmetric matrices*, SIAM J. Sci. Comput., 23 (2001), pp. 1050–1051, https://doi.org/10.1137/S1064827500377435.

[16] D. Li, *Numerical Solution of the Time-Harmonic Maxwell Equations and Incompressible Magnetohydrodynamics Problems*, Ph.D. thesis, University of British Columbia, 2010, http://doi.org/10.14288/1.0051989.

[17] D. Li, C. Greif, and D. Schötzau, *Parallel numerical solution of the time-harmonic Maxwell equations in mixed form*, Numer. Linear Algebra Appl., 19 (2012), pp. 525–539, https://doi.org/10.1002/nla.782.

[18] L. Li, M. Ni, and W. Zheng, *A charge-conservative finite element method for inductionless MHD equations. Part* II*: A robust solver*, SIAM J. Sci. Comput., 41 (2019), pp. B816–B842, https://doi.org/10.1137/19M1260372.

[19] A. Logg, K. A. Mardal, and G. N. Wells, eds., *Automated Solution of Differential Equations by the Finite Element Method*, Lect. Notes Comput. Sci. Eng. 84, Springer, Berlin, 2012, https://doi.org/10.1007/978-3-642-23099-8.

[20] P. Monk, *Finite Element Methods for Maxwell's Equations*, Oxford University Press, Oxford, 2003.

[21] U. Müller and L. Bühler, *Magnetofluiddynamics in Channels and Containers*, Springer, Berlin, 2001, https://doi.org/10.1007/978-3-662-04405-6.

[22] M. F. Murphy, G. H. Golub, and A. J. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput., 21 (2000), pp. 1969–1972, https://doi.org/10.1137/S1064827599355153.

[23] J. C. Nédélec, *Mixed finite elements in* $\mathbb{R}^3$, Numer. Math., 35 (1980), pp. 315–341, https://doi.org/10.1007/BF01396415.

[24] 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 J. Sci. Comput., 36 (2014), pp. B930–B951, https://doi.org/10.1137/140955082.

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

[26] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469, https://doi.org/10.1137/0914028.

[27] D. Schötzau, *Mixed finite element methods for stationary incompressible magneto–hydrodynamics*, Numer. Math., 96 (2004), pp. 771–800, https://doi.org/10.1007/s00211-003-0487-4.

[28] C. Taylor and P. Hood, *A numerical solution of the Navier-Stokes equations using the finite element technique*, Comput. & Fluids, 1 (1973), pp. 73–100, https://doi.org/10.1016/0045-7930(73)90027-3.

[29] M. Wathen, *Iterative Solution of a Mixed Finite Element Discretisation of an Incompressible Magnetohydrodynamics Problem*, Master's thesis, University of British Columbia, 2014, https://doi.org/10.14288/1.0135538.

[30] M. Wathen, *Preconditioners for Incompressible Magnetohydrodynamics*, Ph.D. thesis, University of British Columbia, 2018, https://doi.org/10.14288/1.0375762.

[31] M. Wathen, C. Greif, and D. Schötzau, *Preconditioners for mixed finite element discretizations of incompressible MHD equations*, SIAM J. Sci. Comput., 39 (2017), pp. A2993–A3013, https://doi.org/10.1137/16M1098991.