Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 404 (2023) 115775
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

A multilevel block preconditioner for the HDG trace system applied to incompressible resistive MHD

Sriramkrishnan Muralikrishnan[a,*], Stephen Shannon[b], Tan Bui-Thanh[a,b], John N. Shadid[c,d]

[a] *Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, TX 78712, USA*
[b] *Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX 78712, USA*
[c] *Computational Mathematics Department, Sandia National Laboratories, P.O. Box 5800, MS 1321, Albuquerque, NM 87185, USA*
[d] *Department of Mathematics and Statistics, University of New Mexico, Albuquerque, NM 87131, USA*

Received 22 January 2022; received in revised form 25 October 2022; accepted 13 November 2022
Available online 8 December 2022

Abstract

We present a scalable block preconditioning strategy for the trace system arising from the high-order hybridized discontinuous Galerkin (HDG) discretization of incompressible resistive magnetohydrodynamics (MHD). We construct the block preconditioner with a least squares commutator (BFBT) approximation for the inverse of the Schur complement that segregates out the pressure unknowns of the trace system. The remaining velocity, magnetic field, and Lagrange multiplier unknowns form a coupled nodal unknown block for which a system algebraic multigrid (AMG) is used to approximate the inverse. The complexity of the MHD equations together with the algebraic nature of the statically condensed HDG trace system makes the choice of smoother in the system AMG part critical for the convergence and performance of the block preconditioner. Our numerical experiments show GMRES preconditioned by ILU(0) of overlap zero as a smoother inside system AMG performs best in terms of robustness, time per nonlinear iteration and memory requirements. With several transient test cases in 2D and 3D including the island coalescence problem at high Lundquist number we demonstrate the robustness and parallel scalability of the block preconditioner. Additionally for the upper block a preliminary study of an alternate nodal block system solver based on a multilevel approximate nested dissection is presented. On a 2D island coalescence problem the multilevel approximate nested dissection preconditioner shows better scalability with respect to mesh refinement than the system AMG, but is relatively less robust with respect to Lundquist number scaling. In the Appendix B, we rigorously show: (1) the uniqueness of the solution of the nonlinear MHD system for small time, (2) the convergence of the Picard iterations for each backward Euler step, and (3) the convergence of the entire time stepping procedure.

Keywords: Block preconditioners; HDG; High-order; MHD; AMG; Multilevel

1. Introduction

Incompressible visco-resistive magnetohydrodynamics (MHD) equations play an important role in modeling low Lundquist number liquid metal flows, high Lundquist number large-guide-field fusion plasmas and low flow-Mach-number compressible flows [1–3]. The mathematical basis for the continuum modeling of these systems is

* Correspondence to: Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, 52425 Jülich, Germany.
E-mail address: s.muralikrishnan@fz-juelich.de (S. Muralikrishnan).

https://doi.org/10.1016/j.cma.2022.115775
0045-7825/

the solution of the governing partial differential equations (PDEs) describing conservation of mass, momentum, and energy, augmented by the low-frequency Maxwell's equations. The multiphysics phenomena produced by this strongly coupled nonlinear system includes convective transport, momentum forces evolving from pressure gradients and Lorentz forces from the magnetic field, Alfven wave propagation, and diffusion effects from the viscosity and resistivity of the plasma system. Additionally the coupling of the conservation equations to elliptic constraints from the incompressibility assumption and the solenoidal involution property of the magnetic field add significant complexity to the block structured interactions of the unknowns. Finally the resulting response of these systems is also characterized by a wide range of spatial and temporal scales which makes efficient solution of these systems extremely challenging. In the context of time integration of these systems some form of implicitness is required to deal with the infinite pressure wave speed and to remove one or more sources of stiffness arising from diffusion and fast-waves in the MHD system. The potential benefit of implicit methods however requires robust, efficient, and scalable nonlinear and linear iterative solvers/preconditioners which can handle the highly stiff algebraic systems generated from the incompressible MHD equations.

Over the past few years there has been significant progress in the area of fully implicit robust and scalable MHD simulations as evidenced in [2–8] among others. The vast majority of current methods employ low-order stabilized FEM, mixed FEM or finite volume methods for the spatial discretization and hence require larger numbers of unknowns for high accuracy. Low-order methods have low computation-to-communication ratio which is not preferable for achieving high performance (in terms of percentage of peak performance) in modern extreme scale computing architectures.

High-order spatial discretizations are attractive in this context as they have high computation-to-communication ratio and can simulate the same problem with a reduced number of unknowns compared to low-order discretizations for similar or higher accuracy. Hybridized discontinuous Galerkin (HDG) methods introduced a decade ago are promising candidates for high-order spatial discretization due to the following reasons. It combines the important advantages of DG methods namely arbitrary high-order with compact stencil, flux upwinding on element boundaries and the ability to handle complex geometries together with the smaller and sparser linear system which involves only trace unknowns, a characteristic of hybridized methods.

The first HDG method for incompressible resistive MHD equations is proposed in [9,10] and recently HDG methods for ideal and resistive compressible MHD equations are introduced in [11]. A current challenge in the context of high-order HDG discretizations of MHD is the availability of robust, efficient and scalable iterative solvers/preconditioners and that is the goal of our work. The availability of such a solver will enable large scale 3D high-order HDG simulations of MHD on realistic geometries and this will have significant impact in many areas of plasma physics research including, but not limited to, nuclear fusion.

Towards that goal, we propose a block preconditioning strategy for trace systems [12] arising from HDG discretizations of incompressible resistive MHD [10]. Even though the concept of block preconditioning for saddle point systems has been studied by various researchers [4,6,7,13–20], it is mostly in the context of volume based low-order discretizations such as mixed FEM, stabilized FEM, finite volume and finite difference.

In this paper, the linear system coming from HDG discretization of incompressible resistive MHD [10] poses several challenges which are as follows. First, it involves only trace unknowns which live on the skeleton of the mesh and moreover they are not of the same type. Second, since it is obtained after static condensation of the volume unknowns the linear system is mostly algebraic and it is non-trivial to identify the nature of different blocks in the linear system as opposed to linear systems coming from stabilized and mixed FEM. In [21,22] block preconditioners for HDG discretization of Stokes and incompressible Navier–Stokes equations are presented. However, the authors eliminated only velocity volume unknowns by static condensation and the final linear system contains velocity and pressure trace unknowns together with the pressure volume unknowns. This has been improved in their recent work in [23] where they devised a block preconditioner for the Stokes equations by completely eliminating all the volume unknowns. Still, it is not clear whether such a preconditioner can be extended to multiphysics systems like incompressible resistive MHD, where the linear system is significantly different and complex. *To the best of our knowledge we are not aware of preconditioners for high order HDG discretizations of incompressible resistive MHD, which completely eliminate all the volume unknowns and solve the linear system with only trace unknowns and in that aspect our work is a first step in this direction.*

In this effort we employ the BFBT framework of block preconditioning first introduced in [15] for low order finite difference and finite element discretizations of incompressible Navier–Stokes equations. It has not been extended

and studied in the context of incompressible resistive MHD equations even for low order discretizations. In this regard, the novelty of our work lies in the careful choice of different components involved in the framework so that it leads to an effective preconditioner for high order HDG discretization of incompressible resistive MHD. We demonstrate this numerically by means of several 2D and 3D benchmark problems and a theoretical analysis of the current approach is left for future work.

This paper is organized as follows. In Section 2, we introduce the notations which will be followed throughout the rest of the paper. We present the incompressible resistive MHD equations and the relevant non-dimensional parameters in Section 3. Section 4 presents a Picard iteration for the MHD equations and an HDG scheme for the resulting linear MHD equations. The block structure of the HDG system is also identified in this section. We then proceed to introduce a block preconditioning strategy for the linear system in Section 5. Section 6 presents various 2D and 3D transient test cases to test the robustness and scalability of the block preconditioner. Section 7 concludes the paper with directions for future research. Appendix B contains the convergence proofs of the Picard iterations and the entire time stepping scheme.

2. Notation

In this section we introduce the common notations and conventions which will be followed in the rest of the paper. Let us partition $\Omega \in \mathbb{R}^d$, an open and bounded domain, into N_T nonoverlapping elements T_j, $j = 1, \ldots, N_T$ with Lipschitz boundaries such that $\Omega_h := \cup_{j=1}^{N_T} T_j$ and $\overline{\Omega} = \overline{\Omega}_h$. Here, h is defined as $h := \max_{j\in\{1,\ldots,N_T\}} diam\left(T_j\right)$. We denote the skeleton of the mesh by $\mathcal{E}_h := \cup_{j=1}^{N_T} \partial T_j$: the set of all (uniquely defined) interfaces e between elements. We conventionally identify $\boldsymbol{n}^-$ as the outward normal vector on the boundary ∂T of element T (also denoted as T^-) and $\boldsymbol{n}^+ = -\boldsymbol{n}^-$ as the outward normal vector of the boundary of a neighboring element (also denoted as T^+). Furthermore, we use $\boldsymbol{n}$ to denote either $\boldsymbol{n}^-$ or $\boldsymbol{n}^+$ in an expression that is valid for both cases, and this convention is also used for other quantities (restricted) on a face $e \in \mathcal{E}_h$. For the sake of convenience, we denote by $\mathcal{E}_h^\partial$ the set of all boundary faces on $\partial\Omega$, by $\mathcal{E}_h^o := \mathcal{E}_h \setminus \mathcal{E}_h^\partial$ the set of all interior faces, and $\partial\Omega_h := \{\partial T : T \in \Omega_h\}$.

For simplicity in writing we define $(\cdot,\cdot)_T$ as the $\mathcal{L}^2$-inner product on a domain $T \in \mathbb{R}^d$ and $\langle\cdot,\cdot\rangle_T$ as the $\mathcal{L}^2$-inner product on a domain T if $T \in \mathbb{R}^{d-1}$. We shall use $\|\cdot\|_T := \|\cdot\|_{\mathcal{L}^2(T)}$ as the induced norm for both cases and the particular value of T in a context will indicate which inner product the norm is coming from. We also denote the ε-weighted norm of a function u as $\|u\|_{\varepsilon,T} := \left\|\sqrt{\varepsilon}u\right\|_T$ for any positive ε. Moreover, we define $(u,v)_{\Omega_h} := \sum_{T\in\Omega_h}(u,v)_T$ and $\langle u,v\rangle_{\mathcal{E}_h} := \sum_{e\in\mathcal{E}_h}\langle u,v\rangle_e$ whose induced (weighted) norms are clear, and hence their definitions are omitted. We shall use boldface lowercase letters, e.g. $\boldsymbol{u}$, for vector-valued functions and boldface uppercase letters, e.g. $\boldsymbol{L}$, for matrices and tensors. The notations of inner products and norms are naturally extended for these cases in a component-wise manner. We use the terms "skeletal unknowns" and "trace unknowns" interchangeably and they both refer to the unknowns on the mesh skeleton. They are denoted with hats to differentiate it from the corresponding volume unknowns.

We define the gradient of a vector, the divergence of a matrix, and the outer product symbol $\otimes$ as:

$$(\nabla\boldsymbol{u})_{ij} = \frac{\partial u_i}{\partial x_j}, \quad (\nabla\cdot\boldsymbol{L})_i = \nabla\cdot\boldsymbol{L}\,(i,:) = \sum_{j=1}^{3}\frac{\partial \boldsymbol{L}_{ij}}{\partial x_j}, \quad (\boldsymbol{a}\otimes\boldsymbol{b})_{ij} = a_i b_j = \left(\boldsymbol{a}\boldsymbol{b}^T\right)_{ij}.$$

We define $\mathcal{P}^p(T)$ as the space of polynomials of degree at most p on a domain T. Similarly, $\mathcal{P}^p(e)$ denotes the polynomials of degree at most p on the mesh skeleton edge e and the extensions to vector- or matrix-valued polynomials $[\mathcal{P}^p(T)]^d$, $[\mathcal{P}^p(T)]^{d\times d}$, $[\mathcal{P}^p(e)]^d$, etc, are straightforward. We define the "jump" operator for any quantity $(\cdot)$ as $[\![(\cdot)]\!] := (\cdot)^- + (\cdot)^+$.

3. Incompressible visco-resistive MHD system

The visco-resistive, incompressible MHD equations in non-dimensional form are given by

$$\frac{\partial\boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nabla p - \frac{1}{\mathrm{Re}}\Delta\boldsymbol{u} - \kappa(\nabla\times\boldsymbol{b})\times\boldsymbol{b} = \boldsymbol{f}, \tag{1a}$$

$$\nabla\cdot\boldsymbol{u} = 0, \tag{1b}$$

$$\kappa\frac{\partial\boldsymbol{b}}{\partial t} + \frac{\kappa}{\mathrm{Rm}}\nabla\times(\nabla\times\boldsymbol{b}) - \kappa\nabla\times(\boldsymbol{u}\times\boldsymbol{b}) + \nabla r = \boldsymbol{g}, \tag{1c}$$

$$\nabla \cdot \boldsymbol{b} = 0. \tag{1d}$$

Here, $\boldsymbol{u}$ is the fluid velocity field, $\boldsymbol{b}$ is the magnetic field, p is the mechanical pressure[1] and r denotes the Lagrange multiplier which is used to enforce the solenoidality of the magnetic field constraint [3,24]. More details about this approach can be found in [3,10,24–26].

The non-dimensional parameters in (1) are as follows. $\mathrm{Re} := \rho l_0 u_0/\mu$ is the fluid Reynolds number, $\mathrm{Rm} := \mu_0 u_0 l_0/\eta$ is the magnetic Reynolds number, $\kappa := b_0^2/\rho\mu_0 u_0^2$ is the coupling parameter. Here, ρ is the fluid density, μ is the viscosity, μ_0 is the permeability of free space and η the resistivity. The characteristic length, velocity and magnetic scales are given by l_0, u_0 and b_0 respectively. We refer the readers to [1,27] for details on non-dimensional parameters in the MHD system.

We put (1) into first order form for discretizing with HDG and towards that end let us define the auxiliary variables $\boldsymbol{L}$ and $\boldsymbol{J}$ which represents the velocity gradient and curl of magnetic field respectively [10]. The first order system is given by

$$\mathrm{Re}\boldsymbol{L} - \nabla \boldsymbol{u} = \boldsymbol{0}, \tag{2a}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p - \nabla \cdot \boldsymbol{L} - \kappa(\nabla \times \boldsymbol{b}) \times \boldsymbol{b} = \boldsymbol{f}, \tag{2b}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2c}$$

$$\frac{\mathrm{Rm}}{\kappa}\boldsymbol{J} - \nabla \times \boldsymbol{b} = \boldsymbol{0}, \tag{2d}$$

$$\kappa \frac{\partial \boldsymbol{b}}{\partial t} + \nabla r - \kappa \nabla \times (\boldsymbol{u} \times \boldsymbol{b}) + \nabla \times \boldsymbol{J} = \boldsymbol{g}, \tag{2e}$$

$$\nabla \cdot \boldsymbol{b} = 0. \tag{2f}$$

We refer to $\boldsymbol{J}$ as the current density or simply the current and it should be understood in a non-dimensional sense with the characteristic value defined by $J_0 = \frac{Rm}{\kappa}\frac{b_0}{\mu_0 l_0}$.

The MHD system (2) is equipped with the following set of initial conditions

$$\boldsymbol{u}(t=0) = \boldsymbol{u}_0, \qquad \boldsymbol{b}(t=0) = \boldsymbol{b}_0. \tag{3}$$

We also need to specify boundary conditions for the fluid components, magnetic components and the Lagrange multiplier. Since it is not important for the current discussion we will defer this till Section 6 where we specify these details for each numerical experiment separately.

As shown in Appendix B, we will use the Picard nonlinear solver in our study. The following MHD system [10,24,28], a linearization of (1) about a prescribed velocity $\boldsymbol{w}$ and a prescribed magnetic field $\boldsymbol{d}$,

$$\mathrm{Re}\boldsymbol{L} - \nabla \boldsymbol{u} = \boldsymbol{0}, \tag{4a}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{w}) + \nabla p - \nabla \cdot \boldsymbol{L} - \kappa(\nabla \times \boldsymbol{b}) \times \boldsymbol{d} = \boldsymbol{f}, \tag{4b}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{4c}$$

$$\frac{\mathrm{Rm}}{\kappa}\boldsymbol{J} - \nabla \times \boldsymbol{b} = \boldsymbol{0}, \tag{4d}$$

$$\kappa \frac{\partial \boldsymbol{b}}{\partial t} + \nabla r - \kappa \nabla \times (\boldsymbol{u} \times \boldsymbol{d}) + \nabla \times \boldsymbol{J} = \boldsymbol{g}, \tag{4e}$$

$$\nabla \cdot \boldsymbol{b} = 0. \tag{4f}$$

is the first order form of (B.4), and here we keep the time derivative for the clarity of the HDG discretization that follows. We assume $\boldsymbol{w}$ to reside in $H(div, \Omega)$ and $\boldsymbol{d}$ to reside in $H(div, \Omega) \cap H(curl, \Omega)$, and both to be divergence-free. In the Picard iteration, when we take $[\boldsymbol{w}, \boldsymbol{d}]$ as the value of $[\boldsymbol{u}, \boldsymbol{b}]$ from the previous iteration this is only approximately true unless a projection as in [10,29] is performed after every time step. In the current work we do not perform this projection, and even though from our numerical results in Section 6 the Picard iteration appears to be effective without postprocessing these variables, it is not guaranteed that this will always lead to a stable solution procedure as in Appendix B. Hence it will be considered in our future work.

[1] We use the same notation p for both polynomial order and mechanical pressure as it is the most natural notation for both these entities and which one we refer to will be clear from the context.

4. An HDG discretization for incompressible MHD

In this section we present the HDG scheme first proposed in [10,28], and for simplicity we use the same notations as the exact solution in (4) for the HDG solution. Find $(\boldsymbol{L}, \boldsymbol{u}, p, \boldsymbol{J}, \boldsymbol{b}, r, \hat{\boldsymbol{u}}, \widehat{\boldsymbol{b}}^t, \widehat{r}, \rho)$ in $\boldsymbol{G}_h \times \boldsymbol{V}_h \times Q_h \times \boldsymbol{C}_h \times \boldsymbol{H}_h \times S_h \times \widehat{\boldsymbol{V}}_h \times \widehat{\boldsymbol{H}}_h^t \times \widehat{S}_h \times \mathcal{P}^0(\partial T)$ such that the local equations

$$\mathrm{Re}\,(\boldsymbol{L}, \boldsymbol{G})_T + (\boldsymbol{u}, \nabla \cdot \boldsymbol{G})_T - \langle \hat{\boldsymbol{u}}, \boldsymbol{G}\boldsymbol{n} \rangle_{\partial T} = 0, \tag{5a}$$

$$\left(\frac{\partial \boldsymbol{u}}{\partial t}, \boldsymbol{v}\right)_T - (\nabla \cdot \boldsymbol{L}, \boldsymbol{v})_T + (\nabla p, \boldsymbol{v})_T - \frac{1}{2}(\boldsymbol{u} \otimes \boldsymbol{w}, \nabla \boldsymbol{v})_T + \frac{1}{2}(\nabla \boldsymbol{u}, \boldsymbol{v} \otimes \boldsymbol{w})_T + (\nabla \times \boldsymbol{b}, \boldsymbol{v} \times \kappa \boldsymbol{d})_T + \left\langle \frac{1}{2}(\boldsymbol{w} \cdot \boldsymbol{n})\hat{\boldsymbol{u}} + \boldsymbol{S}_u(\boldsymbol{u} - \hat{\boldsymbol{u}}), \boldsymbol{v} \right\rangle_{\partial T} - [1 - \xi]\left\langle \boldsymbol{n} \times \left(\boldsymbol{b}^t - \widehat{\boldsymbol{b}}^t\right), \boldsymbol{v} \times \kappa \boldsymbol{d} \right\rangle_{\partial T} = (\boldsymbol{f}, \boldsymbol{v})_T, \tag{5b}$$

$$-(\boldsymbol{u}, \nabla q)_T + \langle \hat{\boldsymbol{u}} \cdot \boldsymbol{n}, q - \overline{q} \rangle_{\partial T} + \langle \overline{p} - \rho, \overline{q} \rangle_{\partial T} = 0, \tag{5c}$$

$$\frac{\mathrm{Rm}}{\kappa}(\boldsymbol{J}, \boldsymbol{C})_T - (\boldsymbol{b}, \nabla \times \boldsymbol{C})_T - \left\langle \boldsymbol{n} \times \widehat{\boldsymbol{b}}^t, \boldsymbol{C} \right\rangle_{\partial T} = 0, \tag{5d}$$

$$\kappa\left(\frac{\partial \boldsymbol{b}}{\partial t}, \boldsymbol{h}\right)_T + (\nabla \times \boldsymbol{J}, \boldsymbol{h})_T - (r, \nabla \cdot \boldsymbol{h})_T - (\boldsymbol{u} \times \kappa \boldsymbol{d}, \nabla \times \boldsymbol{h})_T + \langle \widehat{r}, \boldsymbol{h} \cdot \boldsymbol{n} \rangle_{\partial T} + \left\langle ([1 - \xi]\boldsymbol{u} + \xi \hat{\boldsymbol{u}}) \times \kappa \boldsymbol{d}, \boldsymbol{n} \times \boldsymbol{h} \right\rangle_{\partial T} + \left\langle \beta_t \left(\boldsymbol{b}^t - \widehat{\boldsymbol{b}}^t\right), \boldsymbol{h} \right\rangle_{\partial T} = (\boldsymbol{g}, \boldsymbol{h})_T, \tag{5e}$$

$$(\nabla \cdot \boldsymbol{b}, s)_T + \left\langle \frac{1}{\beta_n}(r - \widehat{r}), s \right\rangle_{\partial T} = 0, \tag{5f}$$

the conservation equations

$$-\left\langle [\![-\boldsymbol{L}\boldsymbol{n} + p\boldsymbol{n} + \frac{1}{2}(\boldsymbol{w} \cdot \boldsymbol{n})\boldsymbol{u} + \boldsymbol{S}_u(\boldsymbol{u} - \hat{\boldsymbol{u}}) + \kappa \boldsymbol{d} \times (\boldsymbol{n} \times \xi \boldsymbol{b})]\!], \widehat{\boldsymbol{v}} \right\rangle_e = 0, \tag{5g}$$

$$-\left\langle [\![\boldsymbol{n} \times \boldsymbol{J} + \beta_t \left(\boldsymbol{b}^t - \widehat{\boldsymbol{b}}^t\right) - \boldsymbol{n} \times ([1 - \xi]\boldsymbol{u} \times \kappa \boldsymbol{d})]\!], \widehat{\boldsymbol{h}}^t \right\rangle_e = 0, \tag{5h}$$

$$-\left\langle [\![\boldsymbol{b} \cdot \boldsymbol{n} + \frac{1}{\beta_n}(r - \widehat{r})]\!], \widehat{s} \right\rangle_e = 0, \tag{5i}$$

and the additional constraint

$$\langle \hat{\boldsymbol{u}} \cdot \boldsymbol{n}, \psi \rangle_{\partial T} = 0 \tag{5j}$$

hold for all $(\boldsymbol{G}, \boldsymbol{v}, q, \boldsymbol{C}, \boldsymbol{h}, s, \widehat{\boldsymbol{v}}, \widehat{\boldsymbol{h}}^t, \widehat{s}, \psi)$ in $\boldsymbol{G}_h \times \boldsymbol{V}_h \times Q_h \times \boldsymbol{C}_h \times \boldsymbol{H}_h \times S_h \times \widehat{\boldsymbol{V}}_h \times \widehat{\boldsymbol{H}}_h^t \times \widehat{S}_h \times \mathcal{P}^0(\partial T)$. In addition, the pressure is subject to the constraint

$$(p, 1)_{\Omega_h} = 0.$$

In the above, the notation $\overline{p}$ is defined by $\overline{p} := |\partial T|^{-1} \langle p, 1 \rangle_{\partial T}$ as the ∂T-wise average of p, and $|\partial T|$ is the length of the perimeter of element T. The new unknowns ρ which are sought in $\mathcal{P}^0(\partial T)$ represent the ∂T-wise or edge average pressure. Also, $\widehat{\boldsymbol{b}}^t$ represents the skeletal unknowns corresponding to the tangent magnetic field.

Here, the volume spaces are defined as

$$\boldsymbol{G}_h := \left\{ \boldsymbol{G} \in \left[L^2(\Omega_h)\right]^{d \times d} \ : \ \boldsymbol{G}|_T \in \left[\mathcal{P}^p(T)\right]^{d \times d}, \forall T \in \Omega_h \right\}, \tag{6a}$$

$$\boldsymbol{V}_h := \left\{ \boldsymbol{v} \in \left[L^2(\Omega_h)\right]^d \ : \ \boldsymbol{v}|_T \in \left[\mathcal{P}^p(T)\right]^d, \forall T \in \Omega_h \right\}, \tag{6b}$$

$$Q_h := \left\{ q \in L^2(\Omega_h) \ : \ q|_T \in \mathcal{P}^p(T), \forall T \in \Omega_h \right\}, \tag{6c}$$

$$\boldsymbol{C}_h := \left\{ \boldsymbol{C} \in \left[L^2(\Omega_h)\right]^{\tilde{d}} \ : \ \boldsymbol{C}|_T \in \left[\mathcal{P}^p(T)\right]^{\tilde{d}}, \forall T \in \Omega_h \right\}, \tag{6d}$$

$$\boldsymbol{H}_h := \left\{ \boldsymbol{h} \in \left[L^2(\Omega_h)\right]^d \ : \ \boldsymbol{h}|_T \in \left[\mathcal{P}^p(T)\right]^d, \forall T \in \Omega_h \right\}, \tag{6e}$$

$$S_h := \left\{ s \in L^2(\Omega_h) \ : \ s|_T \in \mathcal{P}^p(T), \forall T \in \Omega_h \right\}, \tag{6f}$$

where $\tilde{d}$ takes the value of one for 2D and three for 3D. We define the skeletal spaces as follows,

$$\widehat{\boldsymbol{V}}_h := \left\{ \widehat{\boldsymbol{v}} \in \left[L^2(\mathcal{E}_h)\right]^d \ : \ \widehat{\boldsymbol{v}}|_e \in \left[\mathcal{P}^p(e)\right]^d, \forall e \in \mathcal{E}_h \right\}, \tag{7}$$

$$\widehat{\boldsymbol{H}}^t_h := \left\{ \widehat{\boldsymbol{h}}^t \in \left[L^2(\mathcal{E}_h)\right]^{d-1} \ : \ \widehat{\boldsymbol{h}}^t|_e \in \widehat{\boldsymbol{H}}^t_h(e) \right\}, \tag{8}$$

$$\widehat{S}_h := \left\{ \widehat{s} \in L^2(\mathcal{E}_h) \ : \ \widehat{s}|_e \in \mathcal{P}^p(e), \forall e \in \mathcal{E}_h \right\}. \tag{9}$$

Here $\widehat{\boldsymbol{H}}^t_h(e)$ is a vector valued polynomial space with no normal component, defined by

$$\widehat{\boldsymbol{H}}^t_h(e) = \left\{ \sum_{i=1}^{d-1} \boldsymbol{t}^i \widehat{h}_{h,i} \ : \ \widehat{h}_{h,i} \in \mathcal{P}^p(e), \forall e \in \mathcal{E}_h \right\}, \tag{10}$$

where $\boldsymbol{t}^i$ are tangent vectors to e. The values $\xi = \frac{1}{2}$, $\beta_n = \beta_t = 1$ are chosen and the stabilization $\boldsymbol{S}_u$ is taken as

$$\boldsymbol{S}_u := \tau_t \boldsymbol{T} + \tau_n \boldsymbol{N}, \tag{11}$$

where $\boldsymbol{N} := \boldsymbol{n} \otimes \boldsymbol{n}$, $\boldsymbol{T} := -\boldsymbol{n} \times (\boldsymbol{n} \times .) = \boldsymbol{I} - \boldsymbol{N}$, $\tau_t = \frac{1}{2}\sqrt{4 + (\boldsymbol{w} \cdot \boldsymbol{n})^2}$ and $\tau_n = \frac{1}{2}\sqrt{8 + (\boldsymbol{w} \cdot \boldsymbol{n})^2}$.

The well-posedness of the local and global solvers of the scheme (5) and also the error analysis is shown in [10,28]. For this scheme the volume velocity and magnetic fields converge optimally with rate $\mathcal{O}(h^{p+1})$, whereas all the other volume variables converge with rate $\mathcal{O}(h^{p+1/2})$. The verification of this scheme for a number of prototypical MHD problems is shown in [9,10,28].

After discretizing the time derivative terms in Eq. (5) by means of some time discretization scheme e.g., backward Euler, for each Picard iteration we need to solve a linear system (see Appendix B for the details). Since the local equations are well-posed [10,28], we can express the volume unknowns in terms of the skeletal unknowns. We then use the conservation conditions to generate the global linear system which has the following block form [28]

$$\begin{bmatrix} A & -D^\top & E & G \\ D & 0 & 0 & 0 \\ F & 0 & C & J \\ H & 0 & K & L \end{bmatrix} \begin{bmatrix} \widehat{U} \\ \rho \\ \widehat{B}^t \\ \widehat{R} \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}. \tag{12}$$

In Algorithm 1, we show the overall solution procedure for the discretized incompressible resistive MHD system using HDG. The convergence of this algorithm is shown in Appendix B. We will exploit the block structure in Eq. (12) to develop a preconditioning strategy in the next section.

Algorithm 1 A solution of incompressible resistive MHD system using HDG.

Ensure: Assign initial conditions $\left[\boldsymbol{u}^0, \boldsymbol{b}^0\right] = [\boldsymbol{u}_0, \boldsymbol{b}_0]$, and $n = 1$
1: **for** $t_n < T_{end}$ **do**
2: Assign initial guess for the Picard solver based on previous time step. Set $m = 0$
3: **while** not converged **do**
4: Initialize the linear solver based on previous Picard iteration. set $m = m + 1$,
5: Solve the linear trace system (12).
6: Recover volume unknowns from trace unknowns using (5).
7: Check Picard convergence. If yes, **exit**, otherwise **continue**.
8: **end while**
9: Return $\left[\boldsymbol{u}_h^{n,m}, \boldsymbol{b}_h^{n,m}\right]$ as an approximation of $[\boldsymbol{u}(t_n), \boldsymbol{b}(t_n)]$ and set $n = n + 1$,
10: **end for**

5. A block preconditioner for the HDG system

Before delving into the construction of the block preconditioner we first briefly explain the need for it. If we want to precondition the linear system (12) using multigrid or multilevel methods, we cannot apply it directly because of the difference in the nature of the trace unknowns. The unknowns $(\widehat{U}, \widehat{B}^t, \widehat{R})$ are all nodal skeletal unknowns belonging to $\mathcal{P}^p(e)$, whereas the edge average pressure ρ is an element-wise constant and is independent

of the solution order p. Thus with standard multigrid or multilevel methods, coarsening becomes an issue unless different strategies are employed for the different types of trace unknowns which is complicated when one prefers to use black box AMG methods. This problem is also encountered in the linear systems arising from mixed finite element methods and a strategy to tackle this issue is block preconditioning. The idea is to identify and group blocks corresponding to different unknowns and use approximate block inverses for preconditioning [4,6,7,14,15,17,18]. Here we use similar techniques to develop a preconditioner for the linear system (12).

Towards this goal we first rewrite Eq. (12) into the saddle point form as follows

$$\begin{bmatrix} A & E & G & -D^\top \\ F & C & J & 0 \\ H & K & L & 0 \\ D & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \widehat{U} \\ \widehat{B}^t \\ \widehat{R} \\ \rho \end{bmatrix} = \begin{bmatrix} F_1 \\ F_3 \\ F_4 \\ F_2 \end{bmatrix}. \tag{13}$$

Denoting the 3×3 block corresponding to the unknowns $(\widehat{U}, \widehat{B}^t, \widehat{R})$ as $\mathcal{F}$ and $[D \quad 0 \quad 0]$ as $\mathcal{B}$ we can write the above matrix as

$$\begin{bmatrix} \mathcal{F} & -\mathcal{B}^\top \\ \mathcal{B} & 0 \end{bmatrix}. \tag{14}$$

For a 2×2 block matrix such as (14) its block inverse (assuming $\mathcal{F}^{-1}$ and $\mathcal{S}^{-1}$ exists) can be written as [30,31]

$$\begin{bmatrix} \mathcal{F} & -\mathcal{B}^\top \\ \mathcal{B} & 0 \end{bmatrix}^{-1} = \begin{bmatrix} \mathcal{F}^{-1} & \mathcal{F}^{-1}\mathcal{B}^\top\mathcal{S}^{-1} \\ 0 & \mathcal{S}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -\mathcal{B}\mathcal{F}^{-1} & I \end{bmatrix}, \tag{15}$$

where $\mathcal{S} := \mathcal{B}\mathcal{F}^{-1}\mathcal{B}^\top$ is the Schur complement. Now, when we use the block upper triangular matrix of the inverse (15) as a right preconditioner for the saddle point matrix (14) we get

$$\begin{bmatrix} \mathcal{F} & -\mathcal{B}^\top \\ \mathcal{B} & 0 \end{bmatrix} \begin{bmatrix} \mathcal{F}^{-1} & \mathcal{F}^{-1}\mathcal{B}^\top\mathcal{S}^{-1} \\ 0 & \mathcal{S}^{-1} \end{bmatrix} = \begin{bmatrix} I & 0 \\ \mathcal{B}\mathcal{F}^{-1} & I \end{bmatrix}. \tag{16}$$

All the eigenvalues of the preconditioned matrix have the value 1, and hence with a Krylov subspace method such as GMRES with at most two iterations we can solve the system [14,15,17,30].

However, the problem with this ideal preconditioner is that we need inverses of $\mathcal{F}$ and $\mathcal{S}$ which are expensive to compute. Hence a natural idea is to use approximations of these inverses in the construction of the preconditioner. First, let us consider the approximation for the inverse of the Schur complement matrix $\mathcal{S}$. We will follow the approach shown in [14,15,17] for incompressible Navier–Stokes equations and use the following approximation $(\tilde{\mathcal{S}}^{-1})$ for the inverse of the Schur complement

$$\tilde{\mathcal{S}}^{-1} = \left(\mathcal{B}\mathcal{B}^\top\right)^{-1} \mathcal{B}\mathcal{F}\mathcal{B}^\top \left(\mathcal{B}\mathcal{B}^\top\right)^{-1}. \tag{17}$$

This is called BFBT approximation[2] of the inverse Schur complement in the literature [14,15,17,18] and has been successfully used in the context of Stokes and incompressible Navier–Stokes equations discretized with finite volume, finite difference and mixed FEM methods. In those cases, $\mathcal{B}\mathcal{B}^\top$ corresponds to the Poisson operator and instead of taking inverses one typically use a geometric multigrid or AMG cycle [14,15,17]. Also, for approximating $\mathcal{F}^{-1}$, AMG or geometric multigrid cycles are used [14,15,17]. Thus these choices provide a cheap approximation to the ideal preconditioner in Eq. (15) for a certain class of problems and discretizations. Scaled versions of the BFBT approximation are proposed in [14,17,18,32] in order to improve the robustness in certain cases.

The HDG linear system for skeletal unknowns in Eq. (12) which comes after static condensation of volume unknowns is mostly algebraic in nature. Thus it is more difficult to analyze than stabilized or mixed FEM linear systems, especially for complex systems like MHD. Moreover, our grouping of unknowns in Eq. (13) to arrive at the saddle point form is also algebraic. We can see that already in the $\mathcal{F}$ matrix, which contains part of the fluid block (without pressure) and the magnetic block. We base our choices for $\left(\mathcal{B}\mathcal{B}^\top\right)^{-1}$ and $\mathcal{F}^{-1}$ considering this aspect.

In the context of HDG discretization of the incompressible MHD system, $\mathcal{B}\mathcal{B}^\top$ corresponds to a matrix of size $N_T \times N_T$ which is much smaller in size compared to $\mathcal{F}$. It is independent of the solution order p and the bandwidth is also small since it corresponds to the edge-average pressure which is piecewise constant per element. Furthermore, in

[2] The naming comes from the notations used conventionally for the blocks and the middle term sandwiched between the two $\left(\mathcal{B}\mathcal{B}^\top\right)^{-1}$.

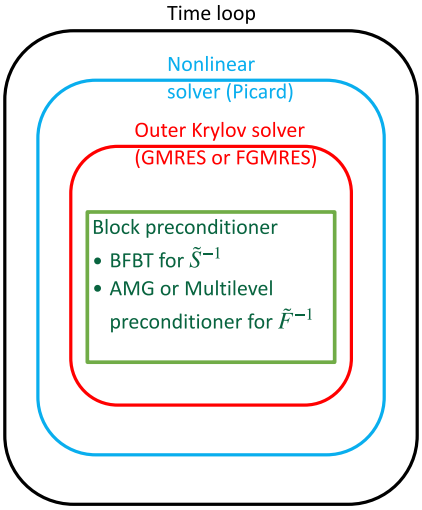


Fig. 1. Schematic of the entire solution algorithm including block preconditioner for incompressible resistive MHD system using HDG.

high order HDG we try to minimize the number of elements while increasing the order to increase the computation-to-communication ratio, leading to small $\mathcal{B}\mathcal{B}^\top$ matrix. Thus, in this work we use a parallel sparse direct solver `SuperLU_DIST` [33] for taking inverse of this matrix in the BFBT approximation. Though not considered in the current work, it may be worthwhile to investigate replacing this part with preconditioned conjugate gradient solver of lenient tolerance to improve scalability and time to solution.

For approximating $\mathcal{F}^{-1}$, we use one v-cycle of a system projection AMG solver from the ML library [34] of Trilinos project [35]. Similar to [3], we order the unknowns in the nodal block such that the degrees-of-freedom within each node appear consecutively. This helps to preserve the coupling between different variables during coarsening. The ordering of unknowns within each node is $(\widehat{U}, \widehat{B}^t, \widehat{R})$, i.e. velocity trace unknowns are first, followed by the tangent magnetic field and then the Lagrange multiplier. Other orderings are also possible and it can affect the performance of the AMG cycle. Since it is beyond the current scope of this manuscript, we do not consider them here. We use non-smoothed and uncoupled aggregation with a sparse direct solver on the coarsest level as in [3]. More details about the aggregation strategy employed in AMG can be found in [3]. The choice of smoothers inside the AMG cycle is critical. The mixed parabolic–hyperbolic nature of MHD together with the algebraic nature of the HDG trace system demands us to use strong smoothers such as ILU(0) and GMRES instead of standard smoothers like Jacobi and Gauss–Seidel. This is based on our numerical experiments in Section 6.

Another choice for $\mathcal{F}^{-1}$ is the multilevel preconditioner introduced in [36]. Since the coarse solver in the multilevel preconditioner is obtained by an approximation of the nested dissection direct solver, it can also provide an effective approximation to $\mathcal{F}^{-1}$. We compare the performance of the multilevel preconditioner to AMG in approximating $\mathcal{F}^{-1}$ in Section 6.4. In our numerical studies we observed that the performance of the scaled BFBT approximation is very similar to the non-scaled one (17) and hence we use the latter in all of numerical experiments. In summary we use a right preconditioner

$$\begin{bmatrix} \tilde{\mathcal{F}}^{-1} & \tilde{\mathcal{F}}^{-1}\mathcal{B}^\top\tilde{\mathcal{S}}^{-1} \\ 0 & \tilde{\mathcal{S}}^{-1} \end{bmatrix}, \tag{18}$$

where $\tilde{\mathcal{F}}^{-1}$ is one v-cycle of AMG or the multilevel algorithm (Algorithm 1 in [36]) and the approximate Schur complement inverse $\tilde{\mathcal{S}}^{-1}$ is given by the BFBT approximation (17). The computation of $\left(\mathcal{B}\mathcal{B}^\top\right)^{-1}$ in the BFBT approximation is carried out by the parallel direct solver `SuperLU_DIST`. A schematic of the entire solution Algorithm 1, including the block preconditioner, is shown in Fig. 1.

6. Numerical results

In this section we test the performance of the block preconditioner on some of the transient test cases in incompressible resistive MHD considered in [3,4,6]. In particular, we consider 2D and 3D versions of the island

coalescence problem, hydromagnetic Kelvin–Helmholtz (HMKH) instability and hydromagnetic lid-driven cavity problems. We use quadrilateral elements in 2D and hexahedral elements in 3D. For time integration, we use the backward Euler time stepping for all the 3D test cases and stiffly-accurate, singly diagonal, five-stage, fourth-order diagonally implicit Runge–Kutta (SDIRK) method of [37] for all the 2D test cases. The reason for using low-order backward Euler for the 3D test cases is because high-order time integrators involve many stages and thus require more computational time. Since we are interested in the performance of preconditioners we elected to just use backward Euler as it will not have significant influence on our results. For the nonlinear solver we employ the Picard iteration scheme with a stopping criterion based on the discrete norm of the solution update vector given in [3]:

$$\sqrt{\frac{1}{N_u}\sum_{i=1}^{N_u}\left[\frac{|\Delta\chi_i|}{\varepsilon_r|\chi_i|+\varepsilon_a}\right]^2} < 1.$$

Here, χ_i is any variable, $\Delta\chi_i$ is its corresponding update in the Picard iteration, N_u is the total number of unknowns and ε_a and ε_r are the absolute and relative tolerances which are set to 10^{-6} and 10^{-4}, respectively. For the stopping criterion of the linear solver, apart from the HMKH test case, we set the norm of the residual divided by the norm of the right-hand side of the Picard linear system to be 10^{-6}. For the HMKH test case, it turns out we need a stricter tolerance of 10^{-9} to make the Picard iterations converge.[3] In addition we also checked that all our simulations satisfy the criteria $\|\mathcal{R}^m\|/\|\mathcal{R}^0\| < 10^{-2}$ as in [3], where $\mathcal{R}^m$ is the residual at the mth Picard iteration and $\mathcal{R}^0$ is the initial residual. This ensures sufficient reduction in the relative nonlinear residual norm.

For all the parallel results, we implemented our algorithms on top of the deal.II FEM library version 9.0.1 [38,39]. The weak scaling studies are conducted in the Knights Landing (KNL) nodes of the Stampede2 supercomputer at the Texas Advanced Computing Center. Each node of KNL consists of 68 intel Xeon Phi 7250 1.4 GHz processors with 4 threads per core. It has 96 GB DDR4 RAM along with 16 GB high speed MCDRAM which acts as L3 cache. We have used only MPI parallelism even though our deal.II code has task based parallelism using thread building blocks (TBB) in addition to MPI. The reasons for this choice are: (i) to have memory locality and avoid memory contention which may complicate the weak scaling studies (ii) the main focus of this study, which is the linear solver part, uses ML from Trilinos which does not have threads support.

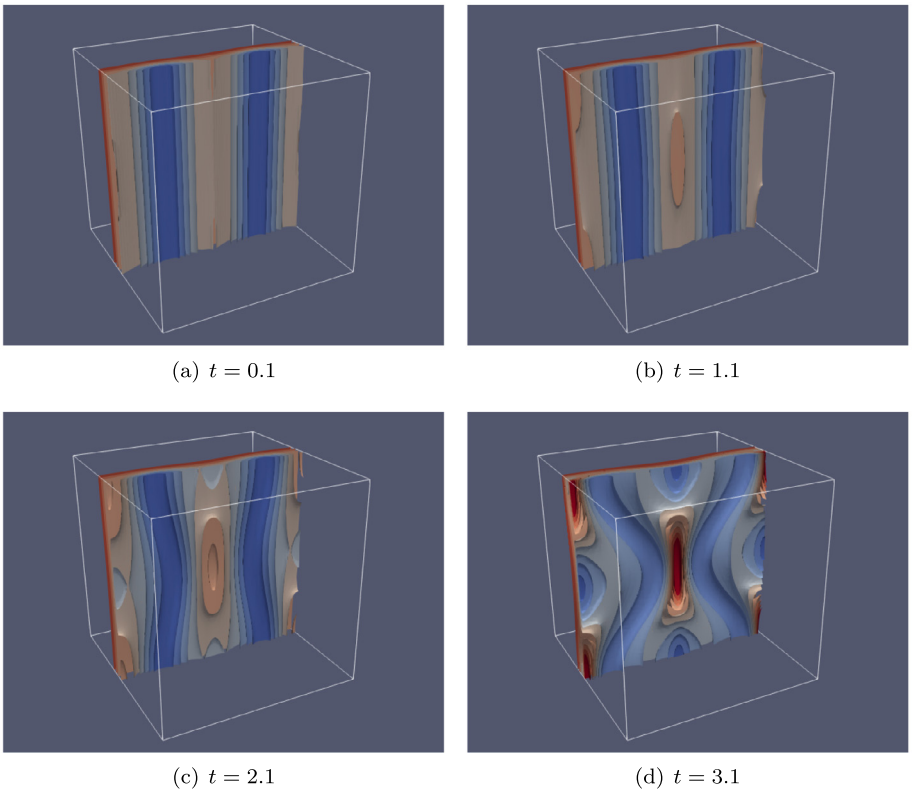
6.1. Island coalescence

Magnetic reconnection is a fundamental phenomenon by which a magnetic field changes its structure and is accompanied by conversion of magnetic field energy into plasma energy and transport [2,3]. Many physical phenomena which occurs in outer space such as solar flares, coronal mass ejections involve magnetic reconnection as an important driving mechanism. It is also important in a laboratory scenario, especially in fusion experiments to understand and control plasma disruptions which can lead to loss in plasma confinement and also damages to the machine. More details about magnetic reconnection can be found in [1,40,41].

While magnetic reconnection is important for its physical significance, it is also characterized by disparate spatial and temporal scales which serves as an ideal test bed for testing the robustness of our preconditioners/solvers. In this section we consider a specific reconnection problem which is the island coalescence studied in [2,3]. It initially consists of two islands embedded in a Harris current sheet as shown in Fig. 2(a). A perturbation to the initial configuration and the combined magnetic field causes the center of the islands (referred as the o-points) to move towards each other and eventually coalesce to form one island. When the reconnection happens, the islands form a 'x' structure in the center of the domain as shown in Figs. 2(d) and 5. In what follows we will briefly describe the settings of this problem and then evaluate the performance of our preconditioner in this case.

The domain is $[-1, 1]^d$, where d is the spatial dimension. The boundary conditions are periodic in the x-direction (the left and right faces) and also in the z-direction (the back and front faces) for $d = 3$. On the top and bottom faces in the y-direction, for the magnetic part, perfect conducting boundary conditions—described by zero normal magnetic field $\boldsymbol{b}\cdot\boldsymbol{n} = 0$ and zero tangential electric field $\boldsymbol{n}\times\boldsymbol{E} = 0$—are applied. For the fluid part on the top and bottom faces, mirror boundary conditions—described by zero normal velocity $\boldsymbol{u}\cdot\boldsymbol{n} = 0$ and zero shear

[3] In this case we do not divide the norm of the residual by the norm of the right hand side as its value is greater than 1 and we needed a more accurate linear solver solution for the convergence of the Picard solver.



(a) $t = 0.1$ (b) $t = 1.1$

(c) $t = 2.1$ (d) $t = 3.1$

Fig. 2. Evolution of the z-component of the current (J_z) with time. The contours of J_z show a highly kinked state at $t = 3.1$ due to the perturbation in the $z-$direction.

stress—are enforced. The Lagrange multiplier r is set as zero on all the boundaries. We refer the readers to [28] for an application of these boundary conditions in the context of HDG discretizations.

The initial condition consists of zero fluid velocity ($\boldsymbol{u}_0 = \boldsymbol{0}$), and the magnetic field given by

$$\boldsymbol{b}_0 = \left(\frac{\sinh(2\pi y)}{\cosh(2\pi y) + \varepsilon \cos(2\pi x)}, \frac{\varepsilon \sin(2\pi x)}{\cosh(2\pi y) + \varepsilon \cos(2\pi x)}, 0 \right),$$

where in 2D the first two components of the field values are used. Here, ε refers to the width of the island and we choose it to be 0.2 as in [3]. In order for the initial configuration to be in equilibrium, a forcing of $\boldsymbol{g} = \nabla \times \boldsymbol{J}_0$ is used where

$$\boldsymbol{J}_0 := \boldsymbol{J}(t = 0) = \left(0, 0, -\frac{2\pi\kappa\left(1 - \varepsilon^2\right)}{\mathrm{Rm}\left(\cosh(2\pi y) + \varepsilon \cos(2\pi x)\right)^2} \right).$$

For the momentum equation zero forcing ($\boldsymbol{f} = \boldsymbol{0}$) is selected. To set the islands into motion in a reproducible manner rather than relying on the accumulation of round-off error, an initial perturbation of

$$\delta\boldsymbol{b}_0 = \sigma \left(\frac{\pi}{2} \cos(\pi x) \sin\left(\frac{\pi y}{2}\right) \cos(\pi z), -\pi \sin(\pi x) \cos\left(\frac{\pi y}{2}\right) \cos(\pi z), 0 \right),$$

is applied with $\sigma = 10^{-3}$, which sets the magnitude of perturbation [28]. For 2D, the first two components are used without the $z-$term. As described in [28] we choose the characteristic velocity as Alfven speed which gives $\kappa = 1$. Also in all numerical experiments we set the fluid Reynolds number and magnetic Reynolds number (which is Lundquist number in this case) equal to each other.

In Appendix A, we show a validation of the high-order HDG method with the conventional finite element [42] and finite volume methods [43] as well as the theoretical Sweet–Parker model scaling in the literature for the 2D island coalescence problem.

Now let us consider the 3D island coalescence problem and choose a Lundquist number $S = 10^3$, and a $16 \times 12 \times 14$ clustered mesh of order $p = 5$. Fig. 2, shows the evolution of the $z-$component of the current by taking a slice of the contour plot at mid-plane $z = 0$. As can be seen, the perturbation in the $z-$direction causes the current contours to bend and results in a highly kinked state as evidenced in Fig. 2(d). Our results show good agreement (visually) with the results in [2,3].

Now we compare the performance of three preconditioners for this problem in a weak scaling sense, i.e. we increase the problem size proportionally with the increase in number of processors so that the number of elements per processor remains constant. We take 4^3, 8^3, 16^3 and 32^3 elements and 2, 16, 128 and 1024 processors respectively so that we have 32 elements per core. For $p = 4$, this amounts to 32,000 primal volume degrees of freedom ($\boldsymbol{u}$, $\boldsymbol{b}$, p and r) and roughly 15,000 trace degrees of freedom per core respectively. A time stepsize of 0.1 is selected and all the results are averaged over six time steps. The nonlinear Picard solver takes on average $3.2 - 4$ iterations in all the cases.

First of the preconditioners is a one level domain decomposition method with an incomplete factorization sub-domain solver with zero fill-in (ILU(0)) and overlap of one[4] (denoted as DD, ILU(0) in Fig. 3 and let us refer to this as DD with ILU(0)). The other two preconditioners are two variants of the block preconditioner given in Eq. (18) with BFBT approximation for $\tilde{\mathcal{S}}^{-1}$ and one AMG v-cycle for $\tilde{\mathcal{F}}^{-1}$. The difference between them is the smoothing inside AMG, in one of them we use the ILU(0) smoother of overlap one (denoted as BFBT+AMG, ILU(0) in Fig. 3 and let us refer to this as BFBT+AMG with ILU(0)), whereas in the other one we use the GMRES right preconditioned by ILU(0) of overlap zero[5] as smoother (denoted as BFBT+AMG, GMRES in Fig. 3 and let us refer to this as BFBT+AMG with GMRES). The reason for these non-standard choice of smoothers is, classical smoothers like Jacobi, Chebyshev and Gauss–Seidel did not result in converging iterations. This is also observed in [3,44] for linear systems arising from stabilized FEM discretization of MHD. Thus it indicates that strong smoothers are needed for AMG cycles applied to linear systems coming from incompressible resistive MHD.

We perform three pre- and three post-smoothing steps (in the case of GMRES smoother, these denote the number of inner iterations), whereas in the DD with ILU(0) preconditioner we perform three smoothing steps. The outer iterations are performed with GMRES for both the DD with ILU(0) and the BFBT+AMG with ILU(0) preconditioners. For the BFBT+AMG with GMRES, we use flexible GMRES (FGMRES) [45] for the outer iterations due to the nonlinear nature of the inner iterations. In both cases, similar to [3] we use non-restarted versions as it may result in degradation of convergence. All the preconditioners are applied as right preconditioners to the outer GMRES or FGMRES.

We compare the performance in terms of average iterations per Picard step and average time per Picard step in Fig. 3. From Fig. 3(a) for solution order $p = 4$, we can see that the iterations of DD with ILU(0) preconditioner increases much faster than the other two block preconditioners and this is due to the lack of coarse solvers. Similar results are also observed in [3] for the stabilized FEM discretization. The average linear iterations per Picard step for the two block preconditioners are less than the one level domain decomposition preconditioner. The growth with number of unknowns/processors (here the number of unknowns refers to the number of trace unknowns only) is also very mild. Between ILU(0) and GMRES smoothers we can see that the GMRES smoother takes slightly more iterations than the ILU(0).

In Fig. 3(b), we compare the average time per Picard step for the three preconditioners and again DD with ILU(0) preconditioner takes more time compared to the block preconditioners. However, the large difference in iteration counts evidenced in Fig. 3(a) between DD with ILU(0) and BFBT+AMG with ILU(0) is not much reflected in the timing in Fig. 3(b). The reason is that ILU(0) with overlap one spends most time in the setup cost and very little time in the application of the preconditioners. Since the setup cost in both DD with ILU(0) and BFBT+AMG with ILU(0) are very similar, the difference comes only from the application of the preconditioners, which is high for DD with ILU(0) due to larger number of iterations. The scenario for the GMRES smoother is exactly opposite with very little time spent for the setup and most of the cost coming from the application or solving part of the preconditioner. This reflects in the timing results in Fig. 3(b) with BFBT+AMG with GMRES smoother taking the least time (in spite of its iteration counts bit higher than ILU(0) smoother) and more than two times faster than the other two preconditioners. We remark that even though the number of volume unknowns per core is the same in this weak scaling study, the number of trace unknowns per core is not exactly the same and the cases of 16 cores,

[4] Here, overlap one means each processor will include both its own set of rows and the rows corresponding to its non-zero columns.

[5] Here, overlap zero means each processor will include only its own set of rows and thus no communication is needed.

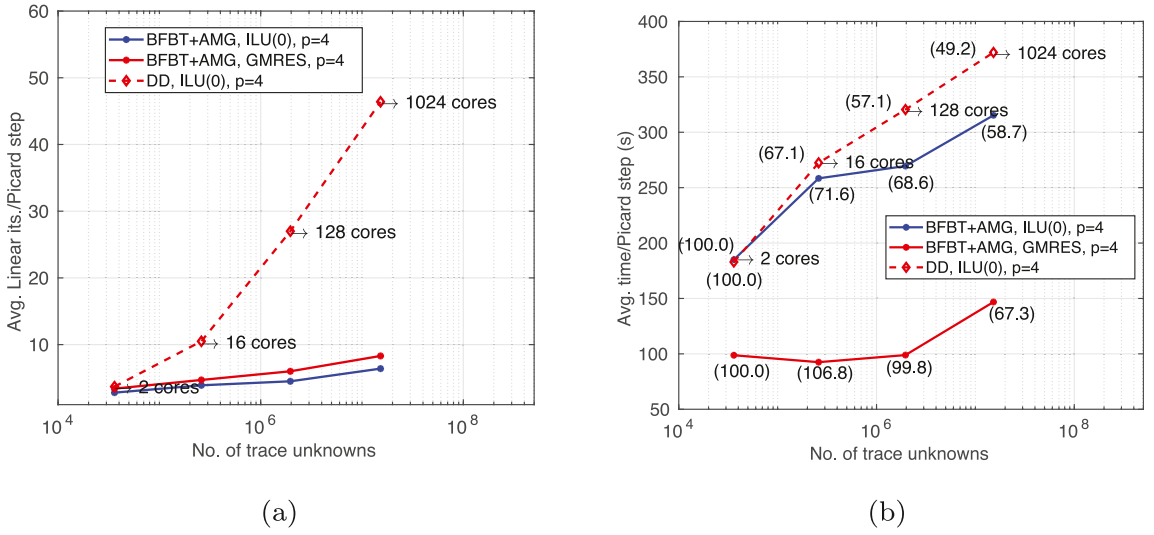


Fig. 3. 3D island coalescence problem: weak scaling study of average iterations per Picard step (left) and average time per Picard step (right) for three preconditioners and solution order $p = 4$. The markers in BFBT+AMG with ILU(0) and GMRES curves also represent the same number of processors as those in DD with ILU(0) curve. The values within parentheses in the right figure represent weak scaling parallel efficiencies.

Table 1
3D island coalescence problem. Robustness study with respect to Lundquist number S: iterations and time represent the average number of iterations taken per Picard step and the average time per Picard step, respectively. "*" indicates the preconditioners did not reach the specified tolerance for the maximum number of iterations which is taken as 1000 for DD with ILU(0) and 200 for BFBT+AMG with ILU(0). The time step in this study is $\Delta t = 0.05$ and the mesh, solution order and number of cores are 32^3, $p = 4$ and 1024, respectively. The average number of Picard iterations taken is approximately 3.

S	DD, ILU(0)		BFBT+AMG, ILU(0)		BFBT+AMG, GMRES	
	Iterations	Time (s)	Iterations	Time (s)	Iterations	Time (s)
10^3	94	489.4	3.5	287.8	5.3	124.4
10^4	*	*	*	*	6	127.6
10^5	*	*	*	*	7.5	128.2
10^6	*	*	*	*	7.6	131.2

128 cores and 1024 cores have slightly less number of trace unknowns per core compared to the two-core case. This is the reason why in Fig. 3(b) the time per Picard step for GMRES smoother decreases slightly for 16 cores case even though the iterations increase a bit compared to that for 2 cores. However, asymptotically this difference gets negligible and hence does not affect much our inferences.

In Table 1 we compare the robustness of the preconditioners with respect to Lundquist number S. Clearly, BFBT+AMG with GMRES is more robust than the other two preconditioners and takes nearly constant number of iterations and time per Picard step for the parameters mentioned in Table 1.

Now, in addition to computational time and robustness, another important aspect to take into account is memory. ILU(0) smoothers with overlap one require significantly more memory and this is much pronounced at high solution orders. In our numerical experiments we found that, due to memory limitations in the KNL nodes of Stampede2, for solution orders greater than four we can use only 8 cores per node or less, even though the system has 68 cores per node. The GMRES smoother (GMRES preconditioned by ILU(0) with overlap zero) on the other hand has less memory requirements than the ILU(0) smoother with overlap one, and this is the reason we test only BFBT+AMG with GMRES smoother for solution orders $p = 5, 6$ in Fig. 4.

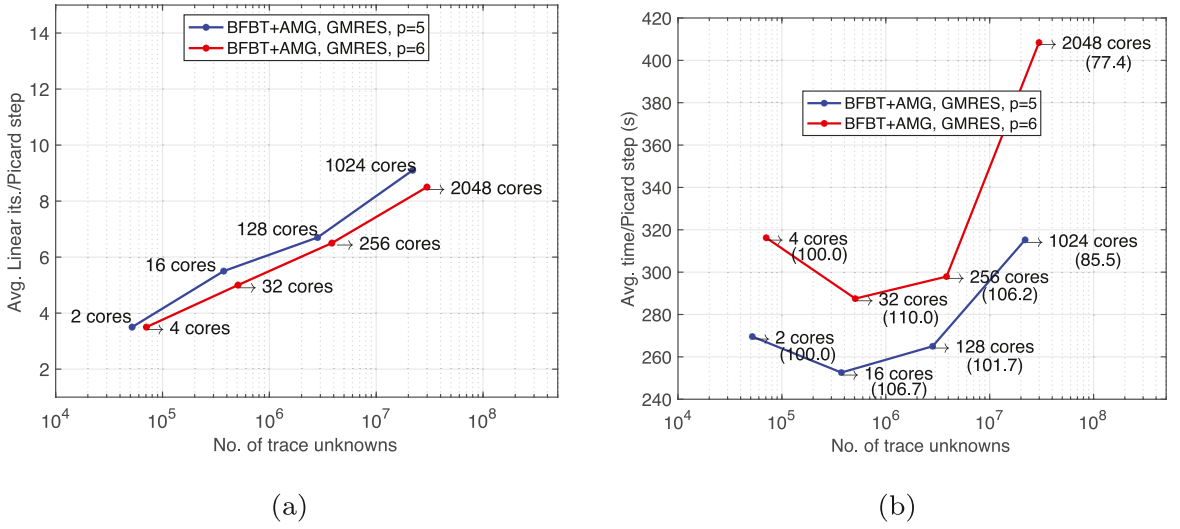


Fig. 4. 3D island coalescence problem. BFBT+AMG with GMRES smoother: weak scaling study of average iterations per Picard step (left) and average time per Picard step (right) for solution orders $p = 5, 6$. The values within parentheses in the right figure represent weak scaling parallel efficiencies.

In Fig. 4, the average number of iterations and time per Picard step are shown for the BFBT+AMG with GMRES smoother for solution orders $p = 5, 6$. For $p = 5$, we have 32 elements per core and this amounts to 55,296 primal volume degrees of freedom and roughly 22,000 trace degrees of freedom per core respectively. For $p = 6$, with the selected numbers of cores we have 16 elements per core which leads to 43,904 primal volume degrees of freedom and roughly 15,000 trace degrees of freedom per core respectively. The iteration counts are very similar to those observed for $p = 4$ in Fig. 3(a) with a mild increase with respect to mesh refinements. The time per Picard step however, for $p = 4, 5$ with 1024 cores and $p = 6$ with 2048 cores show a significant increase compared to 128 and 256 cores, respectively. This is a consequence of two things: (i) increase in iteration count; (ii) coarsening in AMG. Since the increase in iteration count is very small its contribution to the increase in solve time is only minor. The major component comes from the coarsening in AMG which is explained as follows. We use uncoupled aggregation strategy in AMG which means each processor aggregates its local grid and no aggregate can span between two processors. Thus the number of aggregates can never be less than the number of processors. This presents a problem at high processor counts as it results in fewer levels (3 or 4) in the AMG hierarchy with larger problem sizes on the coarsest level. Since we use a centralized serial sparse direct solver on the coarsest level, the timing increases. This trend is observed in the other two numerical experiments and especially in Fig. 12 for the lid driven cavity problem. In that case we have almost constant iteration counts, which is reflected in the timing till 256 processors. For both 1024 and 2048 processors we see some increase in timing.

We found that this issue can be alleviated to a certain extent by using the parallel sparse direct solver `SuperLU_DIST`. However, since its scalability is also limited and the size of the matrix is not large enough to hide the communication costs a better solution to this issue might be to perform repartitioning during aggregation as in [3]. Repartitioning the coarse operators, will help us to create more levels in the multigrid hierarchy with smaller problem sizes in each level by employing only a processor subset on some coarse levels. Thus the coarsest level where we employ the serial sparse direct solver will have a significantly smaller size and hence can be solved with less computational time. Other than the repartitioning for uncoupled aggregation there are other AMG coarsening strategies such as uncoupled MIS (maximal independent set) which must be explored and these will be the objective of a future work.

In Tables 2 and 3 we study the strong scalability of BFBT+AMG with GMRES smoother for solution orders $p = 4, 6$. We see a good strong scaling performance and the efficiency improves with increase in the solution order for similar number of trace unknowns/core due to increased computation-to-communication ratio. For the strong scaling study we used the Skylake nodes of Stampede2 supercomputer and since it has a higher clock speed

Table 2
3D island coalescence problem. Strong scaling study for BFBT+AMG with GMRES smoother for solution order $p = 4$. The simulation is carried out in the Skylake nodes of Stampede2 supercomputer.

$N_T = 32,768$, $N_{trace} = 15.2M$, p = 4, $\Delta t = 0.1$, averaged over 6 time steps				
#cores	Time/Picard [s]	N_T/core	Trace DOF/core	Efficiency [%]
128	211.6	256	118.4 K	–
256	126.8	128	59.2 K	83.4
512	64.2	64	29.6 K	82.4
1024	36.3	32	14.8 K	73
2048	19.4	16	7.4 K	68
4096	16.6	8	3.7 K	40
6144	14.9	5.3	2.5 K	29.6

Table 3
3D island coalescence problem. Strong scaling study for BFBT+AMG with GMRES smoother for solution order $p = 6$. The simulation is carried out in the Skylake nodes of Stampede2 supercomputer.

$N_T = 32,768$, $N_{trace} = 29.8M$, p = 6, $\Delta t = 0.1$, averaged over 6 time steps				
#cores	Time/Picard [s]	N_T/core	Trace DOF/core	Efficiency [%]
512	290	64	58.4 K	–
1024	129.5	32	29.2 K	112
2048	82.6	16	14.6 K	87.7
4096	46.6	8	7.3 K	77.7
8192	39.4	4	3.7 K	46

(2.1 GHz) the average time per Picard step is $4-5$ times less than that for KNL nodes (1.4 GHz). Of the three preconditioners the BFBT+AMG with GMRES smoother is faster, more robust and less memory consuming. *For these reasons, we will consider only BFBT+AMG with GMRES in the subsequent numerical studies. Moreover, since the strong scaling performance is not very much problem dependent we only study weak scaling.*

We would like to comment on the percentage of timings for different parts of the HDG method. In all our finest resolution simulations, the global solver for the trace system accounts for more than 80% of the total time, the local Schur complements together with the assembly of the linear system account for roughly 10% and finally the local recovery of volume unknowns from the trace unknowns contributes to approximately 5%. All other components contribute only to a negligible portion of the total time. In the block preconditioner for the global solver, the AMG preconditioner which is used to approximate $\mathcal{F}^{-1}$ is the dominant cost followed by the BFBT approximation for the inverse of the Schur complement.

We now use the BFBT+AMG with GMRES preconditioner to simulate a challenging problem which is 2D island coalescence at high Lundquist numbers. Here as an example, we consider a Lundquist number of $S = 10^7$. The significance of this problem is that at high Lundquist numbers a thin current sheet that forms at the center of the domain during reconnection breaks and gives rise to small structures called plasmoids [46–48]. The dynamics of this problem is very different from the low Lundquist number cases where the islands gradually approach each other and coalesce to a single island.

The settings of this experiment are as follows. While for the scaling studies we used a fixed time stepsize of $\Delta t = 0.1$, in this case we use an adaptive time step where we initially start with a stepsize of 0.05 and if the Picard solver fails to converge in 20 iterations we cut the stepsize by half and try again. This is essential for capturing the highly nonlinear evolution of this problem and we observed a stepsize of 0.0015625 during the plasmoid formation and evolution stages. Once the time stepsize is cut by half we do not increase it later in the simulation, while increasing the stepsize may help in reducing the overall time of the simulation our focus here is to simulate the correct physics and also test the robustness of our preconditioner. The domain is discretized by 128×128 uniform elements and solution order $p = 9$. For time stepping we use the five-stage fourth-order SDIRK scheme. The problem is run on 512 cores in the Skylake nodes of Stampede2 supercomputer.

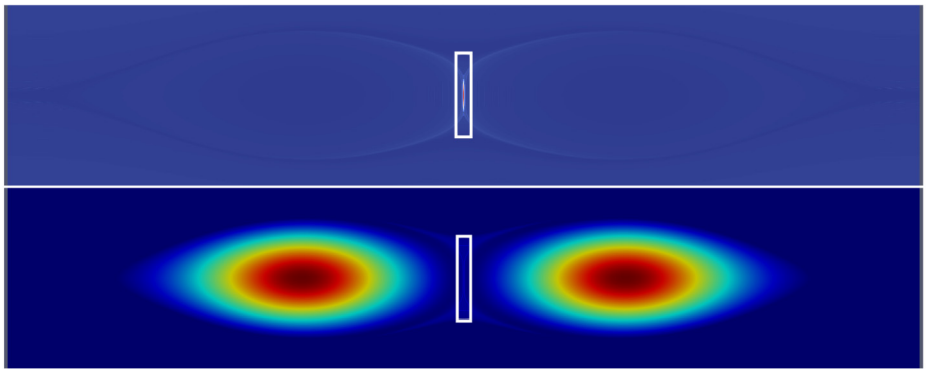
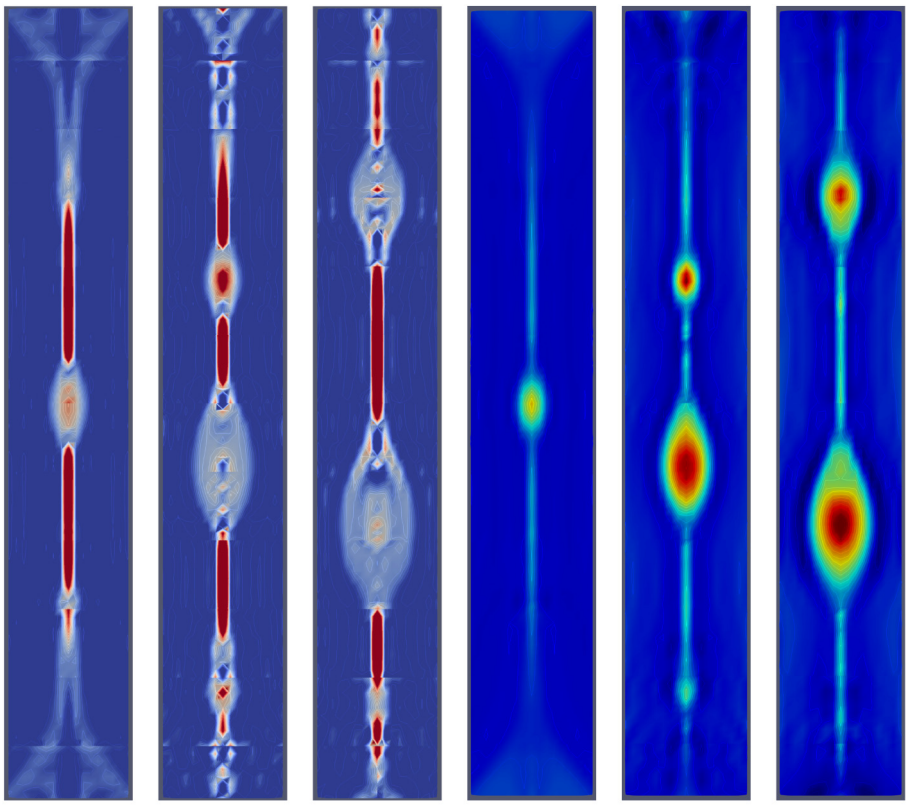


Fig. 5. 2D island coalescence problem: Current (top) and pressure (bottom) at time $t = 5.2$ showing the coalescence of the islands. In Fig. 6 we will focus on the box region marked in the central portion of the above figures.



(a) $t = 5.85$ (b) $t = 6.22$ (c) $t = 6.3$ (d) $t = 5.85$ (e) $t = 6.22$ (f) $t = 6.3$

Fig. 6. 2D island coalescence problem: Current ((a), (b) and (c)) and pressure ((d), (e) and (f)) plots at the indicated times showing the formation and evolution of plasmoids with time.

Fig. 5 shows the formation of x-structure in the center of the domain at $t = 5.2$ similar to the 3D problem in Fig. 2(d). We zoom into the box marked at the center of the domain in Fig. 5 to see clearly the formation and evolution of plasmoids. Fig. 6 shows the current and pressure within the box marked in Fig. 5 at different times. From Fig. 6 we can clearly see bubble like structures which are called plasmoids emanating from the breakdown of the thin current sheet and moving in the vertical direction. We can at least see two prominent plasmoids: one moving downwards and the other upwards in Fig. 6. However, there are many tiny plasmoids which continuously

appear and merge as time proceeds. This problem clearly shows the multiscale nature of the magnetic reconnection phenomenon in both space and time. Indeed, the island widths are of $\mathcal{O}(0.1)$, whereas from the Sweet–Parker model the width of the current sheet and hence the size of plasmoids should be $\mathcal{O}(L/S^{1/2})$ which for this case is $\mathcal{O}(6 \times 10^{-4})$.[6] Similarly the time scale in which the island moves is almost an order of magnitude larger than the plasmoid time scales. The linear solver iterations for this problem is mostly less than 10 for majority of the Picard steps with some exceptions taking between $15-20$. Thus the BFBT+AMG preconditioner with GMRES smoother seems to be fairly robust for this challenging problem which involves multiscale physics in space and time. The current plots in Fig. 6 show the need for $hp-$adaptivity or application of filters/artificial viscosity. In [8], the authors have used adaptive mesh refinement to resolve the plasmoids in the context of high-order stabilized FEM. We will investigate more into adaptivity, slope limiting, filtering, and artificial viscosity strategies for our high-order scheme in the future works.

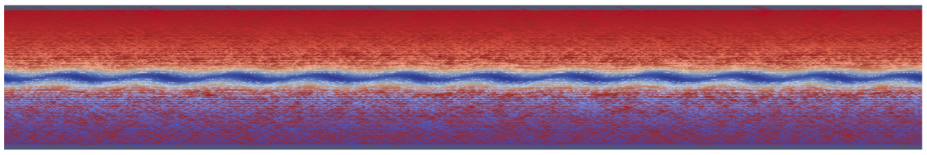
6.2. Hydromagnetic Kelvin–Helmholtz instability

In this section we consider the 2D and 3D versions of hydromagnetic Kelvin–Helmholtz (HMKH) instability problem studied in [3,4,6]. This problem also involves magnetic reconnection similar to the island coalescence problem in the previous section. The domain we consider is $[0,4] \times [-2,2]$ in 2D and $[0,4] \times [-2,2] \times [0,2]$ in 3D. The initial condition consists of two counter flowing conducting fluids of constant velocities given by $\boldsymbol{u}_0(x, y \geq 0, z) = (1,0,0)$ and $\boldsymbol{u}_0(x, y < 0, z) = (-1,0,0)$ and a Harris sheet magnetic field defined by $\boldsymbol{b}_0(x,y,z) = (B_0 \tanh(y/\delta), 0, 0)$. We choose a zero forcing for both fluid and magnetic equations. Similar to the island coalescence problem, the boundary conditions are periodic in x- and z-directions (3D). On the top and bottom boundaries, the fluid boundary conditions are zero normal velocity and zero shear stress, while the magnetic field is defined by the Harris sheet in the initial condition. The Lagrange multiplier r is set to 0 on all the boundaries. Following [3], we take : $\kappa = 1$, $\mathrm{Re} = \mathrm{Rm} = 10^4$, $B_0 = 0.3333$ and $\delta = 0.1$. These values along with $\rho = \mu_0 = 1$ gives a super Alfvenic Mach number of $M_A = u/u_A = 3$. If $M_A > 1$ then the magnetic field is not strong enough to suppress the instabilities and the shear layer is Kelvin–Helmholtz unstable. Thus the initial disturbances eventually grow to form vortices which roll up and merge as time goes on.

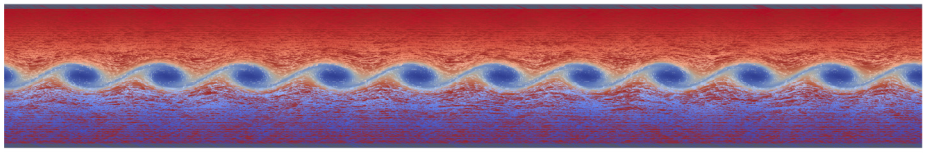
First, we consider the 2D HMKH problem discretized in a 128×128 uniform mesh with solution order $p = 6$. Time stepsize of $\Delta t = 0.001$ is chosen for the five stage fourth order SDIRK method. As mentioned in the beginning of the numerical section we need a stricter tolerance of 10^{-9} in the linear solver to make the Picard iterations converge for this problem. The preconditioner is BFBT+AMG with GMRES smoother and the outer iterations are carried out by FGMRES. In Fig. 7, the evolution of vorticity with time is shown along with the magnetic vectors marked by arrows. Vortices are rolled up to form the familiar cat-eye pattern, and the magnetic vectors bend and follow the fluid evolution as time proceeds. In Fig. 8, we show the 3D HMKH solution with a $20 \times 24 \times 7$ mesh clustered around the region of solution and solution order $p = 5$. An initial time stepsize of $\Delta t = 0.025$ is selected for the backward Euler time stepping, and the adaptive time stepping procedure described in the previous section is employed. Similar to the 2D solutions, we see the rollup of vortices followed by the magnetic vectors.

Next, we study the weak scaling performance of the BFBT+AMG with GMRES smoother for the 3D HMKH problem. To that end, we used a fixed time stepsize of $\Delta t = 0.01$ in the backward Euler time stepping and the results were averaged over six time steps. Four different meshes $8 \times 8 \times 10$, $16 \times 16 \times 10$, $32 \times 32 \times 10$ and $64 \times 64 \times 10$ corresponding to 32, 128, 512 and 2048 processors are considered respectively. Thus the number of elements per processor in this case is 20. For $p = 4, 5$, this leads to 20,000 and 34,560 primal volume degrees of freedom and roughly 10,000 and 14,000 trace degrees of freedom per core respectively. The fluid CFL number for high-order methods with LGL quadrature points is given by $CFL_u = up^2\Delta t/h_{min}$ as in [49,50]. The fluid velocity scale $u = 1$ in this case and hence the CFL_u ranges from 0.8 to 2.56 for solution order $p = 4$ and 1.25 to 4 for $p = 5$. The Alfvenic CFL is given by $CFL_{Alfven} = u_A p^2 \Delta t/h$ and since $u_A = u/3$ in this case hence $CFL_{Alfven} = CFL_u/3$. The Picard solver took approximately 2 iterations in all cases. Fig. 9 shows the average iteration counts and time per Picard step. Since the tolerance of the iterative solver in this case (10^{-9}) is stricter than for the island coalescence problem (10^{-6}) we see an increase in overall iteration counts. However, the number of iterations still lies mostly between $10-20$ which is moderate given the tight tolerance of 10^{-9} and the maximum

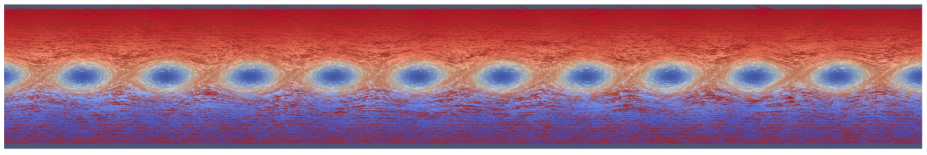
[6] In our simulations in Fig. 6, we observe plasmoids of $\mathcal{O}(0.01)$ and hence it is under-resolved and calls for mesh refinement.



(a) $t = 1.51$



(b) $t = 1.73$



(c) $t = 2.61$

Fig. 7. 2D HMKH problem: Vorticity plots at the indicated times along with the magnetic vectors (marked as arrows). The magnetic vectors are scaled by their magnitude and are colored by the x-component of the magnetic field (b_x). The red arrows on the top represent the positive values of b_x and blue arrows on the bottom represent the negative values. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

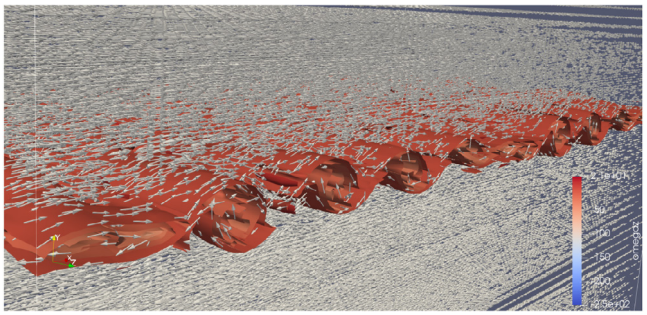


Fig. 8. 3D HMKH problem: $Z-$component of vorticity contours along with magnetic vectors at time $t = 2.0625$. The magnetic vectors are not colored and scaled in this figure to improve visibility. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

fluid CFL numbers of 2.56 and 4 for $p = 4$ and 5, respectively. The average time per Picard step reflects the trend in the iteration count together with the decrease in scalability in the 2048 processors regime. This is due to the lack of repartitioning during coarsening in the AMG as discussed in the previous section on the island coalescence problem.

6.3. Lid driven cavity

In this section we consider a hydromagnetic version of the classical lid driven cavity problem. The settings of this problem follow [6] closely. Even though we simulated the 2D version of this problem, here we present the results only for the 3D problem. The domain is $[-0.5, 0.5]^3$, with no slip boundary conditions of $\boldsymbol{u} = \boldsymbol{0}$ applied on all the walls except the top one where we apply a velocity of $\boldsymbol{u} = (1, 0, 0)$ to drive the flow. For the magnetic field we set the tangential component on each wall as $\boldsymbol{b} \times \boldsymbol{n} = (-1, 0, 0) \times \boldsymbol{n}$ which acts from right to left. The Lagrange multiplier r is set as zero on all the boundaries. Both initial conditions and forcings are taken as zero.

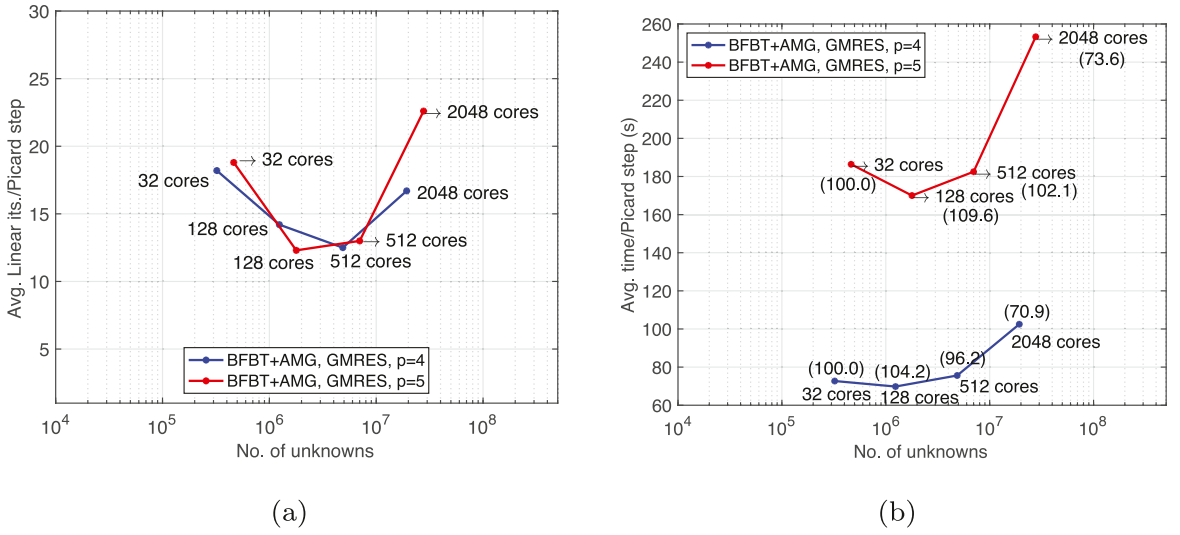
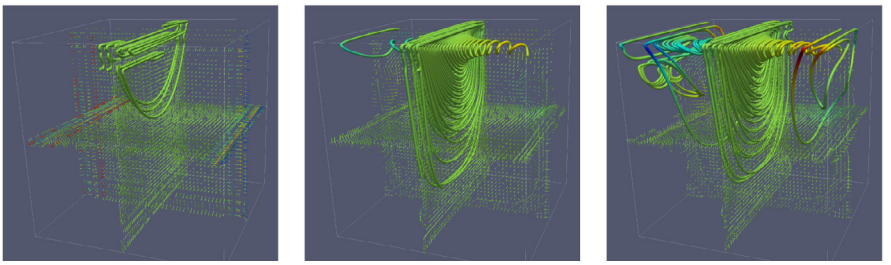


Fig. 9. 3D HMKH problem. BFBT+AMG with GMRES smoother: weak scaling study of average iterations per Picard step (left) and average time per Picard step (right) for solution orders $p = 4, 5$. The values within parentheses in the right figure represent weak scaling parallel efficiencies.



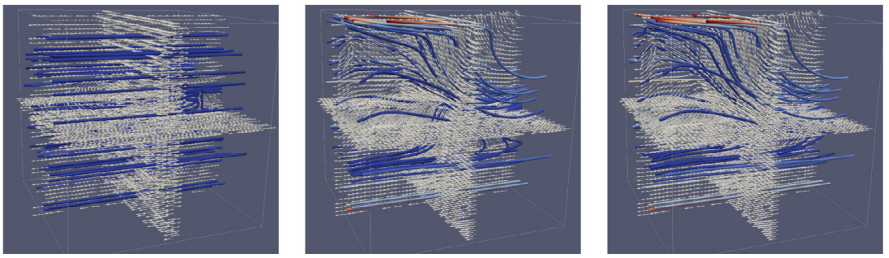
(a) $t = 0.0125$ (b) $t = 5.075$ (c) $t = 5.6625$

Fig. 10. 3D lid driven cavity problem: Evolution of the streamlines and velocity vectors with time. Both the streamlines and the velocity vectors are colored by the $z-$component of velocity, and the velocity vectors are not scaled. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We choose the following parameters: $\kappa = 1$, $\mathrm{Re} = \mathrm{Rm} = 1000$ which correspond to a Hartmann number of $Ha = \sqrt{\kappa}\mathrm{ReRm} = 1000$.

First, we consider a 8^3 uniform mesh, solution order $p = 6$ and an initial time stepsize of $\Delta t = 0.0125$. The linear and nonlinear solver tolerances and the stopping criteria are same as the ones used for the island coalescence problem. Fig. 10 shows the evolution of the streamlines together with the velocity vectors in the $x = 0$, $y = 0$ and $z = 0$ planes. The presence of the third dimension allows the streamlines to curl in the $z-$direction in Figs. 10(a) and 10(b), which gives rise to complex velocity patterns. In Fig. 11 the corresponding magnetic field lines along with the magnetic vectors are shown. Since the applied tangential magnetic field goes from right to left, so do the magnetic lines and the vectors with some bending caused by the interaction with the fluid components.

For the weak scaling study we consider a fixed time stepsize of $\Delta t = 0.0025$ for solution orders $p = 4, 5$, and $\Delta t = 0.001$ for $p = 6$. The meshes we consider are 4^3, 8^3, 16^3 and 32^3 corresponding to 4, 32, 256 and 2048 processors, respectively, with 16 elements per core. For $p = 4, 5$ and 6, this leads to 16,000, 27,648 and 43,904 primal volume degrees of freedom and roughly 7500, 11,000 and 15,000 trace degrees of freedom per core respectively. In this case $CFL_u = CFL_{Alfven}$ as $u = u_A = 1$. It varies in the range $0.16 - 1.28$ for $p = 4$, $0.25 - 2$ for $p = 5$, and $0.14 - 1.15$ for $p = 6$. The number of Picard iterations in all the cases varies from 3.5 to 4.7. In Fig. 12 we show the average iterations and time per Picard step for the BFBT+AMG preconditioner with GMRES



(a) $t = 0.0125$ (b) $t = 5.075$ (c) $t = 5.6625$

Fig. 11. 3D lid driven cavity problem: Evolution of the magnetic field lines and magnetic vectors with time. The magnetic vectors are not colored nor scaled, and the field lines are colored by b_x. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

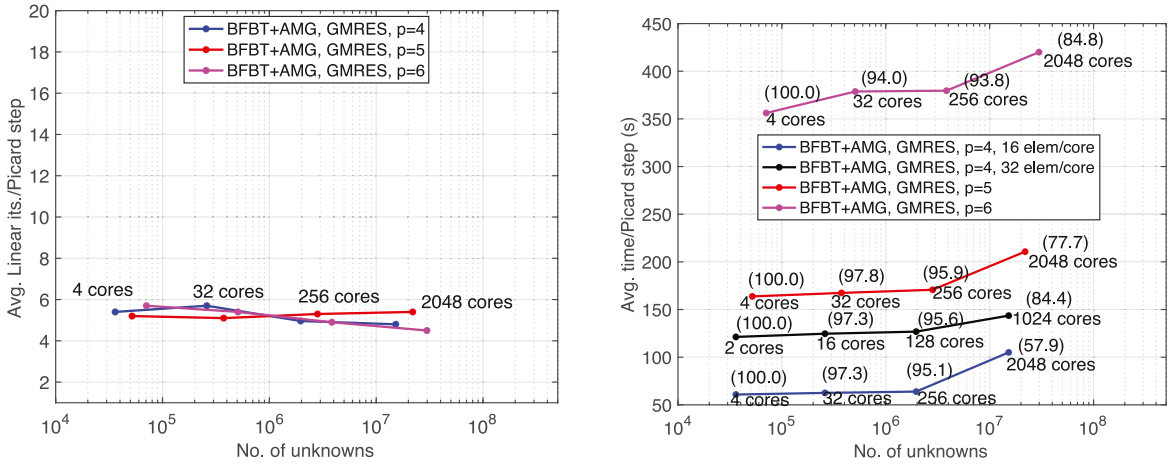


Fig. 12. 3D lid driven cavity problem. BFBT+AMG with GMRES smoother: weak scaling study of average iterations per Picard step (left) and average time per Picard step (right) for solution orders $p = 4, 5, 6$. The values within parentheses in the right figure represent weak scaling parallel efficiencies.

smoother and the results are averaged over six time steps. Compared to the previous two experiments, this case has almost constant iteration count for all meshes and all solution orders.[7] In Fig. 12(b) we again see some increase in time per Picard step for 2048 cores. This is due to the coarsening in AMG as explained in Section 6.1 on island coalescence and other components of the block preconditioner, as we have a flat iteration count. We consider another example with solution order $p = 4$ and with 2, 16, 128 and 1024 cores in Fig. 12(b). This case has 32 elements, 32,000 primal volume degrees of freedom and roughly 15,000 trace degrees of freedom per core respectively. It shows better weak scaling performance than the other case with 2048 cores for the finest mesh size. This once again highlights the need for investigation of better coarsening strategies in the AMG preconditioner as explained in Section 6.1 in order to achieve better scalability at large number of cores.

6.4. BFBT+Multilevel preconditioner

The AMG preconditioner used to approximate $\mathcal{F}^{-1}$ in the block preconditioner dominates the cost and in this section we consider the multilevel preconditioner introduced in [36] as an alternative. The problem we consider

[7] We also tested with $\Delta t = 0.01$ for $p = 4$ and observed that the iterations of the linear solver remains more or less the same as that for $\Delta t = 0.0025$ whereas the average Picard iterations are 5.3, 6.5, 7.5 and 9.3 corresponding to the four mesh sizes.

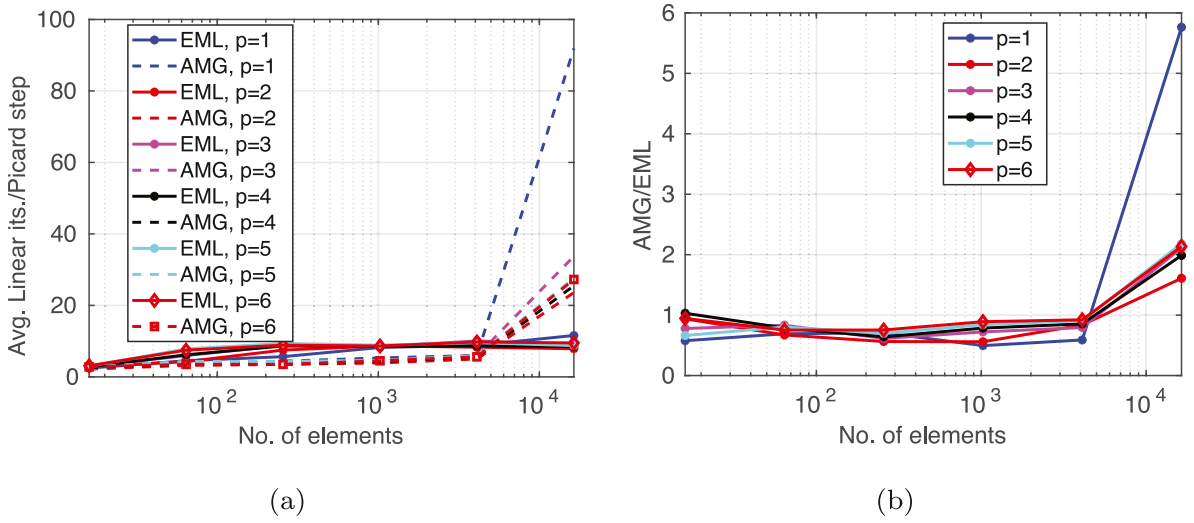


Fig. 13. 2D island coalescence problem: Comparison of AMG and multilevel (EML) preconditioners as the mesh is refined in h and p. In both cases we use the BFBT approximation for the inverse of the Schur complement. On the left is the number of average iterations per Picard step and on the right is the ratio of average time taken per Picard step for BFBT+AMG over BFBT+EML.

is the 2D island coalescence studied in Section 6.1. We will also compare the performance of the iterative solver against nested dissection direct solver as the latter is one of the popular alternatives for 2D problems.

In [36] the multilevel preconditioner is introduced for problems with scalar trace unknowns and here we will extend the idea in a natural manner to vector-valued trace unknowns. Similar to AMG, we order the unknowns such that on each edge all the trace unknowns (except edge average pressure) corresponding to the first node are ordered first, followed by the unknowns in the second node and so on. The ordering of unknowns within each nodal point is $(\widehat{U}, \widehat{B}^t, \widehat{R})$.

We then apply one v-cycle of the iterative multilevel algorithm (Algorithm 1 in [36]) for $\tilde{\mathcal{F}}^{-1}$. For the coarse-solver in the multilevel algorithm we use the enriched multilevel approach (EML) due to its robustness and better performance compared to the non-enriched version [36]. By means of several numerical experiments we observed that no pre-smoothing and one post-smoothing in the block-Jacobi part of the multilevel algorithm gives the least number of outer GMRES iterations and we use that setting in all cases. We also observed that increasing the number of smoothing steps generally leads to more number of iterations in this case, and post-smoothing performs better than pre-smoothing. Since the nodal block ($\mathcal{F}$) in the HDG discretization of MHD is a non-symmetric mixed parabolic–hyperbolic system, and for the most part it is purely algebraic, we cannot in general expect better performance by increasing the number of smoothing steps. The number of iterations also depends on the interaction between the fine scale solver (block-Jacobi) and the coarse-scale solver (EML) in terms of capturing the overall spectrum, and we do not have a clear understanding of this yet. Though out of the scope of this work, Fourier analysis could guide us to select the appropriate number of smoothing steps as well as the choice of the fine scale solver.

Having described the specifications of the multilevel algorithm for $\tilde{\mathcal{F}}^{-1}$ we now compare BFBT+multilevel preconditioner (referred as BFBT+EML) to BFBT+AMG with GMRES smoother. The outer iterations are carried out with GMRES in the case of BFBT+multilevel preconditioner and FGMRES for BFBT+AMG with GMRES smoother. For the time discretization we use the backward Euler time stepping with a fixed time stepsize of $\Delta t = 0.1$. The results are averaged over six time steps for all the cases except for 128×128 and 256×256 meshes where we average over three time steps. We choose a Lundquist number of $S = 10^3$. We run the AMG solver serially in this case to compare with the serial implementation of the multilevel method. So this study is mainly to assess the numerical scalability of multilevel and AMG preconditioners with respect to mesh refinement, solution order, and Lundquist number.

Fig. 13(a) shows the average number of iterations for both the AMG and the multilevel (EML) preconditioners together with the BFBT approximation for the inverse Schur complement in the block preconditioning (18) as the mesh and solution order are refined. In calculating this average we have omitted the iteration counts for the first

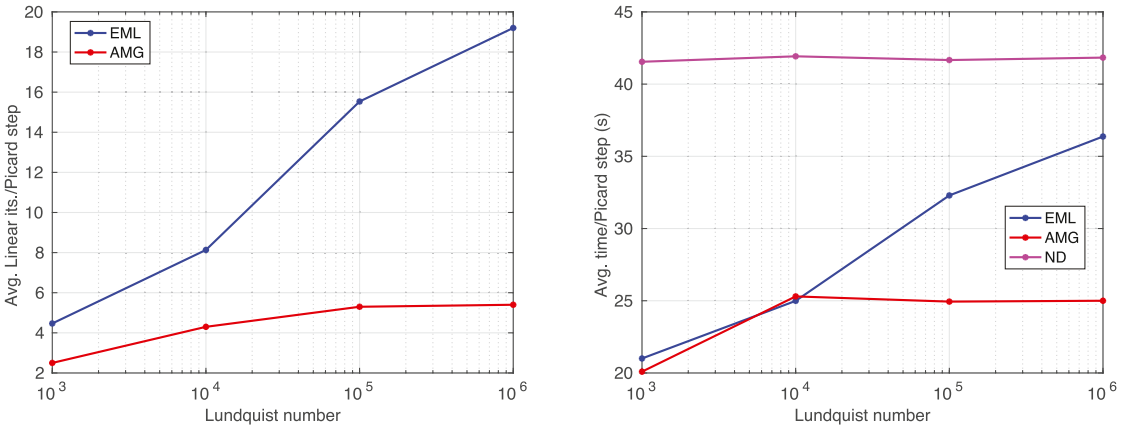


Fig. 14. 2D island coalescence problem: Comparison of AMG and multilevel (EML) preconditioners with increase in Lundquist number S. In both cases we use the BFBT approximation for the inverse of the Schur complement. On the left is the average iterations per Picard step and on the right is the average time taken per Picard step. We also show timings for the nested dissection (ND) direct solver on the right for comparison.

Picard step in the first time step. This is because the iteration counts for both EML and AMG solvers are higher for this case than the rest of the steps as we start from a zero initial guess. Both the algorithms show almost constant iteration count until 64×64 mesh with AMG taking slightly less iteration counts than EML. However, for 128×128 mesh (the last marker) and all solution orders, AMG shows a sudden increase in iteration count whereas the performance of EML remains almost the same. Thus the EML solver is more scalable than AMG in terms of mesh refinements in the settings of this experiment. Fig. 13(b) shows the ratio of average time taken per Picard step for BFBT+AMG over the time taken for BFBT+EML and it reflects the trend observed in the iteration counts in Fig. 13(a). We get approximately $2 - 6$ times speedup with BFBT+EML over BFBT+AMG for the 128×128 mesh.

Now we test the robustness of AMG and EML preconditioners with respect to Lundquist number for this problem. To that end, we consider a 64×64 mesh with $p = 6$ and $\Delta t = 0.05$. The results are again averaged over six time steps. Fig. 14 shows the average iteration counts and time per Picard step for Lundquist numbers in the range $[10^3, 10^6]$. We can see that the BFBT+AMG preconditioner is more robust with respect to the increase in Lundquist numbers than the BFBT+EML preconditioner. Nevertheless, the growth in iterations for the BFBT+EML preconditioner is still moderate, and in all the cases both the preconditioners take less time than the nested dissection (ND) direct solver. This is important as it is very hard to beat the timings of the ND direct solver with iterative solvers for complex PDEs such as MHD in 2D. In our future work, we like to improve the robustness of the BFBT+EML preconditioner by exploring other fine scale solvers in addition to block-Jacobi.

Finally, we will compare the BFBT+EML preconditioned GMRES with the nested dissection (abbreviated as ND and it is one of the fast direct solvers in 2D). We use the UMFPACK library [51] for this purpose and order the trace system matrix in Eq. (13) in nested dissection manner. In Fig. 15 we compare the average time taken per Picard step with mesh refinements for the BFBT+EML preconditioned GMRES and the nested dissection direct solver with different solution orders. The dominant cost in the BFBT+EML preconditioned solver comes from the factorization involved in EML. From the theoretical complexities derived in [36], the iterative solver at all orders has almost linear scaling with the number of elements whereas the ND direct solver shows an asymptotic scaling of $\mathcal{O}({N_T}^{3/2})$.

In Fig. 16 we show the speedup of BFBT+EML preconditioned GMRES relative to ND and we can see at all solution orders, when the number of elements is greater than 10^3 (after 5 uniform refinements), the iterative solver is faster than ND. We obtain a maximum speedup of approximately 23 for $p = 1$ and 256×256 elements, beyond which the ND solver ran into out of memory in the factorization step. In terms of memory for $p = 1$ and 256×256

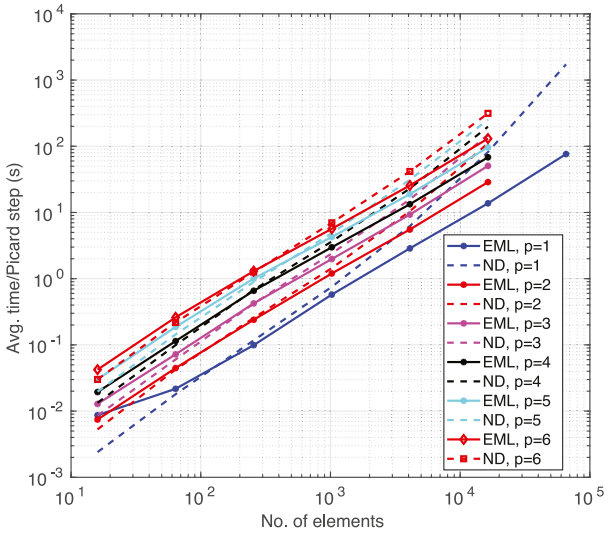


Fig. 15. 2D island coalescence problem: Comparison of scaling of BFBT+EML preconditioned GMRES (denoted as EML) and ND direct solver with number of elements for solution orders $p = 1 - 6$.

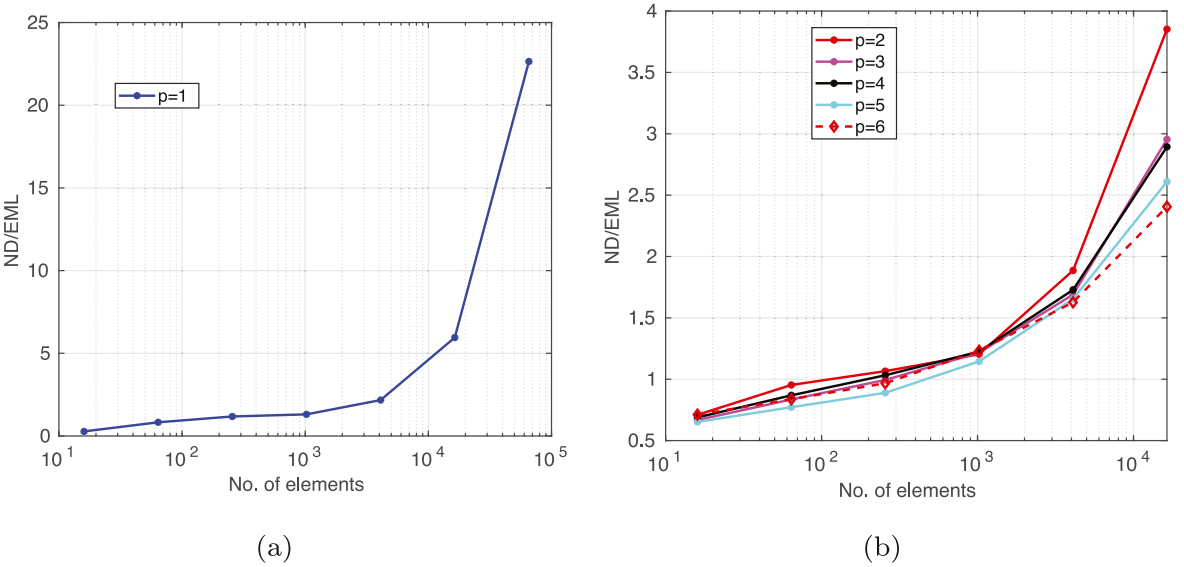


Fig. 16. 2D island coalescence problem: Ratio of time taken per Picard step for ND solver over BFBT+EML preconditioned GMRES for solution order $p = 1$ (left) and for orders $p = 2 - 6$ (right). The ND solver ran into out of memory for orders $p > 1$ and 256^2 elements.

mesh, ND needed 98.2 GB for the L, U factors whereas EML needed only 2.4 GB which is 41 times less memory. The speedup and memory reduction we observed here are significantly higher than the ones observed for the scalar problems studied in [36]. Thus BFBT+EML preconditioned GMRES can deliver significant speedups and memory savings compared to the ND direct solver for vector-valued problems even in 2D. Since 3D problems require a parallel implementation of the EML part in the BFBT+EML preconditioned GMRES, it is left as a part of future work.

7. Conclusion

In this work we present a block preconditioning strategy for the trace system coming from the HDG discretization of the incompressible resistive MHD equations. In the block preconditioner, we use least squares commutator

(BFBT) approximation for the inverse of the Schur complement and a system projection AMG v-cycle for the approximate inverse of the nodal block. For the smoother inside AMG cycle, we compare preconditioned GMRES and ILU(0) smoother of overlap one and conclude that the GMRES smoother is faster, requires less memory and is more robust. We test the performance of the block preconditioner on several 2D and 3D transient test cases (including the island coalescence problem at high Lundquist numbers) and demonstrate robustness and parallel scalability up to thousands of cores. We also conduct a preliminary study of the application of the multilevel approximate nested dissection preconditioner introduced in [36] for the approximate inverse of the nodal block, and compare its performance with AMG and a full nested dissection direct solver. The BFBT+multilevel preconditioner shows better numerical scalability compared to BFBT+AMG with respect to mesh refinements. In terms of robustness with respect to Lundquist numbers BFBT+AMG performs better and strong smoothers are needed in the BFBT+multilevel preconditioner. In comparison with the nested dissection direct solver BFBT+multilevel preconditioned GMRES is significantly faster and requires significantly less memory up to an order of magnitude.

For future work, improving the parallel performance and scalability of the block preconditioner is one of the primary areas of focus. This would enable large scale 3D simulation of realistic geometries that are of interest in fusion research as well as other areas of plasma physics research. Since the block preconditioner developed here is also applicable for Stokes and incompressible Navier–Stokes equations studying the performance in those contexts is also of interest.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to thank Dr. Paul Lin for help regarding various aspects of Trilinos. We also would like to thank Jau-Uei Chen for many useful discussions on this work. SM and SS are partially supported by the U.S. Department of Energy grant DE-SC0018147. T.B.T is partially funded by the National Science Foundation awards NSF-OAC-2212442, NSF-2108320, NSF-1808576 and NSF-CAREER-1845799; by the Department of Energy award DE-SC0018147 and DE-26083989; and by 2020 ConTex award; and by 2021 UT-Portugal CoLab award. We are grateful for the support. At Sandia this work was partially supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics Program and by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research and Office of Fusion Energy Sciences, Scientific Discovery through the Advanced Computing (SciDAC) program. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC, visualization and storage resources that have contributed to the research results reported within this paper. The authors would like to thank the anonymous referees for their critical and useful comments that significantly improved the paper.

Appendix A. Reconnection rate validation study for 2D island coalescence problem

In this section we present a validation study of the high-order HDG method used in this manuscript with the conventional finite element [42] and finite volume methods [43] as well as the theoretical Sweet–Parker scaling in the literature. Fig. A.17 shows that the high-order HDG method requires significantly less number of globally coupled unknowns (trace nodes) compared to the finite element and finite volume methods. All the simulations use one quarter of the domain specified in Section 6.1 by taking advantage of the symmetry and the boundary conditions specified in [42,43]. All the simulations have elements clustered near the 'x'-point and they increase in size away from it.

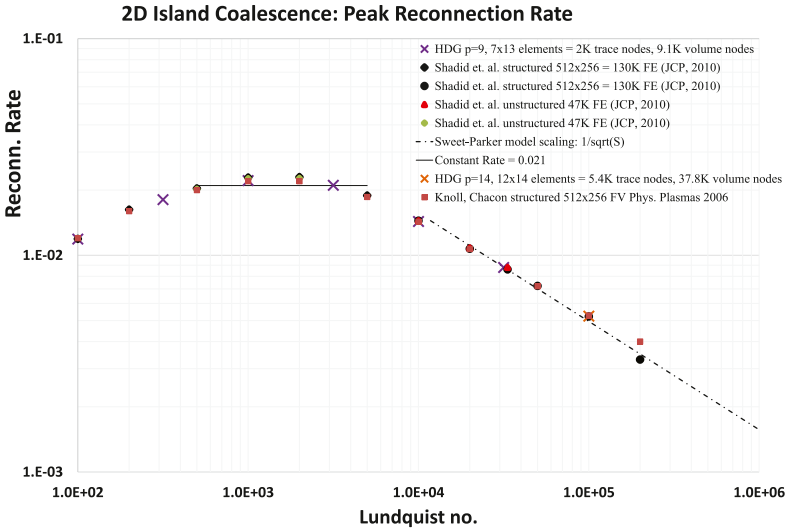


Fig. A.17. 2D island coalescence problem: Validation study of HDG with conventional (linear) stabilized finite element [42] and finite volume methods [43].

Appendix B. Convergence of Algorithm 1

In this section we will (i) show that the unsteady MHD system (1) has a unique solution for small time horizon T, (ii) prove that the Picard iterations for one-step backward Euler MHD system has a unique solution, (iii) show that the backward Euler method for the unsteady MHD system (1) converge to its unique solution, and (iv) the convergence of Algorithm 1 with any converging numerical method for the linearized MHD. For the clarity of the exposition, we consider $d = 3$. Though the results in this section can be adapted to accommodate other boundary conditions, our analysis considers the following particular set of boundary conditions

$$\boldsymbol{u}|_{\partial\Omega} = 0, \quad \boldsymbol{n} \times \boldsymbol{b}|_{\partial\Omega} = 0, \quad (p,1)_\Omega = 0, \text{ and } \quad \text{and } r|_{\partial\Omega} = 0.$$

For simplicity, we assume that Ω is a simply-connected polyhedron with Lipschitz boundary.

To begin, we recall some standard spaces and norms

$$\begin{aligned}
&\mathcal{L}^m := \left\{ q : \int_\Omega |q|^m \, d\Omega < \infty \right\}, && \|q\|_{\mathcal{L}^m}^m := \int_\Omega |q|^m \, d\Omega, \\
&\mathcal{L}_0^2 := \left\{ q \in \mathcal{L}^2 : (q,1)_\Omega = 0 \right\}, && \|q\|_0 := \|q\|_{\mathcal{L}^2}, \\
&\mathcal{H}^1 := \left\{ s \in \mathcal{L}^2 : \nabla s \in \mathcal{L}^2(\Omega) \right\}, && \|s\|_1^2 := \|s\|_0^2 + \|\nabla s\|_0^2, \\
&\mathcal{H}^{\text{curl}} := \left\{ \boldsymbol{v} \in \mathcal{L}^2 : \nabla \times \boldsymbol{v} \in \mathcal{L}^2 \right\}, && \|\boldsymbol{v}\|_{\text{curl}}^2 := \|\boldsymbol{v}\|_0^2 + \|\nabla \times \boldsymbol{v}\|_0^2,
\end{aligned}$$

$$\mathcal{H}_0^1 := \left\{ s \in \mathcal{H}^1 : s|_{\partial\Omega} = 0 \right\}, \quad \mathcal{H}_0^{\text{curl}} := \left\{ \boldsymbol{v} \in \mathcal{H}^{\text{curl}} : \boldsymbol{n} \times \boldsymbol{v}|_{\partial\Omega} = 0 \right\},$$

where, for simplicity of notations, we use the same notations for scalar-valued and vector-valued function spaces, as it will be clear from the context which one we refer to. We also use the short form notation $(\cdot,\cdot)$ for L^2-inner product $(\cdot,\cdot)_\Omega$, $(\cdot,\cdot)_1$ for $\mathcal{H}^1$-inner product, $(\cdot,\cdot)_{curl}$ for $\mathcal{H}^{\text{curl}}$-inner product, and $\langle\cdot,\cdot\rangle$ for duality pairings.

B.1. Existence and uniqueness of the solution of the MHD system

Existence, uniqueness, and regularity of the solution of time dependent MHD system (1), typically using the Galerkin approach, have been investigated (see, e.g., [52–54] and the references therein). Here, we are interested in the existence and uniqueness of the solution and for that purpose we present a simple proof using the classical Picard-Lindelöf theorem. Moreover, this approach provides similar ingredients for the convergence proof of the Picard iterations in Appendix B.2. To that end, we introduce

$$\mathcal{V} := \mathcal{H}_0^1 \times \mathcal{H}_0^{\text{curl}}, \quad \mathcal{Q} := \mathcal{L}_0^2 \times \mathcal{H}_0^1,$$

with the corresponding inner products $([\boldsymbol{u},\boldsymbol{b}],[\boldsymbol{v},\boldsymbol{h}])_{\mathcal{V}} := (\boldsymbol{u},\boldsymbol{v})_1+(\boldsymbol{b},\boldsymbol{h})_{curl}$ and $([p,r],[q,s])_{\mathcal{Q}} := (p,q)+(r,s)_1$, and the induced norms $\|[\boldsymbol{u},\boldsymbol{b}]\|^2_{\mathcal{V}} := \|\boldsymbol{u}\|^2_1 + \|\boldsymbol{b}\|^2_{\text{curl}}$ and $\|[p,r]\|^2_{\mathcal{Q}} := \|p\|^2_0 + \|r\|^2_1$. We incorporate divergence free conditions (1b) and (1d) for both velocity and magnetic fields by defining the following subspaces.

$$\mathcal{H}^1_0\left(\text{div}^0\right) := \left\{\boldsymbol{v}\in\mathcal{H}^1_0 : \nabla\cdot\boldsymbol{v}=0\right\}, \quad \mathcal{H}^{\text{curl}}_0\left(\text{div}^0\right) := \left\{\boldsymbol{h}\in\mathcal{H}^{\text{curl}}_0 : \nabla\cdot\boldsymbol{h}=0\right\},$$

$$\mathcal{V}^0 := \mathcal{H}^1_0\left(\text{div}^0\right)\times\mathcal{H}^{\text{curl}}_0\left(\text{div}^0\right),$$

$$\mathcal{P}_{a,b} := \{[t,\boldsymbol{v},\boldsymbol{h}] : |t-t_0|\le a,\ \|[\boldsymbol{v},\boldsymbol{h}]-[\boldsymbol{u}_0,\boldsymbol{b}_0]\|_{\mathcal{V}}\le b\},$$

where $[\boldsymbol{u}_0,\boldsymbol{b}_0]\in\mathcal{V}^0$ is some initial condition at t_0 for (1), and a, b are positive real numbers.

By searching for a weak solution within $\mathcal{V}^0$, we can rewrite (1) as

$$\begin{aligned}\frac{d\left([\boldsymbol{u},\boldsymbol{b}],[\boldsymbol{v},\boldsymbol{h}]\right)_{\mathcal{V}}}{dt} &= \langle\boldsymbol{f},\boldsymbol{v}\rangle - (\boldsymbol{u}\cdot\nabla\boldsymbol{u},\boldsymbol{v}) - \frac{1}{\text{Re}}(\nabla\boldsymbol{u},\nabla\boldsymbol{v}) + \kappa\left((\nabla\times\boldsymbol{b})\times\boldsymbol{b},\boldsymbol{v}\right)\\ &+\langle\boldsymbol{g},\boldsymbol{h}\rangle + \kappa(\nabla\times(\boldsymbol{u}\times\boldsymbol{b}),\boldsymbol{h}) - \frac{\kappa}{\text{Rm}}(\nabla\times\boldsymbol{b},\nabla\times\boldsymbol{h}) =: \langle\boldsymbol{F}(\boldsymbol{u},\boldsymbol{b}),[\boldsymbol{v},\boldsymbol{h}]\rangle,\end{aligned} \tag{B.1a}$$

where we have defined $\boldsymbol{F}(\boldsymbol{u},\boldsymbol{b})$, given by the right hand side of (B.1a), as a functional in the dual of $\mathcal{V}^0$, i.e. $\boldsymbol{F}(\boldsymbol{u},\boldsymbol{b})\in\left(\mathcal{V}^0\right)'$ for each $[\boldsymbol{u},\boldsymbol{b}]$. Note that the dependence of $\boldsymbol{F}$ on $[\boldsymbol{u},\boldsymbol{b}]$ is nonlinear. In the dual space $\left(\mathcal{V}^0\right)'$, (B.1a) is equivalent to

$$\frac{\partial[\boldsymbol{u},\boldsymbol{b}]}{\partial t} = \boldsymbol{F}(\boldsymbol{u},\boldsymbol{b}). \tag{B.1b}$$

Thanks to the Riesz representation theorem (and thus the linear isometry Riesz map), the existence and uniqueness of the solution of (B.1a) is equivalent to that of (B.1b). The latter is guaranteed by the Picard-Lindelöf theorem if $\boldsymbol{F}(\boldsymbol{u},\boldsymbol{b})$ is bounded and Lipschitz continuous on $\mathcal{P}_{a,b}$ for some a, b.

Lemma 1 (*Boundedness and Lipschitz Continuity of $\boldsymbol{F}(\boldsymbol{u},\boldsymbol{b})$ on $\left(\mathcal{V}^0\right)'$*). *For any $(t,\boldsymbol{u},\boldsymbol{b}), \left(t,\boldsymbol{u}^1,\boldsymbol{b}^1\right), \left(t,\boldsymbol{u}^2,\boldsymbol{b}^2\right)\in\mathcal{P}_{a,b}$, there exists a constant c depending only on Ω, a and b such that*

$$\begin{aligned}\|\boldsymbol{F}(\boldsymbol{u},\boldsymbol{b})\|_{\left(\mathcal{V}^0\right)'} &\le c\left(\|\boldsymbol{f}\|^2_{\left(\mathcal{H}^1_0\right)'} + \|\boldsymbol{u}\|^4_1 + \|\boldsymbol{u}\|^2_1 + \|\boldsymbol{b}\|^4_{curl} + \|\boldsymbol{g}\|^2_{\mathcal{H}^{\text{curl}'}} + \|\boldsymbol{b}\|^2_{curl}\right)^{1/2},\\ \left\|\boldsymbol{F}\left(\boldsymbol{u}^1,\boldsymbol{b}^1\right)-\boldsymbol{F}\left(\boldsymbol{u}^2,\boldsymbol{b}^2\right)\right\|_{\left(\mathcal{V}^0\right)'} &\le c\left\|\left[\boldsymbol{u}^1-\boldsymbol{u}^2,\boldsymbol{b}^1-\boldsymbol{b}^2\right]\right\|_{\mathcal{V}}.\end{aligned}$$

Proof. The main ingredients are the following compact imbeddings [26,55–57]

$$\mathcal{H}^1\hookrightarrow\mathcal{L}^4, \quad \mathcal{H}^1_0\left(\text{div}^0\right)\hookrightarrow\mathcal{L}^4, \quad \mathcal{H}^{\text{curl}}_0\left(\text{div}^0\right)\hookrightarrow\mathcal{L}^4, \tag{B.2}$$

together with Hölder and Poincaré (for both $\mathcal{H}^1_0$ and $\mathcal{H}^{\text{curl}}_0$) inequalities. Indeed we have

$$\begin{aligned}\langle\boldsymbol{F}(\boldsymbol{u},\boldsymbol{b}),[\boldsymbol{v},\boldsymbol{h}]\rangle &\lesssim \|\boldsymbol{f}\|_{\left(\mathcal{H}^1_0\right)'}\|\boldsymbol{v}\|_1 + \|\boldsymbol{u}\|^2_1\|\boldsymbol{v}\|_1 + \|\boldsymbol{u}\|_1\|\boldsymbol{v}\|_1 + \|\boldsymbol{b}\|^2_{curl}\|\boldsymbol{v}\|_1\\ &\quad + \|\boldsymbol{g}\|_{\mathcal{H}^{\text{curl}'}}\|\boldsymbol{h}\|_{curl} + \|\boldsymbol{u}\|_1\|\boldsymbol{b}\|_{curl}\|\boldsymbol{h}\|_{curl} + \|\boldsymbol{b}\|_{curl}\|\boldsymbol{h}\|_{curl}\\ &\lesssim \left(\|\boldsymbol{f}\|^2_{\left(\mathcal{H}^1_0\right)'} + \|\boldsymbol{u}\|^4_1 + \|\boldsymbol{u}\|^2_1 + \|\boldsymbol{b}\|^4_{curl} + \|\boldsymbol{g}\|^2_{\mathcal{H}^{\text{curl}'}} + \|\boldsymbol{b}\|^2_{curl}\right)^{1/2}\|[\boldsymbol{v},\boldsymbol{h}]\|_{\mathcal{V}},\end{aligned}$$

where $\lesssim$ denote the fact that we have ignored all the constants depending only on the domain Ω. Thus

$$\|\boldsymbol{F}(\boldsymbol{u},\boldsymbol{b})\|_{\left(\mathcal{V}^0\right)'} \lesssim \left(\|\boldsymbol{f}\|^2_{\left(\mathcal{H}^1_0\right)'} + \|\boldsymbol{u}\|^4_1 + \|\boldsymbol{u}\|^2_1 + \|\boldsymbol{b}\|^4_{curl} + \|\boldsymbol{g}\|^2_{\mathcal{H}^{\text{curl}'}} + \|\boldsymbol{b}\|^2_{curl}\right)^{1/2},$$

from which, together with the assumption $(t,\boldsymbol{u},\boldsymbol{b})\in\mathcal{P}_{a,b}$, the first assertion follows. The proof for the second assertion follows similarly (see also the proof of Lemma 2). □

The following is the direct consequence of Lemma 1 and the Picard-Lindelöf theorem.

Corollary 1 (*Existence and Uniqueness*). *Let* $M := \max_{(t,\boldsymbol{u},\boldsymbol{b})\in\mathcal{P}_{a,b}} \|\boldsymbol{F}(\boldsymbol{u},\boldsymbol{b})\|_{(\mathcal{V}^0)'}$ *and* $\theta := \min\{a, b/M\}$. *There exists a unique continuously differentiable solution* $(\boldsymbol{u}, p, \boldsymbol{b}, r) \in \mathcal{V}^0 \times \mathcal{Q}$ *for* (1) *in the interval* $[t_0 - \theta, t_0 + \theta]$.

Note that the uniqueness of the fluid pressure p and the Lagrange multiplier r are part of the conclusion, but their recovery from $\boldsymbol{u}, \boldsymbol{b}$ are omitted as they are not explicitly involved in our proofs.

B.2. Convergence of the picard iterations for one-step backward Euler MHD system

In this section, we first discuss the existence and uniqueness of the solution of a generic one step of the backward Euler time discretization for (B.1b). We then describe Picard iterations to approximate the solution of the one-step backward Euler time discretization. We shall show that the linearized MHD system corresponding to each Picard iteration has a unique solution and that the Picard iterations converge to the solution of the one-step backward Euler time discretization. There is no need to introduce new notations for these aforementioned steps to avoid the burden of managing too many notations. Specifically, we will use the same notation for solutions at all these steps and it will be clear from the context.

We now consider one step of the backward Euler method applied to (1), which reads

$$\delta_1\boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nabla p - \frac{1}{\mathrm{Re}}\nabla\cdot\nabla\boldsymbol{u} - \kappa(\nabla\times\boldsymbol{b})\times\boldsymbol{b} = \tilde{\boldsymbol{f}}, \tag{B.3a}$$

$$\nabla\cdot\boldsymbol{u} = 0, \tag{B.3b}$$

$$\delta_2\boldsymbol{b} + \nabla r - \kappa\nabla\times(\boldsymbol{u}\times\boldsymbol{b}) + \frac{\kappa}{\mathrm{Rm}}\nabla\times(\nabla\times\boldsymbol{b}) = \tilde{\boldsymbol{g}}, \tag{B.3c}$$

$$\nabla\cdot\boldsymbol{b} = 0. \tag{B.3d}$$

Here, $\delta_1 = (\Delta t)^{-1}$, where Δt is the time stepsize, $\delta_2 = \kappa\,(\Delta t)^{-1}$, $\tilde{\boldsymbol{f}} = \boldsymbol{f} + \delta_1\overline{\boldsymbol{u}}$ and $\tilde{\boldsymbol{g}} = \boldsymbol{g} + \delta_2\overline{\boldsymbol{b}}$, where $\left[\overline{\boldsymbol{u}}, \overline{\boldsymbol{b}}\right]$ is the pair of velocity and the magnetic fields from the previous time step.

We briefly note that the existence and uniqueness of a weak solution of the system (B.3) when $\delta_1 = \delta_2 = 0$ have been proved for different types of boundary conditions and different functional settings (see, e.g., [26,53,54,58–60] and the references therein), using the theory of trilinear maps [26,59] developed in [56], and using the Banach fixed point theorem [54,58], to name a few. The existence and uniqueness have also been proved using the Schauder-fixed point theorem with regularization for continuous-time density-dependent MHD system, and continuous-time MHD system with steady-state induction equation in [53,54]. Here we provide a proof using the Leray–Schauder principle [61,62], which allows us to show the convergence of the Picard iterations in a straightforward manner.

The corresponding linearized system (1) reads

$$\delta_1\boldsymbol{u} + \boldsymbol{w}\cdot\nabla\boldsymbol{u} + \nabla p - \frac{1}{\mathrm{Re}}\nabla\cdot\nabla\boldsymbol{u} - \kappa(\nabla\times\boldsymbol{b})\times\boldsymbol{d} = \tilde{\boldsymbol{f}}, \tag{B.4a}$$

$$\nabla\cdot\boldsymbol{u} = 0, \tag{B.4b}$$

$$\delta_2\boldsymbol{b} + \nabla r - \kappa\nabla\times(\boldsymbol{u}\times\boldsymbol{d}) + \frac{\kappa}{\mathrm{Rm}}\nabla\times(\nabla\times\boldsymbol{b}) = \tilde{\boldsymbol{g}}, \tag{B.4c}$$

$$\nabla\cdot\boldsymbol{b} = 0. \tag{B.4d}$$

The plan is first to show that the linearized system (B.4) has a unique solution $[\boldsymbol{u}, \boldsymbol{b}, p, r]$ that depends continuously on $\tilde{\boldsymbol{f}}, \tilde{\boldsymbol{g}}, \boldsymbol{w}$ and $\boldsymbol{d}$. Let us denote by $\boldsymbol{\Phi}$ the map from $[\boldsymbol{w}, \boldsymbol{d}]$ to $[\boldsymbol{u}, \boldsymbol{b}]$, i.e., $[\boldsymbol{u}, \boldsymbol{b}] := \boldsymbol{\Phi}([\boldsymbol{w}, \boldsymbol{d}])$, and the Picard iteration can be written as $\left[\boldsymbol{u}^{m+1}, \boldsymbol{b}^{m+1}\right] := \boldsymbol{\Phi}\left(\left[\boldsymbol{u}^m, \boldsymbol{b}^m\right]\right)$, where $\left[\boldsymbol{u}^0, \boldsymbol{b}^0\right] = \left[\overline{\boldsymbol{u}}, \overline{\boldsymbol{b}}\right]$. Next we show that $\boldsymbol{\Phi}$ is compact and then apply the Leray–Schauder principle [61,62] to conclude the existence of a fixed point $\left[\boldsymbol{u}^*, \boldsymbol{d}^*\right]$ of $\boldsymbol{\Phi}$, which is exactly a solution of one-step backward Euler MHD system (B.3). The uniqueness is then ensured by proving the Lipschitz continuity of $\boldsymbol{\Phi}$. The convergence of the Picard iterations is readily available if we can show the contraction of $\boldsymbol{\Phi}$, assuming $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{g}}$ are small.

We search for a weak solution $[\boldsymbol{u}, \boldsymbol{b}, p, r] \in \mathcal{H}_0^1 \times \mathcal{H}_0^{\text{curl}} \times \mathcal{L}_0^2 \times \mathcal{H}_0^1$ of (B.4) such that

$$\delta_1 (\boldsymbol{u}, \boldsymbol{v}) + (\boldsymbol{w} \cdot \nabla \boldsymbol{u}, \boldsymbol{v}) - (p, \nabla \cdot \boldsymbol{v}) + \frac{1}{\text{Re}} (\nabla \boldsymbol{u}, \nabla \boldsymbol{v}) - \kappa ((\nabla \times \boldsymbol{b}) \times \boldsymbol{d}, \boldsymbol{v}) = \left\langle \tilde{\boldsymbol{f}}, \boldsymbol{v} \right\rangle, \tag{B.5a}$$

$$-(\nabla \cdot \boldsymbol{u}, q) = 0, \tag{B.5b}$$

$$\delta_2 (\boldsymbol{b}, \boldsymbol{h}) + (\nabla r, \boldsymbol{h}) - \kappa (\nabla \times (\boldsymbol{u} \times \boldsymbol{d}), \boldsymbol{h}) + \frac{\kappa}{\text{Rm}} (\nabla \times \boldsymbol{b}, \nabla \times \boldsymbol{h}) = \langle \tilde{\boldsymbol{g}}, \boldsymbol{h} \rangle, \tag{B.5c}$$

$$(\boldsymbol{b}, \nabla s) = 0, \tag{B.5d}$$

for all $[\boldsymbol{v}, \boldsymbol{h}, q, s] \in \mathcal{H}_0^1 \times \mathcal{H}_0^{\text{curl}} \times \mathcal{L}_0^2 \times \mathcal{H}_0^1$. The existence and uniqueness of the weak solution of the linearized system (B.5), and hence (B.4), via Brezzi theorem [63] are discussed in [58,64] (and the references therein). Since we are interested in keeping track of the constants that depend on the time stepsize Δt and other parameters (including the forcings $\boldsymbol{f}, \boldsymbol{g}$), we provide more details to pave the way to the continuity of $\boldsymbol{\Phi}$. This, in turn, prepares necessary ingredients to show the existence and uniqueness of the original nonlinear one step Euler MHD system (B.3).

We define two operators $\mathbb{A} := \mathcal{V} \to \mathcal{V}'$ and $\mathbb{B} := \mathcal{V} \to \mathcal{Q}'$ via

$$\langle \mathbb{A} [\boldsymbol{u}, \boldsymbol{b}], [\boldsymbol{v}, \boldsymbol{h}] \rangle := \delta_1 (\boldsymbol{u}, \boldsymbol{v}) + (\boldsymbol{w} \cdot \nabla \boldsymbol{u}, \boldsymbol{v}) + \frac{1}{\text{Re}} (\nabla \boldsymbol{u}, \nabla \boldsymbol{v}) - \kappa ((\nabla \times \boldsymbol{b}) \times \boldsymbol{d}, \boldsymbol{v}) + \delta_2 (\boldsymbol{b}, \boldsymbol{h}) - \kappa (\nabla \times (\boldsymbol{u} \times \boldsymbol{d}), \boldsymbol{h}) + \frac{\kappa}{\text{Rm}} (\nabla \times \boldsymbol{b}, \nabla \times \boldsymbol{h}),$$

and

$$\left\langle \mathbb{B}' [p, r], [\boldsymbol{v}, \boldsymbol{h}] \right\rangle := -(p, \nabla \cdot \boldsymbol{v}) + (\nabla r, \boldsymbol{h}).$$

The weak equation (B.5) is thus equivalent to

$$\mathbb{A} [\boldsymbol{u}, \boldsymbol{b}] + \mathbb{B}' [p, r] = \boldsymbol{F}, \tag{B.6a}$$

$$\mathbb{B} [\boldsymbol{u}, \boldsymbol{b}] = 0, \tag{B.6b}$$

where $\boldsymbol{F} : \mathcal{V} \to \mathbb{R}$ given by $\langle \boldsymbol{F}, [\boldsymbol{v}, \boldsymbol{h}] \rangle = \left\langle \tilde{\boldsymbol{f}}, \boldsymbol{v} \right\rangle + \langle \tilde{\boldsymbol{g}}, \boldsymbol{h} \rangle$. The inf–sup condition for $\mathbb{B}$

$$\inf_{[q,s] \in \mathcal{Q}} \sup_{[\boldsymbol{v}, \boldsymbol{h}] \in \mathcal{V}} \frac{\left\langle \mathbb{B}' [q, s], [\boldsymbol{v}, \boldsymbol{h}] \right\rangle}{\|[q, s]\|_{\mathcal{Q}} \|[\boldsymbol{v}, \boldsymbol{h}]\|_{\mathcal{V}}} \geq \beta,$$

where β depends only on the domain Ω, is readily available from [56,65]. It can be shown (see, e.g. [26,56,57]) that the kernel of $\mathbb{B}$ is given by $\mathcal{V}^0$. Now if $\boldsymbol{w} \in \mathcal{H}_0^1 \left(\text{div}^0\right)$ and $\boldsymbol{d} \in \mathcal{H}_0^{\text{curl}}$, it is easy to see that

$$\langle \mathbb{A} [\boldsymbol{u}, \boldsymbol{b}], [\boldsymbol{u}, \boldsymbol{b}] \rangle \geq \alpha \|[\boldsymbol{u}, \boldsymbol{b}]\|_{\mathcal{V}}^2, \tag{B.7}$$

where $\alpha = \min \{\delta_1, \delta_2, 1/\text{Re}, \kappa/\text{Rm}\}$, and thus $\mathbb{A}$ is coercive on $\mathcal{V}^0$. The system (B.6), and hence equivalently (B.4) and (B.5), thus has a unique solution owing to the Brezzi theorem [63]. Furthermore, $[\boldsymbol{u}, \boldsymbol{b}] \in \mathcal{V}^0$ and

$$\|[\boldsymbol{u}, \boldsymbol{b}]\|_{\mathcal{V}} \leq \gamma := \frac{1}{\alpha} \left(\|\boldsymbol{f}\|_1 + \|\boldsymbol{g}\|_{\text{curl}} + \delta_1 \|\overline{\boldsymbol{u}}\|_1 + \delta_2 \left\| \overline{\boldsymbol{b}} \right\|_{\text{curl}} \right). \tag{B.8}$$

Therefore $[\boldsymbol{u}, \boldsymbol{b}] := \boldsymbol{\Phi} ([\boldsymbol{w}, \boldsymbol{d}])$ is a well-defined map from $\mathcal{V}^0$ into $\mathcal{V}^0$.

We are now in the position to show the continuity of $\boldsymbol{\Phi}$ with respect to $[\boldsymbol{w}, \boldsymbol{d}]$.

Lemma 2. *Let* $\left[\boldsymbol{w}^1, \boldsymbol{d}^1\right], \left[\boldsymbol{w}^2, \boldsymbol{d}^2\right] \in \mathcal{V}^0$, *and* $\left[\boldsymbol{u}^1, \boldsymbol{b}^1\right] = \boldsymbol{\Phi} \left(\left[\boldsymbol{w}^1, \boldsymbol{d}^1\right]\right), \left[\boldsymbol{u}^2, \boldsymbol{b}^2\right] = \boldsymbol{\Phi} \left(\left[\boldsymbol{w}^2, \boldsymbol{d}^2\right]\right)$ *be the two solutions of* (B.5), *respectively. There exists a constant* c *depending only on* Ω *such that*

$$\left\| \boldsymbol{\Phi} \left(\left[\boldsymbol{w}^1, \boldsymbol{d}^1\right]\right) - \boldsymbol{\Phi} \left(\left[\boldsymbol{w}^2, \boldsymbol{d}^2\right]\right) \right\|_{\mathcal{V}} \leq c \frac{\gamma}{\alpha} \left(\left\| \boldsymbol{d}^2 - \boldsymbol{d}^1 \right\|_{\mathcal{L}^4}^2 + \left\| \boldsymbol{w}^2 - \boldsymbol{w}^1 \right\|_{\mathcal{L}^4}^2 \right)^{1/2} \leq c \frac{\gamma}{\alpha} \left\| \left[\boldsymbol{w}^2, \boldsymbol{d}^2\right] - \left[\boldsymbol{w}^1, \boldsymbol{d}^1\right] \right\|_{\mathcal{V}}, \tag{B.9}$$

that is, $\boldsymbol{\Phi} : \mathcal{V}^0 \to \mathcal{V}^0$ *is Lipschitz continuous.*

Proof. Subtracting the two systems corresponding to $\left[\boldsymbol{w}^1, \boldsymbol{d}^1\right]$ and $\left[\boldsymbol{w}^2, \boldsymbol{d}^2\right]$, taking $[\boldsymbol{v}, \boldsymbol{h}] = \left[\boldsymbol{u}^1 - \boldsymbol{u}^2, \boldsymbol{b}^1 - \boldsymbol{b}^2\right]$, and performing some simple algebraic manipulations give

$$\alpha \left\|\left[\boldsymbol{u}^2, \boldsymbol{b}^2\right] - \left[\boldsymbol{u}^1, \boldsymbol{b}^1\right]\right\|_{\mathcal{V}}^2 \leq \underbrace{\left|\left(\left(\boldsymbol{w}^2 - \boldsymbol{w}^1\right) \cdot \nabla \boldsymbol{u}^1, \boldsymbol{u}^2 - \boldsymbol{u}^1\right)\right|}_{T_1 :=} +$$
$$\kappa \underbrace{\left|\left((\nabla \times \boldsymbol{b}^2) \times \left(\boldsymbol{d}^2 - \boldsymbol{d}^1\right), \boldsymbol{u}^2 - \boldsymbol{u}^1\right)\right|}_{T_2 :=} + \kappa \underbrace{\left|\left(\nabla \times (\boldsymbol{u}^2 \times \left(\boldsymbol{d}^2 - \boldsymbol{d}^1\right)), \boldsymbol{b}^2 - \boldsymbol{b}^1\right)\right|}_{T_3 :=} \tag{B.10}$$

The compact imbeddings in (B.2), the coercivity of $\mathbb{A}$, and Hölder and Poincaré inequalities (for both $\mathcal{H}_0^1$ and $\mathcal{H}_0^{\text{curl}}$) give

$$\begin{aligned}
T_1 &\leq c \left\|\boldsymbol{w}^2 - \boldsymbol{w}^1\right\|_{\mathcal{L}^4} \left\|\boldsymbol{u}^1\right\|_1 \left\|\boldsymbol{u}^2 - \boldsymbol{u}^1\right\|_1 \leq c \left\|\boldsymbol{w}^2 - \boldsymbol{w}^1\right\|_1 \left\|\boldsymbol{u}^1\right\|_1 \left\|\boldsymbol{u}^2 - \boldsymbol{u}^1\right\|_1, \\
T_2 &\leq c \left\|\boldsymbol{d}^2 - \boldsymbol{d}^1\right\|_{\mathcal{L}^4} \left\|\boldsymbol{u}^2 - \boldsymbol{u}^1\right\|_1 \left\|\boldsymbol{b}^2\right\|_{\text{curl}} \leq c \left\|\boldsymbol{d}^2 - \boldsymbol{d}^1\right\|_{\text{curl}} \left\|\boldsymbol{u}^2 - \boldsymbol{u}^1\right\|_1 \left\|\boldsymbol{b}^2\right\|_{\text{curl}}, \\
T_3 &\leq c \left\|\boldsymbol{d}^2 - \boldsymbol{d}^1\right\|_{\mathcal{L}^4} \left\|\boldsymbol{b}^2 - \boldsymbol{b}^1\right\|_{\text{curl}} \left\|\boldsymbol{u}^2\right\|_1 \leq c \left\|\boldsymbol{d}^2 - \boldsymbol{d}^1\right\|_{\text{curl}} \left\|\boldsymbol{b}^2 - \boldsymbol{b}^1\right\|_{\text{curl}} \left\|\boldsymbol{u}^2\right\|_1,
\end{aligned}$$

where c is a constant depending only Ω. Now combining (B.10), (B.8), and Cauchy–Schwarz inequality gives

$$\begin{aligned}
\alpha \left\|\left[\boldsymbol{u}^2, \boldsymbol{b}^2\right] - \left[\boldsymbol{u}^1, \boldsymbol{b}^1\right]\right\|_{\mathcal{V}} &\leq c\gamma \left(\left\|\boldsymbol{d}^2 - \boldsymbol{d}^1\right\|_{\mathcal{L}^4}^2 + \left\|\boldsymbol{w}^2 - \boldsymbol{w}^1\right\|_{\mathcal{L}^4}^2\right)^{1/2} \\
&\leq c\gamma \left\|\left[\boldsymbol{w}^2, \boldsymbol{d}^2\right] - \left[\boldsymbol{w}^1, \boldsymbol{d}^1\right]\right\|_{\mathcal{V}},
\end{aligned}$$

and this end the proof. □

Theorem 1 (*Existence and uniqueness*)**.** *There exists weak solution* $[\boldsymbol{u}, \boldsymbol{b}, p, r]$ *in* $\mathcal{V} \times \mathcal{Q}$ *for one step backward Euler MHD system* (B.3)*. Furthermore, if* Δt, $\boldsymbol{f}$, $\boldsymbol{g}$, $\overline{\boldsymbol{u}}$ *and* $\overline{\boldsymbol{b}}$ *are such that* $c\gamma/\alpha < 1$*, then the solution is unique.*

Proof. We first show that the map $\boldsymbol{\Phi} : \mathcal{V}^0 \to \mathcal{V}^0$ of the weak system (B.5) is compact. To that end, let $\left\{\left[\boldsymbol{u}^m, \boldsymbol{b}^m\right]\right\}_{m=1}^{\infty}$ be a bounded sequence in $\mathcal{V}^0$ such that $\left\|\left[\boldsymbol{u}^m, \boldsymbol{b}^m\right]\right\|_{\mathcal{V}} \leq M < \infty$ for some M. Due to the compact imbeddings (B.2), there exists a subsequence, again denoted $\left\{\left[\boldsymbol{u}^m, \boldsymbol{b}^m\right]\right\}_{m=1}^{\infty}$ for simplicity, converges in $\mathcal{L}^4$ to some $\left[\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{b}}\right] \in \mathcal{V}^0 \cap \mathcal{L}^4$, that is, $\left\|\boldsymbol{d}^m - \tilde{\boldsymbol{d}}\right\|_{\mathcal{L}^4} \to 0$ and $\|\boldsymbol{w}^m - \tilde{\boldsymbol{w}}\|_{\mathcal{L}^4} \to 0$ as $n \to \infty$. Now using the continuity of $\boldsymbol{\Phi}$ in (B.9) we have

$$\left\|\boldsymbol{\Phi}\left(\left[\boldsymbol{u}^m, \boldsymbol{b}^m\right]\right) - \boldsymbol{\Phi}\left(\left[\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{b}}\right]\right)\right\|_{\mathcal{V}} \leq c\frac{\gamma}{\alpha}\left(\left\|\boldsymbol{d}^m - \tilde{\boldsymbol{d}}\right\|_{\mathcal{L}^4}^2 + \left\|\boldsymbol{w}^m - \tilde{\boldsymbol{w}}\right\|_{\mathcal{L}^4}^2\right)^{1/2} \xrightarrow{m\to\infty} 0,$$

that is, the sequence $\left\{\boldsymbol{\Phi}\left(\left[\boldsymbol{u}^m, \boldsymbol{b}^m\right]\right)\right\}_{m=1}^{\infty} \subset \mathcal{V}^0$ convergences to $\boldsymbol{\Phi}\left(\left[\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{b}}\right]\right) \in \mathcal{V}^0$, and hence compact. The application of the Leray–Schauder principle is now immediate, owing to the uniform *a priori* estimate (B.8), to conclude that there exist a weak solution for the one-step backward Euler MHD system (B.3). The uniqueness is obvious by the Lipschitz continuity of $\boldsymbol{\Phi}$ in (B.9) and the assumption $c\gamma/\alpha < 1$. □

Corollary 2 (*Convergence of the Picard iterations*)**.** *Suppose* $\left[\overline{\boldsymbol{u}}, \overline{\boldsymbol{b}}\right] \in \mathcal{V}^0$*, then the Picard iterations*

$$\left[\boldsymbol{u}^{m+1}, \boldsymbol{b}^{m+1}\right] := \boldsymbol{\Phi}\left(\left[\boldsymbol{u}^m, \boldsymbol{b}^m\right]\right), \quad \left[\boldsymbol{u}^1, \boldsymbol{b}^1\right] = \left[\overline{\boldsymbol{u}}, \overline{\boldsymbol{b}}\right],$$

converges to the unique weak solution of (B.3) *provided that* Δt, $\boldsymbol{f}$, $\boldsymbol{g}$, $\overline{\boldsymbol{u}}$ *and* $\overline{\boldsymbol{b}}$ *are such that* $c\gamma/\alpha < 1$.

B.3. Convergence of Algorithm 1

We are in the position to show the convergence of Algorithm 1. For the simplicity of the exposition, we assume $t_0 = 0$ and the time stepsize Δt is the same for all backward Euler steps. Unlike Appendix B.2, we ought to introduce new notations for each step of Algorithm 1 in order to have the clarity in the proof. To that end, let $t_n = n\Delta t$, $n = 0, 1, \ldots, N$, where N is the total number of time steps such that $\theta = N\Delta t$. We denote by $\left[\boldsymbol{u}^n, \boldsymbol{b}^n\right]$ the backward Euler temporal approximation (see (B.3)) of the exact solution at time t_n, i.e. $[\boldsymbol{u}(t_n), \boldsymbol{b}(t_n)]$ (see (B.1b)), by $\left[\boldsymbol{u}^{n,m}, \boldsymbol{b}^{n,m}\right]$ the Picard approximation (see (B.5)) of $\left[\boldsymbol{u}^n, \boldsymbol{b}^n\right]$ after m Picard iterations, and by

$\left[\boldsymbol{u}_h^{n,m}, \boldsymbol{b}_h^{n,m}\right]$ some spatial numerical approximation of $\left[\boldsymbol{u}^{n,m}, \boldsymbol{b}^{n,m}\right]$ with (the largest) meshsize h. We assume that the numerical solution $\left[\boldsymbol{u}_h^{n,m}, \boldsymbol{b}_h^{n,m}\right]$ is convergent, in particular,

$$\left\|\left[\boldsymbol{u}^{n,m}, \boldsymbol{b}^{n,m}\right] - \left[\boldsymbol{u}_h^{n,m}, \boldsymbol{b}_h^{n,m}\right]\right\|_{\mathcal{V}} = \mathcal{O}(h), \tag{B.11}$$

and we shall discuss how the HDG discretization can fit this assumption. Regardless of what convergent numerical method is chosen, we need to solve a (possibly large) linear system of equations in order to obtain $\left[\boldsymbol{u}_h^{n,m}, \boldsymbol{b}_h^{n,m}\right]$. For example, in this paper the linear system resulting from the HDG discretization is (12), and for large scale problems one has to develop an efficient solver/preconditioner for it, as we have presented in this paper. We assume the error in computing $\left[\boldsymbol{u}_h^{n,m}, \boldsymbol{b}_h^{n,m}\right]$ (e.g., Krylov solver in Fig. 1) is negligible. Our goal is to characterize the error between the exact solution at time t_n and the corresponding approximate solution at the mth Picard iteration with a convergent numerical method, namely $\left\|[\boldsymbol{u}(t_n), \boldsymbol{b}(t_n)] - \left[\boldsymbol{u}_h^{n,m}, \boldsymbol{b}_h^{n,m}\right]\right\|_{\mathcal{V}}$, in terms of Δt, h, and m.

Theorem 2. *Let θ be defined as in Corollary 1, c be the constant in Lemma 1 (and Lemma 2), α and γ be defined as in (B.7) and (B.8), and Δt, $\boldsymbol{f}$, $\boldsymbol{g}$, $\overline{\boldsymbol{u}}$ and $\overline{\boldsymbol{b}}$ are such that $\zeta := c\gamma/\alpha < 1$. Suppose $[\boldsymbol{u}, \boldsymbol{b}]$ is twice continuously differentiable and*

$$\max_{t\in[0,0+\theta]} \left\| \frac{\partial^2 [\boldsymbol{u}, \boldsymbol{b}]}{\partial t^2} \right\|_{\mathcal{V}} \leq L.$$

Furthermore, assume that a convergent numerical method is deployed to compute $\left[\boldsymbol{u}_h^{n,m}, \boldsymbol{b}_h^{n,m}\right]$. Then

$$\left\|[\boldsymbol{u}(t_n), \boldsymbol{b}(t_n)] - \left[\boldsymbol{u}_h^{n,m}, \boldsymbol{b}_h^{n,m}\right]\right\|_{\mathcal{V}} = \mathcal{O}\left(\Delta t + \zeta^m + h\right).$$

Consequently, $\left[\boldsymbol{u}_h^{n,m}, \boldsymbol{b}_h^{n,m}\right]$ converges to $[\boldsymbol{u}(t_n), \boldsymbol{b}(t_n)]$ when $\Delta t \to 0$, $m \to \infty$, and $h \to 0$.

Proof. We divide the proof in two steps.

Step 1: estimation of the error $\left\|[\boldsymbol{u}(t_n), \boldsymbol{b}(t_n)] - \left[\boldsymbol{u}^n, \boldsymbol{b}^n\right]\right\|_{\mathcal{V}}$. We show that the entire backward Euler time discretization converges to the unique solution of the MHD system (1) (see Corollary 1). A straightforward extension of the standard convergence proof of the backward Euler (see, e.g., [66] and the references therein) gives

$$\left\|[\boldsymbol{u}(t_n), \boldsymbol{b}(t_n)] - \left[\boldsymbol{u}^n, \boldsymbol{b}^n\right]\right\|_{\mathcal{V}} \leq L t_n e^{2t_n c} \Delta t \leq L\theta e^{2\theta c} \Delta t.$$

Step 2: estimation of the error $\left\|\left[\boldsymbol{u}^n, \boldsymbol{b}^n\right] - \left[\boldsymbol{u}^{n,m}, \boldsymbol{b}^{n,m}\right]\right\|_{\mathcal{V}}$. Due to the assumption $\zeta < 1$, the Picard iteration is contractive (see Corollary 2) and, similar to the Banach fixed point iterations, simple algebra yields the following *a priori* error estimate

$$\left\|\left[\boldsymbol{u}^n, \boldsymbol{b}^n\right] - \left[\boldsymbol{u}^{n,m}, \boldsymbol{b}^{n,m}\right]\right\|_{\mathcal{V}} \leq \frac{\zeta^m}{1-\zeta} \left\|\left[\boldsymbol{u}^{n,1}, \boldsymbol{b}^{n,1}\right] - \left[\boldsymbol{u}^{n,0}, \boldsymbol{b}^{n,0}\right]\right\|_{\mathcal{V}}.$$

Now combining the results of Step 1, Step 2, and (B.11) concludes the proof. □

Remark 1. Note that the assumption (B.11) is quite conservative as it corresponds to a first order spatial discretization. For the HDG method employed in this paper the convergence rate is much faster. Indeed, we have showed [10]

$$\left\|\left[\boldsymbol{u}^{n,m}, \boldsymbol{b}^{n,m}\right] - \left[\boldsymbol{u}_h^{n,m}, \boldsymbol{b}_h^{n,m}\right]\right\|_{\mathcal{V}} = \mathcal{O}\left(h^{p+\frac{1}{2}}\right),$$

where p is the solution order.

References

[1] J.P. Goedbloed, S. Poedts, Principles of Magnetohydrodynamics: With Applications To Laboratory and Astrophysical Plasmas, Cambridge University Press, 2004.

[2] L. Chacón, An optimal, parallel, fully implicit newton–krylov solver for three-dimensional viscoresistive magnetohydrodynamics, Phys. Plasmas 15 (5) (2008) 056103.

[3] J.N. Shadid, R.P. Pawlowski, E.C. Cyr, R.S. Tuminaro, L. Chacón, P.D. Weber, Scalable implicit incompressible resistive MHD with stabilized FE and fully-coupled Newton–Krylov-AMG, Comput. Methods Appl. Mech. Engrg. 304 (2016) 1–25.

[4] E.C. Cyr, J.N. Shadid, R.S. Tuminaro, R.P. Pawlowski, L. Chacón, A new approximate block factorization preconditioner for two-dimensional incompressible (reduced) resistive MHD, SIAM J. Sci. Comput. 35 (3) (2013) B701–B730.

[5] L. Chacón, A. Stanier, A scalable, fully implicit algorithm for the reduced two-field low-β extended MHD model, J. Comput. Phys. 326 (2016) 763–772.
[6] E.G. Phillips, J.N. Shadid, E.C. Cyr, H.C. Elman, R.P. Pawlowski, Block preconditioners for stable mixed nodal and edge finite element representations of incompressible resistive MHD, SIAM J. Sci. Comput. 38 (6) (2016) B1009–B1031.
[7] M.P. Wathen, Preconditioners for Incompressible Magnetohydrodynamics (Ph.D. thesis), University of British Columbia, 2018.
[8] Q. Tang, L. Chacón, T.V. Kolev, J.N. Shadid, X.-Z. Tang, An adaptive scalable fully implicit algorithm based on stabilized finite element for reduced visco-resistive MHD, J. Comput. Phys. (2022) 110967.
[9] S. Shannon, J.N. Shadid, T. Bui-Thanh, J.J. Lee, A hybridized discontinuous Galerkin method for resistive incompressible magnetohydrodynamics, in: CCR Summer Proceedings, Sandia, 2016.
[10] J.J. Lee, S.J. Shannon, T. Bui-Thanh, J.N. Shadid, Analysis of an HDG method for linearized incompressible resistive MHD equations, SIAM J. Numer. Anal. 57 (4) (2019) 1697–1722.
[11] C. Ciucua, P. Fernandez, A. Christophe, N.C. Nguyen, J. Peraire, Implicit hybridized discontinuous galerkin methods for compressible magnetohydrodynamics, J. Comput. Phys.: X 5 (2020) 100042.
[12] S. Muralikrishnan, Fast and Scalable Solvers for High-Order Hybridized Discontinuous Galerkin Methods with Applications To Fluid Dynamics and Magnetohydrodynamics (Ph.D. thesis), 2019.
[13] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1.
[14] H. Elman, V.E. Howle, J. Shadid, R. Shuttleworth, R. Tuminaro, Block preconditioners based on approximate commutators, SIAM J. Sci. Comput. 27 (5) (2006) 1651–1668.
[15] H.C. Elman, Preconditioning for the steady-state Navier–Stokes equations with low viscosity, SIAM J. Sci. Comput. 20 (4) (1999) 1299–1316.
[16] R. Planas Badenas, Stabilized Finite Element Formulations for Solving Incompressible Magnetohydrodynamics (Ph.D. thesis), 2013.
[17] H.C. Elman, D.J. Silvester, A.J. Wathen, Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics, Oxford University Press, USA, 2014.
[18] J. Rudi, Global Convection in Earth's Mantle: Advanced Numerical Methods and Extreme-Scale Simulations (Ph.D. thesis), 2019.
[19] M. Wathen, C. Greif, D. Schotzau, Preconditioners for mixed finite element discretizations of incompressible mhd equations, SIAM J. Sci. Comput. 39 (6) (2017) A2993–A3013.
[20] M. Wathen, C. Greif, A scalable approximate inverse block preconditioner for an incompressible magnetohydrodynamics model problem, SIAM J. Sci. Comput. 42 (1) (2020) B57–B79.
[21] S. Rhebergen, G.N. Wells, Preconditioning of a hybridized discontinuous Galerkin finite element method for the Stokes equations, J. Sci. Comput. 77 (3) (2018) 1936–1952.
[22] B.S. Southworth, A.A. Sivas, S. Rhebergen, On fixed-point, krylov, and 2*2 block preconditioners for nonsymmetric problems, SIAM J. Matrix Anal. Appl. 41 (2) (2020) 871–900.
[23] S. Rhebergen, G.N. Wells, Preconditioning for a pressure-robust hdg discretization of the stokes equations, arXiv preprint arXiv:2105.09152.
[24] R. Codina, N. Hernández-Silva, Stabilized finite element approximation of the stationary magneto-hydrodynamics equations, Comput. Mech. 38 (4–5) (2006) 344–355.
[25] N.B. Salah, A. Soulaimani, W.G. Habashi, M. Fortin, A conservative stabilized finite element method for the magneto-hydrodynamic equations, Internat. J. Numer. Methods Fluids 29 (5) (1999) 535–554.
[26] D. Schötzau, Mixed finite element methods for stationary incompressible magneto–hydrodynamics, Numer. Math. 96 (4) (2004) 771–800.
[27] U. Müller, L. Bühler, Magnetofluiddynamics in Channels and Containers, Springer Science & Business Media, 2013.
[28] S. Shannon, Hybridized Discontinuous Galerkin Methods for Magnetohydrodynamics (Ph.D. thesis), 2018.
[29] N.C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations, J. Comput. Phys. 230 (2011) 1147–1170.
[30] M.F. Murphy, G.H. Golub, A.J. Wathen, A note on preconditioning for indefinite linear systems, SIAM J. Sci. Comput. 21 (6) (2000) 1969–1972.
[31] P.S. Vassilevski, Multilevel Block Factorization Preconditioners: Matrix-Based Analysis and Algorithms for Solving Finite Element Equations, Springer Science & Business Media, 2008.
[32] D.A. May, L. Moresi, Preconditioned iterative methods for Stokes flow problems arising in computational geodynamics, Phys. Earth Planet. Inter. 171 (1–4) (2008) 33–47.
[33] X.S. Li, J.W. Demmel, SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems, ACM Trans. Math. Softw. 29 (2) (2003) 110–140.
[34] M.W. Gee, C.M. Siefert, J.J. Hu, R.S. Tuminaro, M.G. Sala, ML 5.0 Smoothed Aggregation User's Guide, Tech. rep. Technical Report SAND2006-2649, Sandia National Laboratories, 2006.
[35] M.A. Heroux, R.A. Bartlett, V.E. Howle, R.J. Hoekstra, J.J. Hu, T.G. Kolda, R.B. Lehoucq, K.R. Long, R.P. Pawlowski, E.T. Phipps, et al., An overview of the trilinos project, ACM Trans. Math. Softw. 31 (3) (2005) 397–423.
[36] S. Muralikrishnan, T. Bui-Thanh, J.N. Shadid, A multilevel approach for trace system in HDG discretizations, J. Comput. Phys. 407 (2020) 109240.
[37] C.A. Kennedy, M.H. Carpenter, Diagonally implicit Runge-Kutta methods for ordinary differential equations. A review.
[38] W. Bangerth, R. Hartmann, G. Kanschat, deal. II – a general purpose object oriented finite element library, ACM Trans. Math. Software 33 (4) (2007) 24/1–24/27.
[39] G. Alzetta, D. Arndt, W. Bangerth, V. Boddu, B. Brands, D. Davydov, R. Gassmöller, T. Heister, L. Heltai, K. Kormann, et al., The deal. ii library, version 9.0, J. Numer. Math. 26 (4) (2018) 173–183.

[40] D. Biskamp, Magnetic reconnection in plasmas, Astrophys. Space Sci. 242 (1–2) (1996) 165–207.
[41] J.P. Goedbloed, R. Keppens, S. Poedts, Advanced Magnetohydrodynamics: With Applications To Laboratory and Astrophysical Plasmas, Cambridge University Press, 2010.
[42] J.N. Shadid, R.P. Pawlowski, J.W. Banks, L. Chacón, P.T. Lin, R.S. Tuminaro, Towards a scalable fully-implicit fully-coupled resistive mhd formulation with stabilized fe methods, J. Comput. Phys. 229 (20) (2010) 7649–7671.
[43] D. Knoll, L. Chacón, Coalescence of magnetic islands, sloshing, and the pressure problem, Phys. Plasmas 13 (3) (2006) 032307.
[44] P. Lin, J.N. Shadid, E.G. Phillips, J.J. Hu, E.C. Cyr, R.P. Pawlowski, A. Prokopenko, P. Tsuji, Performance of Scalable AMG-Based Preconditioners for MHD and Multifluid Plasma Simulations, Tech. rep. Sandia National Lab.(SNL-NM), Sandia, Albuquerque, NM (United States), 2017.
[45] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, SIAM J. Sci. Comput. 14 (2) (1993) 461–469.
[46] A. Bhattacharjee, Y.-M. Huang, H. Yang, B. Rogers, Fast reconnection in high-lundquist-number plasmas due to the plasmoid instability, Phys. Plasmas 16 (11) (2009) 112102.
[47] Y.-M. Huang, A. Bhattacharjee, Scaling laws of resistive magnetohydrodynamic reconnection in the high-lundquist-number, plasmoid-unstable regime, Phys. Plasmas 17 (6) (2010) 062104.
[48] J. Shadid, R. Pawlowski, L. Chacón, D. Knoll, Current sheet break-up via fast plasmoid formation in the island coalescence problem the ultra-high lundquist number regime ($S10^9$), in: APS Division of Plasma Physics Meeting Abstracts, Vol. 52, 2010, pp. CP9–152.
[49] S. Muralikrishnan, M.-B. Tran, T. Bui-Thanh, An improved iterative hdg approach for partial differential equations, J. Comput. Phys. 367 (2018) 295–321.
[50] M. Taylor, J. Tribbia, M. Iskandarani, The spectral element method for the shallow water equations on the sphere, J. Comput. Phys. 130 (1) (1997) 92–108.
[51] T.A. Davis, Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method, ACM Trans. Math. Software 30 (2) (2004) 196–199.
[52] M. Sermange, R. Temam, Some mathematical questions related to the mhd equations, Comm. Pure Appl. Math. 36 (5) (1983) 635–664.
[53] J.-F. Gerbeau, ProblÈMes MathÉMatiques Et NumÉRiques PosÉS Par la ModÉLisation de L'Electrolyse de L'Aluminium (Theses), Ecole des Ponts ParisTech, 1998.
[54] J.-F. Gerbeau, C. Le Bris, T. Leliévre, Mathematical Methods for the Magnetohydrodynamics of Liquid Metals, Numerical Mathematics and Scientific Computation (Ph.D.thesis), University of Minnesota, 2006.
[55] C. Amrouche, C. Bernardi, M. Dauge, V. Girault, Vector potentials in three-dimensional non-smooth domains, Math. Methods Appl. Sci. 21 (9) (1998) 823–864.
[56] V. Girault, P.-A. Raviart, Finite Element Methods for Navier-STokes Equations, Vol. 5, in: Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1986, theory and algorithms.
[57] P. Fernandes, G. Gilardi, Magnetostatic and electrostatic problems in inhomogeneous anisotropic media with irregular boundary and mixed boundary conditions, Math. Models Methods Appl. Sci. 07 (07) (1997) 957–991.
[58] G.-D. Zhang, Y. He, D. Yang, Analysis of coupling iterations based on the finite element method for stationary magnetohydrodynamics on a general domain, Comput. Math. Appl. 68 (7) (2014) 770–788.
[59] M.D. Gunzburger, A.J. Meir, J.S. Peterson, On the existence, uniqueness, and finite element approximation of solutions of the equations of stationary, incompressible magnetohydrodynamics, Math. Comp. 56 (194) (1991) 523–563.
[60] A.J. Meir, P.G. Schmidt, Analysis and numerical approximation of a stationary mhd flow problem with nonideal boundary, SIAM J. Numer. Anal. 36 (4) (1999) 1304–1332.
[61] E. Zeidler, Applied Functional Analysis: Applications To Mathematical Physics, Vol. 108, Springer Science & Business Media, 2012.
[62] P.G. Ciarlet, Linear and Nonlinear Functional Analysis with Applications, Vol. 130, Siam, 2013.
[63] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Element Methods, Computational Mathematics, Vol. 15, Springer–Verlag, Berlin, 1991.
[64] S. Badia, R. Codina, R. Planas, Analysis of an unconditionally convergent stabilized finite element formulation for incompressible magnetohydrodynamics, Arch. Comput. Methods Eng. 22 (4) (2015) 621–636.
[65] R. Hiptmair, Finite elements in computational electromagnetism, Acta Numer. 11 (2002) 237–339.
[66] I. Faragó, Note on the convergence of the implicit euler method, in: I. Dimov, I. Faragó, L. Vulkov (Eds.), Numerical Analysis and Its Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 1–11.