Preconditioning of Hybridizable Discontinuous Galerkin Discretizations of the Navier–Stokes Equations

by

Abdullah Ali Sivas

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Applied Mathematics

Waterloo, Ontario, Canada, 2021

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Matthew G. Knepley Professor, Dept. of Computer Science and Engineering, University at Buffalo
Supervisor(s):	Sander Rhebergen Professor, Dept. of Applied Mathematics, University of Waterloo
Internal Member:	Edward R. Vrscay Professor, Dept. of Mathematics, University of Waterloo
Internal-External Member:	Justin W. L. Wan Professor, Dept. of Computer Science, University of Waterloo
Other Member(s):	Hans De Sterck Professor, Dept. of Applied Mathematics, University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Chapter 1 contains parts solely authored by Abdullah Ali Sivas, in addition to co-authored parts which are published or submitted for publication. Chapter 2 is published [150] and Chapter 3 is submitted for publication [146] (see https://arxiv.org/abs/2010.11130). Abdullah Ali Sivas is the sole author of Chapters 4 and 5 which were written under the supervision of Dr. Sander Rhebergen. Below, I explain my contributions to Chapters 2 and 3 in more detail.

Chapter 2: This research is a product of collaboration between Abdullah Ali Sivas, Dr. Ben Southworth and Dr. Sander Rhebergen, under the supervision of Dr. Sander Rhebergen. Abdullah Ali Sivas has contributed to Theorems 2.2.3 to 2.2.5 in collaboration with Dr. Ben Southworth. Abdullah Ali Sivas also contributed the numerical results. The document has intellectual contributions from all authors.

Chapter 3: The research presented here is primarily done by Abdullah Ali Sivas under the supervision of Dr. Sander Rhebergen. Lemma 3.2.1 is a result of collaboration between Abdullah Ali Sivas and Dr. Ben Southworth. The article was first drafted by Abdullah Ali Sivas and it evolved to its final form through proofreading, rewriting, and other intellectual contributions of all authors.

Abstract

The incompressible Navier–Stokes equations are of major interest due to their importance in modelling fluid flow problems. However, solving the Navier–Stokes equations is a difficult task. To address this problem, in this thesis, we consider fast and efficient solvers. We are particularly interested in solving a new class of hybridizable discontinuous Galerkin (HDG) discretizations of the incompressible Navier–Stokes equations, as these discretizations result in exact mass conservation, are locally conservative, and have fewer degrees of freedom than discontinuous Galerkin methods (which is typically used for advection dominated flows). To achieve this goal, we have made various contributions to related problems, as I discuss next.

Firstly, we consider the solution of matrices with 2×2 block structure. We are interested in this problem as many discretizations of the Navier–Stokes equations result in block linear systems of equations, especially discretizations based on mixed-finite element methods like HDG. These systems also arise in other areas of computational mathematics, such as constrained optimization problems, or the implicit or steady state treatment of any system of PDEs with multiple dependent variables. Often, these systems are solved iteratively using Krylov methods and some form of block preconditioner. Under the assumption that one diagonal block is inverted exactly, we prove a direct equivalence between convergence of 2×2 block preconditioned Krylov or fixed-point iterations to a given tolerance, with convergence of the underlying preconditioned Schur-complement problem. In particular, results indicate that an effective Schur-complement preconditioner is a necessary and sufficient condition for rapid convergence of 2×2 block-preconditioned GMRES, for arbitrary relative-residual stopping tolerances. A number of corollaries and related results give new insight into block preconditioning, such as the fact that approximate block-LDU or symmetric block-triangular preconditioners offer minimal reduction in iteration over block-triangular preconditioners, despite the additional computational cost. We verify the theoretical results numerically on an HDG discretization of the steady linearized Navier–Stokes equations. The findings also demonstrate that theory based on the assumption of an exact inverse of one diagonal block extends well to the more practical setting of inexact inverses.

Secondly, as an initial step towards solving the time-dependent Navier–Stokes equations, we investigate the efficiency, robustness, and scalability of approximate ideal restriction (AIR) algebraic multigrid as a preconditioner in the *all-at-once* solution of a space-time HDG discretization of the scalar advection-diffusion equation. The motivation for this study is two-fold. First, the HDG discretization of the velocity part of the momentum block of the linearized Navier–Stokes equations is the HDG discretization of the vector advection-

diffusion equation. Hence, efficient and fast solution of the advection-diffusion problem is a prerequisite for developing fast solvers for the Navier–Stokes equations. The second reason to study this all-at-once space-time problem is that the time-dependent advection-diffusion equation can be seen as a "steady" advection-diffusion problem in $(d+1)$-dimensions and AIR has been shown to be a robust solver for steady advection-dominated problems. We present numerical examples which demonstrate the effectiveness of AIR as a preconditioner for time-dependent advection-diffusion problems on fixed and time-dependent domains, using both slab-by-slab and all-at-once space-time discretizations, and in the context of uniform and space-time adaptive mesh refinement. A closer look at the geometric coarsening structure that arises in AIR also explains why AIR can provide robust, scalable space-time convergence on advective and hyperbolic problems, while most multilevel parallel-in-time schemes struggle with such problems.

As the final topic of this thesis, we extend two state-of-the-art preconditioners for the Navier–Stokes equations, namely, the pressure convection-diffusion and the grad-div/augmented Lagrangian preconditioners to HDG discretizations. Our preconditioners are simple to implement and our numerical results show that these preconditioners are robust in h and only mildly dependent on the Reynolds numbers.

Acknowledgements

It has been a strenuous journey which seemed never-ending at times. I want to thank everyone who encouraged me to push forward.

My supervisor, Dr. Sander Rhebergen, was always patient and supportive. He is one of the most diligent and meticulous people I have ever met. I am grateful to him for the guidance he provided.

Dr. Ben Southworth is a great research partner and we had many great talks. I would like to thank him for being always ready to engage in meaningful and scholarly discussions.

To the great friends I made here, Giselle Sosa Jones, Tamás Horváth, Keegan Kirk, Eve Workman, Andrew Giuliani, Greg Wang and Krishna Dutt; their influence in my life and my appreciation of their company can not be overstated. Also to Dr. Lilia Krivodonova; she is one of the smartest and funnest people I know, TA'ing for her courses taught me a lot, and I appreciate our discussions greatly.

I want to thank my committee members: Dr. Knepley, Dr. Vrscay, Dr. Wan and Dr. De Sterck for their time, feedback and suggestions.

Chapter 2 is reprinted with some modifications from the following article

> B. S. Southworth, A. A. Sivas, and S. Rhebergen, *On fixed-point, Krylov, and* 2×2 *block preconditioners for nonsymmetric problems*, SIAM Journal on Matrix Analysis and Applications, 41 (2020), pp. 871–900. https://doi.org/10.1137/19M1298317

with permission from Society of Industrial and Applied Mathematics (SIAM). Chapter 3 is reprinted from the following preprint

> A. A. Sivas, B. S. Southworth, and S. Rhebergen, *AIR algebraic multigrid for a space-time hybridizable discontinuous Galerkin discretization of advection(-diffusion)*, 2020. https://arxiv.org/abs/2010.11130

which is currently under review for publication in the SIAM Journal on Scientific Computing. I would like to thank SIAM for granting their permission to use these materials.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The research presented in this thesis is on efficient solution techniques for the numerical solution of the Navier–Stokes equations. The Navier–Stokes equations are useful in modelling many fluid flows of interest and the efficient solution of these equations is an increasingly important area of research. The goal of the research in this thesis is parameter-robust preconditioners for linear systems resulting from the hybridizable discontinuous Galerkin (HDG) discretization [89, 132] of the Navier–Stokes equations. The details of the HDG discretizations are discussed in following chapters, while in the remainder of this chapter, we provide a gentle discussion to emphasize the research problem.

The time-dependent incompressible Navier–Stokes problem is given by

$$\partial_t \vec{u} - \nu \nabla^2 \vec{u} + (\vec{u} \cdot \nabla)\vec{u} + \nabla p = \vec{f} \qquad \text{in } \Omega, \tag{1.1a}$$

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

$$\vec{u}|_{t=0} = \vec{u}_0, \qquad \text{in } \Omega, \tag{1.1c}$$

$$\vec{u} = \vec{g}_D \qquad \text{on } \partial\Omega_D, \tag{1.1d}$$

$$\nu \frac{\partial \vec{u}}{\partial \vec{n}} - \vec{n} p = 0 \qquad \text{on } \partial\Omega_N, \tag{1.1e}$$

where $\vec{u}$ is a vector-valued function representing the velocity of a fluid, the scalar function p represents the kinematic pressure, ν is a given constant called the kinematic viscosity, $\vec{f}$ is the given source term, $\vec{g}_D$ is given boundary data, $\Omega \subset \mathbb{R}^d$ is the domain of the problem in dimension $d = 2, 3$, the boundary of Ω is partitioned as $\partial\Omega = \partial\Omega_D \bigcup \partial\Omega_N$ with $\partial\Omega_D \bigcap \partial\Omega_N = \emptyset$ and $\vec{n}$ denotes the outward normal vector to the boundary. In case of large ν, there is a stable steady state solution as $t \to \infty$ which can be obtained by solving

the stationary Navier–Stokes problem:

$$-\nu\nabla^2\vec{u} + (\vec{u}\cdot\nabla)\vec{u} + \nabla p = \vec{f} \qquad \text{in } \Omega, \tag{1.2a}$$

$$\nabla\cdot\vec{u} = 0 \qquad \text{in } \Omega, \tag{1.2b}$$

$$\vec{u} = \vec{g}_D \qquad \text{on } \partial\Omega_D, \tag{1.2c}$$

$$\nu\frac{\partial\vec{u}}{\partial\vec{n}} - \vec{n}p = 0 \qquad \text{on } \partial\Omega_N. \tag{1.2d}$$

Equation (1.2a) is called the momentum equation and Equation (1.2b) is called the continuity equation. Note that, in the absence of Neumann boundary conditions, i.e. if $\partial\Omega = \partial\Omega_D$, the pressure p is only unique up to a constant. Therefore, if $\partial\Omega = \partial\Omega_D$, we impose that the pressure mean on Ω is zero for the problem to be well-posed.

Usually the equations are normalised with respect to the size of the domain and the magnitude of the velocity to measure relative contributions of convection and diffusion. Let L be the characteristic length of the domain, then $\vec{\xi} = \vec{x}/L$ are the points in a normalised domain where $\vec{x}$ are the points in Ω. Furthermore, let U be some reference value for the magnitude of the velocity. We non-dimensionalize the velocity and pressure according to $\vec{u} = U\vec{u}_*$ and $p(L\vec{\xi}) = U^2 p_*(\vec{\xi})$, respectively, where $*$ denotes the dimensionless variable. Substituting these into Equation (1.2), we get

$$-\frac{1}{Re}\nabla^2\vec{u}_* + (\vec{u}_*\cdot\nabla)\vec{u}_* + \nabla p_* = \frac{L}{U^2}\vec{f} \quad \text{in } \Omega. \tag{1.3}$$

Here $Re := UL/\nu$ is the *Reynolds number* which is used to measure the relative contributions of convection and diffusion. Assuming L and U are chosen suitably, then $Re \leq 1$ means that the flow is diffusion dominated. As Re grows, the flow becomes more convection dominated and it approaches the incompressible Euler equations as $Re \to \infty$. We will not directly use this form of the equations, however we will repeatedly refer to the Reynolds number.

The Navier–Stokes equations can be solved using non-linear iterations, solving a linear problem at each step. Given an initial guess $(\vec{u}_0, p_0)$, a sequence of iterates are computed which converges to the solution. These iterations most commonly take the form of Newton or Picard linearizations. In this thesis, we use Picard iterations due to their large radius of convergence and simplicity in implementation (see [37] and references therein for a discussion on linearising the convection term $(\vec{u}\cdot\nabla)\vec{u}$).

Now we introduce the linearised Navier–Stokes equations, also known as the Oseen

equations,

$$-\nu\nabla^2\vec{u} + (\vec{w}\cdot\nabla)\vec{u} + \nabla p = \vec{f} \quad \text{in } \Omega, \tag{1.4a}$$
$$\nabla\cdot\vec{u} = 0 \quad \text{in } \Omega, \tag{1.4b}$$

in which the operator $(\vec{u}\cdot\nabla)$ in Equation (1.2) has been replaced by $(\vec{w}\cdot\nabla)$ in Equation (1.4a) and where $\vec{w}$ is a given velocity field. We define our Picard iterations by choosing $\vec{w} = \vec{u}_{n-1}$ and solving

$$-\nu\nabla^2\vec{u}_n + (\vec{u}_{n-1}\cdot\nabla)\vec{u}_n + \nabla p_n = \vec{f} \quad \text{in } \Omega, \tag{1.5}$$
$$\nabla\cdot\vec{u}_n = 0 \quad \text{in } \Omega, \tag{1.6}$$

to generate a sequence of iterates $(\vec{u}_i, p_i)$, for $i = 1, 2, \cdots, k$. At each step, we compute $\varepsilon = \max\{||\vec{u}_i - \vec{u}_{i-1}||/||\vec{u}_i - \vec{u}_0||, ||p_i - p_{i-1}||/||p_i - p_0||\}$. When ε is less than some given tolerance, we stop the iteration and set $\vec{u} = \vec{u}_k$. Picard iterations are not likely to converge if used to solve the strong form of the Navier–Stokes problem, however, some convergence results for the Picard iterations applied to weak formulations of the Navier–Stokes problem are available, for example, see [95, 37]. Furthermore, in the case of elliptic nonlinear weak problems, we know that the number of Picard iterations required to convergence will be independent of the mesh size given that the mesh is fine enough and the resulting linear problem is solved sufficiently accurately [66, 70]. While the Navier–Stokes problem is not elliptic, we observe the same phenomenon; as we refine the mesh, the number of Picard iterations to convergence stay fixed.

The finite element method does not directly discretize the Oseen equations. Instead, the finite element method discretizes the weak formulation of this problem. Letting $V_{\vec{\alpha}} = \{\vec{u} \in [\mathcal{H}^1(\Omega)]^d | \vec{u} = \vec{\alpha} \text{ on } \partial\Omega_D\}$ and $Q = L_2(\Omega)$ with $\mathcal{H}^1(\Omega) = \{u \in L_2(\Omega) | \nabla u \in L_2(\Omega)\}$, the weak formulation of the Oseen equations is given by:

Find $\vec{u} \in V_{g_D}$ and $p \in Q$ such that

$$a(\vec{u}, \vec{v}) + n(\vec{w}; \vec{u}, \vec{v}) - b(p, \vec{v}) = (\vec{f}, \vec{v})_\Omega \quad \text{for all } \vec{v} \in V_0, \tag{1.7}$$
$$b(q, \vec{u}) = 0 \quad \text{for all } q \in Q, \tag{1.8}$$

where

$$a(\vec{u}, \vec{v}) = \nu \int_\Omega \nabla\vec{u} \colon \nabla\vec{v}\, \mathrm{d}\vec{x},$$
$$n(\vec{w}; \vec{u}, \vec{v}) = \int_\Omega (\vec{w}\cdot\nabla\vec{u})\cdot\vec{v}\, \mathrm{d}\vec{x},$$
$$b(p, \vec{v}) = \int_\Omega p(\nabla\cdot\vec{v})\, \mathrm{d}\vec{x},$$

and where the dyadic operator : is the double dot product, and $(\cdot,\cdot)$ is the L^2 inner product.

Conforming finite element methods are obtained by introducing finite-dimensional subspaces $V_{\vec{\alpha}}^h \subset V_{\vec{\alpha}}$ and $Q^h \subset Q$. Here h signifies that the cardinality of these spaces depends on the grid size. See [22, Chapters 0-2] for a good and in-depth explanation. The discrete problem is: find $\vec{u_h} \in V_{g_D}^h$ and $p_h \in Q^h$ such that

$$a(\vec{u_h},\vec{v_h}) + n(\vec{w};\vec{u_h},\vec{v_h}) - b(p_h,\vec{v_h}) = (\vec{f},\vec{v_h})_\Omega \quad \text{for all } \vec{v_h} \in V_0^h, \tag{1.9a}$$

$$b(q_h,\vec{u_h}) = 0 \quad \text{for all } q_h \in Q^h. \tag{1.9b}$$

Remark 1.0.1. *Here we would like to clarify what we mean by a grid. A grid is a non-overlapping subdivision of the domain. We further impose the following conditions:*

- *shape-regularity: the ratios of the diameters of the inscribed and the circumscribed circles for each element are bounded; and*
- *quasi-uniformity: the ratio of the sizes of any two elements in the subdivision is bounded.*

An example of a grid is given in Figure 1.1.

To obtain the underlying matrix formulation of this problem, let $\{\vec{\phi_i}\}$ and $\{\psi_i\}$ be bases, respectively, of the spaces V^h and Q^h so that $\exists \mathbf{u}_i, \mathbf{p}_j$ s.t. $\vec{u}_h = \sum_{i=1}^n \mathbf{u}_i \vec{\phi_i}$ and $p_h = \sum_{i=1}^m \mathbf{p}_i \psi_i$. Since we are free to pick the test functions $\vec{v}_h$ and q_h from their respective spaces, we can write the discrete problem Equation (1.9) as a square linear system of the form

$$\begin{bmatrix} A(\nu)+N(\vec{w}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} U \\ P \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix}, \tag{1.10}$$

where $U = [\mathbf{u}_0, \cdots, \mathbf{u}_n]^T$ and $P = [\mathbf{p}_0, \cdots, \mathbf{p}_m]^T$. We will refer to the matrices A, N and B, respectively, as the discrete vector Laplacian, the discrete vector convection and the discrete divergence (the weak gradient is the adjoint of the weak divergence operator so we simply denote it as B^T). These are named after the continuous operators they have been derived from, namely, $a(\vec{u},\vec{v}), n(\vec{w};\vec{u},\vec{v})$ and $b(p,\vec{v})$ respectively. Note that while A is symmetric, N is non-symmetric, and so the full system matrix is also non-symmetric.

We conclude this section by summarizing the discrete problem in Algorithm 1 and pointing out the focus of my research. Picard iterations are used to solve the non-linear Navier–Stokes equations. At every Picard iteration we need to solve the discrete form

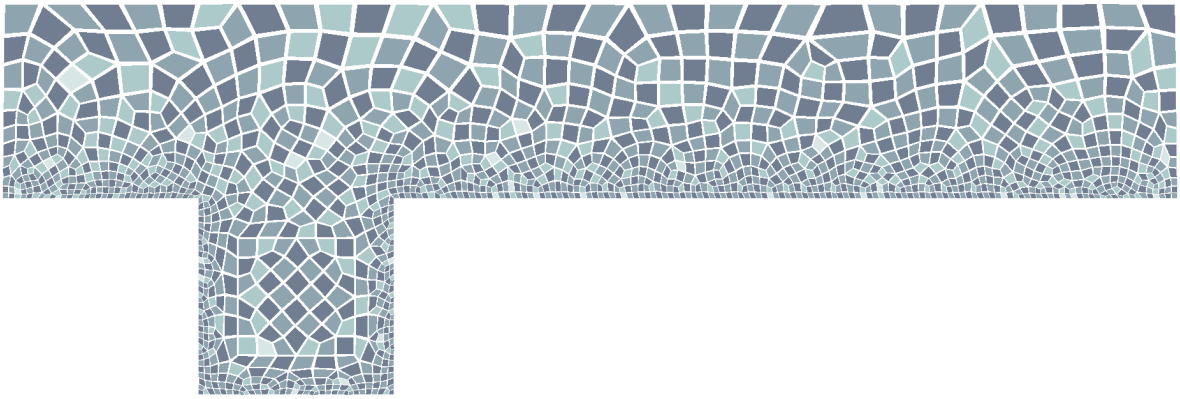


Figure 1.1: An example of a grid.

of the linearised Navier–Stokes equations (Equation (1.10)). Often, many Picard iterations are required to obtain a solution for the non-linear Navier–Stokes problem. Since Equation (1.10) needs to be solved at each of these Picard iterations, the efficiency of the algorithm strongly depends on the efficiency of the solution of these linear systems. However, efficiency depends highly on two problem parameters, namely, the mesh size h and the Reynolds number Re. The ultimate goal of this thesis is to find a solver for Equation (1.10) that is robust both in h and Re.

1.1 Convergence of Krylov subspace methods

In this section, we discuss some error bounds of Krylov subspace methods, particularly GMRES (Generalized Minimal RESidual method) [139, 137, 138] for the solution of linear systems of the form $Ax = b$. A Krylov subspace method is a projection method: given the linear system $Ax = b$ with $A \in \mathbb{R}^{n\times n}$, initial guess x_0 and two m-dimensional subspaces K_m and L_m, we seek

$$\hat{x} \in x_0 + K_m \text{ such that } b - A\hat{x} \perp L_m,$$

Algorithm 1 An algorithm to solve the Navier–Stokes equations

Initialize $i = 1$, $rres = 1$. Given an initial guess $(\vec{u}_0, p_0)$

while $rres > tol$ and $i < maxit$ **do**

Create and solve the linear system

$$\begin{bmatrix} A(\nu) + N(\vec{u}_{i-1}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} U^{(i)} \\ P^{(i)} \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix}.$$

Construct $\vec{u}_j = \sum_{i=1}^{n} \mathbf{u}_i^{(j)} \vec{\phi}_i$ and $p_j = \sum_{i=1}^{m} \mathbf{p}_i^{(j)} \psi_i$ using $U^{(j)} = \left[\mathbf{u}_0^{(j)}, \cdots, \mathbf{u}_n^{(j)}\right]^T$ and $P^{(j)} = \left[\mathbf{p}_0^{(j)}, \cdots, \mathbf{p}_m^{(j)}\right]^T$

Compute $rres = \max\{||\vec{u}_i - \vec{u}_{i-1}||/||\vec{u}_i - \vec{u}_0||, ||p_i - p_{i-1}||/||p_i - p_0||\}$ and increment i.

end while

where $K_m = \text{span}\{r_0, Ar_0, \ldots, A^{m-1}r_0\}$ with $r_0 = b - Ax_0$. As a result, Krylov subspace methods look for an approximate solution x_m in the search space K_m with the condition that the residual vector $b - Ax_m$ is orthogonal to L_m. Depending on the choice of the subspace L_m, we obtain different Krylov subspace methods ($L_m = AK_m$ for GMRES). Moreover, the approximate solution obtained at the m-th iteration of a Krylov subspace method can be written as $x_m = x_0 + p_{m-1}(A)r_0$ where p_{m-1} is a $m-1$-st degree consistent polynomial, i.e. $p_{m-1}(0) = 1$.

The discussion on the error bounds is the foundation of work presented in this thesis, as some of these error bounds are very similar to the error bounds of finite element methods which allows us to use functional analysis tools to develop and analyze preconditioners. The classical and most well-known results on the convergence of Krylov subspace methods are the error bounds of the conjugate gradient method [138, pg. 176] given in Theorems 1.1.1 and 1.1.2.

Theorem 1.1.1. *Given the problem $Ax = b$, where A is a real symmetric positive definite matrix, after k steps of the conjugate gradient method, the following bound holds*

$$\left\|x - x^{(k)}\right\|_A \leq \min_{p_k \in \Pi_k, p_k(0)=1} \left\|p_k(A)(x - x^{(0)})\right\|_A \leq \min_{p_k \in \Pi_k, p_k(0)=1} \max_j |p_k(\lambda_j)| \left\|(x - x^{(0)})\right\|_A,$$

where $x^{(k)}$ and $x^{(0)}$ are, respectively, k-th and 0-th iterates, λ_i are the eigenvalues of A, and Π_k is the set of real polynomials up to and of degree k.

Theorem 1.1.2. *Given the problem $Ax = b$, where A is a real symmetric positive definite matrix, after k steps of the conjugate gradient method, the following bound holds*

$$\left\|x - x^{(k)}\right\|_A \leq 2\left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}\right)^k \left\|x - x^{(0)}\right\|_A,$$

where $x^{(k)}$ and $x^{(0)}$ are, respectively, k-th and 0-th iterates, and $\kappa(A) = \lambda_{\max}(A)/\lambda_{\min}(A)$ which is called the condition number of A.

Theorem 1.1.2 implies that well-conditioned problems (e.g. the condition number κ of A is small) will reach desired tolerance levels rapidly. However, it should be noted that a large condition number does not necessarily imply slow convergence. For example, the conjugate gradient method will solve the problem *exactly* in at most $|\sigma(A)|$ iterations, where $\sigma(A)$ is the set of eigenvalues (spectrum) of A (see Theorem 1.1.1). Hence, if the coefficient matrix has only two eigenvalues, the conjugate gradient method will converge in two iterations independent of the condition number.

For finite element problems, the condition number of the coefficient matrix A depends on the mesh size h. For example, in the case of a continuous Galerkin discretization of the Poisson problem, $\kappa(A) = O(h^{-2})$. Keeping in mind that the error bound in Theorem 1.1.2 is a pessimistic upper bound (and Theorem 1.1.1), this observation on $\kappa(A)$ hints that the number of iterations to convergence will increase as the grid is refined, i.e., as $h \to 0$. Numerical experiments confirm this prediction; the number of iterations to convergence doubles as the grid size h is halved. It is desirable to find a way to alleviate, or all together eliminate, h-dependence, as many practical problems require very fine grids with h small. We achieve this through preconditioning. A preconditioner, without loss of generality, is an operator P such that $\kappa(P^{-1/2}\mathbf{A}P^{-1/2}) \ll \kappa(\mathbf{A})$, see [119] for a survey. In this sense, the matrix $\mathbf{A}$ itself is the "perfect" preconditioner as $\kappa(\mathbf{A}^{-1}\mathbf{A}) = 1$ independent of h and any other problem parameters. However, inverting $\mathbf{A}$ is equivalent to solving the original system directly, therefore, defeating the purpose of iterative solvers. Hence, we additionally want preconditioners to be cheap to apply.

We can use the concept of *spectral equivalence* to develop and rigorously analyze preconditioners for finite element discretizations of some PDEs. Two symmetric positive definite matrices $A, P \in \mathbb{R}^{n\times n}$ are called spectrally equivalent if there exist constants $C, c > 0$ independent of some problem parameters such that

$$c \leq \frac{\langle x, Ax\rangle}{\langle x, Px\rangle} \leq C \qquad \forall x \in \mathbb{R}^n.$$

Generally, the problem parameter of interest is the grid size h, hence, spectral equivalence is commonly defined with respect to h.

We now include a very useful theorem which ties the concept of spectral equivalence to the condition number.

Theorem 1.1.3 (Rayleigh, can be found in [79] pp. 234-235)**.** *The eigenvalues of an $n \times n$ Hermitian matrix H, given in order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, can be characterized as*

$$\lambda_1 = \max_{0 \neq x \in \mathbb{C}^n} \frac{(x, Hx)}{(x, x)}, \quad \lambda_n = \min_{0 \neq x \in \mathbb{C}^n} \frac{(x, Hx)}{(x, x)}.$$

Using Theorem 1.1.3 and the definition of the condition number, we see that, given two symmetric positive definite matrices A and P of the same dimensions

$$\kappa(P^{-1/2}AP^{-1/2}) = \frac{\lambda_{\max}(P^{-1/2}AP^{-1/2})}{\lambda_{\min}(P^{-1/2}AP^{-1/2})},$$

with

$$\lambda_{\min}(P^{-1/2}AP^{-1/2}) = \min_{0 \neq x \in \mathbb{C}^n} \frac{(x, P^{-1/2}AP^{-1/2}x)}{(x, x)} = \min_{0 \neq y \in \mathbb{C}^n} \frac{(y, Ay)}{(y, Py)},$$
$$\lambda_{\max}(P^{-1/2}AP^{-1/2}) = \max_{0 \neq x \in \mathbb{C}^n} \frac{(x, P^{-1/2}AP^{-1/2}x)}{(x, x)} = \max_{0 \neq y \in \mathbb{C}^n} \frac{(y, Ay)}{(y, Py)}.$$

Therefore, given spectrally equivalent symmetric positive definite matrices A and P, $\kappa(P^{-1/2}AP^{-1/2})$ will be bounded from above by the constant C/c independent of the grid size h and Theorem 1.1.2 implies that the conjugate gradient method will *asymptotically* converge in the same number of iterations independently of the grid size. Such a preconditioner P is called h-robust, or h-optimal.

Note that h-robustness itself may not be practical if the application of the preconditioner is expensive (see the discussion related to the choice $P = A$ above). Hence, we are motivated to seek cheap, h-robust preconditioners. For example, for many discretizations of the Poisson problem, it is well-known that appropriate multigrid cycles are spectrally equivalent to the coefficient matrix (see, for example, [50, pp. 91-112]). This fact, together with linear computational complexity of multigrid methods [138, pg. 443] with respect to the mesh size, further motivates research in this direction.

Unfortunately, we can not appeal to these results directly in the case of the Navier–Stokes equations because discretizations of this problem give rise to non-symmetric and

indefinite coefficient matrices. Hence, the bounds in Theorems 1.1.1 and 1.1.2 are not applicable. Fortunately, many Krylov subspace methods have been developed for solving linear systems of equations where the coefficient matrix may be non-symmetric and indefinite. Among these, we consider GMRES (and flexible GMRES) [139, 137] due to the availability of many error bounds for these methods as we discuss next. We want to note that if the coefficient matrix is symmetric then GMRES produces the same iterates as MINRES (MINimal RESidual method) [120] (albeit at a higher computational cost), so the bounds discussed below are also valid for MINRES.

The earliest GMRES error bounds (Theorem 1.1.4) are due to [48, 42]. Elman's PhD thesis [48] in 1982 is the first time the bounds are published, one year after the same bounds appeared in his paper with Eisenstat and Schultz [42]. In both cases, however, these bounds were presented for GCR. Saad and Schultz published their paper on GMRES [139] almost three years later. They proved that GCR iterates are exactly GMRES iterates. Hence, Elman's bounds are also valid for GMRES.

Theorem 1.1.4 (Elman's bound)**.** *Let A be a positive real matrix, that is, its symmetric part $M = \frac{1}{2}(A + A^T)$ is positive definite. Define its skew-symmetric part $R = \frac{1}{2}(A - A^T)$. If $\{r_k\}_{k=0}^N$ are the residuals generated by GMRES, then*

$$||r_k||_2 \leq \min_{p_k \in \Pi_k, p_k(0)=1} ||p_k(A)||_2 ||r_0||_2 \leq \left[1 - \frac{\lambda_{\min}(M)^2}{\lambda_{\max}(A^T A)}\right]^{k/2} ||r_0||_2,$$

and

$$||r_k||_2 \leq \left[1 - \frac{\lambda_{\min}(M)^2}{\lambda_{\min}(M)\lambda_{\max}(M) + \rho(R)^2}\right]^{k/2} ||r_0||_2,$$

where $\rho(R)^2 = ||R^T R||_2$. If A is diagonalizable, i.e. $A = X\Lambda X^{-1}$, where X is the matrix whose columns are the eigenvectors of A and Λ is a diagonal matrix whose diagonal entries are the eigenvalues, then

$$\frac{||r_k||_2}{||r_0||_2} \leq ||X||_2 ||X^{-1}||_2 \min_{p_k \in \Pi_k, p_k(0)=1} \max_{\Lambda_{jj}} |p_k(\Lambda_{jj})|.$$

Furthermore, this bound can be relaxed such that if the set $\mathcal{E}$ contains the eigenvalues of A then

$$\frac{||r_k||_2}{||r_0||_2} \leq ||X||_2 ||X^{-1}||_2 \min_{p_k \in \Pi_k, p_k(0)=1} \max_{\lambda \in \mathcal{E}} |p_k(\lambda)|.$$

Proof. See [48, Theorems 5.4, 5.9] and [42, Theorems 3.3, 3.4]. □

Notice that the bounds given above require the coefficient matrix to be positive real or diagonalizable. We can relax the conditions on the coefficient matrix A by introducing field of values (FOV), $\mathcal{W}(A) = \{\langle Ax, x\rangle : x \in \mathbb{C}^n, ||x|| = 1\}$. It is known that $\mathrm{Re}\mathcal{W}(A) = \mathcal{W}(\frac{A+A^T}{2})$ and $\mathrm{Im}\mathcal{W}(A) = \mathcal{W}(\frac{A-A^T}{2})$, hence, if $0 \notin \mathcal{W}(A)$ Theorem 1.1.4 can be rewritten as follows:

Theorem 1.1.5 (Elman's FOV bound)**.** *Let A be a square matrix such that $0 \notin \mathcal{W}(A)$, then the residual generated at the k-th step of GMRES satisfies*

$$\frac{||r_k||}{||r_0||} \leq \left[1 - \frac{\mu(A)^2}{||A||^2}\right]^{k/2},$$

where $\mu(A) = \min\{|z| : z \in \mathcal{W}(A)\}$.

If the ratio $\mu(A)^2/||A||^2$ can be bounded by a constant independent of problem parameters (e.g., h and Re for the Navier–Stokes equations), we can guarantee robust performance. If A is positive real, then $\mu(A) = \lambda_{\min}(\frac{A+A^T}{2})$ so Theorems 1.1.4 and 1.1.5 are equivalent, but the latter is applicable under slightly more general conditions. However, these bounds can be very pessimistic for some problems. In such cases, Theorem 1.1.5 is predictive of the performance of GMRES in an asymptotical sense and usually convergence is observed much earlier than predicted. Starke offers another bound [151] and Eiermann and Ernst [41] show that it is an improvement over Theorem 1.1.5.

Theorem 1.1.6 (Starke)**.** *Let A be a square matrix such that $0 \notin \mathcal{W}(A)$, which implies $0 \notin \mathcal{W}(A^{-1})$, then the residual generated at the k-th step of GMRES satisfies*

$$\frac{||r_k||_2}{||r_0||_2} \leq \left[1 - \mu(A)\mu(A^{-1})\right]^{k/2},$$

where $\mu(A) = \min\{|z| : z \in \mathcal{W}(A)\}$.

Proof. See [151, Theorem 3.2] or [41, Theorem 6.1, Corollary 6.2]. □

Remark 1.1.1. *We remark that Starke's paper considers a GMRES which minimizes a different norm than the usual (discrete) ℓ_2-norm, similar to weighted GMRES of Pestana [123]. The results demonstrate optimal convergence with respect to the chosen norm. However, this is not a big problem as noted in [93, 92]. The idea is to minimize GMRES*

residuals in a norm induced by one of the inner products, but measure these residuals in the norm induced by the other inner product. Depending on the pair of norms, their equivalence constants (or the ratio of the constants) may depend on h. Nevertheless, the equivalence constants come in only by an additive logarithmic value into the estimate of the number of iterations [93, pg. 581]. Therefore, optimal convergence in one norm implies almost optimal convergence in the l_2-norm.

Another improvement over Theorem 1.1.5 is by Beckermann et al. [9]. Their idea is to find a circular segment $\mathcal{K}$ such that $\mathbb{C} \supset \mathcal{K} \supset \mathcal{W}(A)$ and

$$\min_{p_k \in \Pi_k, p_k(0)=1} \|p(A)\| \leq C \min_{p_k \in \Pi_k, p_k(0)=1} \max_{z \in \mathcal{K}} |p(z)|,$$

with C a constant. As a result, they obtain an asymptotically sharper bound, see in particular [9, Corollary 2.4].

Theorem 1.1.7 (BGT Bound)**.** *Let A be a square matrix such that $0 \notin \mathcal{W}(A)$, and let $\beta \in (0, \frac{\pi}{2})$ with $\cos(\beta) = \mu(A)/||A||_2$ and $\mu(A)$ is as defined in Theorem 1.1.5. Then the residual generated at the k-th step of GMRES satisfies*

$$\frac{||r_k||_2}{||r_0||_2} \leq (2 + 2/\sqrt{3})(2 + \gamma_\beta)\gamma_\beta^k,$$

where

$$\gamma_\beta := 2 \sin\left(\frac{\beta}{4 - 2\beta/\pi}\right).$$

Proof. See [9]. □

Theorem 1.1.7 has recently been improved by Tichy and Liesen [99, Theorem 3.1]. The improvement is due to recent work by Crouzeix and Palencia [30] and the replacement of $||A||_2$ by $r(A) = \max\{|z| : z \in \mathcal{W}(A)\}$.

Theorem 1.1.8 (BGT Bound Improved)**.** *Let A be a square matrix such that $0 \notin \mathcal{W}(A)$, and let $\beta \in (0, \frac{\pi}{2})$ with $\cos(\beta) = \mu(A)/r(A)$ and $\mu(A)$ is as defined in Theorem 1.1.5. Then the residual generated at the k-th step of GMRES satisfies*

$$\frac{||r_k||_2}{||r_0||_2} \leq (1 + \sqrt{2})(2 + \gamma_\beta)\gamma_\beta^k,$$

where

$$\gamma_\beta := 2 \sin\left(\frac{\beta}{4 - 2\beta/\pi}\right).$$

Proof. See [99, 9, 30] for details. □

Tichy and Liesen, in addition to Theorem 1.1.8, give a tighter bound under stricter conditions on $\mathcal{W}(A)$. We will not repeat this bound here as it is not of interest to us.

Let $A_\xi x = b$ be a linear system that depends on a parameter ξ, and let P_ξ be a preconditioner for this problem. Then the theorems in this section, under certain conditions on $\mathcal{W}(P_\xi^{-1}A_\xi)$, show that it is possible to guarantee that GMRES applied to $P_\xi^{-1}A_\xi x = P_\xi^{-1}b$ will *asymptotically* converge in a fixed number of iterations independent of the parameter ξ. As mentioned previously, the ultimate goal of this thesis is to find a solver for Equation (1.10) that is robust both in h and Re.

1.2 Krylov subspace methods and block preconditioners

In Chapter 2, we explore the relations between block preconditioning and the corresponding convergence of fixed-point and Krylov methods applied to nonsymmetric systems of the form

$$A\mathbf{x} = \mathbf{b}, \qquad \mathbf{x}, \mathbf{b} \in \mathbb{R}^n, \quad A \in \mathbb{R}^{n\times n}, \tag{1.11}$$

where the matrix A has a 2×2 block structure,

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}. \tag{1.12}$$

Such systems arise in numerous areas, including mixed finite elements [14, 31, 91, 167], constraint optimization problems [141, 121, 40], and the solution of neutral particle transport [148]. More generally, the discretization of just about any systems of PDEs with multiple dependent variables can be expressed as a 2×2 block operator by the grouping of variables into two sets. Although iterative methods for saddle-point problems, in which $A_{22} = \mathbf{0}$, have seen extensive research, in this paper we take a more general approach, making minimal assumptions on the submatrices of A.

The primary contribution of Chapter 2 is to prove a direct equivalence between the convergence of a block-preconditioned fixed-point or Krylov iteration applied to Equation (1.11), with convergence of a similar method applied directly to a preconditioned Schur complement of A, where the Schur complements of A are defined as $S_{11} := A_{11} - A_{12}A_{22}^{-1}A_{21}$

and $S_{22} := A_{22} - A_{21}A_{11}^{-1}A_{12}$. In particular, results in Chapter 2 prove that a good approximation to the Schur complement of the 2×2 block matrix Equation (1.12) is a necessary and sufficient condition for rapid convergence of preconditioned GMRES applied to Equation (1.11), for arbitrary relative residual stopping tolerances.

The main assumption in derivations here is that at least one of A_{11} or A_{22} is nonsingular and that the action of its inverse can be computed. Although in practice it is often not advantageous to solve one diagonal block to numerical precision every iteration, it *is* typically the case that the inverse of at least one diagonal block can be reliably computed using some form of iterative method, such as multigrid. The theory developed in Chapter 2 provides a guide for ensuring a convergent and practical preconditioner for Equation (1.11). Once the iteration and convergence are well understood, the time to solution can be reduced by solving the diagonal block(s) to some tolerance. Numerical results further demonstrate how ideas motivated by the theory, where one block is inverted exactly, extend to inexact preconditioners.

1.2.1 Previous work

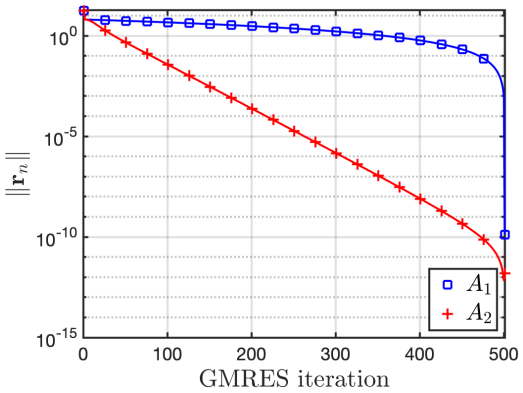
For nonsymmetric 2×2 block operators, most theoretical results in the literature are not necessarily indicative of practical performance. There is also a lack of distinction in the literature between a Krylov convergence result and a fixed-point convergence result.

Theoretical results on block preconditioning generally fall in to one of two categories. First are results based on the assumption that the inverse action of the Schur complement is available, and/or results that show an asymptotic equivalence between the preconditioned 2×2 operator and the preconditioned Schur complement. It is shown in [85, 114] that GMRES (or other minimal residual methods) is guaranteed to converge in two or four iterations for a block-triangular or block-diagonal preconditioned system, respectively, when the diagonal blocks of the preconditioner consist of a Schur complement and the respective complementary block of A (A_{11} or A_{22}). However, computing the action of the Schur complement inverse is generally very expensive. In [6], it is shown that if the minimal polynomial of the preconditioned Schur complement is degree k, then the minimal polynomial of the preconditioned 2×2 system is at most degree $k+1$. Although this does not require the action and inverse of the Schur complement, it is almost never the case that GMRES is iterated until the true minimal polynomial is achieved. As a consequence, the minimal polynomial equivalence also does not provide practical information on convergence of the 2×2 system, as demonstrated in the following example.

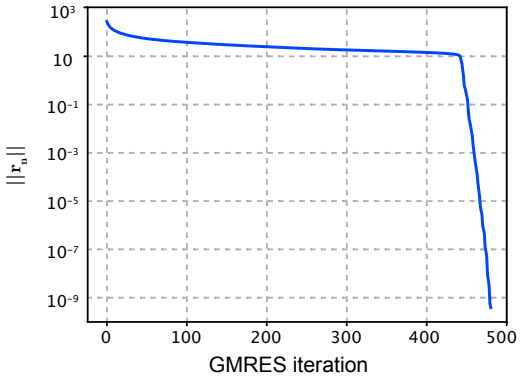
Example 1.2.1. *Define two matrices* $A_1, A_2 \in \mathbb{R}^{1000\times 1000}$,

$$A_1 := \begin{bmatrix} I_{500} & \mathbf{0} \\ \mathbf{0} & D_1 \end{bmatrix}, \qquad A_2 := \begin{bmatrix} I_{500} & \mathbf{0} \\ \mathbf{0} & D_2 \end{bmatrix},$$

where $D_1 \in \mathbb{R}^{500\times 500}$ *is tridiagonal with stencil* $[-1, 2, -1]$ *and* $D_2 \in \mathbb{R}^{500\times 500}$ *is tridiagonal with stencil* $[-1, 2.0025, -1]$. *Note that the minimal polynomials of* A_1 *and* A_2 *in the* ℓ^2*-norm have degree at most* $k = 501$. *Figure 1.2a shows results from applying GMRES with no restarts to* A_1 *and* A_2, *with right-hand side* $\mathbf{b} = (1, 2, ..., 1000)^T/1000$. *Note that neither operator reaches exact convergence in the first 500 iterations, indicating that the minimal polynomial in both cases is degree* $k = 501$. *However, despite having the same degree minimal polynomial (which is less than the size of the matrix), at iteration 250,* A_2 *has reached a residual of* $\|\mathbf{r}\| \approx 10^{-5}$, *while* A_1 *still has residual* $\|\mathbf{r}\| > 1$.



(a) Minimal polynomial does not necessarily provide practical information on convergence of a 2×2 system (Example 1.2.1).



(b) Eigenvalues do not necessarily provide practical information on convergence of a 2×2 system (Example 1.2.2).

Figure 1.2: Convergence of GMRES.

Second, many papers have used eigenvalue analyses in an attempt to provide more practical information on convergence. In the symmetric setting, this has proven effective (see, for example, [116]). Spectral analyses have also been done for various nonsymmetric 2×2 block matrices and preconditioners [45, 91, 5, 6, 94, 144] and eigenvectors for preconditioned operators derived in [124]. However, eigenvalue analyses are asymptotic, guaranteeing eventual convergence but, in the nonsymmetric setting, giving no guarantee of practical performance. In certain cases, a nonsymmetric operator is symmetric in

a non-standard inner product, and some papers have looked at block preconditioning in modified norms [125, 163, 127, 111] that yield self-adjointness. Nevertheless, there are many nonsymmetric problems that are not easily symmetrized and/or where eigenvalues provide little to no practical information on convergence of iterative methods. The following provides one such example in the discretization of differential operators. A formal analysis as in [69, 155] proves that for any set of eigenvalues, there is a matrix such that GMRES converges arbitrarily slowly.

Example 1.2.2. *Consider an upwind discontinuous Galerkin (DG) discretization of linear advection with Dirichlet inflow boundaries [24] and a velocity field* $\mathbf{b}(x, y) := (\cos(\pi y)^2, \cos(\pi x)^2)$ *(see Figure 5a in [107]); more generally, similar results hold for any velocity field with no cycles). In the appropriate ordering, the resulting matrix is block triangular, where "block" refers to the DG element blocks. Then, if we apply block-diagonal (Jacobi) preconditioning, the spectrum of the preconditioned operator is given by* $\sigma(M^{-1}A) = \{1\}$*, and the spectrum of the fixed-point iteration is given by* $\sigma(I - M^{-1}A) = \{0\}$*. Despite all zero eigenvalues, block-Jacobi preconditioned fixed-point or GMRES iterations on such a matrix can converge arbitrarily slowly, until the degree of nilpotency is reached and exact convergence is immediately obtained. Figure 1.2b shows convergence of DG block-Jacobi preconditioned GMRES applied to 2d linear transport, with* 200×200 *finite elements. Convergence occurs very rapidly at around 450 iterations (without restart), approximately the diameter of the mesh (as expected [107]).*

This is not the first work to recognize that eigenvalue analyses of nonsymmetric block preconditioners may be of limited practical use. Norm and field-of-values equivalence are known to provide more accurate measures of convergence for nonsymmetric operators, used as early as [106], and applied recently for specific problems in [13, 93, 101, 104]. Here we stay even more general, focusing directly on the relation between polynomials of a general preconditioned 2×2 system and the preconditioned Schur complement.

1.2.2 Overview of results of Chapter 2

A brief overview of theoretical contributions of Chapter 2 are listed next.

- Fixed-point and minimal residual Krylov iterations preconditioned with a 2×2 block-triangular, block-Jacobi, or approximate block-LDU preconditioner converge to a given tolerance $C\rho$ after n iterations if and only if an equivalent method applied to the underlying preconditioned Schur complement converges to tolerance ρ after n

iterations, for constant C (Section 2.2). Such results do *not* hold for general block-diagonal preconditioners [149].

- A symmetric block-triangular or approximate block-LDU preconditioner offers little to no improvement in convergence over block-triangular preconditioners, when one diagonal block is inverted exactly (Section 2.1.1.1). Numerical results demonstrate the same behaviour for inexact inverses, suggesting that symmetric block-triangular or block-LDU preconditioners are probably not worth the added computational cost in practice.

- The worst-case number of iterations for a block-Jacobi preconditioner to converge to a given tolerance ρ is twice the number of iterations for a block-triangular preconditioner to converge to $C\rho$, for some constant factor C (Section 2.2.3). Numerical results suggest that for non-saddle point problems (nonzero (2,2)-block), this double in iteration count is not due to the staircasing effect introduced in [59] for saddle-point problems.

- With an exact Schur-complement inverse, a *fixed-point* iteration with a block triangular preconditioner converges in two iterations, while a fixed-point iteration with a block-diagonal preconditioner does not converge (Section 2.1).

1.3 Iterative solution of time-dependent advection-diffusion equations

After discretization, the (linearized) Navier–Stokes equations can be written as the linear system in Equations (1.11) and (1.12). In Section 1.2, we discussed block-preconditioning for this linear system. One of the main assumptions there is that the action of the inverse of at least one of A_{11} or A_{22} can be computed. In practice, however, a good approximation to the action of the inverse of A_{11} or A_{22} is sufficient.

In the case of the time-dependent Navier–Stokes equations, A_{11} corresponds to the discretization of the vector advection-diffusion equation. For this reason, in Chapter 3, we consider novel preconditioning for this equation. In particular, we present our investigation in the fast parallel solution of the time-dependent advection(-diffusion) problem on a time-dependent domain $\Omega(t)$,

$$\partial_t u + a \cdot \nabla u - \nu \nabla^2 u = f \qquad \text{in } \Omega(t),\ t^0 < t < t^N, \tag{1.13}$$

where a is the advective velocity, f is a source term, and $\nu \geq 0$ is the diffusion constant. We are particularly interested in the advection-dominated regime where $0 \leq \nu \ll 1$.

To discretize Equation (1.13), we consider the space-time framework in which the problem is recast into a space-time domain as follows. Let $x = (x_1, \ldots, x_d)$ be the spatial variables in spatial dimension d. A point at time $t = x_0$ with position x then has Cartesian coordinates $\hat{x} = (x_0, x)$ in space-time. Defining the space-time domain $\mathcal{E} := \{\hat{x} \ : \ x \in \Omega(x_0), \ t^0 < x_0 < t^N\}$, the space-time advective velocity $\widehat{a} := (1, a)$ and the space-time gradient $\widehat{\nabla} := (\partial_t, \nabla)$, the space-time formulation of Equation (1.13) is given by

$$\widehat{a} \cdot \widehat{\nabla} u - \nu \nabla^2 u = f \qquad \text{in } \mathcal{E}. \tag{1.14}$$

There are multiple reasons to consider space-time finite element methods over traditional discretizations. First, space-time methods provide a natural framework for the discretization of partial differential equations on time-dependent domains [83, 110, 156, 157, 161]. This is because the domain and mesh movement are automatically accounted for by the space-time finite element spaces, which are defined on a triangulation of the space-time domain $\mathcal{E}$. Furthermore, since there is no distinction between spatial and temporal variables, it is relatively straightforward to allow local time stepping and adaptive space-time mesh refinement (see, for example [160]). This is particularly interesting from an efficiency perspective for problems that require locally small time steps and fine mesh resolution to achieve high levels of accuracy in only some parts of the domain. These properties are non-trivial within the context of traditional time-integration techniques. Finally, space-time finite elements allow for greater parallelization by solving for the entire space-time solution simultaneously, rather than in a sequential time-stepping process. This ends up being particularly relevant for hyperbolic PDEs, as will be discussed later.

Space-time discontinuous Galerkin (DG) finite element methods are well suited for solving Equation (1.14) in the advection-dominated limit (see [130, 147, 152, 153, 158, 159, 162] and references therein). This is because space-time DG methods incorporate upwinding in their numerical fluxes, are locally conservative, and automatically satisfy the geometric conservation law (GCL) [97], which requires that the uniform flow remains uniform under grid motion. We point out that alternative discretizations (such as arbitrary Lagrangian–Eulerian methods) may require additional constraints to satisfy the GCL [122]. One downside of space-time DG methods is the large number of globally coupled degrees-of-freedom (DOFs) that arise when applying DG finite elements in $(d+1)$-dimensional space. However, the space-time hybridizable discontinuous Galerkin (HDG) method [128, 129], introduced as a space-time extension of the HDG method [28], can attenuate this problem. The space-time HDG method, like the HDG method, introduces approximate traces of

the solution on the element faces. The DOFs on the interior of an element are then eliminated from the system, resulting in a (significantly smaller) global system of algebraic equations only for the approximate traces. However, it should be noted that a reduction in the number of globally coupled DOFs does not necessarily imply a more efficient time to solution – the linear system still needs to be solved.

In practice, a *slab-by-slab* approach is almost exclusively used to obtain the solution of space-time discretizations, which is analogous to traditional time-integration techniques: the space-time domain is partitioned into space-time slabs and local systems are solved sequentially one time step after the other (e.g. [86, 110, 159]). Although commonly used, such an approach is limited to spatial parallelism, which eventually plateaus in the sense that using more processors does not speed up the time to solution (see, e.g., [52]). With an increasing number of processors available for use and stagnating core clock speeds, there has been significant research on parallel-in-time (PinT) methods in recent years.

Some of the most effective PinT methods are multigrid-in-time methods, where a parallel multilevel method is applied over the time domain, which is then coupled with traditional spatial solves to perform time steps of varying sizes (in particular, see Parareal [100] and multigrid-reduction-in-time [52]). Such methods are effective on parabolic-type problems, but tend to not be robust or just not convergent on advection-dominated and hyperbolic problems without special treatment (for example, see [136, 62, 36, 32, 35]). The simplest explanation for the difficulties such methods have with hyperbolic problems is the separation of space and time. By treating space and time separately, the multilevel coarsening cannot respect the underlying characteristics that propagate in space-time.

A more general approach is to consider space-time multigrid, that is, multigrid methods applied to the full space-time domain. To our knowledge, such an approach has only been applied to parabolic problems, primarily the heat equation [164, 80, 65]. However, even there, space-time multigrid has demonstrated superior performance over PinT methods that use multigrid in space and time separately [53]. Recently, auxiliary-space preconditioning techniques have also been proposed for space-time finite-element discretizations [68], which has the potential to provide more general space-time solvers. Continuing with the above discussion, the *all-at-once* approach to space-time finite elements constructs and solves a single global linear system for the solution in the whole space-time domain. From a solver's perspective, we claim that the all-at-once approach is particularly well suited for advective and hyperbolic problems.

The main contribution of Chapter 3 is demonstrating the suitability of the nonsymmetric algebraic multigrid (AMG) method based on Approximate Ideal Restriction (AIR) [107, 109] for the solution of slab-by-slab and, in particular, all-at-once space-time HDG dis-

cretizations of the advection-diffusion problem in advection-dominated regimes. Advection-dominated problems are typically difficult to solve due to the non-symmetric nature of the problem. Nevertheless, significant developments in multigrid methods for non-symmetric problems have been made in recent years [115, 168, 23, 140, 108]. In particular, AIR has shown to be a robust solver for steady advection-dominated problems. This motivates us to study AIR for a space-time HDG discretization of the advection-diffusion problem, since Equation (1.14) can be seen as a "steady" advection-diffusion problem in $(d+1)$-dimensions.

1.4 State of the art for preconditioners

The HDG discretization of the linearised Navier–Stokes equations results in a generalized saddle point problem. While there are many preconditioners developed for 2-by-2 saddle point problems (see the surveys [11, 17, 126, 117]), these do not directly generalize to our problem, particularly when static condensation is employed (see Chapters 2 and 4 for details), and further modifications are necessary. On the other hand, the literature on preconditioners for HDG discretizations of partial differential equations is scarce. We mention [27, 29, 64, 113, 76, 84, 63, 98, 102] for scalar elliptic problems, [21] for the Stokes problem, and [38, 61] for the compressible Navier–Stokes and Euler equations. However, at the time of writing, we are not aware of any preconditioners developed specifically for HDG discretizations of the incompressible Navier–Stokes problem. In [133], the authors develop and rigorously analyze a block preconditioner for an HDG discretization of the Stokes equations by exploiting the properties of the discretization. Following a similar approach, we will construct two preconditioners for the steady Navier–Stokes equations. In particular, in Chapter 4, we extend the grad-div and augmented Lagrangian preconditioners and the pressure convection-diffusion (PCD) framework to HDG discretizations, and present numerical results. In the next two subsections, we review some of the existing literature on these preconditioners.

1.4.1 Pressure Convection-Diffusion Preconditioners

It is well-known that the preconditioner

$$P = \begin{bmatrix} A + N(u) & X \\ 0 & \frac{1}{\nu} M_p \end{bmatrix}, \tag{1.15}$$

where $X = B^T$ or $X = 0$ and where M_p is the pressure mass matrix, is an h-robust preconditioner [46] for the linear system of the form Equation (1.10) obtained from the Taylor–Hood

discretization [154, 20] of the Navier–Stokes equations (Equation (1.2)). While this preconditioner is efficient with respect to the mesh size in the low Reynolds number regime, as the Reynolds number increases the number of GMRES iterations to convergence increases. One framework of preconditioner to reduce the sensitivity of convergence on the Reynolds number is the PCD preconditioner.

PCD preconditioners have been developed over a chain of papers. The main idea is to replace $\frac{1}{\nu}M_p$ in Equation (1.15) by a better approximation to the pressure Schur complement. To achieve this goal, Kay and Loghin [87] (later extended and published as [88]) use Green's tensors to find a continuous "inverse" to the pressure Schur complement and discretize that continuous inverse. The result is the approximation $X_{\mathcal{G}}^{-1} = M_p^{-1}F_pA_p^{-1}$, where F_p and A_p are, respectively, the pressure convection-diffusion and pressure Poisson matrices. While this is the first appearance of pressure convection-diffusion preconditioners, Silvester et al. [145] proposed the approach we use in this thesis. The main idea comes from the observation that the commutator on the differential operators

$$\mathcal{E}_p = \nabla(\alpha + w\cdot\nabla - \nu\Delta)_p - (\alpha + w\cdot\nabla - \nu\Delta)_u\nabla,$$

where $\mathcal{L}_p = (\alpha + w\cdot\nabla - \nu\Delta)_p$ is the convection-diffusion-reaction operator acting on the pressure space and, similarly, $\mathcal{L}_u = (\alpha + w\cdot\nabla - \nu\Delta)_u$ is the convection-diffusion-reaction operator acting on the velocity space, will be zero over unbounded domains and for constant wind w. The equivalent discrete commutator is given by

$$\mathcal{E}_{p,h} = (M_u^{-1}B^T)(M_p^{-1}F_p) - (M_u^{-1}F)(M_u^{-1}B^T),$$

where M_u is the mass matrix defined on the velocity space. By assuming that these commutator errors are small, either at the continuous level or the discrete level, we can approximate $(M_u^{-1}B^T)(M_p^{-1}F_p)$ by $(M_u^{-1}F)(M_u^{-1}B^T)$. By multiplying this expression with $BF^{-1}M_u$ from the left and with $F_p^{-1}M_p$ from the right, we obtain the following approximation to the Schur complement, $BF^{-1}B^T \approx BM_u^{-1}B^TF_p^{-1}M_p$. Now, using reverse inf-sup stability of the Taylor–Hood discretization of the Navier–Stokes equations, which can be interpreted as $BM_u^{-1}B^T \approx A_p$, we further obtain the approximation $(BF^{-1}B^T)^{-1} \approx M_p^{-1}F_pA_p^{-1}$. The numerical results in [145] show robustness of the preconditioner against the mesh size, and only a weak dependence on the Reynolds number.

Notice that the commutator above acts on the pressure space on which no boundary conditions are enforced. Hence, both A_p and F_p are 1-rank-deficient operators and their nullspaces contain the constant element. The construction of these operators in which we do not impose boundary conditions on A_p and F_p is known as the *do-nothing* strategy.

However, in some cases, there may be some benefit in imposing boundary conditions on these operators [47] which is possible by using the commutator

$$\mathcal{E}_u = (\alpha + w \cdot \nabla - \nu\Delta)_p \nabla \cdot - \nabla \cdot (\alpha + w \cdot \nabla - \nu\Delta)_u,$$

which is defined on the velocity space. The numerical results in [47] show a significant improvement in efficiency when imposing boundary conditions over the *do-nothing* strategy. We observed this also holds for our choice of HDG discretization, see Section 4.2 for a discussion.

For the interested reader, other preconditioners based on minimizing the discrete commutator include the BFBt preconditioner [49], the SPAC preconditioner [43], and the least squares commutator preconditioner [44].

1.4.2 Grad-div and Augmented Lagrangian Preconditioners

Grad-div and augmented Lagrangian preconditioners present an alternative to the PCD framework. However, the goal remains the same: find an accurate approximation to the pressure Schur complement. Grad-div and augmented Lagrangian frameworks are closely related, but we will discuss these two preconditioners separately.

For augmented Lagrangian preconditioners, we modify the block linear system in Equation (1.10) by multiplying the second block row (from the left) by $\gamma B^T W^{-1}$, where W is an SPD matrix of compatible dimensions and $\gamma > 0$, and adding the result to the first block row to obtain

$$\begin{bmatrix} A + N(u) + \gamma B^T W^{-1} B & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}. \tag{1.16}$$

Similar to the ideas discussed in Section 1.4.1, we are looking for a good preconditioner based on the approximation of the Schur complement, i.e., we are looking for a preconditioner of the form

$$P = \begin{bmatrix} A + N(u) + \gamma B^T W^{-1} B & X \\ 0 & \hat{S} \end{bmatrix}, \tag{1.17}$$

where $\hat{S}$ is an approximation to the pressure Schur complement $B(A+N(u)+\gamma B^T W^{-1} B)^{-1} B^T$ of Equation (1.16) and $X = B^T$ or $X = 0$. By the Woodbury matrix formula, we note that

$$(B(A + N(u) + \gamma B^T W^{-1} B)^{-1} B^T)^{-1} = (B(A + N(u))^{-1} B^T)^{-1} + \gamma W^{-1}.$$

Hence, for large γ, we see that γW^{-1} is a good approximation to $(B(A+N(u)+\gamma B^T W^{-1} B)^{-1} B^T)^{-1}$. By choosing $\hat{S} = \gamma^{-1} W$ in Equation (1.17), we obtain the augmented Lagrangian preconditioner.

Grad-div preconditioners employ a similar idea. However, rather than adding a consistent term to the momentum block at the discrete level, grad-div preconditioners are derived by first adding the consistent term $-\gamma \nabla(\nabla \cdot \vec{u})$ with $\gamma > 0$ to the momentum equation Equation (1.2a) and then discretizing this term. Since $\nabla \cdot \vec{u} = 0$, addition of $-\gamma \nabla(\nabla \cdot \vec{u})$ does not change the solution of the problem, however, at the discrete level it changes the properties of the problem and the corresponding linear system becomes easier to solve. Grad-div preconditioners further have the added benefit of penalizing the error in the divergence $\|\nabla \cdot \vec{u}\|$ which improves the solution quality for appropriate choices of γ.

Grad-div and augmented Lagrangian preconditioners were a focus of research until the early 2010s, see [60, 12, 34, 14, 13, 15, 77, 96, 16] among many others. Of particular significance is [12], where the authors develop a highly specialized multigrid method for "inverting" the augmented momentum block as a part of their preconditioner. Their numerical results suggest that the preconditioner is both h and Re robust for iso$P2 - P1$ discretizations of the Navier–Stokes equations. In [13], the authors prove that the scaled mass matrix approximation to the pressure Schur complement, indeed, results in an h and Re robust preconditioner given that the parameter γ is chosen appropriately and the momentum block is solved exactly. Furthermore, they prove that if the momentum block is approximated by its upper block-triangular part, the resulting preconditioner is h-robust.

Lately, there has been a new interest in grad-div and augmented Lagrangian preconditioners, for example [26, 112, 118, 74, 75, 73, 58, 56, 55, 169, 57]. We summarize here the contributions of some of these papers. In [74, 75, 73], the authors extend the augmented Lagrangian preconditioners to finite volume discretizations of the Navier–Stokes equations. They find that the augmented Lagrangian preconditioners are neither h nor Re robust for their applications, however, it still outperforms other state-of-the-art preconditioners, especially when combined with a second preconditioner such as SIMPLE. Next, [58] generalizes the multigrid method of [12] to the Taylor–Hood discretization of the three dimensional Navier–Stokes equations. The adoption of this multigrid method for different discretizations and different equations is a continuing effort [56, 55, 169, 57].

Chapter 2

Krylov Subspace Methods and 2×2 Block Preconditioners

In this chapter, we present our investigation of the relationship between Krylov subspace methods and 2-by-2 block preconditioners. The results of our investigation are used to design and justify our work in later chapters. In particular, Theorem 2.2.3 ties the efficiency of Krylov subspace methods for a 2×2 block system, as given in Equations (1.11) and (1.12), to the Schur complement of the problem.

Chapter 2 is structured as follows. Section 2.1 formally introduces various block preconditioners, considers the distinction between fixed-point and Krylov methods, and derives some relationships on polynomials of the preconditioned operators that define Krylov and fixed-point iterations. Proofs and formal statements of results are provided in Section 2.2, and numerical results are examined in Section 2.3, with a discussion on the practical implications of theory developed here. We conclude in Section 2.4.

This chapter is published in [150].

2.1 Block preconditioners

This section considers 2×2 block preconditioners, where one diagonal block is inverted exactly, and the other is some approximation to the Schur complement.

We consider four different kinds of block preconditioners: block diagonal, block upper triangular, block lower triangular, and block LDU, denoted D, U, L, and M, respectively.

If the preconditioners have no subscript, this implies the diagonal blocks of the preconditioners are the diagonal blocks of A. If one of the diagonal blocks is some approximation to the Schur complement S_{kk}, $k \in \{1, 2\}$, then a 11- or 22- subscript denotes in which block the approximation is used. For example, with a Schur-complement approximation in the $(1, 1)$-block, preconditioners take the forms

$$L_{11} := \begin{bmatrix} \widehat{S}_{11} & \mathbf{0} \\ A_{21} & A_{22} \end{bmatrix}, \qquad U_{11} := \begin{bmatrix} \widehat{S}_{11} & A_{12} \\ \mathbf{0} & A_{22} \end{bmatrix}, \tag{2.1}$$

$$D_{11} := \begin{bmatrix} \widehat{S}_{11} & \mathbf{0} \\ \mathbf{0} & A_{22} \end{bmatrix}, \qquad M_{11} := \begin{bmatrix} I & A_{12}A_{22}^{-1} \\ \mathbf{0} & I \end{bmatrix} \begin{bmatrix} \widehat{S}_{11} & \mathbf{0} \\ \mathbf{0} & A_{22} \end{bmatrix} \begin{bmatrix} I & \mathbf{0} \\ A_{22}^{-1}A_{21} & I \end{bmatrix}.$$

The block-diagonal, block upper-triangular, and block lower-triangular preconditioners with a Schur-complement approximation in the $(2, 2)$-block take an analogous form, with $\widehat{S}_{11} \mapsto A_{11}$ and $A_{22} \mapsto \widehat{S}_{22}$, and the approximate block LDU preconditioner M_{22} is given by

$$M_{22} := \begin{bmatrix} I & \mathbf{0} \\ A_{21}A_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{11} & \mathbf{0} \\ \mathbf{0} & \widehat{S}_{22} \end{bmatrix} \begin{bmatrix} I & A_{11}^{-1}A_{12} \\ \mathbf{0} & I \end{bmatrix}. \tag{2.2}$$

Most results here regarding block-diagonal preconditioning are for the specific case of block Jacobi, where $D_{11} = D_{22} = D$ is the block diagonal of A.

Preconditioners are typically used in conjunction with either a fixed-point iteration or Krylov subspace method to approximately solve a linear system Equation (1.11). Krylov methods approximate the solution to linear systems by constructing a Krylov space of vectors and minimizing the error of the approximate solution over this space, in a given norm. The Krylov space is formed as powers of the preconditioned operator applied to the initial residual. For linear system $A\mathbf{x} = \mathbf{b}$, (left) preconditioner M^{-1}, and initial residual $\mathbf{r}_0$, the dth Krylov space takes the form

$$\mathcal{K}_d := \left\{ \mathbf{r}_0, M^{-1}A\mathbf{r}_0, ..., (M^{-1}A)^{d-1}\mathbf{r}_0 \right\}.$$

Minimizing over this space is thus equivalent to constructing a minimizing polynomial $p(M^{-1}A)\mathbf{r}_0$, which is optimal in a given norm. This optimality can be in the operator norm (that is, including a $\sup_{\mathbf{r}_0 \neq \mathbf{0}}$) for a worst-case convergence over all initial guesses and right-hand sides, or optimal for a specific initial residual. Examples include conjugate gradient (CG), which minimizes error in the A-norm, MINRES, which minimizes error in the $AM^{-1}A$-norm [4], left-preconditioned GMRES, which minimizes error in the $(M^{-1}A)^*(M^{-1}A)$-norm, or right-preconditioned GMRES, which minimizes error in the

A^*A-norm. Note that error in the A^*A-norm is equivalent to residual in the ℓ^2-norm, which is how minimal-residual methods are typically presented. Fixed-point iterations also correspond to polynomials of the preconditioned operator, but they are not necessarily optimal in a specific norm.

Analysis in this chapter is focused on polynomials of block-preconditioned operators, particularly deriving upper and lower bounds on minimizing Krylov polynomials of a fixed degree. Section 2.1.1 begins by considering fixed-point iterations and the corresponding matrix polynomials in the nonsymmetric setting, and discusses important differences between the various preconditioners in Equation (2.1) and Equation (2.2). Section 2.1.2 then examines general polynomials of the preconditioned operator, developing the theoretical framework used in Section 2.2 to analyze convergence of block-preconditioned Krylov methods. Due to the equivalence of a Krylov method and a minimizing polynomial of the preconditioned operator, we refer to, for example, GMRES and a minimizing polynomial of $p(M^{-1}A)$ in the ℓ^2-norm, interchangeably.

2.1.1 Observations on fixed-point iterations

For some approximate inverse P to linear operator A, error propagation of a fixed-point iteration takes the form $\mathcal{E} := I - P^{-1}A$ and residual propagation takes the form $\mathcal{R} := A\mathcal{E}^{-1}A = I - AP^{-1}$. Define

$$\begin{aligned} \mathcal{E}_{11} &:= I - \widehat{S}_{11}^{-1}S_{11}, \qquad & \mathcal{R}_{11} &:= I - S_{11}\widehat{S}_{11}^{-1}, \\ \mathcal{E}_{22} &:= I - \widehat{S}_{22}^{-1}S_{22}, \qquad & \mathcal{R}_{22} &:= I - S_{22}\widehat{S}_{22}^{-1}. \end{aligned} \tag{2.3}$$

Consider first block-triangular and approximate block-LDU preconditioners. Powers of fixed-point error and residual propagation with these block preconditioners take the fol-

lowing forms:

$$
\begin{aligned}
(I - L_{11}^{-1}A)^d &= \begin{bmatrix} I \\ -A_{22}^{-1}A_{21} \end{bmatrix} \mathcal{E}_{11}^{d-1} \begin{bmatrix} I - \widehat{S}_{11}^{-1}A_{11} & -\widehat{S}_{11}^{-1}A_{12} \end{bmatrix}, \\
(I - AL_{22}^{-1})^d &= \begin{bmatrix} -A_{12}\widehat{S}_{22}^{-1} \\ I - A_{22}\widehat{S}_{22}^{-1} \end{bmatrix} \mathcal{R}_{22}^{d-1} \begin{bmatrix} -A_{21}A_{11}^{-1} & I \end{bmatrix}, \\
(I - AU_{11}^{-1})^d &= \begin{bmatrix} I - A_{11}\widehat{S}_{11}^{-1} \\ -A_{21}\widehat{S}_{11}^{-1} \end{bmatrix} \mathcal{R}_{11}^{d-1} \begin{bmatrix} I & -A_{12}A_{22}^{-1} \end{bmatrix}, \\
(I - U_{22}^{-1}A)^d &= \begin{bmatrix} -A_{11}^{-1}A_{12} \\ I \end{bmatrix} \mathcal{E}_{22}^{d-1} \begin{bmatrix} -\widehat{S}_{22}^{-1}A_{21} & I - \widehat{S}_{22}^{-1}A_{22} \end{bmatrix},
\end{aligned}
\tag{2.4}
$$

$$
(I - AL_{11}^{-1})^d = \begin{bmatrix} \mathcal{R}_{11}^p & -\mathcal{R}_{11}^{d-1}A_{12}A_{22}^{-1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad (I - L_{22}^{-1}A)^d = \begin{bmatrix} \mathbf{0} & -A_{11}^{-1}A_{12}\mathcal{E}_{22}^{d-1} \\ \mathbf{0} & \mathcal{E}_{22}^d \end{bmatrix}
$$

$$
(I - AU_{22}^{-1})^d = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -\mathcal{R}_{22}^{d-1}A_{21}A_{11}^{-1} & \mathcal{R}_{22}^d \end{bmatrix}, \quad (I - U_{11}^{-1}A)^d = \begin{bmatrix} \mathcal{E}_{11}^d & \mathbf{0} \\ -A_{22}^{-1}A_{21}\mathcal{E}_{11}^{d-1} & \mathbf{0} \end{bmatrix},
$$

$$
(I - M_{11}^{-1}A)^d = \begin{bmatrix} \mathcal{E}_{11}^d & \mathbf{0} \\ -A_{22}^{-1}A_{21}\mathcal{E}_{11}^d & \mathbf{0} \end{bmatrix}, \quad (I - AM_{11}^{-1})^d = \begin{bmatrix} \mathcal{R}_{11}^d & -\mathcal{R}_{11}^d A_{12}A_{22}^{-1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},
$$

$$
(I - M_{22}^{-1}A)^d = \begin{bmatrix} \mathbf{0} & -A_{11}^{-1}A_{12}\mathcal{E}_{22}^d \\ \mathbf{0} & \mathcal{E}_{22}^d \end{bmatrix}, \quad (I - AM_{22}^{-1})^d = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -\mathcal{R}_{22}^d A_{21}A_{11}^{-1} & \mathcal{R}_{22}^d \end{bmatrix}.
$$

Let $\|\cdot\|$ be a given norm on A and $\|\cdot\|_c$ be a given norm on the Schur-complement problem.[1] Note that any of the above fixed-point iterations is convergent in $\|\cdot\|$ for all initial error or residual, if and only if the corresponding Schur-complement fixed-point iteration in Equation (2.3) is convergent in $\|\cdot\|_c$. Moreover, it is well-known that for block-triangular preconditioners with an exact Schur complement, minimal residual Krylov methods converge in two iterations [85, 114]. However, convergence in two iterations actually follows from fixed-point convergence rather than Krylov iterations.

Proposition 2.1.1 (Block triangular-preconditioners with Schur complement)**.** *If $\widehat{S}_{kk} = S_{kk}$, for $k \in \{1, 2\}$, then fixed-point iteration with a (left or right) block upper or block lower-triangular preconditioner converges in two iterations.*

[1] In the case of ℓ^p-norms, $\|\cdot\| = \|\cdot\|_c$, but in general, such as for matrix-induced norms, they may be different.

Proof. The proof follows by noting that if $\widehat{S}_{kk} = S_{kk}$, for $k \in \{1, 2\}$, then all terms defined in (2.3) are zero. □

Now consider block-diagonal preconditioners. Then,

$$I - D_{11}^{-1}A = \begin{bmatrix} I - \widehat{S}_{11}^{-1}A_{11} & -\widehat{S}_{11}^{-1}A_{12} \\ -A_{22}^{-1}A_{21} & \mathbf{0} \end{bmatrix}, \quad I - AD_{11}^{-1} = \begin{bmatrix} I - A_{11}\widehat{S}_{11}^{-1} & -A_{12}A_{22}^{-1} \\ -A_{21}\widehat{S}_{11}^{-1} & \mathbf{0} \end{bmatrix},$$
$$I - D_{22}^{-1}A = \begin{bmatrix} \mathbf{0} & -A_{11}^{-1}A_{12} \\ -\widehat{S}_{22}^{-1}A_{21} & I - \widehat{S}_{22}^{-1}A_{22} \end{bmatrix}, \quad I - AD_{22}^{-1} = \begin{bmatrix} \mathbf{0} & -A_{12}\widehat{S}_{22}^{-1} \\ -A_{21}A_{11}^{-1} & I - A_{22}\widehat{S}_{22}^{-1} \end{bmatrix}.$$

If we simplify to block Jacobi (that is, $\widehat{S}_{kk} := A_{kk}$ for $k \in \{1, 2\}$), both diagonal blocks are zero, and a closed form for powers of block-diagonal preconditioners can be obtained for an arbitrary number of fixed-point iterations,

$$\begin{aligned} (I - D^{-1}A)^{2d} &= \begin{bmatrix} A_{11}^{-1}A_{12}A_{22}^{-1}A_{21} & \mathbf{0} \\ \mathbf{0} & A_{22}^{-1}A_{21}A_{11}^{-1}A_{12} \end{bmatrix}^d, \\ (I - AD^{-1})^{2d} &= \begin{bmatrix} A_{12}A_{22}^{-1}A_{21}A_{11}^{-1} & \mathbf{0} \\ \mathbf{0} & A_{21}A_{11}^{-1}A_{12}A_{22}^{-1} \end{bmatrix}^d. \end{aligned} \tag{2.5}$$

Noting that if $\widehat{S}_{11} := A_{11}$ and $\widehat{S}_{22} := A_{22}$ in Equation (2.3), then

$$(I - D^{-1}A)^{2d} = \begin{bmatrix} \mathcal{E}_{11} & \mathbf{0} \\ \mathbf{0} & \mathcal{E}_{22} \end{bmatrix}^d, \qquad (I - AD^{-1})^{2d} = \begin{bmatrix} \mathcal{R}_{11} & \mathbf{0} \\ \mathbf{0} & \mathcal{R}_{22} \end{bmatrix}^d.$$

It follows that block Jacobi converges if and only if block upper- and lower-triangular preconditioners, with diagonal blocks given by A_{11} and A_{22}, both converge. Furthermore, the expected number of iterations of block Jacobi to converge to a given tolerance are approximately double that of the equivalent block-triangular preconditioning, give or take some independent constant factor (e.g., $A_{11}^{-1}A_{12}$) from the fixed-point operators. A similar result is later shown for preconditioning Krylov methods with block Jacobi (see Theorem 2.2.5). This relation of twice as many iterations for Jacobi/block-diagonal preconditioning has been noted or observed a number of times, perhaps originally in [59] where MINRES/GMRES are proven to stall every other iteration on saddle-point problems.

Remark 2.1.1 (Non-convergent block-diagonal fixed-point)**.** *As mentioned above, fixed-point iteration converges in two iterations for a block-triangular preconditioner if the Schur*

complement is inverted exactly. However, the same does not hold for block Jacobi. Let D_{22} *be a block diagonal preconditioner with* $\widehat{S}_{22} := S_{22}$*. In the case of a saddle-point matrix, say* B*, where* $B_{22} = \mathbf{0}$*,*

$$(I - D_{22}^{-1}B)^{3d} = \begin{bmatrix} \mathbf{0} & -B_{11}^{-1}B_{12} \\ -S_{22}^{-1}B_{21} & I \end{bmatrix}^{3d} = (-1)^d \begin{bmatrix} -B_{11}^{-1}B_{12}S_{22}^{-1}B_{21} & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix},$$

where $S_{22} := -B_{21}B_{11}^{-1}B_{12}$*. Here we see the interesting property that as we continue to iterate, error-propagation of block-diagonal preconditioning does not converge or diverge. In fact,* $(I - D_{22}^{-1}B)^{3d}$ *is actually a periodic point of period two under the matrix-valued mapping* $(I - D_{22}^{-1}B)^3$*, for* $d \geq 1$*. The general* 2×2 *case is more complicated and does not appear to have such a property. However, expanding to up to four powers gave no indication that it would result in a convergent fixed-point iteration, as it does with GMRES acceleration [85].*

Remark 2.1.2 (Non-optimal block-diagonal Krylov)**.** *It was recently shown that for* 2×2 *systems with nonzero diagonal blocks, block-diagonal preconditioning of minimal-residual methods with an exact Schur complement does not necessarily converge in a fixed number of iterations, in contrast to block-triangular preconditioners or block-diagonal preconditioners for matrices with a zero (2,2) block [149].*

2.1.1.1 Symmetric block-triangular preconditioners

The benefit of Jacobi or block-diagonal preconditioning for SPD matrices is that they are also SPD, which permits the use of three-term recursion relations like conjugate gradient (CG) and MINRES, whereas block upper- or lower-triangular preconditioners are not applicable. Approximate block-LDU preconditioners offer one symmetric option. Another option that might be considered, particularly by those that work in iterative or multigrid methods, is a symmetric triangular iteration, consisting of a block-upper triangular iteration followed by a block-lower triangular iteration (or vice versa), akin to a symmetric (block) Gauss–Seidel sweep. Interestingly, this does not appear to be an effective choice. Consider a symmetric block-triangular preconditioner with approximate Schur complement in the (2,2)-block. The preconditioner can take two forms, depending on whether the lower or upper iteration is done first. For example,

$$\begin{aligned} (I - L_{22}^{-1}A)(I - U_{22}^{-1}A) &= I - L_{22}^{-1}(L_{22} + U_{22} + A)U_{22}^{-1}A, \\ (I - U_{22}^{-1}A)(I - L_{22}^{-1}A) &= I - U_{22}^{-1}(L_{22} + U_{22} + A)L_{22}^{-1}A. \end{aligned}$$

Define $\mathcal{H}_{22}^{-1} := L_{22}^{-1}(L_{22}+U_{22}+A)U_{22}^{-1}$ and $\mathcal{G}_{22}^{-1} := U_{22}^{-1}(L_{22}+U_{22}+A)L_{22}^{-1}$, corresponding to upper-lower and lower-upper, symmetric preconditioners respectively. Expanding in block form, we see that preconditioners associated with a symmetric block-triangular iteration are given by

$$\mathcal{H}_{22}^{-1} := \begin{bmatrix} I & \mathbf{0} \\ -\widehat{S}_{22}^{-1}A_{21} & I \end{bmatrix} \begin{bmatrix} A_{11}^{-1} & \mathbf{0} \\ \mathbf{0} & 2\widehat{S}_{22}^{-1} - \widehat{S}_{22}^{-1}A_{22}\widehat{S}_{22}^{-1} \end{bmatrix} \begin{bmatrix} I & -A_{12}\widehat{S}_{22}^{-1} \\ \mathbf{0} & I \end{bmatrix},$$
$$\mathcal{G}_{22}^{-1} := \begin{bmatrix} I & -A_{11}^{-1}A_{12} \\ \mathbf{0} & I \end{bmatrix} \begin{bmatrix} A_{11}^{-1} & \mathbf{0} \\ \mathbf{0} & 2\widehat{S}_{22}^{-1} - \widehat{S}_{22}^{-1}A_{22}\widehat{S}_{22}^{-1} \end{bmatrix} \begin{bmatrix} I & \mathbf{0} \\ -A_{21}A_{11}^{-1} & I \end{bmatrix}.$$

Notice that each of these preconditioners can be expressed as a certain block-LDU type preconditioner; however, it is not clear that either would be as good as or a better preconditioner than block LDU. In the simplest (and also fairly common) case that $\widehat{S}_{22} = A_{22}$, then $\mathcal{H}_{22}^{-1}$ and $\mathcal{G}_{22}^{-1}$ are exactly equivalent to the two variants of block-LDU preconditioning in Equation (2.1) and Equation (2.2), respectively, with diagonal blocks used to approximate the Schur complement. As we will see in Section 2.1.2.3, this is also formally equivalent to a block-triangular preconditioner.

Adding an approximation to the Schur complement in the (2,2)-block, $\mathcal{G}_{22}^{-1}$, is equivalent to block-LDU preconditioning with Schur-complement approximation in the (2,2)-block, except that now we approximate S_{22}^{-1} with the operator $2\widehat{S}_{22}^{-1} - \widehat{S}_{22}^{-1}A_{22}\widehat{S}_{22}^{-1}$, as opposed to $\widehat{S}_{22}^{-1}$ in block-LDU preconditioning Equation (2.2). It is not clear if such an approach would ever be beneficial over standard LDU, although it is possible one can construct such a problem. For $\widehat{S}_{22} \neq A_{22}$, it is even less clear that $\mathcal{H}_{22}^{-1}$ would make a good or better preconditioner compared with LDU or block triangular. Analogous things can be said about Schur-complement approximations in the (1,1)-block. Numerical results in Section 2.3 confirm these observations, where symmetric block-triangular preconditioners offer at best a marginal reduction in total iteration count over block upper- or lower-triangular preconditioners, and sometimes observe worse convergence, at the expense of several additional (approximate) inverses.

2.1.2 Krylov and polynomials of the preconditioned matrix

This section begins by considering polynomials applied to the approximate block-LDU and block-triangular preconditioned operators in Section 2.1.2.1 and Section 2.1.2.2, respectively (the block-diagonal preconditioner is discussed in Section 2.2.3). These results are used in Section 2.1.2.3 to construct a norm in which fixed-point or Krylov iterations

applied to approximate block-LDU or block-triangular preconditioned operators are equivalent to the preconditioned Schur complement. Section 2.1.2.4 uses this equivalence to motivate the key tool used in proofs provided in Section 2.2.

2.1.2.1 Approximate block-LDU preconditioner

In this section we apply a polynomial to the block-LDU preconditioned operator. For an approximate block-LDU preconditioner with approximate Schur complement in the $(2,2)$-block,

$$M_{22}^{-1}A = \begin{bmatrix} I & -A_{11}^{-1}A_{12} \\ \mathbf{0} & I \end{bmatrix} \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & \widehat{S}_{22}^{-1}S_{22} \end{bmatrix} \begin{bmatrix} I & A_{11}^{-1}A_{12} \\ \mathbf{0} & I \end{bmatrix} := P_1 \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & \widehat{S}_{22}^{-1}S_{22} \end{bmatrix} P_1^{-1}. \tag{2.6}$$

The three-term formula reveals the change of basis matrix, P_1, between the LDU-preconditioned operator and the Schur-complement problem. This allows us to express polynomials p of the preconditioned operator as a change of basis applied to the polynomial of the preconditioned Schur complement and the identity,

$$\begin{aligned} p(M_{22}^{-1}A) &= \begin{bmatrix} I & -A_{11}^{-1}A_{12} \\ \mathbf{0} & I \end{bmatrix} \begin{bmatrix} p(I) & \mathbf{0} \\ \mathbf{0} & p(\widehat{S}_{22}^{-1}S_{22}) \end{bmatrix} \begin{bmatrix} I & A_{11}^{-1}A_{12} \\ \mathbf{0} & I \end{bmatrix} \\ &= \begin{bmatrix} p(I) & A_{11}^{-1}A_{12}\left(p(I) - p(\widehat{S}_{22}^{-1}S_{22})\right) \\ \mathbf{0} & p(\widehat{S}_{22}^{-1}S_{22}) \end{bmatrix}. \end{aligned} \tag{2.7}$$

Using right preconditioning, the polynomial takes the form

$$\begin{aligned} p(AM_{22}^{-1}) &= \begin{bmatrix} I & \mathbf{0} \\ A_{21}A_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} p(I) & \mathbf{0} \\ \mathbf{0} & p(S_{22}\widehat{S}_{22}^{-1}) \end{bmatrix} \begin{bmatrix} I & \mathbf{0} \\ -A_{21}A_{11}^{-1} & I \end{bmatrix} \\ &= \begin{bmatrix} p(I) & \mathbf{0} \\ \left(p(I) - p(S_{22}\widehat{S}_{22}^{-1})\right) A_{21}A_{11}^{-1} & p(S_{22}\widehat{S}_{22}^{-1}) \end{bmatrix}. \end{aligned}$$

Similarly, polynomials of the left and right preconditioned operator by a block LDU with approximate Schur complement in the $(1,1)$-block take the form

$$\begin{aligned} p(M_{11}^{-1}A) &= \begin{bmatrix} p(\widehat{S}_{11}^{-1}S_{11}) & \mathbf{0} \\ A_{22}^{-1}A_{21}\left(p(I) - p(\widehat{S}_{11}^{-1}S_{11})\right) & p(I) \end{bmatrix}, \\ p(AM_{11}^{-1}) &= \begin{bmatrix} p(S_{11}\widehat{S}_{11}^{-1}) & \left(p(I) - p(S_{11}\widehat{S}_{11}^{-1})\right)A_{12}A_{22}^{-1} \\ \mathbf{0} & p(I) \end{bmatrix}. \end{aligned} \tag{2.8}$$

2.1.2.2 Block-triangular preconditioner

We now consider polynomials of a block-triangular preconditioned operator. Notice that error- and residual-propagation operators for four of the block-triangular preconditioners in Equation (2.4) take a convenient form, with two zero blocks in the 2×2 matrix. We focus on these operators in particular, looking at the left and right preconditioned operators

$$\begin{aligned} U_{11}^{-1}A &= \begin{bmatrix} \widehat{S}_{11}^{-1}S_{11} & \mathbf{0} \\ A_{22}^{-1}A_{21} & I \end{bmatrix}, & AL_{11}^{-1} &= \begin{bmatrix} S_{11}\widehat{S}_{11}^{-1} & A_{12}A_{22}^{-1} \\ \mathbf{0} & I \end{bmatrix}, \\ L_{22}^{-1}A &= \begin{bmatrix} I & A_{11}^{-1}A_{12} \\ \mathbf{0} & \widehat{S}_{22}^{-1}S_{22} \end{bmatrix}, & AU_{22}^{-1} &= \begin{bmatrix} I & \mathbf{0} \\ A_{21}A_{11}^{-1} & S_{22}\widehat{S}_{22}^{-1} \end{bmatrix}. \end{aligned}$$

These block triangular operators are easy to raise to powers; for example,

$$(U_{11}^{-1}A)^d = \begin{bmatrix} (\widehat{S}_{11}^{-1}S_{11})^d & \mathbf{0} \\ A_{22}^{-1}A_{21}\sum_{\ell=0}^{d-1}(\widehat{S}_{11}^{-1}S_{11})^\ell & I \end{bmatrix}, \tag{2.9}$$

with similar block structures for $(AL_{11}^{-1})^d$, $(L_{22}^{-1}A)^d$, and $(AU_{22}^{-1})^d$.

Now consider some polynomial $p(t)$ of degree d with coefficients $\{\alpha_i\}$ applied to the preconditioned operator. Diagonal blocks are given by the polynomial directly applied to the diagonal blocks, in this case $p(\widehat{S}_{11}^{-1}S_{11})$ and $p(I)$. One off-diagonal block will be zero and the other (for $p(U_{11}^{-1}A)$) takes the form $A_{22}^{-1}A_{21}F$, where

$$F := \sum_{i=1}^{d}\alpha_i\sum_{\ell=0}^{i-1}(\widehat{S}_{11}^{-1}S_{11})^\ell. \tag{2.10}$$

Assume that $p(t)$ is a consistent polynomial, $p(0) = 1$, as is the case in Krylov or fixed-point iterations. Then $\alpha_0 = 1$, and

$$\begin{aligned} F(I - \widehat{S}_{11}^{-1} S_{11}) &= \sum_{i=1}^{d} \alpha_i I - \sum_{i=1}^{d} \alpha_i (\widehat{S}_{11}^{-1} S_{11})^i \\ &= p(I) - I - (p(\widehat{S}_{11}^{-1} S_{11}) - I) \\ &= p(I) - p(\widehat{S}_{11}^{-1} S_{11}). \end{aligned} \tag{2.11}$$

If $I - \widehat{S}_{11}^{-1} S_{11}$ is invertible, not uncommon in practice as preconditioning often does not invert any particular eigenmode exactly, then

$$p(U_{11}^{-1} A) = \begin{bmatrix} I - \widehat{S}_{11}^{-1} S_{11} & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix} \begin{bmatrix} p(\widehat{S}_{11}^{-1} S_{11}) & \mathbf{0} \\ A_{22}^{-1} A_{21} \left(p(I) - p(\widehat{S}_{11}^{-1} S_{11}) \right) & p(I) \end{bmatrix} \begin{bmatrix} (I - \widehat{S}_{11}^{-1} S_{11})^{-1} & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix}. \tag{2.12}$$

Analogous derivations hold for other block-triangular preconditioners.

2.1.2.3 Equivalence of block-triangular and LDU preconditioners

Notice from Equation (2.8) that the middle term in Equation (2.12) exactly corresponds to $p(M_{11}^{-1} A)$. Applying similar techniques to the other triangular preconditioners above yield the following result on equivalence between consistent polynomials of approximate block-LDU preconditioned and block-triangular preconditioned operators. In particular, this applies to polynomials resulting from fixed-point or Krylov iterations.

Proposition 2.1.2 (Similarity of LDU and triangular preconditioning)**.** *Let $p(t)$ be some consistent polynomial. Then*

$$\begin{aligned} p(U_{11}^{-1} A) \begin{bmatrix} I - \widehat{S}_{11}^{-1} S_{11} & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix} &= \begin{bmatrix} I - \widehat{S}_{11}^{-1} S_{11} & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix} p(M_{11}^{-1} A), \\ p(L_{22}^{-1} A) \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & I - \widehat{S}_{22}^{-1} S_{22} \end{bmatrix} &= \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & I - \widehat{S}_{22}^{-1} S_{22} \end{bmatrix} p(M_{22}^{-1} A), \\ \begin{bmatrix} I - S_{11} \widehat{S}_{11}^{-1} & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix} p(A L_{11}^{-1}) &= p(A M_{11}^{-1}) \begin{bmatrix} I - S_{11} \widehat{S}_{11}^{-1} & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix}, \\ \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & I - S_{22} \widehat{S}_{22}^{-1} \end{bmatrix} p(A U_{22}^{-1}) &= p(A M_{22}^{-1}) \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & I - S_{22} \widehat{S}_{22}^{-1} \end{bmatrix}. \end{aligned}$$

If the Schur-complement fixed-point, for example $I - \widehat{S}_{11}^{-1}S_{11}$, is invertible, then the above equalities are similarity relations between a consistent polynomial applied to an LDU-preconditioned operator and a block-triangular preconditioned operator.

Proof. The proof follows from derivations analogous to those in Section 2.1.2.1 and Section 2.1.2.2. □

Combining with a three-term representation of block LDU preconditioners yields the change of basis matrix between block triangular preconditioner operators and the preconditioned Schur complement. For example, consider $p(L_{22}^{-1}A)$. From Equation (2.6) and Proposition 2.1.2,

$$Qp(L_{22}^{-1}A) = p\left(\begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & \widehat{S}_{22}^{-1}S_{22} \end{bmatrix}\right) Q, \quad \text{where} \quad Q := \begin{bmatrix} I & A_{11}^{-1}A_{12}(I - \widehat{S}_{22}^{-1}S_{22}) \\ \mathbf{0} & I - \widehat{S}_{22}^{-1}S_{22} \end{bmatrix}.$$

If we suppose that $I - \widehat{S}_{22}^{-1}S_{22}$ is invertible, then Q is invertible and we can construct the norm in which fixed-point or Krylov iterations applied to $L_{22}^{-1}A$ are equivalent to the preconditioned Schur complement. For any consistent polynomial $p(t)$,

$$\|p(L_{22}^{-1}A)\| = \left\| p\left(\begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & \widehat{S}_{22}^{-1}S_{22} \end{bmatrix}\right) \right\|_{(QQ^*)^{-1}}, \quad \left\| p\left(\begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & \widehat{S}_{22}^{-1}S_{22} \end{bmatrix}\right) \right\| = \|p(L_{22}^{-1}A)\|_{Q^*Q}.$$

Similar results are straightforward to derive for $p(U_{11}^{-1}A)$, $p(AL_{11}^{-1})$, and $p(AU_{22}^{-1})$.

2.1.2.4 On bounding minimizing Krylov polynomials

To motivate the framework used for most of the proofs to follow in Section 2.2, consider block-LDU preconditioning (for example, Equation (2.8)). Observe that a polynomial $p(t)$ of the preconditioned operator is a block-triangular matrix consisting of combinations of $p(t)$ applied to the preconditioned Schur complement, and $p(I)$. A natural way to bound a minimizing polynomial from above is to then define

$$q(t) := \varphi(t)(1 - t), \tag{2.13}$$

for some consistent polynomial $\varphi(t)$. Applying q to the preconditioned operator eliminates the identity terms, and we are left with, for example, terms involving $\varphi(\widehat{S}^{-1}S)(I - \widehat{S}^{-1}S)$.

This is just one fixed-point iteration applied to the preconditioned Schur complement, and some other consistent polynomial applied to the preconditioned Schur complement, which we can choose to be a certain minimizing polynomial.

Proposition 2.1.2 shows that such an approximation is also convenient for block triangular preconditioning. The $(1 - t)$ term applies the appropriate transformation to the off-diagonal term as in Equation (2.11). As in the case of block-LDU preconditioning, we are then left with a block triangular matrix, with terms consisting of φ applied to the preconditioned Schur complement.

In terms of notation, in this chapter $\varphi^{(d)}$ denotes some form of minimizing polynomial, with superscript (d) indicating the polynomial degree d. Subscripts, e.g., $\varphi_{22}^{(d)}$, indicate a minimizing polynomial for the corresponding (preconditioned) $(2, 2)$-Schur complement, and q denotes a polynomial of the form in Equation (2.13).

2.2 Minimizing Krylov polynomials

This section uses the relations derived in Section 2.1.2 to prove a relation between the Krylov minimizing polynomial for the preconditioned 2×2 operator and that for the preconditioned Schur complement. Approximate block-LDU preconditioning is analyzed in Section 2.2.1, followed by block-triangular preconditioning in Section 2.2.2, and block-Jacobi preconditioning in Section 2.2.3. As mentioned previously, the Krylov method, such as left-preconditioned GMRES, is referred to interchangeably with the equivalent minimizing polynomial.

2.2.1 Approximate block-LDU preconditioning

This section first considers approximate block-LDU preconditioning and GMRES in Theorem 2.2.1, proving equivalence between minimizing polynomials of the 2×2 preconditioned operator and the preconditioned Schur complement. Although we are primarily interested in nonsymmetric operators in this chapter (and thus not CG), it is demonstrated in Theorem 2.2.2 that analogous techniques can be applied to analyze preconditioned CG. Due to the induced matrix norm used in CG, the key step is in deriving a reduced Schur-complement induced norm on the preconditioned Schur complement problem.

Theorem 2.2.1 (Block-LDU preconditioning and GMRES)**.** *Let $\varphi^{(d)}$ denote a minimizing polynomial of the preconditioned operator of degree d in the ℓ^2-norm, for initial residual*

$\mathbf{r} = [\mathbf{r}_1; \mathbf{r}_2]$ *(or initial preconditioned residual for right preconditioning). Let* $\varphi_{kk}^{(d)}$ *be the minimizing polynomial for* $\widehat{S}_{kk}^{-1} S_{kk}$ *in the* ℓ^2*-norm, for initial residual* $\mathbf{r}_k$*, and* $k \in \{1, 2\}$*. Then,*

$$\|\varphi_{11}^{(d)}(\widehat{S}_{11}^{-1}S_{11})\mathbf{r}_1\| \leq \|\varphi^{(d)}(M_{11}^{-1}A)\mathbf{r}\| \leq \left\| \begin{bmatrix} I \\ -A_{22}^{-1}A_{21} \end{bmatrix} (I - \widehat{S}_{11}^{-1}S_{11})\varphi_{11}^{(d-1)}(\widehat{S}_{11}^{-1}S_{11})\mathbf{r}_1 \right\|,$$

$$\frac{1}{\sqrt{2}} \left\|\varphi_{11}^{(d)}(S_{11}\widehat{S}_{11}^{-1})\hat{\mathbf{r}}_1\right\| \leq \|\varphi^{(d)}(AM_{11}^{-1})\mathbf{r}\| \leq \left\|(I - S_{11}\widehat{S}_{11}^{-1})\varphi_{11}^{(d-1)}(S_{11}\widehat{S}_{11}^{-1})\hat{\mathbf{r}}_1\right\|,$$

$$\|\varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{r}_2\| \leq \|\varphi^{(d)}(M_{22}^{-1}A)\mathbf{r}\| \leq \left\| \begin{bmatrix} -A_{11}^{-1}A_{12} \\ I \end{bmatrix} (I - \widehat{S}_{22}^{-1}S_{22})\varphi_{22}^{(d-1)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{r}_2 \right\|,$$

$$\frac{1}{\sqrt{2}} \left\|\varphi_{22}^{(d)}(S_{22}\widehat{S}_{22}^{-1})\hat{\mathbf{r}}_2\right\| \leq \|\varphi^{(d)}(AM_{22}^{-1})\mathbf{r}\| \leq \left\|(I - S_{22}\widehat{S}_{22}^{-1})\varphi_{22}^{(d-1)}(S_{22}\widehat{S}_{22}^{-1})\hat{\mathbf{r}}_2\right\|.$$

where $\hat{\mathbf{r}}_1 := \mathbf{r}_1 - A_{12}A_{22}^{-1}\mathbf{r}_2$ *and* $\hat{\mathbf{r}}_2 := \mathbf{r}_2 - A_{21}A_{11}^{-1}\mathbf{r}_1$*.*

Now let $\varphi^{(d)}$ *and* $\varphi_{kk}^{(d)}$ *denote minimizing polynomials of degree* d *over all vectors in the* ℓ^2*-norm. Then,*

$$\|\varphi_{11}^{(d)}(\widehat{S}_{11}^{-1}S_{11})\| \leq \|\varphi^{(d)}(M_{11}^{-1}A)\| \leq \left\| \begin{bmatrix} I \\ -A_{22}^{-1}A_{21} \end{bmatrix} \right\| \left\|(I - \widehat{S}_{11}^{-1}S_{11})\varphi_{11}^{(d-1)}(\widehat{S}_{11}^{-1}S_{11})\right\|,$$

$$\|\varphi_{11}^{(d)}(\widehat{S}_{11}^{-1}S_{11})\| \leq \|\varphi^{(d)}(AM_{11}^{-1})\| \leq \left\| \begin{bmatrix} I & -A_{12}A_{22}^{-1} \end{bmatrix} \right\| \left\|(I - S_{11}\widehat{S}_{11}^{-1})\varphi_{11}^{(d-1)}(S_{11}\widehat{S}_{11}^{-1})\right\|,$$

$$\|\varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22})\| \leq \|\varphi^{(d)}(M_{22}^{-1}A)\| \leq \left\| \begin{bmatrix} -A_{11}^{-1}A_{12} \\ I \end{bmatrix} \right\| \left\|(I - \widehat{S}_{22}^{-1}S_{22})\varphi_{22}^{(d-1)}(\widehat{S}_{22}^{-1}S_{22})\right\|,$$

$$\|\varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22})\| \leq \|\varphi^{(d)}(AM_{22}^{-1})\| \leq \left\| \begin{bmatrix} -A_{21}A_{11}^{-1} & I \end{bmatrix} \right\| \left\|(I - S_{22}\widehat{S}_{22}^{-1})\varphi_{22}^{(d-1)}(S_{22}\widehat{S}_{22}^{-1})\right\|.$$

Proof. First, recall that left-preconditioned GMRES is equivalent to minimizing the initial residual based on a consistent polynomial in $M_{22}^{-1}A$. Let $\varphi_{22}^{(d)}(t)$ be the minimizing polynomial of degree d for $\widehat{S}_{22}^{-1}S_{22}$, where $\varphi(0) = 1$. Define the degree $d+1$ polynomial $q(t) := \varphi_{22}^{(d)}(t)(1-t)$. Notice that $q(0) = 1$, $q(1) = 0$, and from Equation (2.7) we have

$$q(M_{22}^{-1}A) = \begin{bmatrix} \mathbf{0} & -A_{11}^{-1}A_{12}q(\widehat{S}_{22}^{-1}S_{22}) \\ \mathbf{0} & q(\widehat{S}_{22}^{-1}S_{22}) \end{bmatrix}.$$

Let $\varphi^{(d+1)}$ be the minimizing polynomial in $M_{22}^{-1}A$ of degree $d+1$ for initial residual $\mathbf{r}$. Then,

$$\|\varphi^{(d+1)}(M_{22}^{-1}A)\mathbf{r}\| \leq \|q(M_{22}^{-1}A)\mathbf{r}\| = \left\| \begin{bmatrix} -A_{11}^{-1}A_{12} \\ I \end{bmatrix} (I - \widehat{S}_{22}^{-1}S_{22})\varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{r}_2 \right\|.$$

Taking the supremum over $\mathbf{r}$ and noting that $\|\mathbf{r}\| \geq \|\mathbf{r}_2\|$, this immediately yields an ideal GMRES bound as well, where the minimizing polynomial of degree $d+1$ in norm, $\varphi^{(d+1)}$, is bounded via

$$\|\varphi^{(d+1)}(M_{22}^{-1}A)\| \leq \left\| \begin{bmatrix} -A_{11}^{-1}A_{12} \\ I \end{bmatrix} \right\| \left\| (I - \widehat{S}_{22}^{-1}S_{22})\varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22}) \right\|.$$

Right-preconditioned GMRES is equivalent to the ℓ^2-minimizing consistent polynomial in AM_{22}^{-1} applied to the initial preconditioned residual. A similar proof as above for right preconditioning yields

$$\begin{aligned} \|\varphi^{(d+1)}(AM_{22}^{-1})\mathbf{r}\| &\leq \|(I - S_{22}\widehat{S}_{22}^{-1})\varphi_{22}^{(d)}(S_{22}\widehat{S}_{22}^{-1})\hat{\mathbf{r}}_2\|, \\ \|\varphi^{(d+1)}(AM_{22}^{-1})\| &\leq \left\| \begin{bmatrix} -A_{21}A_{11}^{-1} & I \end{bmatrix} \right\| \left\| (I - S_{22}\widehat{S}_{22}^{-1})\varphi_{22}^{(d)}(S_{22}\widehat{S}_{22}^{-1}) \right\|. \end{aligned}$$

where $\mathbf{r}$ now refers to the initial preconditioned residual, φ refers to minimizing polynomials for AM_{22}^{-1}, and $\hat{\mathbf{r}}_2 := \mathbf{r}_2 - A_{21}A_{11}^{-1}\mathbf{r}_1$.

For a lower bound, let $\varphi^{(d)}$ be the minimizing polynomial of degree d in $M_{22}^{-1}A$ for $\mathbf{r}$. Then, for an ℓ^p-norm with $p \in [1, \infty]$,

$$\begin{aligned} \|\varphi^{(d)}(M_{22}^{-1}A)\mathbf{r}\| &= \left\| \begin{bmatrix} \varphi^{(d)}(I)\mathbf{r}_1 + A_{11}^{-1}A_{12}\left(\varphi^{(d)}(I) - \varphi^{(d)}(\widehat{S}_{22}^{-1}S_{22})\right)\mathbf{r}_2 \\ \varphi^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{r}_2 \end{bmatrix} \right\| \\ &\geq \|\varphi^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{r}_2\| \\ &\geq \|\varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{r}_2\|. \end{aligned}$$

This also yields an ideal GMRES bound, where the minimizing polynomial in norm is bounded via $\|\varphi^{(d)}(M_{22}^{-1}A)\| \geq \|\varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22})\|$. For right preconditioning,

$$\|\varphi^{(d)}(AM_{22}^{-1})\mathbf{r}\| = \left\| \begin{bmatrix} \varphi^{(d)}(I)\mathbf{r}_1 \\ \varphi^{(d)}(I)\mathbf{r}_1 + \varphi^{(d)}(S_{22}\widehat{S}_{22}^{-1})(\mathbf{r}_2 - A_{21}A_{11}^{-1}\mathbf{r}_1) \end{bmatrix} \right\|. \tag{2.14}$$

The ideal GMRES bound follows immediately by noting that the supremum over $\mathbf{r}$ is greater than or equal to setting $\mathbf{r}_1 = \mathbf{0}$ and taking the supremum over $\mathbf{r}_2$, which yields

$$\|\varphi^{(d)}(AM_{22}^{-1})\| \geq \|\varphi^{(d)}(\widehat{S}_{22}^{-1}S_{22})\| \geq \|\varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22})\|.$$

Then, note the identity

$$\begin{aligned}\left\|\begin{bmatrix}\mathbf{x}\\ \mathbf{x}+\mathbf{y}\end{bmatrix}\right\|^2 &= 2\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 + \langle \mathbf{x},\mathbf{y}\rangle + \langle \mathbf{y},\mathbf{x}\rangle \\ &\geq 2\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - 2\|\mathbf{x}\|\|\mathbf{y}\| \geq \frac{\|\mathbf{y}\|^2}{2}.\end{aligned} \tag{2.15}$$

Applying Equation (2.15) to Equation (2.14) with $\mathbf{x} := \varphi^{(d)}(I)\mathbf{r}_1$ and $\mathbf{y} := \varphi^{(d)}(S_{22}\widehat{S}_{22}^{-1})(\mathbf{r}_2 - A_{21}A_{11}^{-1}\mathbf{r}_1)$ yields the lower bound on $\|\varphi^{(d)}(AM_{22}^{-1})\mathbf{r}\|$.

Appealing to Equation (2.8) and analogous derivations yield similar results for the block-LDU preconditioner with Schur-complement approximation in the (1,1)-block. □

Remark 2.2.1 (Left vs. right preconditioning)**.** *Interestingly, there exist vectors* $\mathbf{x}$ *and* $\mathbf{y}$ *such that Equation (2.15) is tight, suggesting there may be specific examples where* $\|\varphi^{(d)}(AM_{22}^{-1})\mathbf{r}\| \leq \left\|\varphi_{22}^{(d)}(S_{22}\widehat{S}_{22}^{-1})\hat{\mathbf{r}}_2\right\|$*. If this is the case (rather than a flaw elsewhere in the line of proof), it means there are initial residuals where the preconditioned* 2×2 *operator converges faster than the corresponding preconditioned Schur complement, a scenario that is not possible with left-preconditioning.*

Although the focus of this chapter is general nonsymmetric operators, similar techniques as used in the proof of Theorem 2.2.1 can be applied to analyze CG, resulting in the following theorem.

Theorem 2.2.2 (LDU preconditioning and CG)**.** *Let* $\varphi^{(d)}$ *be a minimizing polynomial in* $M_{kk}^{-1}A$*, of degree* d*, in the* A*-norm, for initial error vector* $\mathbf{e} = [\mathbf{e}_1;\mathbf{e}_2]$*, and* $k \in \{1,2\}$*. Let* $\varphi_{kk}^{(d)}$ *be the minimizing polynomial for* $\widehat{S}_{kk}^{-1}S_{kk}$ *in the* S_{kk} *norm, for initial error vector* $\mathbf{e}_k$*. Then,*

$$\begin{aligned}\|\varphi_{kk}^{(d)}(\widehat{S}_{kk}^{-1}S_{kk})\mathbf{e}_k\|_{S_{kk}} &\leq \|\varphi^{(d)}(M_{kk}^{-1}A)\mathbf{e}\|_A \\ &\leq \|(I-\widehat{S}_{kk}^{-1}S_{kk})\varphi_{kk}^{(d-1)}(\widehat{S}_{kk}^{-1}S_{kk})\mathbf{e}_k\|_{S_{kk}}.\end{aligned}$$

Now, let $\varphi^{(d)}$ denote minimizing polynomials over all vectors in the appropriate norm (A-norm or S_{kk}-norm), representing worst-case CG convergence. Then,

$$\begin{aligned}\|\varphi_{kk}^{(d)}(\widehat{S}_{kk}^{-1}S_{kk})\|_{S_{kk}} &\leq \|\varphi^{(d)}(M_{kk}^{-1}A)\|_A \\ &\leq \|(I-\widehat{S}_{kk}^{-1}S_{kk})\varphi_{kk}^{(d-1)}(\widehat{S}_{kk}^{-1}S_{kk})\|_{S_{kk}}.\end{aligned}$$

Proof. Here we prove the case of $k=2$. An analogous derivation appealing to Equation (2.8) yields equivalent results for $k=1$.

Recall that CG forms a minimizing consistent polynomial of $M_{22}^{-1}A$ in the A-norm. Let $\varphi^{(d)}$ be the minimizing polynomial of degree d in $M_{22}^{-1}A$ for error vector $\mathbf{e}$ in the A-norm. Then, expressing A in a block LDU sense to simplify the term $A\varphi^{(d+1)}(M_{22}^{-1}A)$, we immediately obtain a lower bound:

$$\begin{aligned}\|\varphi^{(d)}(M_{22}^{-1}A)\mathbf{e}\|_A^2 &= \langle A\varphi^{(d)}(M_{22}^{-1}A)\mathbf{e}, \varphi^{(d)}(M_{22}^{-1}A)\mathbf{e}\rangle \\ &= \left\langle \begin{bmatrix} A_{11} & \mathbf{0} \\ \mathbf{0} & S_{22}\end{bmatrix}\begin{bmatrix}\varphi^{(d)}(I)(\mathbf{e}_1+A_{11}^{-1}A_{12}\mathbf{e}_2) \\ \varphi^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{e}_2\end{bmatrix}, \begin{bmatrix}\varphi^{(d)}(I)(\mathbf{e}_1+A_{11}^{-1}A_{12}\mathbf{e}_2) \\ \varphi^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{e}_2 t\end{bmatrix}\begin{bmatrix}\mathbf{e}_1 \\ \mathbf{e}_2\end{bmatrix}\right\rangle \\ &\geq \langle S_{22}\varphi^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{e}_2, \varphi^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{e}_2\rangle \\ &= \|\varphi^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{e}_2\|_{S_{22}}^2 \\ &= \|\varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{e}_2\|_{S_{22}}^2,\end{aligned}$$

where $\varphi_{22}^{(d)}$ is the minimizing polynomial of degree d in $\widehat{S}_{22}^{-1}S_{22}$ for error vector $\mathbf{e}_2$. A lower bound on the minimizing polynomial of degree d in norm follows immediately by noting that

$$\begin{aligned}\|\varphi^{(d)}(M_{22}^{-1}A)\|_A = \sup_{\mathbf{e}\neq\mathbf{0}}\frac{\|\varphi^{(d)}(M_{22}^{-1}A)\mathbf{e}\|_A}{\|\mathbf{e}\|_A} &\geq \sup_{\substack{\mathbf{e}_2\neq\mathbf{0}, \\ \mathbf{e}_1=-A_{11}^{-1}A_{12}}}\frac{\|\varphi^{(d)}(M_{22}^{-1}A)\mathbf{e}\|_A}{\|\mathbf{e}\|_A} \\ &= \|\varphi^{(d)}(\widehat{S}_{22}^{-1}S_{22})\|_{S_{22}} \geq \|\varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22})\|_{S_{22}}.\end{aligned}$$

For an upper bound, let $\varphi_{22}^{(d)}$ be the minimizing polynomial of degree d in $\widehat{S}_{22}^{-1}S_{22}$ for error vector $\mathbf{e}_2$ in the S_{22}-norm. Define the degree $d+1$ polynomial $q(t) := (1-t)\varphi_{22}^{(d)}(t)$, and let $\varphi^{(d+1)}$ be the minimizing polynomial of degree $d+1$ in $M_{22}^{-1}A$ for error vector $\mathbf{e}$ in

the A-norm. Then

$$\begin{aligned}\|\varphi^{(d+1)}(M_{22}^{-1}A)\mathbf{e}\|_A^2 &\leq \|q(M_{22}^{-1}A)\mathbf{e}\|_A^2 \\ &= \left\langle \begin{bmatrix} A_{11} & \mathbf{0} \\ \mathbf{0} & S_{22} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & q(\widehat{S}_{22}^{-1}S_{22}) \end{bmatrix} \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & q(\widehat{S}_{22}^{-1}S_{22}) \end{bmatrix} \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{bmatrix} \right\rangle \\ &= \langle S_{22}q(\widehat{S}_{22}^{-1}S_{22})\mathbf{e}_2, q(\widehat{S}_{22}^{-1}S_{22})\mathbf{e}_2 \rangle \\ &= \|(I - \widehat{S}_{22}^{-1}S_{22})\varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{e}_2\|_{S_{22}}^2.\end{aligned}$$

For a bound in norm, note that for a fixed $\mathbf{e}_2$, $\|\mathbf{e}\|_A^2$ is a quadratic function in $\mathbf{e}_1$, with minimum obtained at $\mathbf{e}_1 := -A_{11}^{-1}A_{12}$. Then,

$$\begin{aligned}\|\varphi^{(d+1)}(M_{22}^{-1}A)\|_A^2 &= \sup_{\mathbf{e}\neq\mathbf{0}} \frac{\|\varphi^{(d+1)}(M_{22}^{-1}A)\mathbf{e}\|_A^2}{\|\mathbf{e}\|_A^2} \leq \sup_{\mathbf{e}\neq\mathbf{0}} \frac{\|q(M_{22}^{-1}A)\mathbf{e}\|_A^2}{\|\mathbf{e}\|_A^2} \\ = \sup_{\mathbf{e}_2} \frac{\|(I - \widehat{S}_{22}^{-1}S_{22})\varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{e}_2\|_{S_{22}}^2}{\inf_{\mathbf{e}_1}\|\mathbf{e}\|_A^2} &= \sup_{\mathbf{e}_2} \frac{\|(I - \widehat{S}_{22}^{-1}S_{22})\varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{e}_2\|_{S_{22}}^2}{\|\mathbf{e}_2\|_{S_{22}}^2} \\ &= \|(I - \widehat{S}_{22}^{-1}S_{22})\varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22})\|_{S_{22}}^2.\end{aligned}$$

□

From Theorem 2.2.2 we note that for CG, upper and lower inequalities prove that after d iterations, the preconditioned 2×2 system converges at least as accurately as $d-1$ CG iterations on the preconditioned Schur complement, $\widehat{S}_{kk}^{-1}S_{kk}$, plus one fixed-point iteration, and not more accurately than d CG iterations on the preconditioned Schur complement. Because there are operators for which convergence of fixed-point and CG are equivalent, this indicates there are cases for which the upper and lower bounds in Theorem 2.2.2 are tight. Note, these bounds also have no dependence on the off-diagonal blocks, a result not shared by other preconditioners and Krylov methods examined in this chapter. It is unclear if the larger upper bound in GMRES in Theorem 2.2.1 is a flaw in the line of proof, or if CG on the preconditioned 2×2 system can achieve slightly better convergence (in the appropriate norm) with respect to the preconditioned Schur complement than GMRES.

2.2.2 Block-triangular preconditioning

In this section we consider block-triangular preconditioning. In particular, we prove equivalence between minimizing polynomials of the 2×2 preconditioned operator and the preconditioned Schur complement for block-triangular preconditioning. We consider separately

left preconditioning in Theorem 2.2.3 and right preconditioning in Theorem 2.2.4. Theorems are stated for the preconditioners that take the simplest form in Equation (2.4) (left vs. right and Schur complement in the (1,1)- or (2,2)-block), the same as those discussed in Section 2.1.2. However, note that, for example, any polynomial $p(U_{22}^{-1}A) = U_{22}^{-1}p(AU_{22}^{-1})U_{22}$. Thus, if we prove a result for left preconditioning with U_{22}^{-1}, a similar result holds for right preconditioning, albeit with modified constants/residual. Such results are not stated here for the sake of brevity.

Theorem 2.2.3 (Left block-triangular preconditioning and GMRES)**.** *Let $\varphi^{(d)}$ denote a minimizing polynomial of the preconditioned operator of degree d in the ℓ^2-norm, for initial residual $\mathbf{r} = [\mathbf{r}_1; \mathbf{r}_2]$. Let $\varphi_{kk}^{(d)}$ be the minimizing polynomial for $\widehat{S}_{kk}^{-1}S_{kk}$ in the ℓ^2-norm, for initial residual $\mathbf{r}_k$, and $k \in \{1, 2\}$. Then,*

$$\|\varphi_{11}^{(d)}(\widehat{S}_{11}^{-1}S_{11})\mathbf{r}_1\| \leq \|\varphi^{(d)}(U_{11}^{-1}A)\mathbf{r}\| \leq \left\| \begin{bmatrix} I - \widehat{S}_{11}^{-1}S_{11} \\ -A_{22}^{-1}A_{21} \end{bmatrix} \varphi_{11}^{(d-1)}(\widehat{S}_{11}^{-1}S_{11})\mathbf{r}_1 \right\|,$$

$$\|\varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{r}_2\| \leq \|\varphi^{(d)}(L_{22}^{-1}A)\mathbf{r}\| \leq \left\| \begin{bmatrix} -A_{11}^{-1}A_{12} \\ I - \widehat{S}_{22}^{-1}S_{22} \end{bmatrix} \varphi_{22}^{(d-1)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{r}_2 \right\|.$$

Now let $\varphi^{(d)}$ and $\varphi_{kk}^{(d)}$ denote minimizing polynomials of degree d over all vectors in the ℓ^2-norm (instead of for the initial residual). Then,

$$\|\varphi_{11}^{(d)}(\widehat{S}_{11}^{-1}S_{11})\| \leq \|\varphi^{(d)}(U_{11}^{-1}A)\| \leq \left\| \begin{bmatrix} I - \widehat{S}_{11}^{-1}S_{11} \\ -A_{22}^{-1}A_{21} \end{bmatrix} \right\| \left\| \varphi_{11}^{(d-1)}(\widehat{S}_{11}^{-1}S_{11}) \right\|,$$

$$\|\varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22})\| \leq \|\varphi^{(d)}(L_{22}^{-1}A)\| \leq \left\| \begin{bmatrix} -A_{11}^{-1}A_{12} \\ I - \widehat{S}_{22}^{-1}S_{22} \end{bmatrix} \right\| \left\| \varphi_{22}^{(d-1)}(\widehat{S}_{22}^{-1}S_{22}) \right\|.$$

Proof. Recall left-preconditioned GMRES is equivalent to the minimizing consistent polynomial in the ℓ^2-norm over the preconditioned operator, for initial residual $\mathbf{r}$. Consider lower-triangular preconditioning with an approximate Schur complement in the (2,2)-block,

$$L_{22}^{-1}A = \begin{bmatrix} I & A_{11}^{-1}A_{12} \\ \mathbf{0} & \widehat{S}_{22}^{-1}S_{22} \end{bmatrix}. \tag{2.16}$$

Let $\varphi(t)$ be some consistent polynomial, and define a second consistent polynomial $q(t) := (1-t)\varphi(t)$. Plugging in $t = L_{22}^{-1}A$ and expanding the polynomial φ analogous to the steps

in Equation (2.9) and Equation (2.10) yields

$$q(L_{22}^{-1}A) = \begin{bmatrix} \mathbf{0} & -A_{11}^{-1}A_{12} \\ \mathbf{0} & I - \widehat{S}_{22}^{-1}S_{22} \end{bmatrix} \begin{bmatrix} \varphi(I) & F \\ \mathbf{0} & \varphi(\widehat{S}_{22}^{-1}S_{22}) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -A_{11}^{-1}A_{12}\varphi(\widehat{S}_{22}^{-1}S_{22}) \\ \mathbf{0} & (I - \widehat{S}_{22}^{-1}S_{22})\varphi(\widehat{S}_{22}^{-1}S_{22}) \end{bmatrix},$$

where F is the upper left block of $q(L_{22}^{-1}A)$, similar to Equation (2.10).

Now let $\varphi^{(d)}$ be the minimizing polynomial in $L_{22}^{-1}A$ of degree d for initial residual $\mathbf{r} = [\mathbf{r}_1; \mathbf{r}_2]$, and $\varphi_{22}^{(d)}$ be the minimizing polynomial in $\widehat{S}_{22}^{-1}S_{22}$ of degree d for initial residual $\mathbf{r}_2$. Define the degree d polynomial $q(t) := (1-t)\varphi_{22}^{(d-1)}(t)$. Then

$$\|\varphi^{(d)}(L_{22}^{-1}A)\mathbf{r}\| \leq \|q(L_{22}^{-1}A)\mathbf{r}\| = \left\| \begin{bmatrix} -A_{11}^{-1}A_{12} \\ I - \widehat{S}_{22}^{-1}S_{22} \end{bmatrix} \varphi_{22}^{(d-1)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{r}_2 \right\|.$$

Taking the supremum over $\mathbf{r}$ and appealing to the submultiplicative property of norms yields an upper bound on the minimizing polynomial in norm as well,

$$\|\varphi^{(d)}(L_{22}^{-1}A)\| \leq \left\| \begin{bmatrix} -A_{11}^{-1}A_{12} \\ I - \widehat{S}_{22}^{-1}S_{22} \end{bmatrix} \right\| \left\| \varphi_{22}^{(d-1)}(\widehat{S}_{22}^{-1}S_{22}) \right\|.$$

A lower bound is also obtained in a straightforward manner for initial residual $\mathbf{r}$,

$$\|\varphi^{(d)}(L_{22}^{-1}A)\mathbf{r}\| = \left\| \begin{bmatrix} \varphi(I)^{(d)}\mathbf{r}_1 + F\mathbf{r}_2 \\ \varphi^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{r}_2 \end{bmatrix} \right\| \geq \left\| \varphi^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{r}_2 \right\| \geq \left\| \varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22})\mathbf{r}_2 \right\|,$$

which can immediately be extended to a lower bound on the minimizing polynomial in norm as well,

$$\|\varphi^{(d)}(L_{22}^{-1}A)\| \geq \left\| \varphi_{22}^{(d)}(\widehat{S}_{22}^{-1}S_{22}) \right\|.$$

Analogous derivations yield bounds for an upper-triangular preconditioner with approximate Schur complement in the (1,1)-block. □

We next consider right block-triangular preconditioning.

Theorem 2.2.4 (Right block-triangular preconditioning and GMRES)**.** *Let $\varphi^{(d)}$ denote a minimizing polynomial of the preconditioned operator of degree d in the ℓ^2-norm, for initial preconditioned residual $\mathbf{r} = [\mathbf{r}_1; \mathbf{r}_2]$. Let $\varphi_{kk}^{(d)}$ be the minimizing polynomial for $\widehat{S}_{kk}^{-1} S_{kk}$ in the ℓ^2-norm, for initial residual $\hat{\mathbf{r}}_k$, and $k \in \{1, 2\}$. Then,*

$$\|\varphi^{(d)}(AL_{11}^{-1})\mathbf{r}\| \leq \left\|\varphi_{11}^{(d-1)}(S_{11}\widehat{S}_{11}^{-1})\hat{\mathbf{r}}_1\right\|,$$
$$\|\varphi^{(d)}(AU_{22}^{-1})\mathbf{r}\| \leq \left\|\varphi_{22}^{(d-1)}(S_{22}\widehat{S}_{22}^{-1})\hat{\mathbf{r}}_2\right\|,$$

where $\hat{\mathbf{r}}_1 := (I - S_{11}\widehat{S}_{11}^{-1})\mathbf{r}_1 - A_{12}A_{22}^{-1}\mathbf{r}_2$ and $\hat{\mathbf{r}}_2 := (I - S_{22}\widehat{S}_{22}^{-1})\mathbf{r}_2 - A_{21}A_{11}^{-1}\mathbf{r}_1$.

Now let $\varphi^{(d)}$ and $\varphi_{kk}^{(d)}$ denote minimizing polynomials of degree d over all vectors in the ℓ^2-norm (instead of for the initial preconditioned residual). Then,

$$\|\varphi_{11}^{(d)}(S_{11}\widehat{S}_{11}^{-1})\| \leq \|\varphi^{(d)}(AL_{11}^{-1})\| \leq \left\|\begin{bmatrix} I - S_{11}\widehat{S}_{11}^{-1} & -A_{12}A_{22}^{-1}\end{bmatrix}\right\| \left\|\varphi_{11}^{(d-1)}(S_{11}\widehat{S}_{11}^{-1})\right\|,$$
$$\|\varphi_{22}^{(d)}(S_{22}\widehat{S}_{22}^{-1})\| \leq \|\varphi^{(d)}(AU_{22}^{-1})\| \leq \left\|\begin{bmatrix} -A_{21}A_{11}^{-1} & I - S_{22}\widehat{S}_{22}^{-1}\end{bmatrix}\right\| \left\|\varphi_{22}^{(d-1)}(S_{22}\widehat{S}_{22}^{-1})\right\|.$$

Proof. Recall right-preconditioned GMRES is equivalent to the minimizing consistent polynomial in the ℓ^2-norm over the right-preconditioned operator, for initial preconditioned residual $\mathbf{r}$. Consider

$$AL_{11}^{-1} = \begin{bmatrix} S_{11}\widehat{S}_{11}^{-1} & A_{12}A_{22}^{-1} \\ \mathbf{0} & I \end{bmatrix}.$$

Defining $q(t) = \varphi(t)(1 - t)$ we note that

$$\begin{aligned} q(AL_{11}^{-1}) &= \varphi(AL_{11}^{-1})(I - AL_{11}^{-1}) \\ &= \begin{bmatrix} \varphi(S_{11}\widehat{S}_{11}^{-1}) & F \\ \mathbf{0} & \varphi(I) \end{bmatrix} \begin{bmatrix} I - S_{11}\widehat{S}_{11}^{-1} & -A_{12}A_{22}^{-1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} (I - S_{11}\widehat{S}_{11}^{-1})\varphi(S_{11}\widehat{S}_{11}^{-1}) & -\varphi(S_{11}\widehat{S}_{11}^{-1})A_{12}A_{22}^{-1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \end{aligned}$$

Then,

$$\begin{aligned}\|\varphi^{(d)}(AL_{11}^{-1})\mathbf{r}\| &\leq \|q(AL_{11}^{-1})\mathbf{r}\| \\ &= \left\|\varphi_{11}^{(d-1)}(S_{11}\widehat{S}_{11}^{-1})\left((I - S_{11}\widehat{S}_{11}^{-1})\mathbf{r}_1 - A_{12}A_{22}^{-1}\mathbf{r}_2\right)\right\|, \\ \|\varphi^{(d)}(AL_{11}^{-1})\| &\leq \left\|\begin{bmatrix} I - S_{11}\widehat{S}_{11}^{-1} & -A_{12}A_{22}^{-1}\end{bmatrix}\right\| \left\|\varphi_{11}^{(d-1)}(S_{11}\widehat{S}_{11}^{-1})\right\|.\end{aligned}$$

The lower bound on the minimizing polynomial in norm is obtained by noting

$$\begin{aligned}\|\varphi^{(d)}(AL_{11}^{-1})\| &= \sup_{\mathbf{r}\neq\mathbf{0}} \frac{\left\|\begin{bmatrix}\varphi^{(d)}(S_{11}\widehat{S}_{11}^{-1}) & F \\ \mathbf{0} & \varphi(I)\end{bmatrix}\begin{bmatrix}\mathbf{r}_1 \\ \mathbf{r}_2\end{bmatrix}\right\|}{\|\mathbf{r}\|} \\ &\geq \sup_{\mathbf{r}_1\neq\mathbf{0}} \frac{\left\|\begin{bmatrix}\varphi^{(d)}(S_{11}\widehat{S}_{11}^{-1}) & F \\ \mathbf{0} & \varphi(I)\end{bmatrix}\begin{bmatrix}\mathbf{r}_1 \\ \mathbf{0}\end{bmatrix}\right\|}{\|\mathbf{r}_1\|} = \|\varphi_{11}^{(d)}(S_{11}\widehat{S}_{11}^{-1})\|.\end{aligned}$$

Analogous derivations as above yield bounds for the right upper triangular preconditioner with Schur complement in the (2,2)-block, AU_{22}^{-1}.

□

As discussed previously, similar results as Theorem 2.2.4 hold for preconditioning with U_{11}^{-1} and L_{22}^{-1}. However, it is not clear if a lower bound for specific initial residual, as proven for block-LDU and left block-triangular preconditioning in Theorem 2.2.1 and Theorem 2.2.3, holds for right block-triangular preconditioning. For block-LDU preconditioning, the lower bound is weaker for right preconditioning, including a factor of $1/\sqrt{2}$.

2.2.3 Block-Jacobi preconditioning

In this section we prove equivalence between minimizing polynomials of the 2×2 preconditioned operator and the preconditioned Schur complement for block-Jacobi preconditioning.

Let $q(t)$ be some polynomial in t, where $q(0) = 1$. Note that q can always be written equivalently as a polynomial $p(1 - t)$, under the constraint that the sum of polynomial

coefficients for p, say $\{\alpha_i\}$, sum to one (to enforce $q(0) = 1$). Thus let $\varphi^{(2d)}(D^{-1}A)$ be the minimizing polynomial of degree $2d$ in the ℓ^2-norm, and let us express this equivalently as a polynomial p, where

$$\begin{aligned} p(I - D^{-1}A) &:= \sum_{i=0}^{2d} \alpha_i (I - D^{-1}A)^i \\ &= \sum_{i=0}^{d} \alpha_{2i} (I - D^{-1}A)^{2i} + \sum_{i=0}^{d-1} \alpha_{2i+1} (I - D^{-1}A)^{2i+1} \\ &= \sum_{i=0}^{d} \alpha_{2i} (I - D^{-1}A)^{2i} + (I - D^{-1}A) \sum_{i=0}^{d-1} \alpha_{2i+1} (I - D^{-1}A)^{2i}. \end{aligned}$$

From Equation (2.5), even powers of $I - D^{-1}A$ take a block diagonal form, and we can write

$$\begin{aligned} p(I - D^{-1}A) &= \begin{bmatrix} \hat{p}(A_{11}^{-1}A_{12}A_{22}^{-1}A_{21}) & \mathbf{0} \\ \mathbf{0} & \hat{p}(A_{22}^{-1}A_{21}A_{11}^{-1}A_{12}) \end{bmatrix} \\ &\quad + \begin{bmatrix} \mathbf{0} & -A_{11}^{-1}A_{12} \\ -A_{22}^{-1}A_{21} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{p}(A_{11}^{-1}A_{12}A_{22}^{-1}A_{21}) & \mathbf{0} \\ \mathbf{0} & \tilde{p}(A_{22}^{-1}A_{21}A_{11}^{-1}A_{12}) \end{bmatrix} \\ &= \begin{bmatrix} \hat{p}(A_{11}^{-1}A_{12}A_{22}^{-1}A_{21}) & -A_{11}^{-1}A_{12}\tilde{p}(A_{22}^{-1}A_{21}A_{11}^{-1}A_{12}) \\ -A_{22}^{-1}A_{21}\tilde{p}(A_{11}^{-1}A_{12}A_{22}^{-1}A_{21}) & \hat{p}(A_{22}^{-1}A_{21}A_{11}^{-1}A_{12}) \end{bmatrix} \\ &= \begin{bmatrix} \hat{p}(I - A_{11}^{-1}S_{11}) & -A_{11}^{-1}A_{12}\tilde{p}(I - A_{22}^{-1}S_{22}) \\ -A_{22}^{-1}A_{21}\tilde{p}(I - A_{11}^{-1}S_{11}) & \hat{p}(I - A_{22}^{-1}S_{22}) \end{bmatrix}, \end{aligned} \tag{2.17}$$

where $\hat{p}$ and $\tilde{p}$ are degree d and $d-1$ polynomials with coefficients $\{\hat{\alpha}_i\} \leftarrow\!\shortmid \{\alpha_{2i}\}$ and $\{\tilde{\alpha}_i\} \leftarrow\!\shortmid \{\alpha_{2i+1}\}$, respectively. This is the primary observation leading to the proof of Theorem 2.2.5. Also note the identities that for any polynomial q,

$$A_{11}^{-1}A_{12}q(I - A_{22}^{-1}S_{22}) = q(I - A_{11}^{-1}S_{11})A_{11}^{-1}A_{12}, \tag{2.18a}$$
$$A_{22}^{-1}A_{21}q(I - A_{11}^{-1}S_{11}) = q(I - A_{22}^{-1}S_{22})A_{22}^{-1}A_{21}, \tag{2.18b}$$

which will be used with Equation (2.17) in the derivations that follow.

Theorem 2.2.5 (Block-Jacobi preconditioning & ideal GMRES)**.** *Let $\varphi(D^{-1}A)$ be the worst-case consistent minimizing polynomial of degree $2d$, in the ℓ^p-norm, $p \in [1, \infty]$, for*

$D^{-1}A$. *Let* $\varphi_{11}^{(d)}$ *and* $\varphi_{22}^{(d)}$ *be the minimizing polynomials of degree* d *in the same norm, for* $A_{11}^{-1}S_{11}$ *and* $A_{22}^{-1}S_{22}$*, respectively. Then,*

$$\|\varphi(D^{-1}A)\| \geq \frac{\min\left\{\|\varphi_{11}^{(d)}(A_{11}^{-1}S_{11})\|, \|\varphi_{11}^{(d)}(A_{22}^{-1}S_{22})\|\right\}}{1+\min\left\{\|A_{11}^{-1}A_{12}\|, \|A_{22}^{-1}A_{21}\|\right\}},$$
$$\frac{\|\varphi(D^{-1}A)\|}{\|A_{22}^{-1}A_{21}\|+\|A_{11}^{-1}A_{12}\|} \leq \min\left\{\|\varphi_{11}^{(d-1)}(A_{11}^{-1}S_{11})\|, \|\varphi_{22}^{(d-1)}(A_{22}^{-1}S_{22})\|\right\}.$$

Similarly, now let $\varphi_{11}^{(d)}$ *and* $\varphi_{22}^{(d)}$ *be the minimizing polynomials of degree* d *for* $S_{11}A_{11}^{-1}$ *and* $S_{22}A_{22}^{-1}$*, respectively. Then,*

$$\|\varphi(AD^{-1})\| \geq \frac{\min\left\{\|\varphi_{11}^{(d)}(S_{11}A_{11}^{-1})\|, \|\varphi_{22}^{(d)}(S_{22}A_{22}^{-1})\|\right\}}{1+\min\left\{\|A_{21}A_{11}^{-1}\|, \|A_{12}A_{22}^{-1}\|\right\}},$$
$$\frac{\|\varphi(AD^{-1})\|}{\|A_{12}A_{22}^{-1}\|+\|A_{21}A_{11}^{-1}\|} \leq \min\left\{\|\varphi_{11}^{(d-1)}(S_{11}A_{11}^{-1})\|, \|\varphi_{22}^{(d-1)}(S_{22}A_{22}^{-1})\|\right\}.$$

Proof. Recall preconditioned GMRES is equivalent to the minimizing consistent polynomial in the ℓ^2-norm over the preconditioned operator. We start with the lower bounds. Let $\varphi^{(2d)}$ be the consistent minimizing polynomial (in norm) of degree $2d$ for $D^{-1}A$, and let $p(t)$ be a polynomial such that $p(I-D^{-1}A)=\varphi(D^{-1}A)$, where coefficients of p, say

$\{\alpha_i\}$ are such that $\sum \alpha_i = 1$. From Equation (2.17) and Equation (2.18),

$$\begin{aligned}
\|p(I - D^{-1}A)\| &= \sup_{\mathbf{r}\neq\mathbf{0}} \frac{\left\| \begin{bmatrix} \hat{p}(I - A_{11}^{-1}S_{11}) & -\tilde{p}(I - A_{11}^{-1}S_{11})A_{11}^{-1}A_{12} \\ -\tilde{p}(I - A_{22}^{-1}S_{22})A_{22}^{-1}A_{21} & \hat{p}(I - A_{22}^{-1}S_{22}) \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{bmatrix} \right\|}{\|\mathbf{r}\|} \\
&\geq \sup_{\mathbf{r}_1\neq\mathbf{0}} \frac{\left\| \begin{bmatrix} \hat{p}(I - A_{11}^{-1}S_{11}) & -\tilde{p}(I - A_{11}^{-1}S_{11})A_{11}^{-1}A_{12} \\ -\tilde{p}(I - A_{22}^{-1}S_{22})A_{22}^{-1}A_{21} & \hat{p}(I - A_{22}^{-1}S_{22}) \end{bmatrix} \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{0} \end{bmatrix} \right\|}{\|\mathbf{r}_1\|} \\
&\geq \max\left\{ \sup_{\mathbf{r}_1\neq\mathbf{0}} \frac{\left\|\hat{p}(I - A_{11}^{-1}S_{11})\mathbf{r}_1\right\|}{\|\mathbf{r}_1\|}, \sup_{\mathbf{r}_1\neq\mathbf{0}} \frac{\left\|\tilde{p}(I - A_{22}^{-1}S_{22})A_{22}^{-1}A_{21}\mathbf{r}_1\right\|}{\|\mathbf{r}_1\|} \right\} \\
&\geq \max\left\{ \sup_{\mathbf{r}_1\neq\mathbf{0}} \frac{\left\|\hat{p}(I - A_{11}^{-1}S_{11})\mathbf{r}_1\right\|}{\|\mathbf{r}_1\|}, \sup_{A_{11}^{-1}A_{12}\tilde{\mathbf{r}}_2\neq\mathbf{0}} \frac{\left\|\tilde{p}(I - A_{22}^{-1}S_{22})A_{22}^{-1}A_{21}A_{11}^{-1}A_{12}\tilde{\mathbf{r}}_2\right\|}{\|A_{11}^{-1}A_{12}\tilde{\mathbf{r}}_2\|} \right\} \\
&\geq \max\left\{ \sup_{\mathbf{r}_1\neq\mathbf{0}} \frac{\left\|\hat{p}(I - A_{11}^{-1}S_{11})\mathbf{r}_1\right\|}{\|\mathbf{r}_1\|}, \sup_{\tilde{\mathbf{r}}_2\neq\mathbf{0}} \frac{\left\|\tilde{p}(I - A_{22}^{-1}S_{22})(I - A_{22}^{-1}S_{22})\tilde{\mathbf{r}}_2\right\|}{\|A_{11}^{-1}A_{12}\|\|\tilde{\mathbf{r}}_2\|} \right\}.
\end{aligned}$$

Note that the step introducing the maximum in the third line holds for ℓ^p-norms, $p \in [1, \infty]$.

Now recall that to enforce $\varphi^{(2d)}(0) = 1$, it must be the case that coefficients of p sum to one. Thus, it must be the case that for coefficients of $\hat{p}$ and $\tilde{p}$, say $\{\hat{\alpha}_i\}$ and $\{\tilde{\alpha}_i\}$, $\sum_i \hat{\alpha}_i + \sum_i \tilde{\alpha}_i := \hat{s} + \tilde{s} = 1$. Let us normalize such that each polynomial within the supremum has coefficients of sum one, which yields

$$\begin{aligned}
&\|p(I - D^{-1}A)\| \\
&\quad\geq \max\left\{ |\hat{s}| \sup_{\mathbf{r}_1\neq\mathbf{0}} \frac{\left\|\hat{p}(I - A_{11}^{-1}S_{11})\mathbf{r}_1\right\|}{|\hat{s}|\|\mathbf{r}_1\|}, |\tilde{s}| \sup_{\tilde{\mathbf{r}}_2\neq\mathbf{0}} \frac{\left\|\tilde{p}(I - A_{22}^{-1}S_{22})(I - A_{22}^{-1}S_{22})\tilde{\mathbf{r}}_2\right\|}{|\tilde{s}|\|A_{11}^{-1}A_{12}\|\|\tilde{\mathbf{r}}_2\|} \right\} \\
&\quad\geq \max\left\{ |\hat{s}| \left\|\varphi_{11}^{(d)}(A_{11}^{-1}S_{11})\right\|, \frac{|1 - \hat{s}|}{\|A_{11}^{-1}A_{12}\|} \left\|\varphi_{22}^{(d)}(A_{22}^{-1}S_{22})\right\| \right\} \\
&\quad:= \max\left\{ |\hat{s}|C_1, \frac{|1 - \hat{s}|}{\|A_{11}^{-1}A_{12}\|}C_2 \right\}, \qquad (2.19)
\end{aligned}$$

where $\varphi_{11}^{(d)}$ is the minimizing polynomial of degree d of $A_{11}^{-1}S_{11}$, and similarly for $\varphi_{22}^{(d)}$ and $A_{22}^{-1}S_{22}$. In the 22-case, note that $\tilde{p}(I - A_{22}^{-1}S_{22})(I - A_{22}^{-1}S_{22})$ can be expressed as

a polynomial of degree d in $A_{22}^{-1}S_{22}$. Furthermore, in expressing the two polynomials, $\tilde{p}(I - A_{22}^{-1}S_{22})$ and the product $\tilde{p}(I - A_{22}^{-1}S_{22})(I - A_{22}^{-1}S_{22})$, as polynomials in $A_{22}^{-1}S_{22}$, the identity coefficients are equal. In particular, when scaling by $1/|\tilde{s}|$, both polynomials are equivalent to consistent polynomials in $A_{22}^{-1}S_{22}$ of degree $d-1$ and d, respectively. This allows us to bound the polynomials in $A_{22}^{-1}S_{22}$ (as well as $A_{11}^{-1}S_{11}$) from below using the true worst-case minimizing polynomial (in norm).

To derive bounds for all p, we now minimize over $\hat{s}$. If $\hat{s} \geq 1$, it follows that $\|p(I - D^{-1}A)\| \geq C_1$, and for $\hat{s} \leq 0$, we have $\|p(I - D^{-1}A)\| \geq \frac{C_2}{\|A_{11}^{-1}A_{12}\|}$. For $\hat{s} \in (0,1)$, the minimum over $\hat{s}$ of the maximum in Equation (2.19) is obtained at $\hat{s}$ such that $\hat{s}C_1 = \frac{1-\hat{s}}{\|A_{11}^{-1}A_{12}\|}C_2$, or $\hat{s}_{min} := \frac{C_2}{\|A_{11}^{-1}A_{12}\|C_1+C_2}$. Evaluating yields

$$\|\varphi^{(2d)}(D^{-1}A)\| = \|p(I - D^{-1}A)\| \geq \frac{C_1C_2}{\|A_{11}^{-1}A_{12}\|C_1 + C_2} \geq \frac{\min\{C_1, C_2\}}{1 + \|A_{11}^{-1}A_{12}\|}.$$

An analogous proof as above, but initially setting $\mathbf{r}_1 = \mathbf{0}$ rather than $\mathbf{r}_2$ yields a similar result,

$$\|\varphi^{(2d)}(D^{-1}A)\| = \|p(I - D^{-1}A)\| \geq \frac{\min\{C_1, C_2\}}{1 + \|A_{22}^{-1}A_{21}\|}.$$

Right preconditioning follows an analogous derivation, where $\varphi^{(2d)}(AD^{-1}) = p(I - AD^{-1})$ instead takes the form

$$p(I - AD^{-1}) = \begin{bmatrix} \hat{p}(I - S_{11}A_{11}^{-1}) & -A_{12}A_{22}^{-1}\tilde{p}(I - S_{22}A_{22}^{-1}) \\ -A_{21}A_{11}^{-1}\tilde{p}(I - S_{11}A_{11}^{-1}) & \hat{p}(I - S_{22}A_{22}^{-1}) \end{bmatrix}.$$

Next, we prove the upper bounds. Similar to previously, from Equation (2.17) we have

$$\begin{aligned} \|p(I - D^{-1}A)\| &= \left\| \begin{bmatrix} \hat{p}(I - A_{11}^{-1}S_{11}) & -\tilde{p}(I - A_{11}^{-1}S_{11})A_{11}^{-1}A_{12} \\ -A_{22}^{-1}A_{21}\tilde{p}(I - A_{11}^{-1}S_{11}) & \hat{p}(I - A_{22}^{-1}S_{22}) \end{bmatrix} \right\| \\ &\leq \left\| \begin{bmatrix} \hat{q}(I - A_{11}^{-1}S_{11}) & -\tilde{q}(I - A_{11}^{-1}S_{11})A_{11}^{-1}A_{12} \\ -A_{22}^{-1}A_{21}\tilde{q}(I - A_{11}^{-1}S_{11}) & \hat{q}(I - A_{22}^{-1}S_{22}) \end{bmatrix} \right\|, \end{aligned}$$

for polynomials $\hat{q}$ and $\tilde{q}$ of degree d and $d-1$ such that coefficients satisfy $\sum_i \hat{\alpha}_i + \sum_i \tilde{\alpha}_i = 1$.

Let $\hat{q} = \mathbf{0}$. Then,

$$\begin{aligned}
\|p(I - D^{-1}A)\| &\leq \left\| \begin{bmatrix} \mathbf{0} & -\tilde{q}(I - A_{11}^{-1}S_{11})A_{11}^{-1}A_{12} \\ -A_{22}^{-1}A_{21}\tilde{q}(I - A_{11}^{-1}S_{11}) & \mathbf{0} \end{bmatrix} \right\| \\
&\leq \left\| \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -A_{22}^{-1}A_{21}\tilde{q}(I - A_{11}^{-1}S_{11}) & \mathbf{0} \end{bmatrix} \right\| + \left\| \begin{bmatrix} \mathbf{0} & -\tilde{q}(I - A_{11}^{-1}S_{11})A_{11}^{-1}A_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right\| \\
&= \|A_{22}^{-1}A_{21}\tilde{q}(I - A_{11}^{-1}S_{11})\| + \|\tilde{q}(I - A_{11}^{-1}S_{11})A_{11}^{-1}A_{12}\| \\
&\leq \|\tilde{q}(I - A_{11}^{-1}S_{11})\| \left(\|A_{22}^{-1}A_{21}\| + \|A_{11}^{-1}A_{12}\| \right).
\end{aligned}$$

Recalling that $\tilde{q}(I - A_{11}^{-1}S_{11})A_{11}^{-1}A_{12} = A_{11}^{-1}A_{12}\tilde{q}(I - A_{22}^{-1}S_{22})$ and $A_{22}^{-1}A_{21}\tilde{q}(I - A_{11}^{-1}S_{11}) = \tilde{q}(I - A_{22}^{-1}S_{22})A_{22}^{-1}A_{21}$ for any polynomial $\tilde{q}$, we also have the equivalent result

$$\|p(I - D^{-1}A)\| \leq \|\tilde{q}(I - A_{22}^{-1}S_{22})\| \left(\|A_{22}^{-1}A_{21}\| + \|A_{11}^{-1}A_{12}\| \right).$$

Let $\varphi_{11}^{(d-1)}(t)$ and $\varphi_{22}^{(d-1)}(t)$ denote the consistent worst-case minimizing polynomials of degree $d-1$ for $A_{11}^{-1}S_{11}$ and $A_{22}^{-1}S_{22}$, respectively. Note, $\tilde{q}$ is also a polynomial of degree $d-1$ in (without loss of generality) $A_{11}^{-1}S_{11}$. Because coefficients of $\tilde{q}$ satisfy $\sum_i \tilde{\alpha}_i = 1$, $\tilde{q}(I - A_{11}^{-1}S_{11})$ can equivalently be expressed as a consistent polynomial in $(A_{11}^{-1}S_{11})$. Thus let $\tilde{q}(I - A_{11}^{-1}S_{11}) := \varphi_{11}^{(d-1)}(A_{11}^{-1}S_{11})$. Analogous steps for $A_{22}^{-1}S_{22}$ yield bounds

$$\begin{aligned}
\|\varphi^{(2d)}(D^{-1}A)\| &\leq \|\varphi_{11}^{(d-1)}(A_{11}^{-1}S_{11})\| \left(\|A_{22}^{-1}A_{21}\| + \|A_{11}^{-1}A_{12}\| \right), \\
\|\varphi^{(2d)}(D^{-1}A)\| &\leq \|\varphi_{22}^{(d-1)}(A_{22}^{-1}S_{22})\| \left(\|A_{22}^{-1}A_{21}\| + \|A_{11}^{-1}A_{12}\| \right).
\end{aligned}$$

Similar to the proof of a lower bound, an analogous derivation as above yields right preconditioning bounds

$$\begin{aligned}
\|\varphi^{(2d)}(AD^{-1})\| &\leq \|\varphi_{11}^{(d-1)}(S_{11}A_{11}^{-1})\| \left(\|A_{12}A_{22}^{-1}\| + \|A_{21}A_{11}^{-1}\| \right), \\
\|\varphi^{(2d)}(AD^{-1})\| &\leq \|\varphi_{22}^{(d-1)}(S_{22}A_{22}^{-1})\| \left(\|A_{12}A_{22}^{-1}\| + \|A_{21}A_{11}^{-1}\| \right),
\end{aligned}$$

where φ now denotes minimizing polynomials associated with right preconditioning. □

Remark 2.2.2 (General block-diagonal preconditioner)**.** *This section proved results for block-Jacobi preconditioners, where the preconditioner inverts the diagonal blocks of the original matrix, and convergence is defined by the underlying preconditioned Schur complement. However, such results do not extend to more general block-diagonal preconditioners*

with Schur-complement approximation $\widehat{S}_{22} \neq A_{22}$. In [149], examples are constructed where block-diagonal preconditioning with an exact Schur complement take several hundred iterations to converge, while block-triangular preconditioning with an exact Schur complement requires only three iterations (the extra iteration over a theoretical max of two is likely due to floating point error).

2.3 The steady linearized Navier–Stokes equations

To demonstrate the new theory in practice, we consider a finite-element discretization of the steady linearized Navier–Stokes equations, which results in a nonsymmetric operator with block structure, to which we apply various block-preconditioning techniques. The finite-element discretization is constructed using the MFEM finite-element library [2], PETSc is used for the block-preconditioning and linear-algebra interface [7], and *hypre* provides the algebraic multigrid (AMG) solvers for various blocks in the operator [54].

Let $\Omega \subset \mathbb{R}^2$ be a polygonal domain with boundary $\partial\Omega$. We consider the steady linearized Navier–Stokes problem for the velocity field $u : \Omega \to \mathbb{R}^2$ and pressure field $p : \Omega \to \mathbb{R}$, given by

$$-\nu\Delta u + \nabla \cdot (w \otimes u) - \gamma\nabla\nabla \cdot u + \nabla p = f \qquad \text{in } \Omega, \tag{2.20a}$$

$$\nabla \cdot u = 0 \qquad \text{in } \Omega, \tag{2.20b}$$

$$u = g \qquad \text{on } \partial\Omega, \tag{2.20c}$$

where $w : \Omega \to \mathbb{R}^2$ is a given solenoidal velocity field, $\nu \in \mathbb{R}^+$ is the kinematic viscosity, $\gamma \geq 0$ is a constant, $g \in \mathbb{R}^2$ is a given Dirichlet boundary condition, and $f : \Omega \to \mathbb{R}^d$ is a forcing term. The consistent grad-div term $-\gamma\nabla\nabla \cdot u$ is added to Equation (2.20a) to improve convergence of the iterative solver when solving the discrete form of Equation (2.20).

As a test case in this section we set $\gamma = 1000$, $\nu = 10^{-4}$, and $f = -\nu\Delta u + \nabla \cdot (w \otimes u) + \nabla p$ and $g = u$ are derived from the exact solution

$$u = \begin{bmatrix} \sin(3x_1)\sin(3x_2) \\ \cos(3x_1)\cos(3x_2) \end{bmatrix}, \qquad p = (1 - 3x_1)x_2, \qquad \text{in } \Omega = [0, 1]^2,$$

with $w = u$.

We discretize the linearized Navier–Stokes problem Equation (2.20) using the pointwise mass-conserving hybridizable discontinuous Galerkin (HDG) method introduced in [132].

This HDG method approximates both the velocity and pressure separately on element interiors and element boundaries. As such, we make a distinction between interior element degrees-of-freedom (DOFs) and the facet DOFs. We discuss this HDG method next, with required modifications for grad-div term.

Let $\mathcal{T} = \{K\}$ be a tessellation of the domain Ω into non-overlapping simplices K. The boundaries of elements K are denoted by ∂K and the outward unit normal vectors on each ∂K are denoted by n. We set $h = \max_{K\in\mathcal{T}} h_K$, where h_K is the characteristic length of an element K. We also need the notion of facets: an interior facet is defined as $F_I := \partial K^+ \cap \partial K^-$ for neighbouring elements K^+, K^- and a boundary facet is defined as $F_B := \partial K \cap \partial\Omega$. The sets of interior facets F_I and boundary facets F_B are respectively denoted by $\mathcal{F}_I$ and $\mathcal{F}_B$. The set of all facets is $\mathcal{F} = \mathcal{F}_I \cup \mathcal{F}_B$ while Γ^0 denotes the union of all facets.

We introduce the following discontinuous finite element spaces for the velocity and pressure on $\mathcal{T}$ and their restriction to Γ^0:

$$V_h = \left\{ v_h \in \left[L^2(\mathcal{T})\right]^d,\ v_h \in \left[P_k(K)\right]^d,\ \forall K \in \mathcal{T} \right\}, \tag{2.21}$$

$$\bar{V}_h = \left\{ \bar{v}_h \in \left[L^2(\mathcal{F})\right]^d,\ \bar{v}_h \in \left[P_k(F)\right]^d,\ \forall F \in \mathcal{F}, \bar{v}_h = g_D \text{ on } \Gamma_D \right\}, \tag{2.22}$$

$$Q_h = \left\{ q_h \in L^2(\mathcal{T}),\ q_h \in P_{k-1}(K),\ \forall K \in \mathcal{T} \right\}, \tag{2.23}$$

$$\bar{Q}_h = \left\{ \bar{q}_h \in L^2(\mathcal{F}),\ \bar{q}_h \in P_k(F),\ \forall F \in \mathcal{F} \right\}, \tag{2.24}$$

where $P_l(D)$ is the space of polynomials of degree $l > 0$ on a domain D and d is the dimensionality of the problem. Note that V_h and Q_h are defined everywhere on the tessellation $\mathcal{T}$ and that $\bar{V}_h$ and $\bar{Q}_h$ are defined only on facets $F \in \mathcal{F}$. For ease of notation we define also $\boldsymbol{V}_h = V_h \times \bar{V}_h$, $\boldsymbol{Q}_h = Q_h \times \bar{Q}_h$ and $\boldsymbol{X}_h = \boldsymbol{V}_h \times \boldsymbol{Q}_h$, and denote function pairs in $\boldsymbol{V}_h$ and $\boldsymbol{Q}_h$ by boldface, for example, $\boldsymbol{u}_h = (u_h, \bar{u}_h) \in \boldsymbol{V}_h$ and $\boldsymbol{p}_h = (p_h, \bar{p}_h) \in \boldsymbol{Q}_h$. We will also frequently use the space

$$V_h^{\mathrm{div}} := \left\{ v_h \in V_h \cap H(\mathrm{div}, \Omega) \ :\ \nabla \cdot v_h = 0\ \forall x \in K,\ \forall K \in \mathcal{T} \right\}. \tag{2.25}$$

The HDG formulation for the linearized Navier–Stokes problem Equation (2.20) is given by: given $f \in [L^2(\Omega)]^d$, $\nu > 0$ and $w_h \in V_h^{\mathrm{div}}$, find $(\boldsymbol{u}_h, \boldsymbol{p}_h) \in \boldsymbol{X}_h$ such that

$$B(\boldsymbol{u}_h, \boldsymbol{p}_h; \boldsymbol{v}_h, \boldsymbol{q}_h) = \sum_{K\in\mathcal{T}} \int_K f \cdot v_h \,\mathrm{d}x \quad \forall (\boldsymbol{v}_h, \boldsymbol{q}_h) \in \boldsymbol{X}_h, \tag{2.26}$$

where

$$\begin{aligned} B(\boldsymbol{u}_h, \boldsymbol{p}_h; \boldsymbol{v}_h, \boldsymbol{q}_h) =& a_h(\boldsymbol{u}_h, \boldsymbol{v}_h) + o_h(w_h; \boldsymbol{u}_h, \boldsymbol{v}_h) + \gamma d_h(u_h, v_h) \\ &+ b_h(\boldsymbol{p}_h, \boldsymbol{v}_h) - b_h(\boldsymbol{q}_h, \boldsymbol{u}_h), \end{aligned} \tag{2.27}$$

and where

$$a_h(\boldsymbol{u},\boldsymbol{v}) := \sum_{K\in\mathcal{T}}\int_K \nu\nabla u : \nabla v \,\mathrm{d}x + \sum_{K\in\mathcal{T}}\int_{\partial K}\frac{\alpha\nu}{h_K}(u-\bar{u})\cdot(v-\bar{v})\,\mathrm{d}s, \tag{2.28a}$$
$$- \sum_{K\in\mathcal{T}}\int_{\partial K}\left[\nu(u-\bar{u})\cdot\partial_n v + \nu\partial_n u\cdot(v-\bar{v})\right]\mathrm{d}s,$$

$$o_h(w;\boldsymbol{u},\boldsymbol{v}) := -\sum_{K\in\mathcal{T}}\int_K u\otimes w : \nabla v\,\mathrm{d}x + \sum_{K\in\mathcal{T}}\int_{\partial K}\tfrac{1}{2}w\cdot n(u+\bar{u})\cdot(v-\bar{v})\,\mathrm{d}s \tag{2.28b}$$
$$+ \sum_{K\in\mathcal{T}}\int_{\partial K}\tfrac{1}{2}|w\cdot n|\,(u-\bar{u})\cdot(v-\bar{v})\,\mathrm{d}s,$$

$$d_h(u,v) := \sum_{K\in\mathcal{T}}\int_K (\nabla\cdot u)(\nabla\cdot v)\,\mathrm{d}x, \tag{2.28c}$$

$$b_h(\boldsymbol{p},\boldsymbol{v}) := -\sum_{K\in\mathcal{T}}\int_K p\nabla\cdot v\,\mathrm{d}x + \sum_{K\in\mathcal{T}}\int_{\partial K}(v-\bar{v})\cdot n\bar{p}\,\mathrm{d}s. \tag{2.28d}$$

In Equation (2.28a) $\alpha > 0$ is a penalty parameter that needs to be chosen sufficiently large to ensure stability [131, 165].

To obtain the equivalent linear system, let $u \in \mathbb{R}^{n_u}$ be the vector of discrete velocity with respect to the basis for V_h, let $\bar{u} \in \mathbb{R}^{n_{\bar{u}}}$ be the vector of the discrete velocity with respect to the basis for $\bar{V}_h$, and let $\mathbf{u} := [u^T\ \bar{u}^T]^T$. Similarly, let $p \in \left\{q \in \mathbb{R}^{n_p} | q \neq 1\right\}$ be the vector of discrete pressure with respect to the basis for Q_h, let $\bar{p} \in \mathbb{R}^{n_{\bar{p}}}$ be the vector of the discrete pressure with respect to the basis for $\bar{Q}_h$, and let $\mathbf{p} := [p^T\ \bar{p}^T]$.

Let $\mathbf{v} := [v^T\ \bar{v}^T]^T$ be any vector with $v \in \mathbb{R}^{n_u}$ and $\bar{v} \in \mathbb{R}^{\bar{n}_u}$ and let $\mathbf{q} := [q^T\ \bar{q}^T]^T$ be any vector with $q \in \left\{q \in \mathbb{R}^{n_p} | q \neq 1\right\}$ and $\bar{q} \in \mathbb{R}^{\bar{n}_q}$. We then define the matrix $A \in \mathbb{R}^{(n_u+n_{\bar{u}})\times(n_u+n_{\bar{u}})}$ by

$$a_h(\boldsymbol{v}_h,\boldsymbol{v}_h) = \|\mathbf{v}\|_A^2 \quad \text{where} \quad A := \begin{bmatrix} A_{uu} & A_{\bar{u}u}^T \\ A_{\bar{u}u} & A_{\bar{u}\bar{u}} \end{bmatrix}, \tag{2.29}$$

where we used the notation $\|\mathbf{v}\|_A^2 = \langle A\mathbf{v},\mathbf{v}\rangle = \mathbf{v}^T A\mathbf{v}$. Similarly, we define the matrices $N, D, J \in \mathbb{R}^{(n_u+n_{\bar{u}})\times(n_u+n_{\bar{u}})}$ by

$$o_h(w_h;\boldsymbol{v}_h,\boldsymbol{v}_h) = \langle N\mathbf{v},\mathbf{v}\rangle \qquad \text{where} \quad N := \begin{bmatrix} N_{uu} & N_{u\bar{u}} \\ N_{\bar{u}u} & N_{\bar{u}\bar{u}} \end{bmatrix}, \tag{2.30}$$

$$d_h(v_h,v_h) = \langle D\mathbf{v},\mathbf{v}\rangle \qquad \text{where} \quad D := \begin{bmatrix} D_{uu} & 0 \\ 0 & 0 \end{bmatrix}. \tag{2.31}$$

We define the matrix $B \in \mathbb{R}^{(n_q+\bar{n}_q)\times(n_u+\bar{n}_u)}$ as

$$b_h(\boldsymbol{q}_h, \boldsymbol{v}_h) = \langle B\mathbf{v}, \mathbf{q}\rangle \quad \text{where} \quad B := \begin{bmatrix} B_{pu} & 0 \\ B_{\bar{p}u} & 0 \end{bmatrix}, \tag{2.32}$$

and we define the vector $L \in \mathbb{R}^{(n_u+n_{\bar{u}})}$ as

$$\sum_{K\in\mathcal{T}} \int_K v_h \cdot f \,\mathrm{d}x = \mathbf{v}^T L \quad \text{where} \quad L := \begin{bmatrix} L_u \\ 0 \end{bmatrix}. \tag{2.33}$$

Defining $F_{ij} = A_{ij} + N_{ij} + \gamma D_{ij}$ and separating the interior DOFs and facet DOFs, the HDG linear system takes the form

$$\begin{bmatrix} F_{uu} & F_{u\bar{u}} & B_{pu}^T & B_{\bar{p}u}^T \\ F_{\bar{u}u} & F_{\bar{u}\bar{u}} & 0 & 0 \\ B_{pu} & 0 & 0 & 0 \\ B_{\bar{p}u} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ \bar{u} \\ p \\ \bar{p} \end{bmatrix} = \begin{bmatrix} L \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2.34}$$

The HDG method in [132] is such that element DOFs are local; a direct consequence is that F_{uu} is a block diagonal matrix. Using this, we eliminate u from Equation (2.34) and get the statically-condensed system

$$\begin{bmatrix} F_{\bar{u}\bar{u}} - F_{\bar{u}u}F_{uu}^{-1}F_{u\bar{u}} & -F_{\bar{u}u}F_{uu}^{-1}B_{pu}^T & -F_{\bar{u}u}F_{uu}^{-1}B_{\bar{p}u}^T \\ -B_{pu}F_{uu}^{-1}F_{u\bar{u}} & -B_{pu}F_{uu}^{-1}B_{pu}^T & -B_{pu}F_{uu}^{-1}B_{\bar{p}u}^T \\ -B_{\bar{p}u}F_{uu}^{-1}F_{u\bar{u}} & -B_{\bar{p}u}F_{uu}^{-1}B_{pu}^T & -B_{\bar{p}u}F_{uu}^{-1}B_{\bar{p}u}^T \end{bmatrix} \begin{bmatrix} \bar{u} \\ p \\ \bar{p} \end{bmatrix} = \begin{bmatrix} -F_{\bar{u}u}F_{uu}^{-1}L \\ -B_{pu}F_{uu}^{-1}L \\ -B_{\bar{p}u}F_{uu}^{-1}L \end{bmatrix}. \tag{2.35}$$

In this section we verify the theory developed in Section 2.1 and Section 2.2 by solving the statically-condensed block system Equation (2.35). Note that this is a 3×3 block system while the theory developed in this chapter is for a 2×2 block system Equation (1.11)–Equation (1.12). For this reason, we lump together the pressure DOFs p and $\bar{p}$ and write Equation (2.35) in the form Equation (1.11)–Equation (1.12) with

$$\begin{aligned} A_{11} &= F_{\bar{u}\bar{u}} - F_{\bar{u}u}F_{uu}^{-1}F_{u\bar{u}}, & A_{12} &= \begin{bmatrix} -F_{\bar{u}u}F_{uu}^{-1}B_{pu}^T & -F_{\bar{u}u}F_{uu}^{-1}B_{\bar{p}u}^T \end{bmatrix}, \\ A_{21} &= \begin{bmatrix} -B_{pu}F_{uu}^{-1}F_{u\bar{u}} \\ -B_{\bar{p}u}F_{uu}^{-1}F_{u\bar{u}} \end{bmatrix}, & A_{22} &= \begin{bmatrix} -B_{pu}F_{uu}^{-1}B_{pu}^T & -B_{pu}F_{uu}^{-1}B_{\bar{p}u}^T \\ -B_{\bar{p}u}F_{uu}^{-1}B_{pu}^T & -B_{\bar{p}u}F_{uu}^{-1}B_{\bar{p}u}^T \end{bmatrix}, \end{aligned} \tag{2.36}$$

and

$$\mathbf{x} = \begin{bmatrix} \bar{u} \\ P \end{bmatrix}, \quad P = \begin{bmatrix} p \\ \bar{p} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -F_{\bar{u}u}F_{uu}^{-1}L \\ \mathbf{b}_p \end{bmatrix}, \quad \mathbf{b}_p = \begin{bmatrix} -B_{pu}F_{uu}^{-1}L \\ -B_{\bar{p}u}F_{uu}^{-1}L \end{bmatrix}.$$

2.3.1 Block preconditioning

We consider block upper- and lower-triangular (respectively, U_{22} and L_{22}), block-diagonal (D_{22}), approximate block-LDU (M_{22}), and both versions of symmetric block-triangular preconditioners discussed in Section 2.1.1.1. In all cases the Schur complement S_{22} of Equation (1.12) is approximated by

$$\widehat{S}_{22} = \begin{bmatrix} -B_{pu}F_{uu}^{-1}B_{pu}^T & \mathbf{0} \\ \mathbf{0} & -B_{\bar{p}u}F_{uu}^{-1}B_{\bar{p}u}^T \end{bmatrix}. \tag{2.37}$$

In Chapter 4 we show that $\widehat{S}_{22}$ is a good approximation to the corresponding Schur complement S_{22}. Note that the diagonal block on p, $-B_{pu}F_{uu}^{-1}B_{pu}^T$, is block diagonal and can be inverted directly. Furthermore, for large γ the diagonal block on $\bar{p}$, $-B_{\bar{p}u}F_{uu}^{-1}B_{\bar{p}u}^T$, is a Poisson-like operator which can be inverted rapidly using classical AMG techniques. Finally, the momentum block A_{11} in all preconditioners is an approximation to an advection-diffusion equation. To this block we apply the nonsymmetric AMG solver based on approximate ideal restriction (AIR) [109, 107], a recently developed nonsymmetric AMG method that is most effective on advection-dominated problems. Altogether, we have fast, scalable solvers for the diagonal blocks in the different preconditioners.

Theory in this chapter proves that convergence of Krylov methods applied to the block-preconditioned system is governed by an equivalent Krylov method applied to the preconditioned Schur complement. Since $\widehat{S}_{22}$ is a good approximation to the corresponding Schur complement S_{22}, we consider block preconditioners based on the diagonal blocks $\{A_{11}, \widehat{S}_{22}\}$:

1. An (approximate) inverse of the momentum block A_{11} using AIR and a block-diagonal inverse of the pressure block for $\widehat{S}_{22}$.
2. An (approximate) inverse of the momentum block A_{11} using AIR and a negative block-diagonal inverse of the pressure block for $\widehat{S}_{22}$.

The diagonal inverses computed in the pressure block $\widehat{S}_{22}$ are solved to a small tolerance. The sign is swapped on the pressure block, a technique often used with symmetric systems to maintain an SPD preconditioner, to study the effect of sign of $\widehat{S}_{22}^{-1}$ on convergence.

Although theory developed here is based on an exact inverse of the momentum block, we present results ranging from an exact inverse to a fairly crude inverse, with a reduction in relative residual of only 0.1 per iteration, and demonstrate that theoretical results extend well to the case of inexact inverses in practice. Although AIR has proven an effective solver

for advection-dominated problems, solving the momentum block can still be challenging. For this reason, a relative-residual tolerance for the momentum block is used (as opposed to doing a fixed number of iterations of AIR) as it is not clear a priori how many iterations would be appropriate. Since this results in a preconditioner that is different each iteration, we use FGMRES acceleration (which uses right preconditioning by definition) [137]. This is used as a practical choice, and we demonstrate that the performance of FGMRES is also consistent with theory.

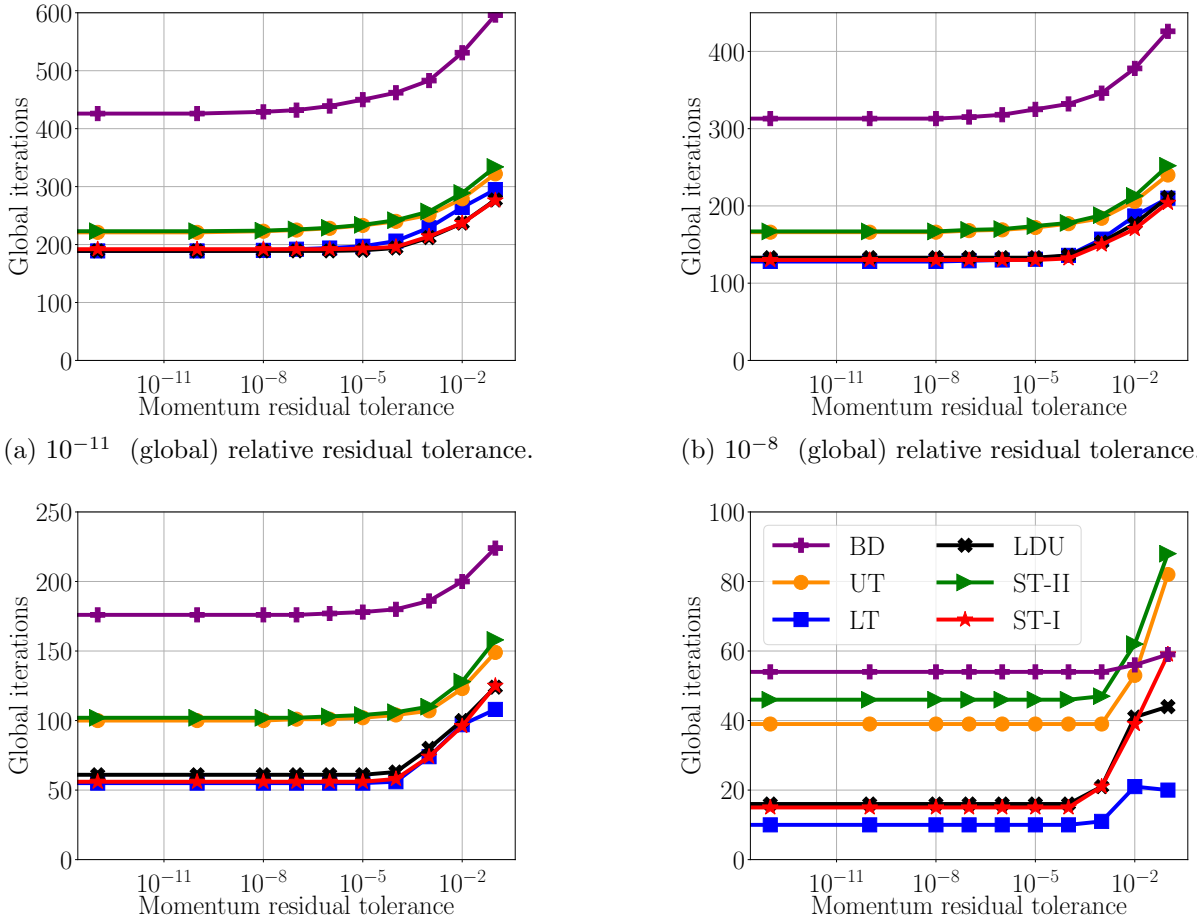
2.3.2 Results

Figure 2.1 shows iterations to various global relative-residual tolerances as a function of relative-residual tolerance of the momentum block for block upper- and lower-triangular, block-diagonal, approximate block-LDU, and both versions of symmetric block triangular preconditioners. In general, theory derived in this chapter based on the assumption of an exact inverse of one diagonal block extends well to the inexact setting. Further points to take away from Figure 2.1 are:

1. For four different relative-residual tolerances of the 2×2 block system, block-diagonal preconditioning takes very close to twice as many iterations as block-triangular preconditioning. For larger tolerances such as 10^{-3}, it is approximately twice the average number of iterations of block upper- and block lower-triangular preconditioning, which is consistent with the derivations and constants in Theorem 2.2.5. Moreover, this relationship holds for almost all tolerances of the momentum block solve, with the exception of considering both large momentum tolerances ($> 10^{-3}$) and large global tolerances (see Figure 2.1d).

2. At no point does a symmetric block-triangular or approximate block-LDU preconditioner offer improved convergence over a block-triangular preconditioner, regardless of momentum or 2×2 system residual tolerance, although the solve times are significantly longer due to the additional applications of the diagonal blocks of the preconditioner. In fact, for a global tolerance of 10^{-3} symmetric block-triangular preconditioning is actually less effective than just block triangular.

3. The block lower-triangular preconditioner is more effective than the block-upper-triangular preconditioner. However, they differ in iteration count by roughly the same 30–40 iterations for all four tolerances tested, indicating it is not a difference in convergence rate (which the theory says it should not be), but rather a difference



(a) 10^{-11} (global) relative residual tolerance.

(b) 10^{-8} (global) relative residual tolerance.

(c) 10^{-5} (global) relative residual tolerance.

(d) 10^{-3} (global) relative residual tolerance.

Figure 2.1: Number of iterations for the 2×2 block preconditioned system to converge to 10^{-11}, 10^{-8}, 10^{-5}, and 10^{-3} relative residual tolerance, as a function of the relative residual tolerance to solve the momentum block. Results are shown for block lower-triangular (LT), block upper-triangular (UT), symmetric lower-then-upper block-triangular (ST-I), symmetric upper-then-lower block-triangular (ST-II), block-diagonal (BD), and approximate block-LDU (LDU) preconditioners.

in the leading constants. Interestingly, it cannot be explained by the norm of off-diagonal blocks (which are similar for upper- and lower-triangular preconditioning in this case). We hypothesize it is due to the initial residual, where for $\mathbf{r}^{(0)} = [\mathbf{r}_1^{(0)}, \mathbf{r}_2^{(0)}]$, we have $\|\mathbf{r}_1^{(0)}\| = 19.14$ and $\|\mathbf{r}_2^{(0)}\| = 0.0063$. Heuristically, it seems more effective in

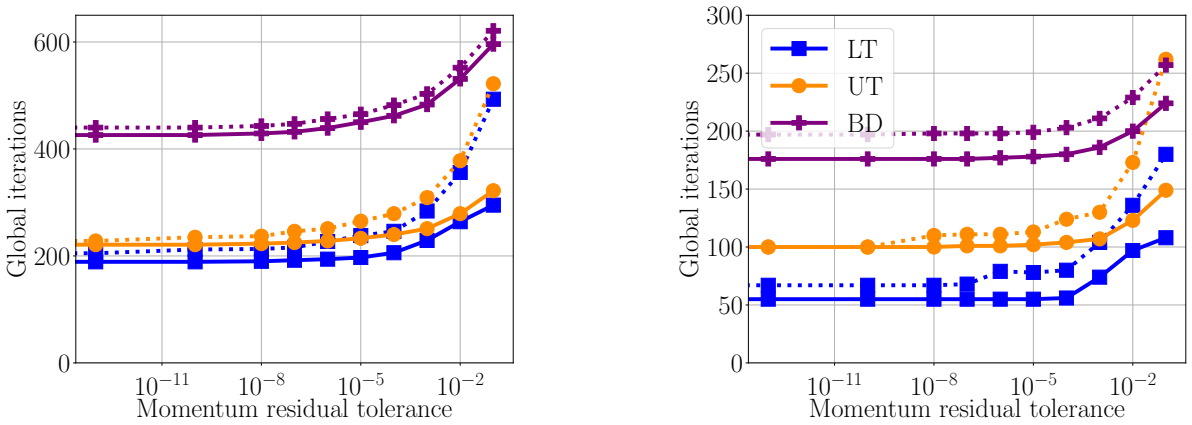
terms of convergence to solve directly on the block with the large initial residual (in this case the (1,1)-block) and lag the variable with a small initial residual (in this case the (2,2)-block), which would correspond to block-lower triangular preconditioning. However, a better understanding of upper vs. lower block-triangular preconditioning is ongoing work.

A common approach for saddle-point problems that are self-adjoint in a given inner product is to use an SPD preconditioner so that three-term recursion formulae, in particular preconditioned MINRES, can be used. For matrices with saddle-point structure, the Schur complement is often negative definite, so this is achieved by preconditioning S_{22} with some approximation $-\widehat{S}_{22}^{-1}$. Although this is advantageous in terms of being able to use MINRES, convergence can suffer compared with GMRES and an indefinite preconditioner.

Results of this chapter indicate a direct correlation between the minimizing polynomial for the 2×2 system and the preconditioned Schur complement. Moreover, convergence on the preconditioned Schur complement should be independent of sign, because Krylov methods minimize over a Krylov space that is invariant to the sign of M^{-1}. Together, this indicates that if the (1,1)-block is inverted exactly, convergence of GMRES applied to the 2×2 preconditioned system should be approximately equivalent, regardless of sign of the Schur-complement preconditioner.

Figure 2.2 demonstrates this property, considering FGMRES iterations on the 2×2 system to relative-residual tolerances of 10^{-11} and 10^{-5}, as a function of momentum relative-residual tolerance. Results are shown for block-diagonal, block lower-triangular, and block-upper-triangular preconditioners, with a natural sign $\widehat{S}_{22}^{-1}$ (solid lines) and swapped sign $-\widehat{S}_{22}^{-1}$ (dotted lines). For accurate solves of the momentum block, we see relatively tight convergence behaviour between $\pm\widehat{S}_{22}^{-1}$. As the momentum solve tolerance is relaxed, convergence of block-triangular preconditioners decay for $-\widehat{S}_{22}^{-1}$. Interestingly, the same phenomenon does not appear to happen for block-diagonal preconditioners, and rather there is a fixed difference in iteration count between $\pm\widehat{S}_{22}^{-1}$. This is likely because a block-diagonal preconditioner does not directly couple the variables of the 2×2 matrix, while the block-triangular preconditioner does. An inexact inverse loses a nice cancellation property of the exact inverse, and the triangular coupling introduces terms along the lines of $I \pm \widehat{S}_{22}^{-1}A_{22}$ (see Equation (2.4)), which clearly depend on the sign of $\widehat{S}_{22}^{-1}$.

In [59] it is proven that minimal residual methods applied to saddle-point problems with a zero (2,2)-block and preconditioned with a block-diagonal preconditioner observe a staircasing effect, where every alternate iteration stalls. This results in approximately twice as many iterations to convergence as a similar block-triangular preconditioner. Although

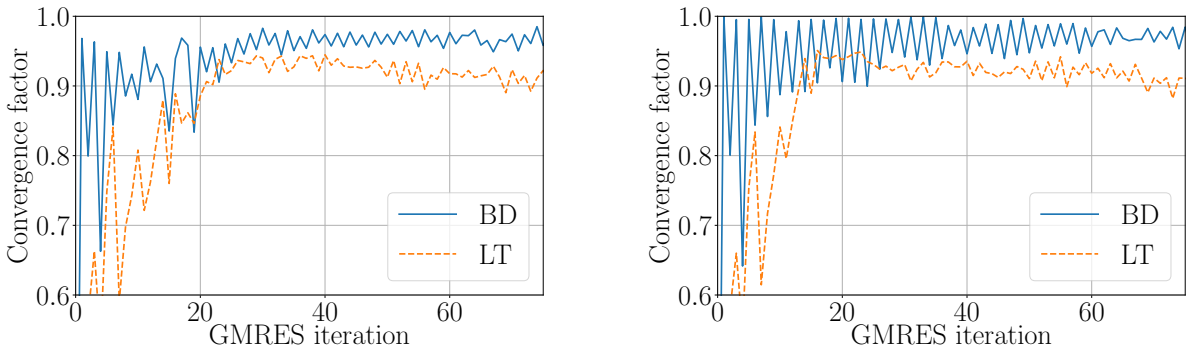


(a) 10^{-11} relative residual tolerance.

(b) 10^{-5} relative residual tolerance.

Figure 2.2: Number of iterations for the 2×2 block preconditioned system to converge to 10^{-11} and 10^{-5} relative residual tolerance, as a function of the relative residual tolerance to solve the momentum block. Results are shown for block lower-triangular (LT), block upper-triangular (UT), and block-diagonal (BD) preconditioners, as in Figure 2.1 (solid lines) and with the sign swapped on the pressure Schur-complement approximation (dotted lines).

the proof appeals to specific starting vectors, the effect is demonstrated in practice as well. Theorem 2.2.5 proved block-diagonal preconditioning is expected to take twice as many iterations as block-triangular preconditioning to reach a given tolerance (within some constant multiplier). Figure 2.3 looks at the GMRES convergence factor as a function of iteration for block-diagonal preconditioning and block lower-triangular preconditioning, with $\pm\widehat{S}_{22}^{-1}$. Interestingly, with $-\widehat{S}_{22}^{-1}$ (see Figure 2.3b), the staircasing effect is clear, where every alternate iteration makes little to no reduction in residual. Although convergence has some sawtooth character for block-diagonal with $\widehat{S}_{22}^{-1}$ as well, it is much weaker, and the staircasing effect is not truly observed. It is possible this explains the slightly better convergence obtained with $\widehat{S}_{22}^{-1}$ in Figure 2.2b, regardless of momentum relative-residual tolerance.



(a) Preconditioning using $\widehat{S}_{22}^{-1}$.

(b) Preconditioning using $-\widehat{S}_{22}^{-1}$.

Figure 2.3: Convergence factor as a function of FGMRES iteration for block-diagonal (BD) and block lower-triangular (LT) preconditioning. Figure 2.3a uses the natural sign on $\widehat{S}_{22}^{-1}$, while Figure 2.3b adds a negative to $\widehat{S}_{22}^{-1}$.

2.4 Conclusions

This chapter analyzes the relationship between Krylov methods with 2×2 block preconditioners and the underlying preconditioned Schur complement. Under the assumption that one of the diagonal blocks is inverted exactly, we prove a direct relationship between the minimizing Krylov polynomial of a given degree for the two systems, thereby proving their equivalence and the fact that an effective Schur complement preconditioner is a necessary and sufficient condition for an effective 2×2 block preconditioner. Theoretical results give further insight into choice of block preconditioner, including that (i) symmetric block-triangular and approximate block-LDU preconditioners offer a minimal reduction in iteration count over block-triangular preconditioners, at the expense of additional computational cost, and (ii) block-diagonal preconditioners take about twice as many iterations to reach a given residual tolerance as block-triangular preconditioners.

Numerical results on an HDG discretization of the steady linearized Navier–Stokes equations confirm the theoretical contributions, and show that the practical implications extend to the case of a Schur-complement approximation coupled with an inexact inverse of the other diagonal block. Although not shown here, it is worth pointing out we have observed similar results with inexact block preconditioners in other applications. For HDG discretizations of symmetric Stokes and Darcy problems, if the pressure Schur complement is approximated by a spectrally equivalent operator, applying two to four multigrid cycles to the momentum block yields comparable convergence on the larger 2×2 system as applying

a direct solve on the momentum block. Classical source iteration and DSA preconditioning for S_N discretizations of neutron transport can also be posed as a 2×2 block preconditioning [148]. There we have also observed that when applying AMG iterations to the (1,1)-block and Schur complement approximation, only 2-3 digits of residual reduction yields convergence on the larger 2×2 system as fast as applying direct solves to each block. In each of these cases, convergence of minimal residual methods applied to the 2×2 system is defined by the preconditioning of the Schur complement.

Chapter 3

Time-Dependent Advection-Diffusion Problems

In Chapter 2, we discussed block-preconditioning for the 2×2-block system Equations (1.11) and (1.12) under the assumption that at least one of the blocks can be inverted exactly, or very accurately (see Figures 2.1 and 2.2). The long term goal of our research is to solve the case where Equations (1.11) and (1.12) represent an HDG discretization of the time-dependent Navier-Stokes equations:

$$\begin{aligned} \vec{u}_t - \nu\nabla^2\vec{u} + (\vec{u}\cdot\nabla)\vec{u} + \nabla p = \vec{f} \quad & \text{in } \Omega(t), \\ \nabla\cdot\vec{u} = 0 \quad & \text{in } \Omega(t), \end{aligned}$$

where $\Omega(t)$ denotes that the domain of the problem may be time-dependent. The traditional approach to discretize in time is to use a classical (implicit or explicit) time stepping strategy, e.g., Euler methods, backwards differentiation formulas, and Runge-Kutta methods, in combination with the HDG discretization of spatial components. More recently, however, the paradigm is shifting towards employing space-time discretizations where time-dependent problems are cast onto a space-time domain, and time variables are treated in the same way as spatial variables. Hence, we use an HDG discretization in both space and time. Advantages of space-time methods over traditional time-stepping techniques include the possibility of local time stepping, adaptive space-time mesh refinement, and natural handling of problems on time-dependent domains.

As discussed previously in Chapter 1, we use Picard iterations to deal with the nonlinearity of the Navier–Stokes equations. This results in having to solve the time-dependent Oseen equations at each Picard iteration. The space-time HDG discretization of the Oseen

equations results in a linear system of the form Equations (1.11) and (1.12). However, to apply the results from Chapter 2, we need a good solver for the diagonal block corresponding to the time-dependent vector advection-diffusion equation,

$$\vec{u}_t - \nu\nabla^2\vec{u} + (\vec{w}\cdot\nabla)\vec{u} = \begin{bmatrix} u_{1,t} - \nu\nabla^2 u_1 + (\vec{w}\cdot\nabla)u_1 \\ u_{2,t} - \nu\nabla^2 u_2 + (\vec{w}\cdot\nabla)u_2 \end{bmatrix}, \tag{3.1}$$

which, in turn, is the concatenation of two time-dependent scalar advection-diffusion equations. Hence, the problem of finding a good preconditioner for the velocity block of space-time HDG discretizations of the time-dependent Navier–Stokes equations can be reduced to finding a good preconditioner for space-time HDG discretizations of the time-dependent advection-diffusion equation. This is the topic of this chapter.

Chapter 3 is organised as follows. In Section 3.1, we present the space-time HDG discretization of the advection-diffusion equation, and approximate ideal restriction (AIR) algebraic multigrid [107, 109] is presented in Section 3.2. A discussion on why AIR can be effective as a space-time solver of advection-dominated problems, while most PinT methods struggle, is provided in Section 3.2.1. Numerical results in Section 3.3 indeed demonstrate that AIR is a robust and scalable solver for space-time HDG discretizations of the advection-diffusion equation. Scalable preconditioning is demonstrated with space-time adaptive mesh refinement (AMR) and on time-dependent domains, and speedups over sequential time stepping are obtained on very small processor counts. We draw conclusions in Section 3.4.

This chapter has recently been submitted for publication, and a preprint is available on arXiv [146].

3.1 The space-time HDG method for the advection-diffusion equation

3.1.1 The advection-diffusion problem on time-dependent domains

Let $\Omega_h(t) \subset \mathbb{R}^d$, an approximation to the domain $\Omega(t)$ in Equation (1.13), be a polygonal ($d = 2$) or polyhedral ($d = 3$) domain whose evolution depends continuously on time $t \in [t^0, t^N]$. We will present numerical results only for the case $d = 2$, but remark that the space-time HDG discretization and solution procedure also hold for $d = 3$. We partition

the boundary of $\Omega_h(t)$, $\partial\Omega_h(t)$, into two sets $\Gamma_D(t)$ (the Dirichlet boundary) and $\Gamma_N(t)$ (the Neumann boundary) such that $\partial\Omega_h(t) = \Gamma_D(t) \cup \Gamma_N(t)$ and $\Gamma_D(t) \cap \Gamma_N(t) = \emptyset$.

As discussed in Section 1.3, a point in space-time at time $t = x_0$ with position x has Cartesian coordinates $\hat{x} = (x_0, x)$. Throughout this chapter, we will use t and x_0 interchangeably. We introduce the $(d+1)$-dimensional computational space-time domain $\mathcal{E}_h := \{\hat{x} : x \in \Omega_h(x_0),\ t^0 < x_0 < t^N\} \subset \mathbb{R}^{d+1}$. The boundary of $\mathcal{E}_h$ is comprised of the hyper-surfaces $\Omega_h(t^0) := \{\hat{x} \in \partial\mathcal{E}_h : x_0 = t^0\}$, $\Omega_h(t^N) := \{\hat{x} \in \partial\mathcal{E}_h : x_0 = t^N\}$, and $\mathcal{Q}_{\mathcal{E}_h} := \{\hat{x} \in \partial\mathcal{E}_h : t^0 < x_0 < t^N\}$. We also introduce the partitioning $\partial\mathcal{E}_h = \partial\mathcal{E}_D \cup \partial\mathcal{E}_N$ where $\partial\mathcal{E}_D := \{\hat{x} : x \in \Gamma_D(x_0),\ t^0 < x_0 < t^N\}$ and $\partial\mathcal{E}_N := \{\hat{x} : x \in \Gamma_N(x_0) \cup \Omega(t^0),\ t^0 < x_0 \leq t^N\}$. The outward unit space-time normal vector to $\partial\mathcal{E}_h$ is denoted by $\widehat{n} = (n_t, n)$, where $n_t \in \mathbb{R}$ is the temporal part of the space-time vector and $n \in \mathbb{R}^d$ the spatial part.

Given the viscosity $\nu \geq 0$, forcing term $f : \mathcal{E}_h \to \mathbb{R}$, and advective velocity $a : \mathcal{E}_h \to \mathbb{R}^d$, the advection-diffusion equation for the scalar $u : \mathcal{E}_h \to \mathbb{R}$ is given by

$$\partial_t u + a \cdot \nabla u - \nu\nabla^2 u = f \qquad \text{in } \mathcal{E}_h, \tag{3.2a}$$
$$-\zeta u(n_t + a \cdot n) + \nu\nabla u \cdot n = g_N \qquad \text{on } \partial\mathcal{E}_N, \tag{3.2b}$$
$$u = g_D \qquad \text{on } \partial\mathcal{E}_D, \tag{3.2c}$$

where $g_N : \mathcal{Q}_N \to \mathbb{R}$ is a suitably smooth function and ζ is an indicator function for the inflow boundary of $\mathcal{E}$, i.e., where $(n_t + a \cdot n) < 0$. Note that the initial condition $u(0, x) = g_N(0, x)$ is imposed by Equation (3.2b). Using the definition of the space-time advective velocity and the space-time gradient introduced in Section 1.3, the space-time formulation of Equation (3.2) is given by

$$\widehat{a} \cdot \widehat{\nabla} u - \nu\nabla^2 u = f \qquad \text{in } \mathcal{E}_h, \tag{3.3a}$$
$$-\zeta u\widehat{a}_n + \nu\nabla u \cdot n = g_N \qquad \text{on } \partial\mathcal{E}_N, \tag{3.3b}$$
$$u = g_D \qquad \text{on } \partial\mathcal{E}_D, \tag{3.3c}$$

where $\widehat{a}_n = \widehat{n} \cdot \widehat{a} = n_t + a \cdot n$. We see that the time-dependent advection-diffusion problem Equation (3.2) is a steady state problem in $(d+1)$-dimensional space-time.

3.1.2 Space-time meshes

The two approaches to meshing a space-time domain $\mathcal{E}_h$ are the slab-by-slab approach and the all-at-once approach. In the *slab-by-slab* approach, the time interval $[t^0, t^N]$ is

partitioned into time levels $t^0 < t^1 < \cdots < t^N$. The n-th time interval is defined as $I^n = (t^n, t^{n+1})$ and its length is the "time-step", denoted by $\Delta t^n = t^{n+1} - t^n$. The space-time domain $\mathcal{E}_h$ is then divided into space-time slabs $\mathcal{E}_h^n = \mathcal{E}_h \cap (I^n \times \mathbb{R}^d)$. Note that each space-time slab $\mathcal{E}_h^n$ is bounded by $\Omega_h(t^n)$, $\Omega_h(t^{n+1})$, and $\mathcal{Q}_{\mathcal{E}_h}^n = \partial \mathcal{E}_h^n \backslash (\Omega_h(t^n) \cup \Omega_h(t^{n+1}))$. A space-time triangulation $\mathcal{T}_h^n$ is then introduced for each space-time slab $\mathcal{E}_h^n$ using standard spatial meshing techniques. In this chapter, we use space-time simplices (see, e.g. [81, 82, 162]) as opposed to space-time hexahedra (see e.g. [3, 159, 160]).

In the *all-at-once* approach, a space-time triangulation $\mathcal{T}_h := \cup_j \mathcal{K}_j$ of the full space-time domain $\mathcal{E}_h$ is introduced. This triangulation consists of non-overlapping space-time simplices $\mathcal{K} \subset \mathbb{R}^{d+1}$. There are no clear time levels except for the time level at $x_0 = t^0$ and $x_0 = t^N$ and the space-time mesh may be fully unstructured. In particular, this naturally allows for arbitrary adaptive mesh refinement (AMR) in space and time. Note, we do not consider hanging nodes in this chapter although hanging nodes in space and time are possible within the space-time framework.

In Figure 3.1 we plot space-time elements in a slab-by-slab approach and in an all-at-once approach in $(1+1)$-dimensional space-time.

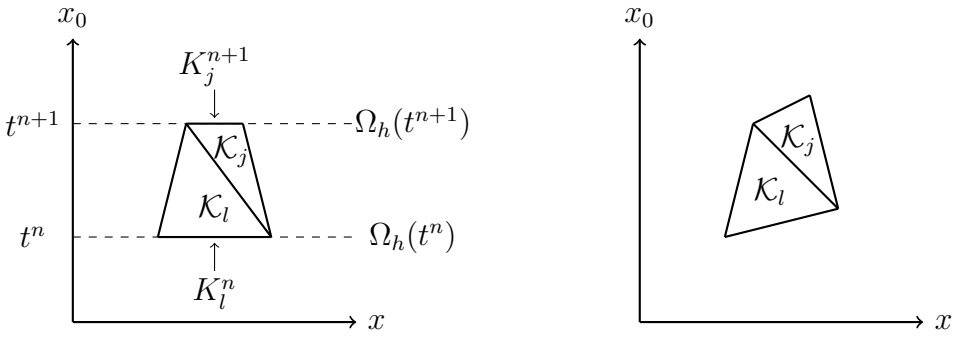


Figure 3.1: Examples of two neighboring elements in $(1+1)$-dimensional space-time. Left: An example of space-time elements in a slab-by-slab approach. The space-time mesh is layered by space-time slabs. Here the elements lie in space-time slab $\mathcal{E}_h^n$. Right: An example of space-time elements in an all-at-once approach. There are no clear time levels for $t^0 < x_0 < t^N$.

3.1.3 The space-time HDG method

Consider a space-time element $\mathcal{K} \in \mathcal{T}_h$ in an all-at-once or slab-by-slab mesh. On the boundary of a space-time element $\partial\mathcal{K}$ we will denote the outward unit space-time normal

vector by $\widehat{n}^{\mathcal{K}} = (n_t^{\mathcal{K}}, n^{\mathcal{K}})$. Two adjacent space-time elements $\mathcal{K}^+$ and $\mathcal{K}^-$ share an interior space-time facet $\mathcal{S} := \partial\mathcal{K}^+ \cap \partial\mathcal{K}^-$. A facet of $\partial\mathcal{K}$ that lies on the space-time boundary $\partial\mathcal{E}_h$ is called a boundary facet. The set of all facets is denoted by $\mathcal{F}$ and the union of all facets by Γ_0. For ease of notation, we will drop the subscripts and superscripts when referring to space-time elements, their boundaries, and outward unit normal vectors in the remainder of this chapter.

We require the following finite element spaces:

$$\begin{aligned} V_h &:= \{v_h \in L^2(\mathcal{E}_h) \,:\, v_h|_{\mathcal{K}} \in P_p(\mathcal{K}),\ \forall \mathcal{K} \in \mathcal{T}_h\}, \\ M_h &:= \{\mu_h \in L^2(\mathcal{F}) \,:\, \mu_h|_{\mathcal{S}} \in P_p(\mathcal{S}),\ \forall \mathcal{S} \in \mathcal{F},\ \mu_h = 0 \text{ on } \partial\mathcal{E}_D\}, \end{aligned}$$

where $P_p(D)$ is the set of polynomials of degree p on a domain D. We furthermore introduce $V_h^\star := V_h \times M_h$. The space-time HDG method for Equation (3.3) is given by [90]: find $(u_h, \lambda_h) \in V_h^\star$ such that

$$\mathcal{B}_h\left((u_h, \lambda_h), (v_h, \mu_h)\right) = \sum_{\mathcal{K}\in\mathcal{T}_h} \int_{\mathcal{K}} f v_h \,\mathrm{d}\hat{x} + \int_{\partial\mathcal{E}_N} g\mu_h \,\mathrm{d}s \quad \forall (v_h, \mu_h) \in V_h^\star, \tag{3.4}$$

where the bilinear form is defined as

$$\begin{aligned} &\mathcal{B}_h\big((u,\lambda),(v,\mu)\big) \\ &\quad := \sum_{\mathcal{K}\in\mathcal{T}_h} \int_{\mathcal{K}} \left(-u\widehat{a}\cdot\widehat{\nabla} v + \nu\nabla u\cdot\nabla v\right) \mathrm{d}\hat{x} + \int_{\partial\mathcal{E}_N} \tfrac{1}{2}\left(\widehat{a}_n + |\widehat{a}_n|\right)\lambda\mu \,\mathrm{d}s \\ &\qquad + \sum_{\mathcal{K}\in\mathcal{T}_h} \int_{\partial\mathcal{K}} \sigma(u,\lambda,\widehat{n})(v-\mu)\,\mathrm{d}s - \sum_{\mathcal{K}\in\mathcal{T}_h} \int_{\partial\mathcal{K}} \nu(u-\lambda)\nabla v\cdot n \,\mathrm{d}s. \end{aligned} \tag{3.5}$$

Here $\sigma(u,\lambda,\widehat{n}) := \sigma_a(u,\lambda,\widehat{n}) + \sigma_d(u,\lambda,n)$ is the "numerical flux" on the cell facets. The advective part of the numerical flux is an upwind flux in both space and time, given by

$$\sigma_a(u,\lambda,\widehat{n}) := \tfrac{1}{2}\left(\widehat{a}_n(u+\lambda) + |\widehat{a}_n|(u-\lambda)\right).$$

The diffusive part of the numerical flux is similar to that of an interior penalty method and is given by

$$\sigma_d(u,\lambda,n) := -\nu\nabla u\cdot n + \frac{\nu\alpha}{h_{\mathcal{K}}}(u-\lambda), \tag{3.6}$$

with $h_{\mathcal{K}}$ the length measure of the element $\mathcal{K}$, and $\alpha > 0$ a penalty parameter. It is shown in [90] that α needs to be sufficiently large to ensure stability of the space-time HDG method.

3.1.4 Sequential time-stepping using the slab-by-slab discretization

The space-time HDG method Equation (3.4) is the same for both the slab-by-slab and all-at-once space-time approaches. However, for the slab-by-slab approach we may write Equation (3.4) in a form similar to traditional time-integration techniques. For this we require the following finite element spaces:

$$V_h^n := \{v_h \in L^2(\mathcal{E}_h^n) \,:\, v_h|_{\mathcal{K}} \in P_p(\mathcal{K}),\ \forall \mathcal{K} \in \mathcal{T}_h^n\},$$
$$M_h^n := \{\mu_h \in L^2(\mathcal{F}^n) \,:\, \mu_h|_{\mathcal{S}} \in P_p(\mathcal{S}),\ \forall \mathcal{S} \in \mathcal{F}^n,\ \mu_h = 0 \text{ on } \partial\mathcal{E}_D^n\},$$

where $\mathcal{F}^n$ is the set of all facets in the slab $\mathcal{E}_h^n$. We furthermore define $V_h^{n,\star} := V_h^n \times M_h^n$. For the slab-by-slab approach, we may write the space-time HDG method for Equation (3.2) as: for each space-time slab $\mathcal{E}_h^n$, $n = 0, 1, \cdots, N-1$, find $(u_h, \lambda_h) \in V_h^{n,\star}$ such that

$$\mathcal{B}_h^n\left((u_h, \lambda_h), (v_h, \mu_h)\right) = \sum_{\mathcal{K} \in \mathcal{T}_h^n} \int_{\mathcal{K}} f v_h \,\mathrm{d}\hat{x} + \int_{\partial\mathcal{E}_N^n} g\mu_h \,\mathrm{d}s, \tag{3.7}$$

for all $(v_h, \mu_h) \in V_h^{n,\star}$, where $\mathcal{B}_h^n(\cdot,\cdot)$ is defined as Equation (3.5) but with $\mathcal{T}_h$ and $\partial\mathcal{E}_N$ replaced by, respectively, $\mathcal{T}_h^n$ and $\partial\mathcal{E}_N^n$. The slab-by-slab approach is similar to traditional time-integration techniques in that the local systems are solved one space-time slab after another. The linear systems arising from space-time finite elements resemble those that arise from fully implicit Runge–Kutta methods (e.g., see [105, 142]).

Well-posedness and convergence of the slab-by-slab space-time HDG method Equation (3.7) was proven in [90]. Furthermore, motivated by the fact that the spatial mesh size h_K and the time-step Δt may be different, an a priori error analysis was presented in [90], resulting in optimal error bounds that are anisotropic in h_K (a measure of the mesh size in spatial direction) and Δt. It is shown, however, that Δt and h_K need to be refined simultaneously to obtain these optimal error bounds, and that refining only in time or only in space may lead to divergence of the error. To this end, all-at-once solvers seem like the natural solution for efficient parallel simulations, where simultaneous local adaptivity in space and time is easily handled.

3.1.5 The discretization

Let $U \in \mathbb{R}^r$ be the vector of expansion coefficients of u_h with respect to the basis for V_h and let $\Lambda \in \mathbb{R}^q$ be the vector of expansion coefficients of λ_h with respect to the basis for

M_h. The space-time HDG method Equation (3.4) can then be expressed as the all-at-once system of linear equations

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} U \\ \Lambda \end{bmatrix} = \begin{bmatrix} F \\ G \end{bmatrix}, \tag{3.8}$$

where A, B, C, and D are matrices obtained from the discretization of $\mathcal{B}_h((\cdot,0),(\cdot,0))$, $\mathcal{B}_h((0,\cdot),(\cdot,0))$, $\mathcal{B}_h((\cdot,0),(0,\cdot))$, and $\mathcal{B}_h((0,\cdot),(0,\cdot))$, respectively.

For the slab-by-slab approach, the linear system Equation (3.8) can be decoupled into smaller linear systems that are solved in each time slab $\mathcal{E}_h^n$. In this case, $U \in \mathbb{R}^r$ is the vector of expansion coefficients of u_h with respect to the basis for V_h^n and $\Lambda \in \mathbb{R}^q$ is the vector of expansion coefficients of λ_h with respect to the basis for M_h^n. Furthermore, A, B, C, and D are then the matrices obtained from the discretization of $\mathcal{B}_h^n((\cdot,0),(\cdot,0))$, $\mathcal{B}_h^n((0,\cdot),(\cdot,0))$, $\mathcal{B}_h^n((\cdot,0),(0,\cdot))$, and $\mathcal{B}_h^n((0,\cdot),(0,\cdot))$, respectively.

The space-time HDG discretization is such that A is a block-diagonal matrix. Using $U = A^{-1}(F - B\Lambda)$ we eliminate U from Equation (3.8) resulting in the following reduced system for Λ:

$$S\Lambda = H, \tag{3.9}$$

where $S = D - CA^{-1}B$ is the Schur complement of the block matrix in Equation (3.8), and $H = G - CA^{-1}F$. Having eliminated the element degrees-of-freedom via static condensation, the linear system Equation (3.9) is significantly smaller than Equation (3.8). However, for the space-time HDG method to be efficient, we still require a fast solver for the reduced non-symmetric problem Equation (3.9), which is discussed in the following section.

3.2 Approximate ideal restriction (AIR) AMG

AMG is traditionally designed for elliptic problems in space or sequential time stepping of parabolic problems, where the resulting linear systems are (nearly) symmetric positive definite or M-matrices. However, a number of papers in recent years have considered extensions of AMG to the nonsymmetric setting, e.g., [115, 168, 23, 140, 108]. In particular, a new AMG method based on a local approximate ideal restriction (ℓAIR; moving forward we simply refer to it as AIR) was developed in [107, 109] specifically for advection-dominated problems and upwinded discretizations. Noting that Equation (3.3) is a "steady" advection-dominated problem in $(d+1)$-dimensional space-time, and that AIR is a robust solver for advection dominated problems, motivates the use of AIR as a preconditioner for the space-time linear system Equation (3.9).

We discuss AIR in Appendix A in more detail for the interested reader. As a brief review, recall that multigrid methods solve $A\mathbf{x} = \mathbf{b}$ by applying a coarse-grid correction based on interpolation and restriction operators, $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + P(RAP)^{-1}R\mathbf{r}^{(i)}$, for matrix A, interpolation P, restriction R, and residual $\mathbf{r}^{(i)} = \mathbf{b} - A\mathbf{x}^{(i)}$. Classical AMG is based on a partitioning of DOFs into fine (F-) and coarse (C-) points, where A can then be expressed in block form as

$$A = \begin{bmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{bmatrix}.$$

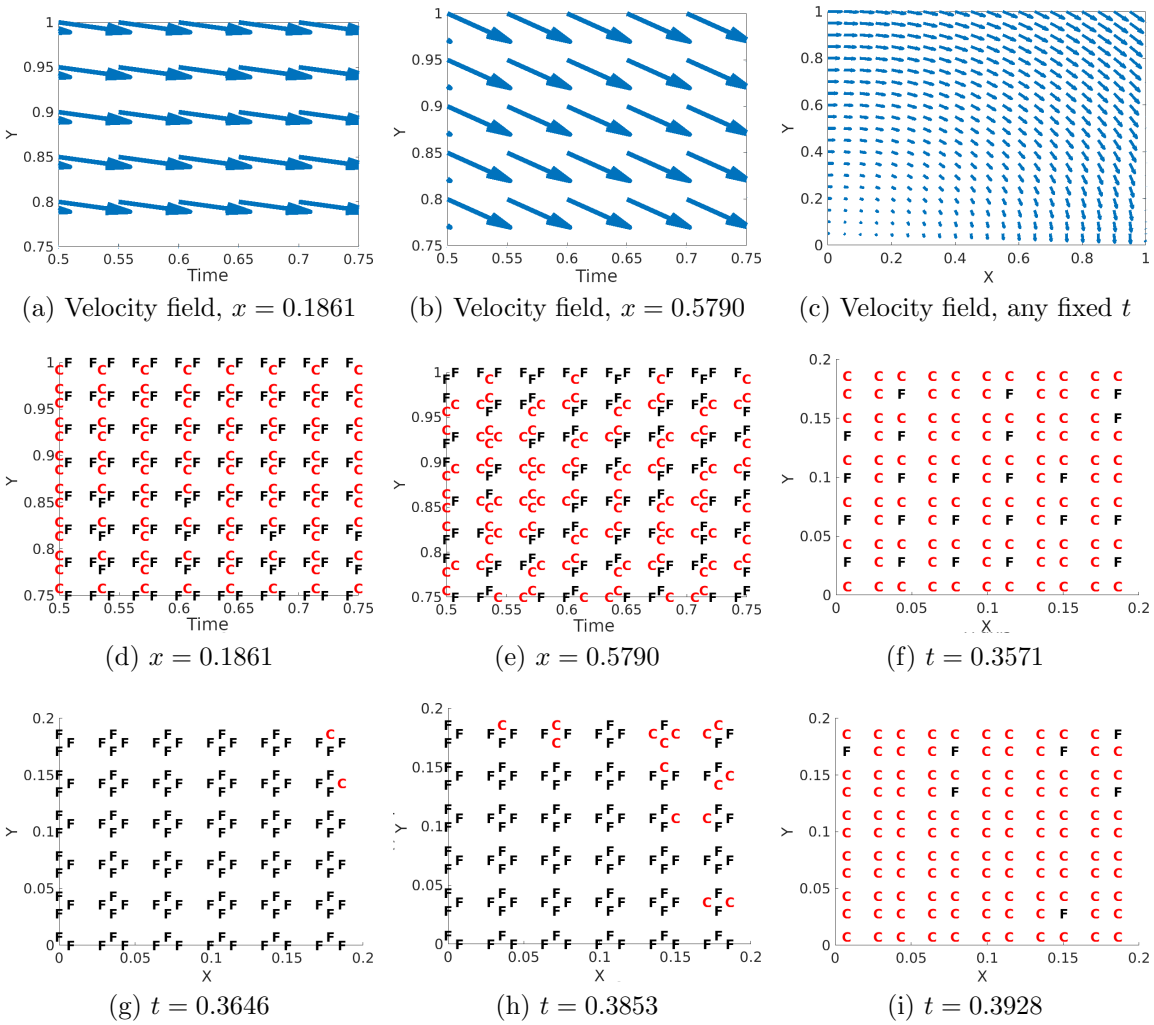
AIR is a reduction method based on the principle that if we use the so-called ideal restriction operator, $R_{\text{ideal}} = [-A_{cf}A_{ff}^{-1} \quad I]$ with any interpolation (in MATLAB notation) $P = [W; I]$, coarse-grid correction eliminates all errors at C-points; following this with an effective relaxation on F-points will guarantee a rapidly convergent method [107, Section 2.3]. Due to the A_{ff}^{-1} term in R_{ideal}, it is not practical to form R_{ideal} explicitly. However, AIR appeals to the observation that for upwinded advective discretizations, one can achieve cheap, accurate, and sparse approximations $R \approx R_{\text{ideal}}$.

3.2.1 Coarsening in space-time

For problems with strong anisotropy or advective components, it is often helpful or even necessary to semi-coarsen along the direction of anisotropy/advection for an effective multigrid method (e.g., [166]). On a high-level, we claim that one of the primary difficulties in applying common (multilevel) PinT schemes to advective/hyperbolic problems is the separate treatment of temporal and spatial variables. A natural result of this is that coarsening performed separately in space and time is often unable to align with hyperbolic characteristics in space-time. Conversely, by treating space and time all-at-once, it is natural for coarsening to align with characteristics, which provides an important piece of a scalable multilevel method.

Figure 3.2 demonstrates how classical AMG coarsening [135] applied to a hyperbolic 2d-space/1d-time HDG discretization naturally applies semi-coarsening along the direction of (space-time) characteristics. For clarity, examples are shown in two-dimensional subdomains for the problem described in Section 3.3.2, with plots for the advective field and the corresponding CF-splitting. The velocity fields given in (a) and (b) correspond to CF-splittings in (d) and (e), respectively. Note that for both cases, we largely see stripes of fine and coarse points orthogonal to the flow direction, which is exactly semi-coarsening along the characteristics. Similarly, in (c), note that in the $[0, 0.2] \times [0, 0.2]$ spatial subdomain (for all time), there is effectively no spatial advection, and thus the space-time

advective field is only traveling forward in time. Plots (f)–(i) demonstrate an effective semi-coarsening in time, where we mark coarse points on the time levels (see (f) and (i)) and fine points on interior time DOFs (see (g) and (h)).



(a) Velocity field, $x = 0.1861$ (b) Velocity field, $x = 0.5790$ (c) Velocity field, any fixed t

(d) $x = 0.1861$ (e) $x = 0.5790$ (f) $t = 0.3571$

(g) $t = 0.3646$ (h) $t = 0.3853$ (i) $t = 0.3928$

Figure 3.2: Red points are C-points and black points are F-points for the hyperbolic problem from Section 3.3.2. Distribution of the C- and F- points follow the velocity fields, showing semi-coarsening along characteristics.

3.2.2 Relaxation and element ordering

Upwinded discontinuous discretizations of linear hyperbolic problems have the benefit that the mesh elements can typically be reordered to be (element) block lower triangular. The corresponding linear system can then be solved directly using a forward solve. Although this is not scalable in parallel (because each process must wait for the previous to finish its solve), it provides an excellent relaxation scheme when each core inverts the subdomain stored on-process. Such a method is commonly used in the transport community to avoid the parallel cost and complexity of a full forward solve, and was shown to provide strong convergence when used with AIR in [71].

Here, we show that an analogous property holds for HDG discretizations of advection (steady or space-time). We do so by noting that the existence of such an ordering is equivalent to proving that the graph of the discretization matrix is acyclic. Assume all mesh elements are convex, let $\{\mathcal{K}_i\}$ denote the set of all n_e elements for a given mesh, and let $\mathbb{E}$ denote the graph of connections between elements, where $\mathbb{E}_{ij} = 1$ if $\exists$ a connection from $\mathcal{K}_i \mapsto \mathcal{K}_j$ with respect to the given velocity field and $\mathbb{E}_{ij} = 0$ otherwise. Let $\{\mathcal{S}_{ij}\}$ denote the set of all n_f outgoing faces, with the subscript $\mathcal{S}_{ij}$ indicating a connection $\mathcal{K}_i \mapsto \mathcal{K}_j \in \mathbb{E}$, and $\mathbb{F}$ denoting the graph of connections between faces. Moreover note that

$$\mathcal{S}_{ij} \mapsto \mathcal{S}_{jk} \quad \text{if and only if} \quad \mathcal{K}_i \mapsto \mathcal{K}_j \text{ and } \mathcal{K}_j \mapsto \mathcal{K}_k. \tag{3.10}$$

Lemma 3.2.1. *Suppose $\mathbb{E}$ is a directed acyclic graph, and the elements $\{\mathcal{K}_i\}$ are ordered such that $\mathbb{E}$ is lower triangular. Furthermore, numerate faces $\mathcal{S}_{ij}$ with respect to index i and then j, for example, $\{\mathcal{S}_{01}, \mathcal{S}_{02}, \mathcal{S}_{12}, \mathcal{S}_{23}, ...\}$. Then, $\mathbb{F}$ is also a directed acyclic graph and lower triangular in this ordering.*

Proof. Because $\mathbb{E}$ is lower triangular, $\nexists\, \mathcal{K}_i \mapsto \mathcal{K}_j$ if $i > j$. It follows by definition that $i < j$ for all faces $\mathcal{S}_{ij}$. Now, suppose there exists a path $\mathcal{S}_{ij} \mapsto \mathcal{S}_{jk}$ in $\mathbb{F}$ such that $i > k$. By Equation (3.10), this is true if and only if $\mathcal{K}_i \mapsto \mathcal{K}_j \mapsto \mathcal{K}_k$. However, this is a contradiction to the assumption of $\mathbb{E}$ being lower triangular. In addition note that by the convexity of elements, $\nexists$ connections $\mathcal{S}_{ij} \mapsto \mathcal{S}_{ik}$, that is, connections between outgoing faces with respect to the velocity field on the same element. Enumerating $\{\mathcal{S}_{ij}\}$ first by index i, then (arbitrarily) by index j as the set of faces $\{\widehat{\mathcal{S}}_\ell\}$ implies $\nexists$ path $g_{ij} \in \mathbb{F}$, $g : \widehat{\mathcal{S}}_i \mapsto \widehat{\mathcal{S}}_j$, such that $i > j$, which completes the proof. □

Lemma 3.2.1 is useful in that an ordering can be determined for an on-process block Gauss–Seidel relaxation which exactly inverts the advective component in the case of no cycles in the mesh, where the block size is given by the number of DOFs in a given element face. Such a relaxation scheme is explored numerically in Section 3.3.2.

3.3 Numerical simulations

This section demonstrates the effectiveness of AIR as a preconditioner for BiCGSTAB to solve the linear system Equation (3.9), including on moving time-dependent domains (Section 3.3.1), and when applying space-time AMR to an interior front problem (Section 3.3.2). All test cases have been implemented in the Modular Finite Element Method (MFEM) library [2] with solver support from HYPRE [1]. Furthermore, we choose the penalty parameter in Equation (3.6) as $\alpha = 10p^2$ where p is the order of the polynomial approximation (see, for example [134]). Unless otherwise specified, AIR is constructed with distance-one connections for building R, with strength tolerance 0.3; 1-point interpolation [107]; no pre-relaxation; post-forward-Gauss-Seidel relaxation (on process), first on F-points, followed by all points; Falgout coarsening, with strength tolerance 0.2; and as an acceleration method for BiCGSTAB (Bi-Conjugate Gradient STABilised), applied to the HDG space-time matrix, scaled on the left by the facet block-diagonal inverse. All parallel simulations are run on the LLNL Quartz machine.

3.3.1 Rotating Gaussian pulse on a time-dependent domain

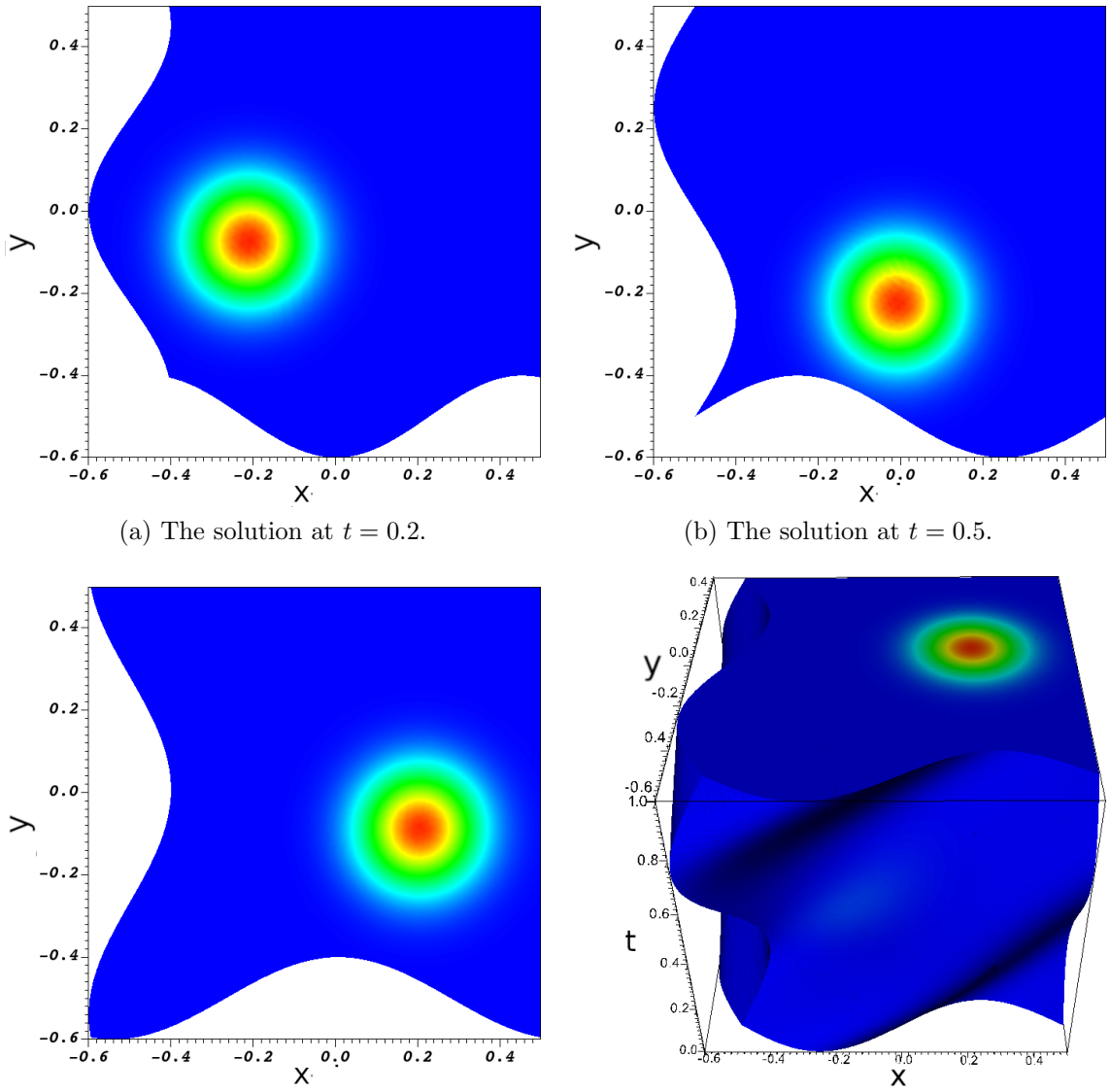
We first consider the solution of a two-dimensional rotating Gaussian pulse on a time-dependent domain [129]. We set $a = (-4x_2, 4x_1)^T$ and $f = 0$. The boundary and initial conditions are chosen such that the analytical solution is given by

$$u(t, x_1, x_2) = \frac{\sigma^2}{\sigma^2 + 2\nu t} \exp\left(-\frac{(\tilde{x}_1 - x_{1c})^2 + (\tilde{x}_2 - x_{2c})^2}{2\sigma^2 + 4\nu t} \right), \tag{3.11}$$

where $\tilde{x}_1 = x_1 \cos(4t) + x_2 \sin(4t)$, $\tilde{x}_2 = -x_1 \sin(4t) + x_2 \cos(4t)$, and $(x_{1c}, x_{2c}) = (-0.2, 0.1)$. Furthermore, we set $\sigma = 0.1$ and consider both a diffusion-dominated case with $\nu = 10^{-2}$ and an advection-dominated case with $\nu = 10^{-6}$. The deformation of the time-dependent domain is based on a transformation of the uniform space-time mesh $(t, x_1^u, x_2^u) \in [0, T] \times [-0.5, 0.5]^2$ given by

$$x_i = x_i^u + A(\tfrac{1}{2} - x_i^u) \sin(2\pi(\tfrac{1}{2} - x_i^* + t)) \quad i = 1, 2, \tag{3.12}$$

where $(x_1^*, x_2^*) = (x_2^u, x_1^u)$, $A = 0.1$, and T is the final time. We show the solution on the time-dependent domain at different time slices and on the full space-time domain (taking $T = 1$) in Figure 3.3.



(a) The solution at $t = 0.2$.

(b) The solution at $t = 0.5$.

(c) The solution at $t = 0.8$.

(d) The solution over the space-time domain $\mathcal{E}_h$.

Figure 3.3: The solution of the rotating Gaussian pulse test case as described in Section 3.3.1 at different time slices and on the full space-time domain when $\nu = 10^{-6}$.

Rates of convergence of the space-time error

In Table 3.1 we compute the rates of convergence of the error in the space-time L^2-norm, i.e.,

$$\|u - u_h\|_{\mathcal{E}_h} := \left(\int_{\mathcal{E}_h} 1 (u - u_h)^2 \, \mathrm{d}\hat{x} \right)^{1/2}.$$

We compute this error taking $T = 1$ and using linear, quadratic, and cubic polynomial approximations to u. We observe optimal rates of convergence, that is, the error in the space-time L^2-norm is of order $\mathcal{O}(h^{p+1})$ when using a p-th order polynomial approximation, for both the all-at-once and slab-by-slab discretizations. This conclusion is true for both the advection- and diffusion-dominated problem.

Table 3.1: Error in the space-time L^2-norm and rate of convergence of a space-time HDG discretization of the advection-diffusion problem described in Section 3.3.1, with $T = 1$.

Slab-by-slab

			$p=1$		$p=2$		$p=3$	
ν	Slabs	Elements per slab	Error	Rate	Error	Rate	Error	Rate
10^{-2}	8	384	1.1e-2	-	2.9e-3	-	8.4e-4	-
	16	1,536	3.4e-3	1.7	4.7e-4	2.6	5.8e-5	3.9
	32	6,144	8.4e-4	2.0	5.9e-5	3.0	3.7e-6	4.0
	64	24,576	2.1e-4	2.0	7.4e-6	3.0	2.3e-7	4.0
10^{-6}	8	384	1.9e-2	-	5.3e-3	-	1.3e-3	-
	16	1,536	6.1e-3	1.6	8.5e-4	2.7	1.3e-4	3.4
	32	6,144	1.6e-3	2.0	1.1e-4	3.0	9.0e-6	3.8
	64	24,576	3.8e-4	2.0	1.4e-5	3.0	5.9e-7	3.9

All-at-once

		$p=1$		$p=2$		$p=3$	
ν	Elements	Error	Rate	Error	Rate	Error	Rate
10^{-2}	2,760	2.0e-2	-	6.0e-3	-	1.7e-3	-
	22,080	5.8e-3	1.8	8.2e-4	2.9	1.2e-4	3.8
	176,640	1.3e-3	2.1	9.5e-5	3.1	7.5e-6	4.0
	1,413,120	3.0e-4	2.1	1.2e-5	3.1	4.6e-7	4.0
10^{-6}	2,760	5.5e-2	-	2.1e-2	-	8.9e-3	-
	22,080	2.0e-2	1.4	3.6e-3	2.6	7.2e-4	3.6
	176,640	5.1e-3	2.0	3.8e-4	3.2	4.5e-5	4.0
	1,413,120	1.0e-3	2.3	4.3e-5	3.2	2.6e-6	4.1

Performance of BiCGSTAB with AIR as preconditioner

This section demonstrates the performance of BiCGSTAB with AIR as a preconditioner in both the advection- and diffusion-dominated regimes. We will use the number of iterations to convergence as the indicator of performance as we know that the setup time and cost of applying AIR per BiCGSTAB iteration are linear with respect to the matrix size, i.e. $\mathcal{O}(N)$ [109, 107]. Hence, the total cost to solve the space-time HDG problem will be linearly dependent on the number of BiCGSTAB iterations. In Table 3.2 we list the total number of BiCGSTAB iterations that are required to reach a relative residual of 10^{-12} in an all-at-once discretization with $T = 1$ and using linear, quadratic, and cubic polynomial approximations to u.

When the problem is close to hyperbolic (when $\nu = 10^{-6}$) we observe perfect scalability, that is, the number of iterations required to converge does not change with the problem size. In the advection-dominated regime, $\nu = 10^{-4}$ and $\nu = 10^{-3}$, the iteration count increases slightly with problem size, but the increase is slow, the iteration counts remain quite low. When more significant diffusion is introduced, $\nu = 10^{-2}$ and $\nu = 10^{-1}$, the iteration count starts to grow more rapidly with increasing problem size. These observations hold for all polynomial degrees considered. It is worth pointing out that for $\nu = 10^{-2}$ and $\nu = 10^{-1}$, using a classical $P^T AP$ AMG approach rather than AIR did result in lower iteration counts (not shown), however, the total time to solution remained notably longer than that of AIR, likely due to denser coarse-grid matrices. Alternatively, an approach similar to [8] could be used to improve the robustness of our approach, by (locally) switching between AIR and a classical AMG approach depending on the parameter ν. However, such an implementation is not available to us at the moment.

From the above observations, we may conclude that BiCGSTAB with AIR as the preconditioner is an excellent iterative solver for the solution of all-at-once space-time HDG discretizations of the advection-diffusion problem in the advection-dominated regime. Unsurprisingly, the solver is suboptimal in the diffusion-dominated regime. To see why, note that we may write Equation (3.3a) as

$$\widehat{a} \cdot \widehat{\nabla} u - \widehat{\nabla} \cdot (\widehat{\nu} \widehat{\nabla} u) = f \qquad \text{in } \mathcal{E}_h,$$

where $\widehat{\nu} = \operatorname{diag}(0, \nu, \nu)$ (note that there is no diffusion in the time direction). This is a "steady" advection-diffusion problem in $(d+1)$-dimensional space-time with completely *anisotropic* diffusion in d dimensions and advection in one dimension. Problems with anisotropic diffusion are known to pose a challenge to multilevel solvers (see, for example, [143] for a literature review on the challenges of using multilevel solvers for problems with

Table 3.2: The number of BiCGSTAB iterations (with AIR as the preconditioner) required to reach a relative residual of 10^{-12} for the test case described in Section 3.3.1 with $T = 1$. The stopping tolerance was not reached within 5000 iterations if a value is missing.

$p = 1$

DOFs	ν				
	10^{-6}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
17,496	7	7	7	8	12
136,224	8	7	8	10	17
1,074,816	8	8	10	13	54
8,538,624	8	9	12	18	-

$p = 2$

DOFs	ν				
	10^{-6}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
34,992	11	8	9	11	19
272,448	8	9	10	14	30
2,149,632	9	11	13	18	46
17,077,248	9	14	15	30	83

$p = 3$

DOFs	ν				
	10^{-6}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
58,320	9	8	10	14	26
454,080	9	11	12	18	38
3,582,720	9	13	15	25	73
28,462,080	10	17	18	46	144

anisotropic diffusion). Robust solvers for the mixed regime of advection and strongly anisotropic diffusion are ongoing work.

Remark 3.3.1 (Stopping tolerance)**.** *Our main goal is to test the performance of BiCGSTAB with AIR as a preconditioner. For this reason, we chose the stopping criteria for BiCGSTAB to be a relative residual of* 10^{-12}*. In practice, however, the stopping tolerance need not be chosen this small (see, for example [51, pages 73, 77–79]). We demonstrate this also in Table 3.3 where, for the case* $p = 2$ *for a problem with 272,448 degrees of freedom. We note that the error in the space-time* L_2*-norm does not improve after the first full BiCGSTAB iteration although it takes 8 iterations to reach a relative residual of* 10^{-12} *(see Table 3.2).*

Table 3.3: Error in the space-time L_2-norm as a function of BiCGSTAB iteration number for the test case from Table 3.2. We use a quadratic ($p = 2$) polynomial approximation and the linear system has 272,448 degrees of freedom. The preconditioned residual presented in the table is the residual of the full-step.

Iteration Number	Preconditioned Residual	$\|u - u_h\|_{L_2(\mathcal{E}_h)}$
0	2.5e-4	1.6e-3
1	8.4e-6	3.6e-3
2	1.0e-6	3.6e-3
3	6.1e-9	3.6e-3
4	1.6e-10	3.6e-3

Remark 3.3.2 (GMRES and other Krylov)**.** *In this section we considered the performance of BiCGSTAB with AIR as a preconditioner. Of course, we may replace BiCGSTAB with any other iterative method for non-symmetric systems of linear equations. GMRES performed equally well in most cases; however, there were several examples that stalled significantly upon GMRES restart, and which also required a moderately high number of iterations to convergence, limiting the use of full-memory GMRES. In our tests, BiCGSTAB has appeared to be slightly more robust and, thus, is used for all numerical tests presented here.*

Scalability and parallel-in-time on moving domains

The current paradigm in scientific computing is to use multiple computing units simultaneously to lower runtime. Hence, the scalability of an algorithm is an important measure

of performance. Ideally, the runtime should be inversely proportional to the number of computing units. Unfortunately, this is not always achievable due to limited fast access memory (caches), limited memory bandwidth, and inter-process and inter-node communication. One advantage of the all-at-once space-time approach over the slab-by-slab space-time approach is the better communication to computation ratio. This is because it is possible to parallelize both in space and time simultaneously, as opposed to the standard parallel-in-time approach of treating space and time separately.

To test the scalability of BiCGSTAB with AIR as a preconditioner applied to the space-time HDG discretization, we will measure the total wall-clock time spent on solving the rotating Gaussian pulse problem discussed at the beginning of this section. For this, we consider a final time of $T = 16$, we consider both an advection- ($\nu = 10^{-6}$) and a diffusion-dominated ($\nu = 10^{-2}$) problem, and we consider both an all-at-once and a slab-by-slab discretization. For the all-at-once discretization, we consider two unstructured space-time meshes; the coarse mesh consists of 45576 tetrahedra and the fine mesh consists of 364608 tetrahedra. For the slab-by-slab approach, we consider a coarse mesh in which the space-time domain is divided into 128 space-time slabs and each slab consists of 384 tetrahedra. The fine slab-by-slab mesh consists of 256 slabs and each slab consists of 1536 tetrahedra. Note that the slab-by-slab meshes were created to have a similar number of tetrahedra as the all-at-once space-time meshes.

The total wall-clock times we measure are the combination of time spent on the following four stages: setup, assembly, solving, and reconstruction. During the setup stage, the mesh is read from a file and refined sequentially and finite element spaces and linear and bi-linear forms are created. We remark that this stage is not parallelizable and it affects the speedup we obtain. The assembly stage contains the computation of elemental matrices, computation of elemental Schur complements, and the assembly of the global linear system Equation (3.9). This stage is almost embarrassingly parallel. The next stage is the solve stage in which the global linear system is solved using BiCGSTAB with AIR as the preconditioner. This stage is weakly scalable. Finally, the element solution $U = A^{-1}(F - B\Lambda)$ is reconstructed in the reconstruction stage (see Section 3.1.5). This step, in theory, does not require any communication as it can be done completely locally.

Parallel speedup in a strong-scaling sense for each combination of mesh resolution and diffusion coefficient is shown in Figure 3.4. We see that, in all cases, the all-at-once approach is the best algorithm sequentially. Hence, the speedups are calculated relative to the sequential timing of the solutions using all-at-once approach for different order of approximations. The best speedup we achieve at 256 processes is slightly more that 100, and just less than 50% efficiency. This can be mostly attributed to the sequential nature of the setup stage, for example, it takes up to 10% of wall-clock time spent for large problems

solved with many (64-256) cores. In addition to this, there is a significant loss of scalability during the solve stage, which is largely due to the algorithm becoming communication bound and thereby less efficient in parallel. For example, the speedup observed on the fine mesh at 256 processes is close to 2× larger than that observed on the coarse mesh. Hence, for larger problems, we expect better speedup with the primary bottleneck being the setup stage. It is worth pointing out that a number of recent works have developed architecture-aware communication algorithms for sparse matrix-vector operations and AMG that can significantly improve scalability in the communication-bound regime (e.g., [18, 19]), but we do not exploit such methods here.

Figure 3.5 plots the relative speedup of the all-at-once approach to the slab approach with respect to wall-clock time, that is, $\text{Time}_{\text{all-at-once}}(n)/\text{Time}_{\text{slab-by-slab}}(n)$. We see that, generally, the all-at-once approach is 20% to 50% faster than the slab-by-slab approach, although in some cases it is up to 2× faster. Note that this comparison is imperfect, and an accurate measure of speedup is nuanced – for example, the slab mesh here has roughly 8% more elements than the all-at-once mesh; however, the slab mesh is also structured in time, while the all-at-once mesh is fully unstructured in space and time, which can degrade performance of multigrid solvers on a fine-grained/memory-access level. They also differ algorithmically; for example, the all-at-once approach does one setup phase for AIR, followed by the solve phase, while using the slab-by-slab approach requires rebuilding the solver each time step. In general, we do not try to isolate where the speedup comes from in this chapter. Rather, we highlight here that by using AIR as a full space-time solver, we are able to see speedups over sequential time stepping for low core counts, a property that is not shared by most parallel-in-time schemes.

3.3.2 Moving internal layer problem

We now consider the moving internal layer problem proposed in [33]. We solve Equation (3.3) on the unit cube space-time domain ($\mathcal{E}_h = [0,1]^3$) with $a = (x_2, -x_1)^T$, $f = 0$, and with $\nu = 0$ (the hyperbolic limit). We impose a Neumann boundary at $t = 0$, on which we set $g_N = 0$, and an outflow boundary at the final time $t = 1$. On the boundary $x_2 = 0$ we set $g_D = 1$ and we set $g_D = 0$ on the remaining boundaries. For the time interval of interest, the exact solution is given by

$$u(t, x_1, x_2) = \begin{cases} 1 & \text{when } \|(x_1, x_2)\|_2 < 1 \text{ and } \operatorname{atan2}(x_2, x_1) > \pi/2 - t, \\ 0 & \text{otherwise,} \end{cases}$$

which describes a front that rotates around the origin as time evolves.

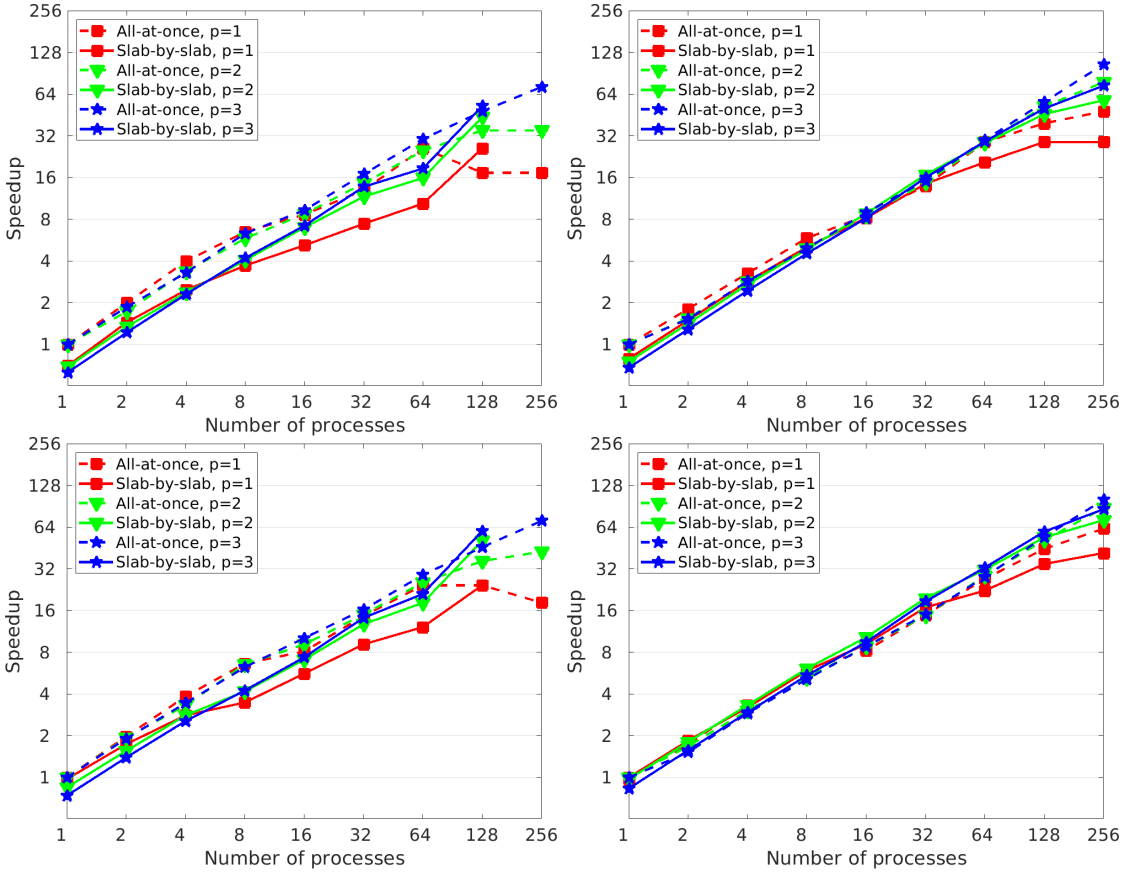


Figure 3.4: Parallel scalability. Top left: $\nu = 10^{-6}$, coarse mesh, top right: $\nu = 10^{-6}$, fine mesh, bottom left: $\nu = 10^{-2}$, coarse mesh, bottom right: $\nu = 10^{-2}$, fine mesh

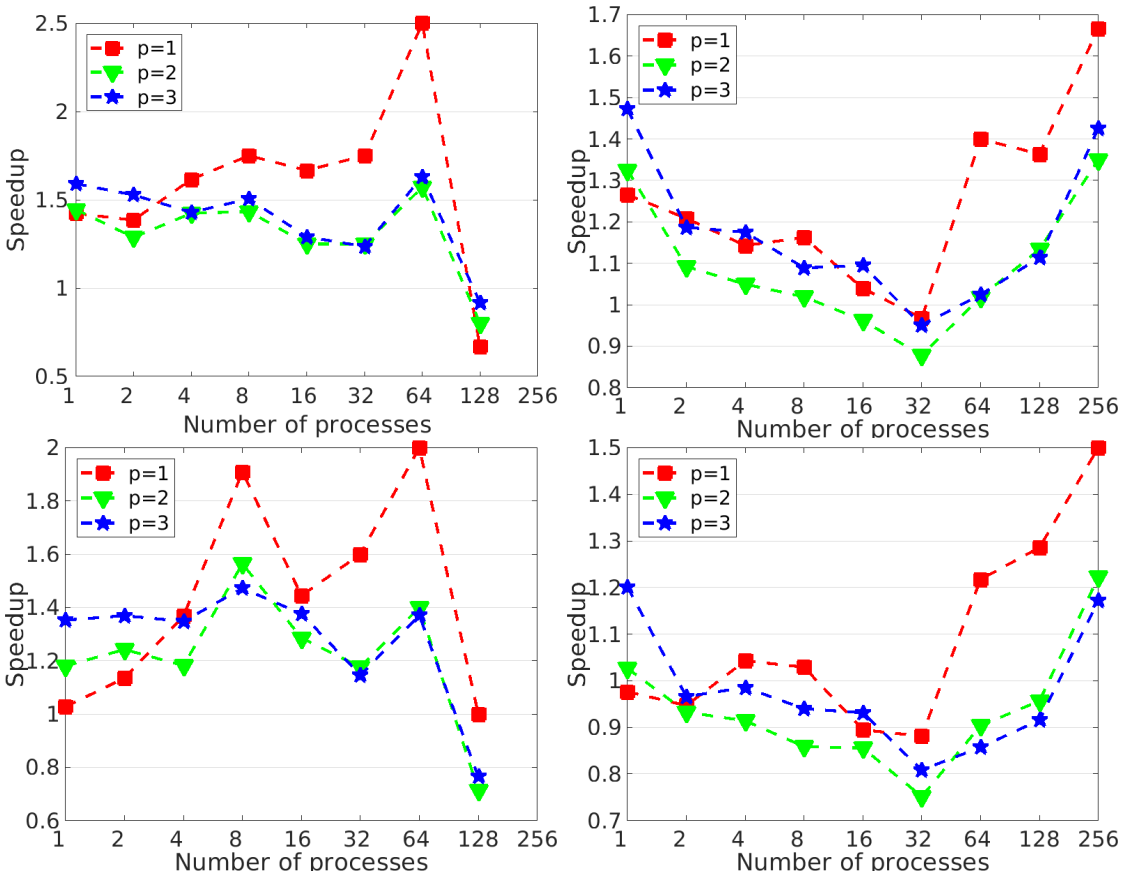


Figure 3.5: Relative speedup of all-at-once approach against slab-by-slab approach. Top left: $\nu = 10^{-6}$, coarse mesh, top right: $\nu = 10^{-6}$, fine mesh, bottom left: $\nu = 10^{-2}$, coarse mesh, bottom right: $\nu = 10^{-2}$, fine mesh

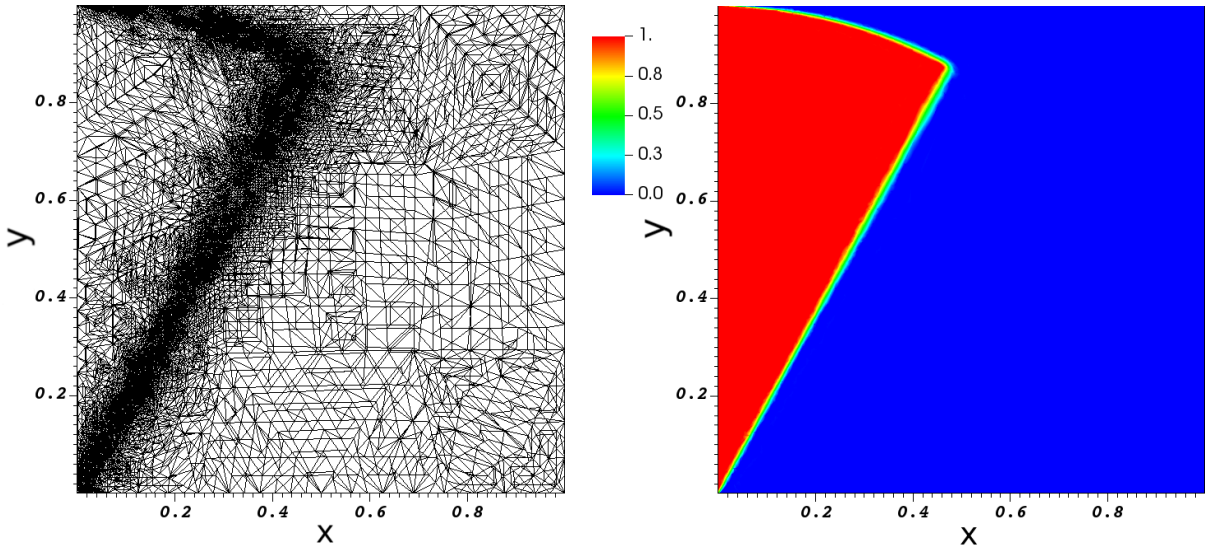
Space-time adaptive mesh refinement

To efficiently solve this problem, we use space-time adaptive mesh refinement (AMR) in an all-at-once discretization, where we refine locally in both space and time. The Zienkiewicz–Zhu (ZZ) error estimator [170, 171, 172] is used to mark space-time elements that need to be refined. Although the ZZ error estimator is not theoretically efficient or reliable for many problems, it is often used heuristically in adaptive finite element codes due to its simplicity, low computational cost, and wide availability. See Figure 3.6 for a plot of the mesh and the solution at two different time slices. A plot of the adaptively refined mesh in space-time is given in Figure 3.7.

In Figure 3.8 we compare the convergence of the error in the space-time L^2-norm using space-time AMR to using uniformly refined meshes. Let N be the total number of globally coupled DOFs. We see that the error on uniform meshes is approximately $\mathcal{O}(N^{-1/9})$ while the error on the space-time AMR meshes is approximately $\mathcal{O}(N^{-1/6})$, i.e., we obtain faster convergence using space-time AMR than when using uniformly refined meshes. We remark that the error when using an efficient and reliable error estimator for this problem is expected to be $\mathcal{O}(N^{-1/3})$ (see, for example, [25] for the analysis of an a posteriori error estimator for a DG discretization of the steady advection equation). However, we are not aware of any efficient and reliable error estimators for space-time HDG discretizations of the time-dependent advection equation.

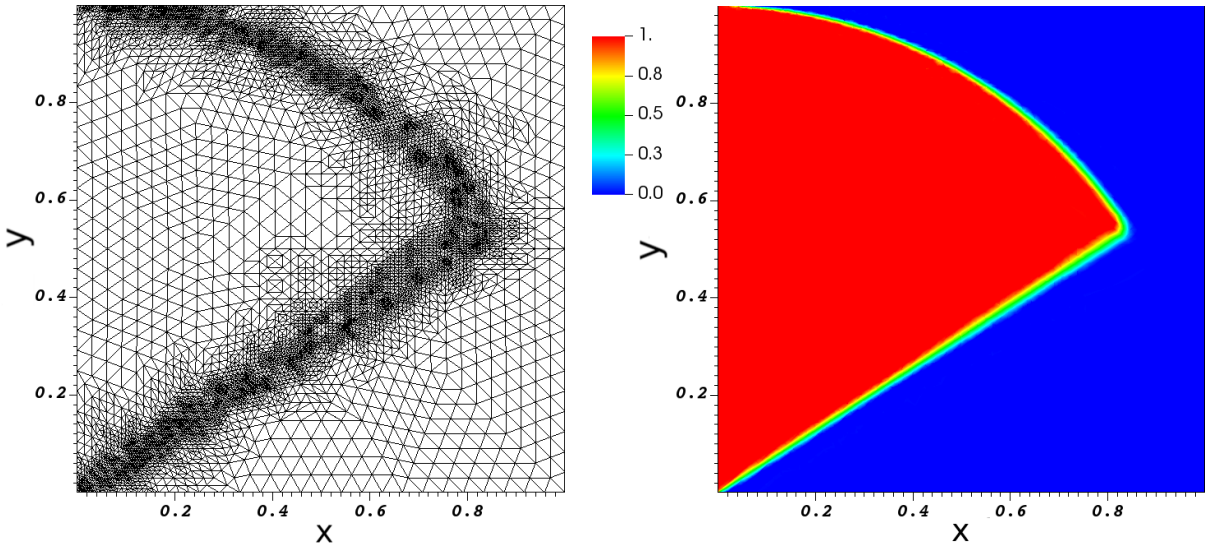
Performance of AIR and on-process solves

Last, we consider the performance of BiCGSTAB with AIR as a preconditioner within the context of space-time AMR, and demonstrate the application of Lemma 3.2.1. It is well known that upwind DG discretizations of advection on convex elements yield matrices that are block triangular in some element ordering. This can serve as a robust on-process relaxation routine, where the triangular element ordering is obtained and an ordered block Gauss–Seidel exactly inverts the on-process subdomain [71]. AIR also relies, in some sense, on having a matrix with dominant lower triangular structure, where it can be shown that triangular structure allows for a good approximation to ideal restriction [107]. Although HDG discretizations are not always thought of as block linear systems in the same way that DG discretizations are, Lemma 3.2.1 proves that by treating DOFs on a given facet as a block in the matrix, an analogous result holds, that is, the matrix is block lower triangular in some ordering. With AIR preconditioning, the block structure can be accounted for by using a block implementation of AIR (e.g., [109]) coupled with block relaxation or, in the



(a) Mesh slice at $t = 0.5$

(b) Solution slice at $t = 0.5$



(c) Mesh slice at $t = 1$

(d) Solution slice at $t = 1$

Figure 3.6: The numerical solution to the interior layer problem at two different time slices. The non-triangular polygons in the top left figure are because we are slicing the space-time mesh at $t = 0.5$; we are cutting through space-time tetrahedra.

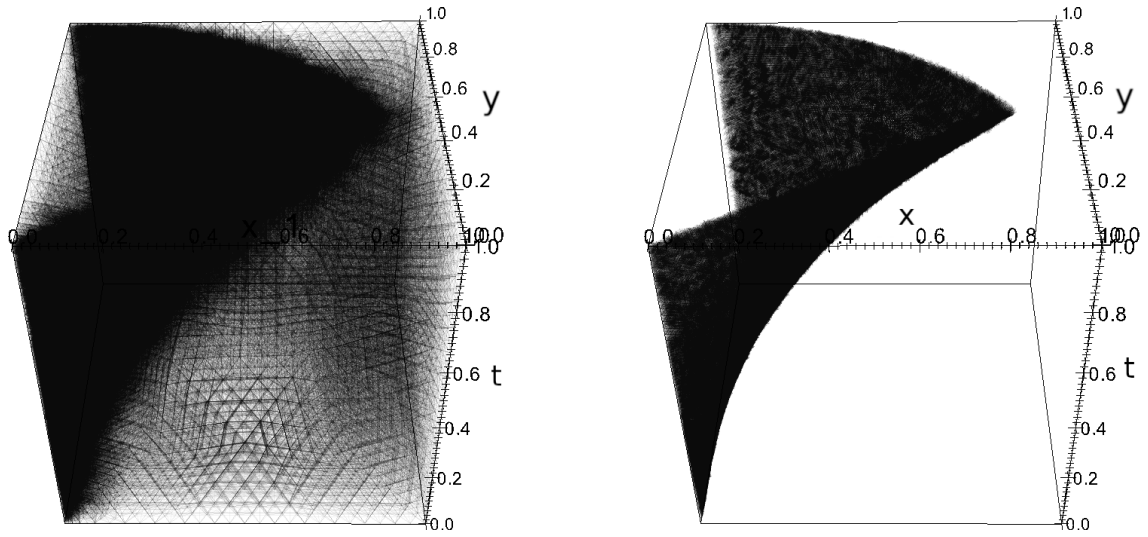


Figure 3.7: Left: the space-time AMR mesh obtained using the ZZ error estimator for the test case described in Section 3.3.2. Right: only the elements below the median element size are shown. Note that the mesh is refined along the space-time interior layer.

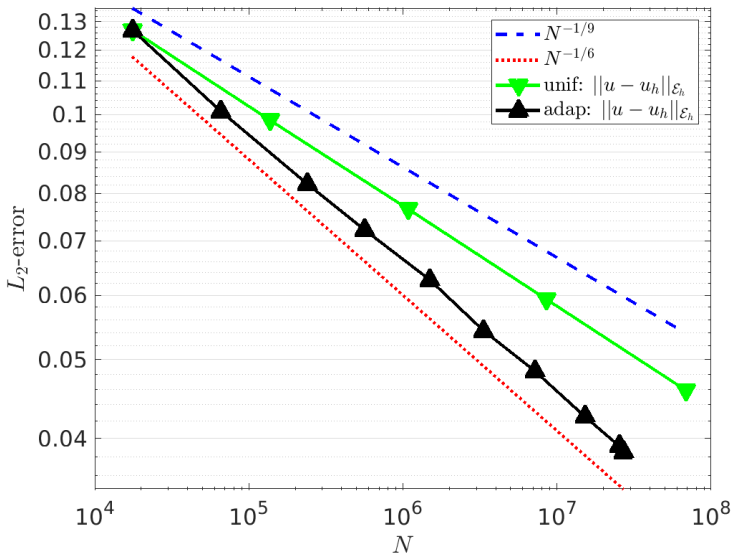


Figure 3.8: We compare the convergence of the error in the space-time L^2-norm using space-time AMR to using uniformly refined space-time meshes. The test case is described in Section 3.3.2. Here N is the total number of globally coupled DOFs.

advection-dominated regime, scaling on the left by the block-diagonal inverse, wherein the scaled matrix is then scalar lower triangular.

Figure 3.9 demonstrates each of these points in practice, applying AIR to a succession of adaptively refined space-time problems with various relaxation and block inverse strategies. The number of DOFs on the x-axis corresponds to successive levels of adaptive space-time mesh refinement (with correspondingly larger number of DOFs). First, note that accounting for the block structure in the matrix is important for scalable convergence at larger problem sizes. Not scaling by the block inverse ("No Block Inv") can lead to an increase in iteration count by more than $3\times$ for the largest problem size, and likely worse as DOFs further increase. On the other hand, after applying the block inverse scaling, even pointwise Jacobi relaxation yields near perfectly scalable convergence. Furthermore, because we are considering a hyperbolic equation with cycle-free space-time velocity field $\widehat{a} = (1, x_2, -x_1)^T$, from Lemma 3.2.1 the scaled matrix is lower triangular. A topological sort of the on-process matrix yields the triangular ordering, and an ordered Gauss–Seidel relaxation then exactly inverts the on-process block. Simulations in Figure 3.9 are run on 128 cores, and we see that with an on-process solve as relaxation, the number of iterations required to converge is half of the second-best relaxation method we tested, forward Gauss–Seidel (although both are still quite good).[1]

Remark 3.3.3 (Relation to PinT)**.** *It is worth pointing out the relation of the on-process solve to PinT methods. In MGRiT and Parareal, the relaxation scheme corresponds to solving the time-propagation problem between F-time-points. If you assign one F-point per process, this is solving the time-propagation problem exactly on process and is coupled with a coarsening in time. Similar to the discussion in Section 3.2.1 here, we actually solve the space-time problem exactly along the characteristics on each process, which is then coupled with a coarsening that aligns with the characteristics (see Figure 3.2). Again, we believe this more holistic treatment of space and time is what allows for perfectly scalable parallel-in-time convergence on hyperbolic problems.*

3.4 Conclusions

AIR algebraic multigrid is known to be a robust preconditioner for discretizations of steady advection-dominated advection-diffusion problems. This research was motivated by the

[1]Note that the moving domain considered in Section 3.3.1 introduces cycles in the matrix-graph, and the resulting matrix is not necessarily block triangular. However, cycle-breaking strategies such as used in [72] for DG transport simulations on curvilinear meshes can find a "good" ordering and provide comparable performance as a direct on-process solve when coupled with the larger AIR algorithm.

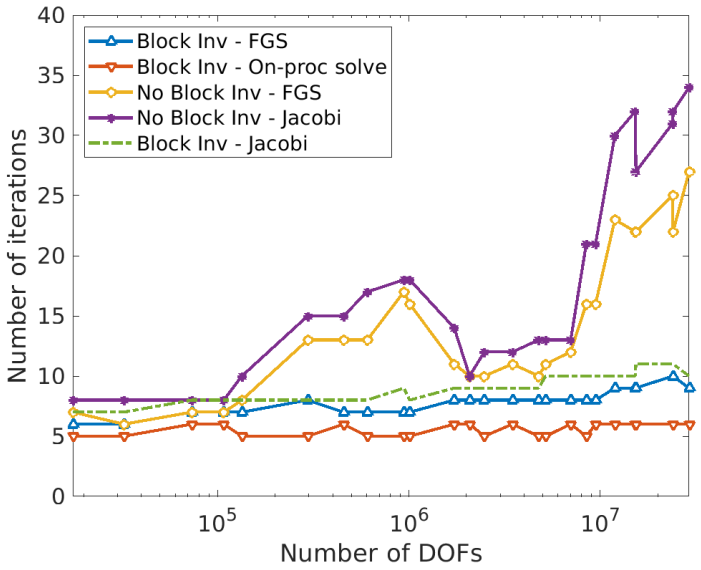


Figure 3.9: A comparison of the number of BiCGSTAB iterations to convergence using AIR as preconditioner with different relaxation strategies. We plot the number of iterations against the number of globally coupled DOFs at different levels of refinement within the AMR algorithm.

question whether AIR AMG is robust as an all-at-once solver for space-time HDG discretizations of time-dependent advection-diffusion problems, since such problems can be seen as "steady" advection-diffusion problems in $(d+1)$-dimensions. By numerical examples, we have indeed demonstrated that AIR provides fast, effective, and scalable preconditioning for space-time discretizations of advection-dominated problems, including robust convergence on space-time AMR and moving, time-dependent domains.

Advection-dominated problems are notoriously difficult for parallel-in-time methods, motivating a number of efforts to develop specialized techniques that can handle advection on coarse time-grids (e.g., [35, 32]). Here, we claim that the best way to provide time parallelism for hyperbolic problems is by treating space and time *together*. In particular, a critical component in multigrid methods is constructing an effective coarse grid. By applying AIR all-at-once to a space-time discretization, coarsening is able to align with hyperbolic characteristics in space-time and provide a coarse-grid that naturally captures these characteristics. Moreover, we proved that for purely hyperbolic problems, the space-time HDG discretization on convex elements is block triangular in some ordering. Using this ordering, a relaxation scheme can be designed that exactly solves along the characteristics on-process, complementing the coarse-grid alignment. Classical parallel-in-time multigrid methods that coarsen in space and time separately are typically unable to align with hyperbolic characteristics, often resulting in slow convergence or divergence for time-dependent advection-dominated problems.

Chapter 4

Preconditioners for an HDG discretization of the Navier–Stokes problem

In this chapter, we introduce two novel preconditioners for an HDG discretization of the stationary Navier–Stokes equations. These preconditioners are based on grad-div (or augmented Lagrangian) preconditioners and pressure convection-diffusion preconditioners, respectively, discussed in Sections 1.4.1 and 1.4.2.

There are multiple challenges when developing preconditioners for HDG discretizations of the Navier–Stokes problem. One of the main challenges is related to static condensation: directly developing preconditioners for the reduced systems obtained after static condensation is difficult due to complexity of the operators involved, especially if both the element velocity and the element pressure unknowns are eliminated. That being said, there are some approaches which can be used to overcome this challenge, see [27] and [133] for two examples. Here, we propose a third approach where we develop preconditioners for the full problem, and modify the preconditioners to apply them to the reduced problems. This is possible by Lemma 4.0.1 together with Theorem 1.1.8.

Lemma 4.0.1. *Let*

$$\mathcal{A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & 0 \end{bmatrix},$$

and

$$\bar{\mathcal{A}} = \begin{bmatrix} A_{22} - A_{21}A_{11}^{-1}A_{12} & A_{23} - A_{21}A_{11}^{-1}A_{13} \\ A_{32} - A_{31}A_{11}^{-1}A_{12} & -A_{31}A_{11}^{-1}A_{13} \end{bmatrix},$$

be non-singular real matrices, where A_{11} and A_{22} are non-singular square matrices, and each A_{ij} is compatible with A_{ji}. Let

$$P = \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{21} & A_{22} & 0 \\ A_{31} & A_{32} & K \end{bmatrix},$$

for some non-singular square matrix K and let

$$\bar{P} = \begin{bmatrix} A_{22} - A_{21}A_{11}^{-1}A_{12} & 0 \\ A_{32} - A_{31}A_{11}^{-1}A_{12} & K \end{bmatrix}.$$

Then $\mathcal{W}(\bar{P}^{-1}\bar{\mathcal{A}}) \subseteq \mathcal{W}(P^{-1}\mathcal{A})$.

Proof. Let $I_1 \in \mathbb{R}^{n_1 \times n_1}$, $I_2 \in \mathbb{R}^{n_2 \times n_2}$, and $I_3 \in \mathbb{R}^{n_3 \times n_3}$ be identity matrices. We then define the following block matrices,

$$E_1 = \begin{bmatrix} A_{11}^{-1} & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix}, \qquad E_2 = \begin{bmatrix} I_1 & 0 & 0 \\ -A_{21} & I_2 & 0 \\ -A_{31} & 0 & I_3 \end{bmatrix}, \tag{4.1}$$

and let $\tilde{P} = E_2E_1P$ and $\tilde{\mathcal{A}} = E_2E_1\mathcal{A}$. We then observe that

$$\tilde{P}^{-1}\tilde{\mathcal{A}} = P^{-1}E_1^{-1}E_2^{-1}E_2E_1\mathcal{A} = P^{-1}\mathcal{A}. \tag{4.2}$$

Noting furthermore that

$$\tilde{P} = \begin{bmatrix} I & A_{11}^{-1}A_{12} & 0 \\ 0 & A_{22} - A_{21}A_{11}^{-1}A_{12} & 0 \\ 0 & A_{32} - A_{31}A_{11}^{-1}A_{12} & K \end{bmatrix}, \tag{4.3}$$

and that

$$\tilde{\mathcal{A}} = \begin{bmatrix} I & A_{11}^{-1}A_{12} & A_{11}^{-1}A_{13} \\ 0 & A_{22} - A_{21}A_{11}^{-1}A_{12} & A_{23} - A_{21}A_{11}^{-1}A_{13} \\ 0 & A_{32} - A_{31}A_{11}^{-1}A_{12} & -A_{31}A_{11}^{-1}A_{13} \end{bmatrix}, \tag{4.4}$$

it follows through direct calculation that the bottom-right 2×2-block principal submatrix of $\tilde{P}^{-1}\tilde{\mathcal{A}}$ is $\bar{P}^{-1}\bar{\mathcal{A}}$. Hence, the bottom-right 2×2-block principal submatrix of $P^{-1}\mathcal{A}$ also equals $\bar{P}^{-1}\bar{\mathcal{A}}$. It is well-known (for example, see [78, 1.2.11] or [10, Submatrix inclusion]) that $\mathcal{W}(A') \subseteq \mathcal{W}(A)$ where A' is a principal submatrix of $A \in \mathbb{C}^{n\times n}$. □

Lemma 4.0.1 implies that if P is a good preconditioner for $\mathcal{A}$, $\bar{P}$ is at least as good of a preconditioner for $\bar{\mathcal{A}}$, if not better. This lemma further predicts that the optimality of P with respect to the problem parameters h and/or Re is preserved under static condensation. In addition, we can interpret Theorems 2.2.3 and 2.2.4 (and other results from Chapter 2) as a relation between the full problem and the statically condensed problem in the context of HDG; observe that the coefficient matrix in Equation (2.35) is the Schur complement of the coefficient matrix in Equation (2.34). Moreover, as discussed at the beginning of Section 2.1, minimizing the residual over a d-dimensional Krylov subspace $\mathcal{K}_d = \{r_0, Ar_0, \ldots, A^{d-1}r_0\}$, obtained at the d-th iteration of the corresponding Krylov subspace method, is equivalent to constructing a consistent polynomial p which minimizes $\|p(A)r_0\|$ or $\|p(A)\| = \sup_{r_0\neq 0} \|p(A)r_0\|$. Therefore, we know that the minimizing polynomials $\psi^{(d)}$ and $\varphi^{(d)}$ satisfy

$$\|\psi^{(d)}(\bar{P}^{-1}\bar{\mathcal{A}})\| \leq \|\varphi^{(d)}(P^{-1}A)\|,$$

i.e., the number of GMRES iterations k required to guarantee that $\|\psi^{(k)}(\bar{P}^{-1}\bar{\mathcal{A}})\| < tol$ is less than or equal to d given that $\|\varphi^{(d)}(P^{-1}A)\| < tol$.

The rest of this chapter is organized as follows. First, in Section 4.1, we introduce grad-div preconditioners for the HDG discretization of Navier–Stokes, followed by required modifications for the statically condensed (reduced) problem. Second, in Section 4.2, we generalize PCD preconditioners for the full and reduced linear system problems. Finally, in Section 4.3, we present our numerical results.

4.1 Grad-div preconditioners for HDG

As discussed in Section 1.4.2, augmented Lagrangian preconditioners are obtained by modifying the linear system

$$\begin{bmatrix} X & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix},$$

by adding $\gamma B^T W^{-1} B$ and $\gamma B^T W^{-1}\mathbf{g}$, respectively, to the top left block of the coefficient matrix and the top block of the right hand side vector, where W is an SPD matrix of compatible dimensions and $\gamma > 0$:

$$\begin{bmatrix} X + \gamma B^T W^{-1} B & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} + \gamma B^T W^{-1}\mathbf{g} \\ \mathbf{g} \end{bmatrix}.$$

Given that the initial linear system is the result of a mixed finite element discretization of the (Navier–)Stokes equations, and choosing $W = M_p$, where M_p is the mass matrix defined on the pressure space, it is not difficult to show that $\gamma B^T W^{-1} B$ has the following integral representation:

$$\sum_K \gamma \int_K \Pi_{Q_h}(\nabla \cdot u)\Pi_{Q_h}(\nabla \cdot v)\mathrm{d}x,$$

where $\Pi_{Q_h}(f)$ is the L^2 projection of a scalar valued function f onto the space Q_h. Furthermore, if $\nabla \cdot V_h \subset Q_h$ then $\Pi_{Q_h}(q_h) = q_h$ for all $q_h \in \nabla \cdot V_h$, in which case, $\gamma B^T W^{-1} B$ simply has the following integral representation:

$$\sum_K \gamma \int_K (\nabla \cdot u)(\nabla \cdot v)\mathrm{d}x,$$

and we obtain grad-div preconditioners. Since we are solely focusing on the HDG discretization of the Navier–Stokes equations and $\nabla \cdot V_h \subset Q_h$ for the HDG method [132], we consider our preconditioners as grad-div preconditioners. However, they can also be classified as augmented Lagrangian preconditioners by the argument above.

We first confirm the relationship between the matrix $\gamma B^T W^{-1} B$ and its integral representation $\sum_K \gamma \int_K (\nabla \cdot u)(\nabla \cdot v)\mathrm{d}x$ for HDG.

Lemma 4.1.1. *Let D and B_{pu} be defined, respectively, as in Equations (2.31) and (2.32) and let M_p be the mass-matrix on Q_h. Then D can be factorized in terms of B_{pu} and M_p^{-1} as follows: $D = B_{pu}^T M_p^{-1} B_{pu}$.*

Proof. Consider first the following grad-div problem for $u : \Omega \to \mathbb{R}^d$:

$$-\gamma \nabla \nabla \cdot u + \mu u = g \qquad \text{in } \Omega, \tag{4.5a}$$

$$u \cdot n = 0 \qquad \text{on } \partial\Omega, \tag{4.5b}$$

where $\gamma \in \mathbb{R}^+$, $\mu \in \mathbb{R}^+$ and $g \in [L^2(\Omega)]^d$ are given. A hybridized divergence-conforming discretization of Equation (4.5) is given by: find $(u_h, \bar{p}_h) \in V_h \times \bar{Q}_h$ such that

$$\gamma \sum_{K \in \mathcal{T}} \int_K (\nabla \cdot u_h)(\nabla \cdot v_h) \,\mathrm{d}x + \mu \sum_{K \in \mathcal{T}} \int_K u_h \cdot v_h \,\mathrm{d}x - \sum_{K \in \mathcal{T}} \int_{\partial K} \bar{p}_h v_h \cdot n \,\mathrm{d}s = \int_\Omega g \cdot v_h \,\mathrm{d}x \quad \forall v_h \in V_h, \tag{4.6a}$$

and

$$- \sum_{K \in \mathcal{T}} \int_{\partial K} \bar{q}_h u_h \cdot n \,\mathrm{d}s = 0 \quad \forall \bar{q}_h \in \bar{Q}_h. \tag{4.6b}$$

This can be expressed as the following system of linear equations:

$$\begin{bmatrix} \gamma D + \mu M_u & B_{\bar{p}u}^T \\ B_{\bar{p}u} & 0 \end{bmatrix} \begin{bmatrix} u \\ \bar{p} \end{bmatrix} = \begin{bmatrix} G \\ 0 \end{bmatrix}, \tag{4.7}$$

where M_u is the mass matrix on V_h.

By introducing the auxiliary variable $p = -\gamma \nabla \cdot u$, we may write Equation (4.5) also as:

$$\mu u + \nabla p = g \qquad \text{in } \Omega, \tag{4.8a}$$

$$\nabla \cdot u + \gamma^{-1} p = 0 \qquad \text{in } \Omega, \tag{4.8b}$$

$$u \cdot n = 0 \qquad \text{on } \partial\Omega. \tag{4.8c}$$

A hybridized discretization of a conforming finite element method for this problem is given by: find $(u_h, \boldsymbol{p}_h) \in V_h \times \boldsymbol{Q}_h$ such that

$$\mu \sum_{K \in \mathcal{T}} \int_K u_h \cdot v_h \,\mathrm{d}x + b_h(\boldsymbol{p}_h, v_h) = \sum_{K \in \mathcal{T}} \int_K g \cdot v_h \,\mathrm{d}x \qquad \forall v_h \in V_h, \tag{4.9a}$$

$$-b_h(\boldsymbol{q}_h, u_h) + \gamma^{-1} \sum_{K \in \mathcal{T}} \int_K p_h q_h \,\mathrm{d}x = 0 \qquad \forall \boldsymbol{q}_h \in \boldsymbol{Q}_h, \tag{4.9b}$$

where $b_h(\cdot, \cdot)$ is defined by Equation (2.28d). The corresponding linear system to this discretization is given by

$$\begin{bmatrix} \mu M_u & B_{pu}^T & B_{\bar{p}u}^T \\ -B_{pu} & \gamma^{-1} M_p & 0 \\ -B_{\bar{p}u} & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ p \\ \bar{p} \end{bmatrix} = \begin{bmatrix} G \\ 0 \\ 0 \end{bmatrix}. \tag{4.10}$$

Eliminating the auxiliary variable p at the algebraic level from Equation (4.10), we find

$$\begin{bmatrix} \gamma B_{pu}^T M_p^{-1} B_{pu} + \mu M_u & B_{\bar{p}u}^T \\ B_{\bar{p}u} & 0 \end{bmatrix} \begin{bmatrix} u \\ \bar{p} \end{bmatrix} = \begin{bmatrix} G \\ 0 \end{bmatrix}. \tag{4.11}$$

Comparing now Equation (4.7) and Equation (4.11), the result follows. □

Using Lemma 4.1.1 and [67, Proposition 2.1], we can write,

$$\begin{aligned} \begin{bmatrix} F_{uu} & F_{u\bar{u}} & B_{pu}^T & B_{\bar{p}u}^T \\ F_{\bar{u}u} & F_{\bar{u}\bar{u}} & 0 & 0 \\ B_{pu} & 0 & 0 & 0 \\ B_{\bar{p}u} & 0 & 0 & 0 \end{bmatrix}^{-1} &= \begin{bmatrix} A_{uu} + N_{uu} + \gamma D & F_{u\bar{u}} & B_{pu}^T & B_{\bar{p}u}^T \\ F_{\bar{u}u} & F_{\bar{u}\bar{u}} & 0 & 0 \\ B_{pu} & 0 & 0 & 0 \\ B_{\bar{p}u} & 0 & 0 & 0 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} A_{uu} + N_{uu} & F_{u\bar{u}} & B_{pu}^T & B_{\bar{p}u}^T \\ F_{\bar{u}u} & F_{\bar{u}\bar{u}} & 0 & 0 \\ B_{pu} & 0 & 0 & 0 \\ B_{\bar{p}u} & 0 & 0 & 0 \end{bmatrix}^{-1} + \gamma \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & M_p^{-1} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \end{aligned}$$

Defining

$$S_\gamma = - \begin{bmatrix} B_{pu} & 0 \\ B_{\bar{p}u} & 0 \end{bmatrix} \begin{bmatrix} F_{uu} & F_{u\bar{u}} \\ F_{\bar{u}u} & F_{\bar{u}\bar{u}} \end{bmatrix}^{-1} \begin{bmatrix} B_{pu}^T & B_{\bar{p}u}^T \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} S_{\gamma,pp} & S_{\gamma,\bar{p}p} \\ S_{\gamma,p\bar{p}} & S_{\gamma,\bar{p}\bar{p}} \end{bmatrix},$$

and

$$S = - \begin{bmatrix} B_{pu} & 0 \\ B_{\bar{p}u} & 0 \end{bmatrix} \begin{bmatrix} A_{uu} + N_{uu} & F_{u\bar{u}} \\ F_{\bar{u}u} & F_{\bar{u}\bar{u}} \end{bmatrix}^{-1} \begin{bmatrix} B_{pu}^T & B_{\bar{p}u}^T \\ 0 & 0 \end{bmatrix},$$

and using [103, Theorem 2.1(i)], we have

$$S_\gamma^{-1} = S^{-1} + \gamma \begin{bmatrix} M_p^{-1} & 0 \\ 0 & 0 \end{bmatrix}.$$

Now, consider the following:

$$S_\gamma^{-1} = S^{-1} + \gamma \begin{bmatrix} M_p^{-1} & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} (S^{-1})_{pp} & (S^{-1})_{p\bar{p}} \\ (S^{-1})_{\bar{p}p} & (S^{-1})_{\bar{p}\bar{p}} \end{bmatrix} + \gamma \begin{bmatrix} M_p^{-1} & 0 \\ 0 & 0 \end{bmatrix} \tag{4.12}$$

$$= \sqrt{\gamma} \begin{bmatrix} M_p^{-1/2} & 0 \\ 0 & I \end{bmatrix} \left(\begin{bmatrix} \gamma^{-1} M_p^{1/2} (S^{-1})_{pp} M_p^{1/2} & \gamma^{-1/2} M_p^{1/2} (S^{-1})_{p\bar{p}} \\ (S^{-1})_{\bar{p}p} \gamma^{-1/2} M_p^{1/2} & (S^{-1})_{\bar{p}\bar{p}} \end{bmatrix} + \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \right) \sqrt{\gamma} \begin{bmatrix} M_p^{-1/2} & 0 \\ 0 & I \end{bmatrix}. \tag{4.13}$$

Assuming γ is large, we now approximate S_γ^{-1} by:

$$S_\gamma^{-1} \approx \sqrt{\gamma}\begin{bmatrix} M_p^{-1/2} & 0 \\ 0 & I \end{bmatrix}\left(\begin{bmatrix} 0 & 0 \\ 0 & (S^{-1})_{\bar{p}\bar{p}} \end{bmatrix} + \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\right)\sqrt{\gamma}\begin{bmatrix} M_p^{-1/2} & 0 \\ 0 & I \end{bmatrix}$$
$$= \begin{bmatrix} \gamma M_p^{-1} & 0 \\ 0 & (S^{-1})_{\bar{p}\bar{p}} \end{bmatrix} = \begin{bmatrix} \gamma M_p^{-1} & 0 \\ 0 & (S_\gamma^{-1})_{\bar{p}\bar{p}} \end{bmatrix} \approx \begin{bmatrix} \gamma M_p^{-1} & 0 \\ 0 & (S_{\gamma,\bar{p}\bar{p}})^{-1} \end{bmatrix}.$$

The last equality is due to the first equality in Equation (4.12) and [103, Theorem 2.1(i)], and the last approximation is valid as $\gamma \to \infty$. Therefore, we have

$$S_\gamma \approx \hat{S}_\gamma = \begin{bmatrix} \gamma^{-1} M_p & 0 \\ 0 & S_{\gamma,\bar{p}\bar{p}} \end{bmatrix}.$$

There are two advantages to the approximation $\hat{S}_\gamma$ over S_γ. First, it decouples the blocks of the Schur complement, hence the application of $\hat{S}_\gamma$ is much cheaper than the application of S_γ. Second, both M_p and $S_{\gamma,\bar{p}\bar{p}}$ are easily invertible; M_p is a block diagonal matrix with small and fixed-size blocks and $S_{\gamma,\bar{p}\bar{p}}$ can be inverted using standard algebraic multigrid methods since it is a Laplacian-like matrix for large γ, see [39, Section 3.2].

With these results, we introduce our preconditioner for the full linear system Equation (2.34) as follows:

$$\mathcal{P}_{GD} := \begin{bmatrix} F_{uu} & F_{u\bar{u}} & 0 & 0 \\ F_{\bar{u}u} & F_{\bar{u}\bar{u}} & 0 & 0 \\ B_{pu} & 0 & \gamma^{-1} M_p & 0 \\ B_{\bar{p}u} & 0 & 0 & S_{\gamma,\bar{p}\bar{p}} \end{bmatrix}. \tag{4.14}$$

The preconditioner for the reduced problem Equation (2.35) can then be obtained from Equation (4.14) resulting in:

$$\bar{\mathcal{P}}_{GD1} := \begin{bmatrix} F_{\bar{u}\bar{u}} - F_{\bar{u}u}F_{uu}^{-1}F_{u\bar{u}} & 0 & 0 \\ -B_{pu}F_{uu}^{-1}F_{u\bar{u}} & -B_{pu}F_{uu}^{-1}B_{pu}^T & 0 \\ -B_{\bar{p}u}F_{uu}^{-1}F_{u\bar{u}} & 0 & -B_{\bar{p}u}F_{uu}^{-1}B_{\bar{p}u}^T \end{bmatrix}.$$

Here, we used Lemma 4.0.1, but with $-B_{pu}F_{uu}^{-1}B_{pu}^T$ instead of $\gamma^{-1}M_p$. Note that $-B_{pu}F_{uu}^{-1}B_{pu}^T = S_{\gamma,pp}$ and $-B_{\bar{p}u}F_{uu}^{-1}B_{\bar{p}u}^T = S_{\gamma,\bar{p}\bar{p}}$. We make this choice because it improves the convergence

of GMRES by virtue of $\text{diag}(S_{\gamma,pp}, S_{\gamma,\bar{p}\bar{p}})$ being a better approximation to S_γ than $\hat{S}_\gamma$. This preconditioner was used in Chapter 2.

For $\bar{\mathcal{P}}_{GD1}$ to be a practical preconditioner, we need a suitable solver to "invert" $F_{\bar{u}\bar{u}} - F_{\bar{u}u}F_{uu}^{-1}F_{u\bar{u}}$ when $\gamma \neq 0$. Significant effort has been spent for other discretizations on that front as discussed in Section 1.4.2. In Chapter 2, we used AIR to implement the action of the inverse, and rather than stopping at a fixed number of multigrid cycles, we iterated until a residual tolerance is reached, see for example Figure 2.2. We chose this approach since fine-tuning the number of fixed iterations for each problem is at best a hassle and at worst a fool's errand due to the large numerical null-space introduced by γD when γ is large. However, using the same null-space argument, we can approximate $F_{uu}^{-1} \approx (A_{uu} + N_{uu})^{-1}$ and since $F_{\bar{u}\bar{u}} - F_{\bar{u}u}(A_{uu} + N_{uu})^{-1}F_{u\bar{u}}$ is simply the hybridization of the vector advection-diffusion equations, we know that a fixed number of AIR iterations will be optimal. Therefore, with the choice of $A_{uu} + N_{uu}$, AIR is an efficient and practical solver for $F_{\bar{u}\bar{u}} - F_{\bar{u}u}(A_{uu} + N_{uu})^{-1}F_{u\bar{u}}$ and it can be used as a part of the preconditioner:

$$\bar{\mathcal{P}}_{GD2} := \begin{bmatrix} F_{\bar{u}\bar{u}} - F_{\bar{u}u}(A_{uu} + N_{uu})^{-1}F_{u\bar{u}} & 0 & 0 \\ -B_{pu}F_{uu}^{-1}F_{u\bar{u}} & -B_{pu}F_{uu}^{-1}B_{pu}^T & 0 \\ -B_{\bar{p}u}F_{uu}^{-1}F_{u\bar{u}} & 0 & -B_{\bar{p}u}F_{uu}^{-1}B_{\bar{p}u}^T \end{bmatrix}. \tag{4.15}$$

4.2 Pressure convection-diffusion preconditioners for HDG

In this section, we present an alternative to the grad-div preconditioner Equation (4.15). The preconditioner in this section is based on pressure convection-diffusion preconditioners, see Chapter 1 for the detailed literature review. For this, consider Equation (2.34) with $\gamma = 0$, i.e.,

$$\begin{bmatrix} A_{uu} + N_{uu} & F_{u\bar{u}} & B_{pu}^T & B_{\bar{p}u}^T \\ F_{\bar{u}u} & F_{\bar{u}\bar{u}} & 0 & 0 \\ B_{pu} & 0 & 0 & 0 \\ B_{\bar{p}u} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ \bar{u} \\ p \\ \bar{p} \end{bmatrix} = \begin{bmatrix} L \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{4.16}$$

Similar to the grad-div preconditioners presented in the previous section, we want to find a good approximation to the pressure Schur complement of the system matrix in Equa-

tion (4.16), i.e., for

$$S = -\begin{bmatrix} B_{pu} & 0 \\ B_{\bar{p}u} & 0 \end{bmatrix} \begin{bmatrix} A_{uu} + N_{uu} & F_{u\bar{u}} \\ F_{\bar{u}u} & F_{\bar{u}\bar{u}} \end{bmatrix}^{-1} \begin{bmatrix} B_{pu}^T & B_{\bar{p}u}^T \\ 0 & 0 \end{bmatrix}.$$

For this purpose, we modify the PCD preconditioner of [51] and extend it to HDG discretizations of the Navier–Stokes equations. First, we give the HDG discretization of the scalar advection-diffusion problem as it is necessary in the construction of the preconditioner.

In conservative form, the scalar advection-diffusion equation is given by

$$\nabla \cdot (\vec{c}p - \nu \nabla p) = f \qquad \text{in } \Omega, \tag{4.17a}$$
$$(-\zeta p\vec{c} + \nu \nabla p) \cdot n = 0 \qquad \text{on } \Gamma_N, \tag{4.17b}$$
$$p = g_D \qquad \text{on } \Gamma_D, \tag{4.17c}$$

where $\nu > 0$ is constant, and $\zeta = 0$ if $\vec{c} \cdot n \geq 0$, otherwise $\zeta = 1$, $\partial\Omega = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$. Assuming $\nabla \cdot \vec{c} = 0$, we get the non-conservative form of the equation:

$$\vec{c} \cdot \nabla p - \nu \Delta p = f \text{ in } \Omega.$$

Similar to the discretization of the Navier–Stokes equations, we define $\mathcal{T} = \{K\}$ as the tessellation of the domain Ω into non-overlapping simplices K. ∂K, n, and h_K denote, respectively, the boundary, the outward unit normal, and the characteristic length of an element K and we set $h = \max_{K \in \mathcal{T}} h_K$. The set of all facets contained in the mesh is denoted by $\mathcal{F} = \{F\}$ while Γ^0 denotes the union of all facets.

Let $P_l(D)$ denote the space of polynomials of degree $l \geq 1$ on a domain D. We then introduce the following discontinuous finite element spaces for the pressure on $\mathcal{T}$ and the restriction of the pressure to Γ^0:

$$Q_h := \left\{ q_h \in L^2(\Omega),\ q_h \in P_{k-1}(K)\ \forall K \in \mathcal{T} \right\}, \tag{4.18a}$$
$$\bar{Q}_h := \left\{ \bar{q}_h \in L^2(\mathcal{F}),\ \bar{q}_h \in P_k(F)\ \forall F \in \mathcal{F}, \bar{q}_h = g_D \text{ on } \partial\Omega \right\}. \tag{4.18b}$$

Furthermore, let $\boldsymbol{Q}_h = Q_h \times \bar{Q}_h$ and $\boldsymbol{p}_h = (p_h, \bar{p}_h) \in \boldsymbol{Q}_h$, and introduce the bilinear forms

$$a_h^p(\boldsymbol{p}_h, \boldsymbol{q}_h) := \sum_{K\in\mathcal{T}} \int_K \nu \nabla p_h \cdot \nabla q_h \,\mathrm{d}x + \sum_{K\in\mathcal{T}} \int_{\partial K} \frac{\alpha\nu}{h_K}(p_h - \bar{p}_h)(q_h - \bar{q}_h)\,\mathrm{d}s \tag{4.19a}$$
$$- \sum_{K\in\mathcal{T}} \int_{\partial K} \left[\nu(p_h - \bar{p}_h)\nabla q_h \cdot n + \nu(q_h - \bar{q}_h)\nabla p_h \cdot n\right] \mathrm{d}s,$$
$$o_h^p(\vec{c}; \boldsymbol{p}_h, \boldsymbol{q}_h) := -\sum_{K\in\mathcal{T}} \int_K \vec{c} p_h \cdot \nabla q_h \,\mathrm{d}x + \sum_{K\in\mathcal{T}} \int_{\partial K} \tfrac{1}{2}\vec{c}\cdot n(p_h + \bar{p}_h)\cdot(q_h - \bar{q}_h)\,\mathrm{d}s \tag{4.19b}$$
$$+ \sum_{K\in\mathcal{T}} \int_{\partial K} \tfrac{1}{2}|\vec{c}\cdot n|\,(p_h - \bar{p}_h)\cdot(q_h - \bar{q}_h)\,\mathrm{d}s.$$

The HDG discretization of Equation (4.17) is given by: given $f \in L^2(\Omega)$, $\nu > 0$ and $\vec{c}$ with $\nabla \cdot \vec{c} = 0$, find $\boldsymbol{p}_h \in \boldsymbol{Q}_h$ such that

$$B^p(\boldsymbol{p}_h, \boldsymbol{q}_h) = \sum_{K\in\mathcal{T}} \int_K f \cdot v_h \,\mathrm{d}x \quad \forall \boldsymbol{q}_h \in \boldsymbol{Q}_h, \tag{4.20}$$

where

$$B^p(\boldsymbol{p}_h, \boldsymbol{q}_h) = a_h^p(\boldsymbol{p}_h, \boldsymbol{q}_h) + o_h^p(\vec{c}; \boldsymbol{p}_h, \boldsymbol{q}_h). \tag{4.21}$$

The resulting linear system is of the block form

$$F_p \begin{bmatrix} p \\ \bar{p} \end{bmatrix} = \begin{bmatrix} F_{pp} & F_{p\bar{p}} \\ F_{\bar{p}p} & F_{\bar{p}\bar{p}} \end{bmatrix} \begin{bmatrix} p \\ \bar{p} \end{bmatrix} = \begin{bmatrix} L \\ 0 \end{bmatrix}. \tag{4.22}$$

If $\vec{c} = 0$ then we use the notation

$$A_p \begin{bmatrix} p \\ \bar{p} \end{bmatrix} = \begin{bmatrix} A_{pp} & A_{p\bar{p}} \\ A_{p\bar{p}}^T & A_{\bar{p}\bar{p}} \end{bmatrix} \begin{bmatrix} p \\ \bar{p} \end{bmatrix} = \begin{bmatrix} L \\ 0 \end{bmatrix}, \tag{4.23}$$

to distinguish the advection-diffusion problem from the Poisson problem and to emphasize symmetry of the Poisson problem. Now, we can follow the same procedure as [51] and use the commutator

$$\mathcal{E} = \nabla \cdot (-\nu\Delta + \vec{w}\cdot\nabla) - (-\nu\Delta + \vec{w}\cdot\nabla)_p \nabla\cdot, \tag{4.24}$$

where $(-\nu\Delta + \vec{w}\cdot\nabla)_p =: \mathcal{L}_p$ is the convection-diffusion operator in the pressure space. The discrete analogue to this commutator will be called $\mathcal{E}_h$. We next derive the expression

for the discrete commutator. For this, we define the following mesh-dependent norm on $\boldsymbol{Q}_h$:

$$|||\boldsymbol{q}_h|||_p^2 := \|q_h\|_\Omega^2 + \|\bar{q}_h\|_p^2 \qquad \text{where} \qquad \|\bar{q}_h\|_p^2 := \sum_{K\in\mathcal{T}} h_K \|\bar{q}_h\|_{\partial K}^2 . \tag{4.25}$$

Now we define the element and facet pressure mass matrices, $M \in \mathbb{R}^{n_p\times n_p}$ and $\bar{M} \in \mathbb{R}^{\bar{n}_p\times \bar{n}_p}$, respectively, as

$$|||\boldsymbol{q}_h|||_p^2 = \|\mathbf{q}\|_{\mathcal{M}}^2 = \langle Mq, q\rangle + \langle \bar{M}\bar{q}, \bar{q}\rangle, \tag{4.26}$$

where $\mathcal{M} := \mathrm{bdiag}(M, \bar{M})$.

Proposition 4.2.1. *Take b_h as it is defined in Equation (2.28d). Let $\mathcal{D}_h$ denote the discrete divergence operator defined on $\boldsymbol{V}_h$ by*

$$(\mathcal{D}_h\boldsymbol{u}_h, \boldsymbol{p}_h) = b_h(\boldsymbol{p}_h, \boldsymbol{u}_h) \qquad \forall \boldsymbol{p}_h \in \boldsymbol{Q}_h,$$

with the condition $\mathcal{D}_h\boldsymbol{u}_h \in \boldsymbol{Q}_h$. Then $\mathcal{D}_h$ has the matrix representation $\mathcal{M}^{-1}B$, where $\mathcal{M}$ is as defined in Equation (4.26) and B is as defined in Equation (2.32).

Proof. Define $\boldsymbol{q}_h := \mathcal{D}_h\boldsymbol{u}_h$. Expanding $\boldsymbol{q}_h$, $\boldsymbol{p}_h$ in terms of the basis $\{\psi_i\}$ of $\boldsymbol{Q}_h$ and $\boldsymbol{u}_h$ in terms of the basis $\{\vec{\phi}_i\}$ of $\boldsymbol{V}_h$, we obtain the discrete problem,

$$\mathcal{M}\mathbf{q} = B\mathbf{u}.$$

The result follows from the definition $\boldsymbol{q}_h = \mathcal{D}_h\boldsymbol{u}_h$ and the observation that $\boldsymbol{q}_h = \sum_i \mathbf{q}_i\psi_i$. □

The following two propositions can be proven similarly to Proposition 4.2.1. Therefore, we omit their proofs.

Proposition 4.2.2. *Let a_h^p and o_h^p be as defined in Equation (4.19). Let $\mathcal{F}_h^p$ denote the discrete convection-diffusion operator defined on $\boldsymbol{Q}_h$ by*

$$(\mathcal{F}_h^p\boldsymbol{p}_h, \boldsymbol{q}_h) = a_h^p(\boldsymbol{p}_h, \boldsymbol{q}_h) + o_h^p(w; \boldsymbol{p}_h, \boldsymbol{q}_h) \qquad \forall \boldsymbol{q}_h \in \boldsymbol{Q}_h,$$

with the condition $\mathcal{F}_h^p\boldsymbol{p}_h \in \boldsymbol{Q}_h$. Then $\mathcal{F}_h^p$ has the matrix representation $\mathcal{M}^{-1}F_p$, where $\mathcal{M}$ is as defined in Equation (4.26) and F_p is as defined in Equation (4.22).

Proposition 4.2.3. *Let a_h and o_h be as defined in Equation (2.28). Let $\mathcal{F}_h$ denote the discrete convection-diffusion operator defined on $\boldsymbol{V}_h$ by*

$$(\mathcal{F}_h\boldsymbol{u}_h, \boldsymbol{v}_h) = a_h(\boldsymbol{u}_h, \boldsymbol{v}_h) + o_h(w; \boldsymbol{u}_h, \boldsymbol{v}_h) \qquad \forall \boldsymbol{v}_h \in \boldsymbol{V}_h,$$

with the condition $\mathcal{F}_h\boldsymbol{u}_h \in \boldsymbol{V}_h$. Then $\mathcal{F}_h$ has the matrix representation $\mathcal{M}_u^{-1}F$, where $\mathcal{M}_u$ is the mass matrix defined on the combined velocity space $\mathbf{V}_h$ similar to $\mathcal{M}$, and $F = A + N + \gamma D$, see Equations (2.29) to (2.31).

Using Propositions 4.2.1 to 4.2.3, we can express the discrete commutator as the matrix representation of $(\mathcal{E}\boldsymbol{u}_h, \boldsymbol{q}_h) = ([\nabla \cdot (-\nu\Delta + \vec{w} \cdot \nabla) - (-\nu\Delta + \vec{w} \cdot \nabla)_p \nabla \cdot]\boldsymbol{u}_h, \boldsymbol{q}_h)$ as

$$\mathcal{E}_h = (\mathcal{M}^{-1}B)(M_u^{-1}F) - (\mathcal{M}^{-1}F_p)(\mathcal{M}^{-1}B). \tag{4.27}$$

If $\|\mathcal{E}_h\|$ is small in some sense (or, equivalently, if $\|\mathcal{E}\|$ is small), we can approximate $(\mathcal{M}^{-1}B)(M_u^{-1}F)$ by $(\mathcal{M}^{-1}F_p)(\mathcal{M}^{-1}B)$. By multiplying this expression with $\mathcal{M}F_p^{-1}\mathcal{M}$ from the left and with $F^{-1}B^T$ from the right, we obtain the following approximation to the Schur complement,

$$BF^{-1}B^T \approx \mathcal{M}F_p^{-1}BM_u^{-1}B^T. \tag{4.28}$$

We *conjecture* that the HDG discretization we use is reverse inf-sup stable, i.e., there exists a constant $\beta_{p,2}$, independent of the mesh size h, such that for all $\boldsymbol{q}_h \in \boldsymbol{Q}_h$

$$\beta_{p,2} \leq \sup_{\boldsymbol{v}_h \in \boldsymbol{V}_h} \frac{b_h(\boldsymbol{q}_h, \boldsymbol{v}_h)}{|||\boldsymbol{v}_h|||_{v*} |||\boldsymbol{q}_h|||_{p*}},$$

where the norms are defined as

$$|||\boldsymbol{v}_h|||_{v*} = \sum_{K \in \mathcal{T}} \|v_h\|_K^2 + \sum_{K \in \mathcal{T}} h_K \|\bar{v}_h\|_{\partial K}^2,$$
$$|||\boldsymbol{q}_h|||_{p*} = \sum_{K \in \mathcal{T}} \|\nabla q_h\|_K^2 + \sum_{K \in \mathcal{T}} \alpha h_K^{-1} \|q_h - \bar{q}_h\|_{\partial K}^2.$$

A consequence of this reverse inf-sup condition is that the HDG diffusion matrix on the pressure space A_p (Equation (4.23)) can be shown to be a good approximation to $BM_u^{-1}B^T$. Therefore, we replace $BM_u^{-1}B^T$ by A_p which simplifies Equation (4.28) and reduces the cost of the preconditioner:

$$BF^{-1}B^T \approx \mathcal{M}F_p^{-1}A_p.$$

As discussed in the literature review (Section 1.4.1), the boundary conditions of the convection-diffusion problem on the pressure space affects the performance of the PCD preconditioner significantly. When constructing the matrices A_p and F_p, special care must be taken. According to the current literature, the proper way of setting the respective scalar problems on the pressure space requires enforcing *natural* boundary conditions on the boundaries where a Dirichlet boundary condition is enforced for the Navier–Stokes equations and by enforcing homogeneous Dirichlet (*essential*) boundary conditions along Neumann and outflow boundaries of the Navier–Stokes problem. While this is also true in

our experience (see Section 4.3), we furthermore observed that using this strategy for the Navier–Stokes problems with Dirichlet conditions everywhere on the boundary (which is equivalent to the *do-nothing* strategy) may cause divergence. In such cases (for example, Section 4.3.2), we set the boundary conditions for the pressure problems as homogenous Dirichlet everywhere, similar to the Navier–Stokes problem, to obtain a working preconditioner.

To conclude this section, we propose the following HDG variant of the PCD preconditioner

$$\mathcal{P}_{PCD} = \begin{bmatrix} F_{uu} & F_{u\bar{u}} & 0 & 0 \\ F_{\bar{u}u} & F_{\bar{u}\bar{u}} & 0 & 0 \\ B_{pu} & 0 & & \\ B_{\bar{p}u} & 0 & & M_S \end{bmatrix}, \tag{4.29}$$

where the approximation M_S to the Schur complement S is chosen as

$$M_S = \mathcal{M} F_p^{-1} A_p, \tag{4.30}$$

with no boundary conditions prescribed on the scalar advection-diffusion and diffusion problems. Using Lemma 4.0.1 once more, we obtain the following preconditioner for the reduced problem Equation (2.35) with $\gamma = 0$:

$$\bar{\mathcal{P}}_{PCD} = \begin{bmatrix} F_{\bar{u}\bar{u}} - F_{\bar{u}u} F_{uu}^{-1} F_{u\bar{u}} & 0 & 0 \\ -B_{pu} F_{uu}^{-1} F_{u\bar{u}} & & \\ -B_{\bar{p}u} F_{uu}^{-1} F_{u\bar{u}} & & M_S \end{bmatrix}. \tag{4.31}$$

4.3 Numerical Tests

The following numerical tests will be used to demonstrate the efficiency of our preconditioners to solve Equation (2.35), i.e., the statically condensed HDG discretization of the Navier–Stokes problem Equation (1.2), in terms of number of iterations. In all runs, we use FGMRES without restart as our iterative solver with stopping criteria on the absolute tolerance $\|r_k\| < 10^{-11}$, the relative tolerance $\|r_k\|/\|r_0\| < 10^{-8}$ and the number of iterations $k < 1000$. The zero initial guess is used every time the linear solver is called. Also, we want to note that the number of Picard iterations required for convergence are dependent on the viscosity and the problem, but not the mesh size. This is similar to observations of [66, 70]. Hence, the cost of the solution process is determined by the complexity of the subsolvers necessary in the application of the preconditioner and the number of preconditioned

GMRES iterations. The cost of the application of the preconditioner depends on the mesh size ($O(n)$ to $O(n^3)$ depending on the choice of subsolvers). Therefore, in this section, by demonstrating that the number of GMRES iterations to convergence is independent of the mesh size, we show that our proposed preconditioner is an optimal preconditioner with respect to the mesh size h in the sense that the total solution cost is a linear function of the number of DoFs $N \approx h^{-d}$ with d being the dimensionality of the problem.

4.3.1 Flow over a Backward Facing Step

In our first test problem, we consider a fast moving fluid in a rectangular pipe with a sudden expansion. The domain of this problem is given by $\Omega = (-1, 5) \times (-1, 1) \setminus [-1, 0] \times [-1, 0]$. A Dirichlet boundary condition with parallel parabolic profile $\vec{u} = [4y(1-y), 0]^T$ is enforced on the inflow boundary $\partial\Omega_{D_{in}} = \{-1\} \times [0, 1]$ and the homogeneous Neumann boundary condition $\nu\frac{\partial\vec{u}}{\partial\vec{n}} - \vec{n}p = 0$ is imposed along the outflow boundary $\partial\Omega_{D_{out}} = \{5\} \times (-1, 1)$. Finally, the no-flow boundary condition $\vec{u} = [0, 0]^T$ is imposed on the walls $\partial\Omega_{D_{wall}} = \partial\Omega \setminus (\partial\Omega_{D_{in}} \bigcup \partial\Omega_{D_{out}})$. The source term is chosen as $\vec{f} = 0$. We plot the velocity and pressure solutions to this problem for reference in Figure 4.1 when $\nu = 1/100$.

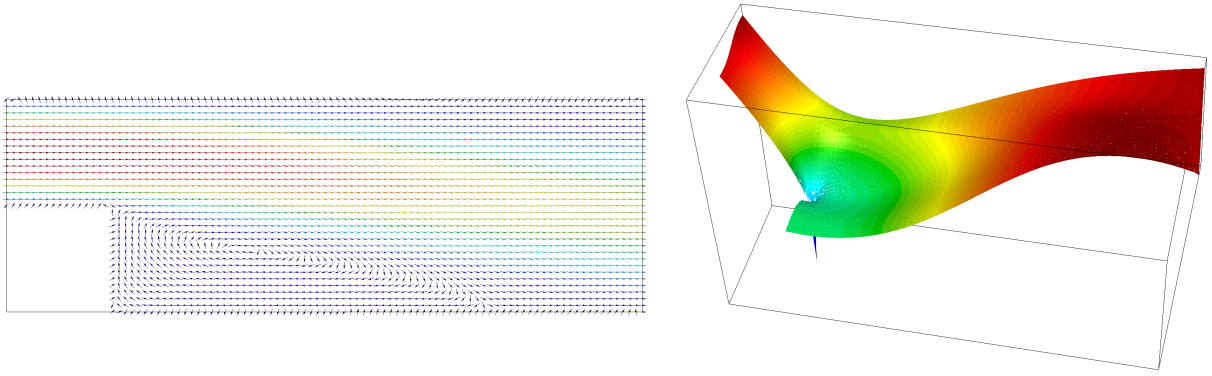


Figure 4.1: The velocity and pressure solutions to the backward facing step problem described in Section 4.3.1.

Note that if we take L equal to the height of the outflow boundary and $U = 1$, the maximum inflow speed, then the viscosity parameter must be $\nu > 1/1000$ for the steady solution to this problem to be stable [51]. In Table 4.1, we present the number of GMRES iterations required to reach the stopping criteria set at the beginning of Section 4.3. Note that the performance of the grad-div preconditioners depend on the choice of γ. We found

that the performance can be quite sensitive with respect to the value of γ. In Table 4.1, we present the results for each ν with γ chosen (by trial and error) such that the iteration counts are minimal.

For the grad-div preconditioner $\bar{P}_{GD1}$, we see h-optimal behaviour for $\nu = 1/5$ and $\nu = 1/50$. For $\nu = 1/100$, the iteration count increases slightly with each level of refinement, however, the increase does not seem to be proportional to the mesh size h.

For the PCD preconditioner $\bar{P}_{PCD}$, first, we see that the number of iterations increase slightly as the viscosity ν decreases. This observation is in line with the existing literature; see [51] for the standard Taylor–Hood discretization of the Navier–Stokes problem. Second, for all tested values of the viscosity ν, we observe h-optimal behaviour. Moreover, the number of GMRES iterations to convergence decreases as the mesh is refined. We expect that this is because M_S in Equation (4.30) approximates the pressure Schur complement better with each new level of refinement.

Table 4.1: The number of GMRES iterations required to reach a relative tolerance of 10^{-8} averaged over the number of iterations of the Picard solver for the problem described in Section 4.3.1 for different values of ν.

	$\bar{\mathcal{P}}_{GD1}$			$\bar{\mathcal{P}}_{PCD}$		
#DOFs	$\nu = 1/5$	1/50	1/100	$\nu = 1/5$	1/50	1/100
	$\gamma = 2$	1/2	1/2			
1,596	29	40.0	46.0	93	128.8	155.0
6,096	30	45.1	57.0	85	108.5	123.7
23,808	30	47.1	67.8	80	88.5	98.7
94,080	29	48.1	73.8	77	76.3	82.7

4.3.2 Lid-driven Cavity Flow

We next consider a leaky version of the lid-driven cavity flow problem. The domain is a square centred at the origin: $\Omega = (-1, 1)^2$. We enforce the Dirichlet boundary condition $\vec{u} = [1, 0]^T$ on $\partial\Omega_{D_{lid}} = [-1, 1] \times \{1\}$ and the no-flow boundary condition $\vec{u} = [0, 0]^T$ on the rest of the boundary $\partial\Omega_{D_{wall}} = \partial\Omega \setminus \partial\Omega_{D_{lid}}$. The source term is chosen as $\vec{f} = 0$. We plot the velocity and pressure solutions to this problem for reference in Figure 4.2 when $\nu = 1/500$.

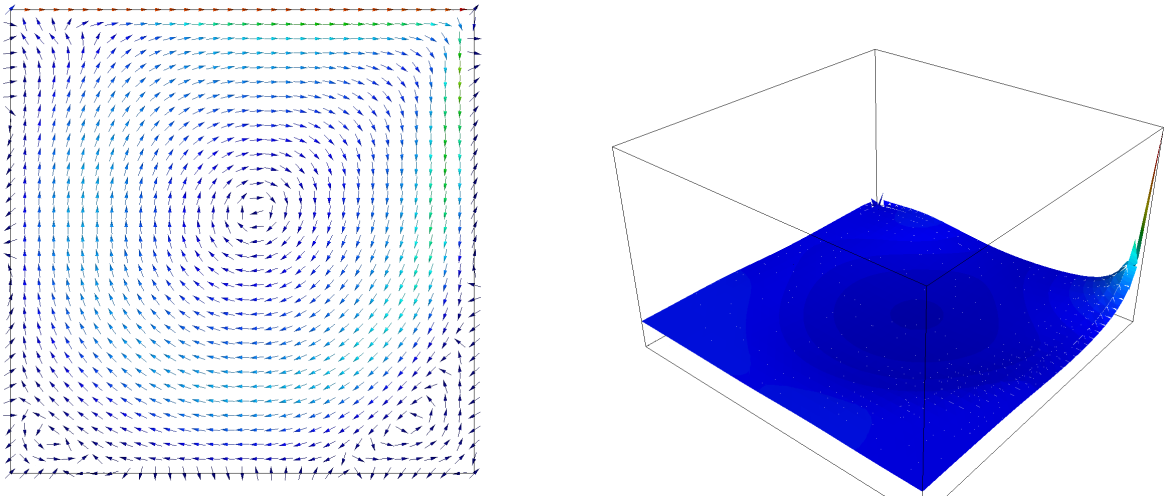


Figure 4.2: The velocity and pressure solutions to the lid-driven cavity problem described in Section 4.3.2.

The steady flow solution in a two-dimensional cavity is stable only for larger values of viscosity. For this reason, we consider $1/500 \leq \nu < 1$ to guarantee that the steady solution we find is physically meaningful. In Table 4.2, we present the number of GMRES iterations required to reach the stopping criteria set in Section 4.3. Similar to the backward facing step problem (Section 4.3.1), the results for $\bar{P}_{GD1}$ are presented for each ν together with the "best" value of γ which we found by trial-and-error.

For the grad-div preconditioner $\bar{P}_{GD1}$, we again see h-optimal behaviour for larger values of viscosity: $\nu = 1/5$ and $\nu = 1/50$. For $\nu = 1/500$, similar to the backward facing step case, we see a small increase in the number of iterations required for convergence as the mesh is refined.

For this problem, FGMRES combined with the PCD preconditioner $\bar{P}_{PCD}$ converges independently of the mesh size h for the tested values of viscosity. However, as the viscosity decreases, the number of iterations to convergence increases.

4.4 Conclusion

In this chapter, we introduced two novel preconditioners for the HDG discretization of the Navier–Stokes problem. The challenge of obtaining efficient preconditioners for the

Table 4.2: The number of GMRES iterations required to reach a relative tolerance of 10^{-8} averaged over the number of iterations of the Picard solver for the problem described in Section 4.3.2 for different values of ν.

	$\bar{\mathcal{P}}_{GD1}$			$\bar{\mathcal{P}}_{PCD}$		
#DOFs	$\nu = 1/5$	$1/50$	$1/500$	$\nu = 1/5$	$1/50$	$1/500$
	$\gamma = 1/4$	$1/20$	$1/20$			
2,256	29.0	29.1	71.8	80.0	99.0	287.0
8,736	28.0	29.1	86.5	76.0	90.9	264.1
34,368	27.0	28.1	94.3	75.0	85.1	212.8
136,320	26.0	27.1	99.1	73.0	82.8	191.5

statically condensed problem (Equation (2.35)) is solved using Lemma 4.0.1, and we numerically tested the resulting preconditioners: $\bar{P}_{GD1}$ and $\bar{P}_{PCD}$. We observed that both these preconditioners are h-robust and only mildly dependent on Re.

For each problem and each value of the viscosity ν, we found that there are values γ such that $\bar{\mathcal{P}}_{GD1}$ is h-robust and the number of GMRES iterations to convergence is minimal. The function determining these γ values does not seem to be a simple function of ν. We suspect that the optimal choice of γ will also be dependent on the domain and the velocity field (rather than the characteristic speed). We also found that choosing γ large deteriorates the convergence in terms of iterations. However, we expect that if the mesh can be refined to a point close to the asymptotic limit, the resulting preconditioner would be h and Re robust, albeit less practical than choosing γ heuristically for each problem and each ν.

Finally, from our simulations, we observed that if the Navier–Stokes problem has Dirichlet boundary conditions prescribed everywhere on the boundary of the domain, building the preconditioner $\bar{P}_{PCD}$ according to conventional wisdom (i.e., the "do-nothing" strategy) results in a divergent preconditioner. We solved this issue by enforcing homogeneous Dirichlet boundary conditions everywhere on the boundary for the pressure subproblems. The lack of convergence is to be investigated as future work. If the Navier–Stokes problem has both natural and essential boundary conditions prescribed, then the strategy of imposing boundary conditions described in [145] works for our problem as well.

Chapter 5

Conclusions

In this thesis, we presented our work on different building blocks for preconditioning of HDG discretizations of the incompressible Navier–Stokes equations. In this chapter, we summarize this work and present recommendations for future work.

5.1 Summary

In Chapter 2, we developed a framework to analyze 2-by-2 block preconditioners combined with Krylov subspace methods. Assuming that one of the blocks is inverted exactly, we showed that a good approximation to the Schur complement is fundamental in obtaining an effective 2-by-2 block preconditioner. Moreover, the theory in Chapter 2 shows that block LDU preconditioning offers minimal improvement over block triangular preconditioning in terms of iterations while having larger computational cost. Block diagonal preconditioners are computationally cheaper than block triangular preconditioners, but the number of iterations to convergence suffer. Hence, if preserving symmetry is not a concern, block triangular preconditioners should be preferred over block diagonal and block LDU preconditioners. These theoretical results are confirmed numerically, hence their predictive power has practical application. Furthermore, we numerically showed that the assumption of availability of the exact inverse of one of the blocks can be relaxed.

In Chapter 3, we concentrated on solving an HDG discretization of a time-dependent advection-diffusion problem on potentially moving domains. Our interest in this problem stems from the fact that "inverting" the momentum block of the linearized Navier–Stokes problem is equivalent to solving a decoupled vector advection-diffusion equation. Hence,

an efficient solver for this problem is crucial in developing a good preconditioner for the time-dependent Navier–Stokes equations. We investigated AIR as preconditioner for this problem, since it has been shown to be effective in solving DG discretizations of time-dependent advection-diffusion problems on fixed domains. We showed, theoretically, that AIR is also effective for HDG discretizations of advection-diffusion problems and that static condensation will not degrade the efficiency of AIR. We furthermore showed, by numerical tests, that AIR is a more efficient and scalable solver/preconditioner in an all-at-once approach to solving space-time formulations of advection-diffusion problems than when used in a slab-by-slab approach. We hypothesize that this is true because combining AIR with the all-at-once approach allows coarsening along space-time characteristics which is a property not enjoyed by other approaches in solving these problems.

Lastly, in Chapter 4, we presented two novel preconditioners for an HDG discretization of the stationary Navier–Stokes equations. These preconditioners are based on two state-of-the-art preconditioners, namely grad-div and pressure convection-diffusion preconditioners. In both cases, the objective is to find an approximation to the pressure Schur complement. First, we showed that if we can find a good preconditioner for the full problem, we can directly obtain a good preconditioner for the reduced problem through static condensation (Lemma 4.0.1). Next, we extended the grad-div and PCD preconditioners to HDG discretizations of the Navier–Stokes problem. For grad-div preconditioners, a consistent grad-div term is added to the momentum equation to aid in finding an approximation to the pressure Schur complement. In our case, we showed that the discrete form of this grad-div term can be factorized into very simple and already available matrices, which leads to a simple approximation of the pressure Schur complement. Finally, we considered the PCD preconditioner. This preconditioner is based on assuming small commutators in approximating the pressure Schur complement. We extended this framework to HDG discretizations. Our numerical results show that these preconditioners are h-optimal and mildly dependent on the Reynolds number of the problem.

5.2 Future work

In this thesis we considered different aspects of preconditioning for an HDG discretization of the Navier–Stokes equations. However, preconditioning for discretizations of the Navier–Stokes equations is still far from being a solved problem. Here I will discuss a few topics which are a direct extension of this thesis.

First, in Chapter 4, we derived grad-div and PCD preconditioners for HDG discretizations of the stationary Navier–Stokes problem. Numerical examples showed that for mod-

erate Reynolds numbers these are effective preconditioners, however, a theoretical analysis of these preconditioners is still missing. Of special interest is to rigorously show whether these preconditioners are h and Re robust.

The final goal, however, is a preconditioner for space-time HDG discretizations of the time-dependent Navier–Stokes equations on moving domains. Two major components for an effective preconditioner for such a problem is a good approximation to the (pressure) Schur complement of the discretization (see Chapter 2) and an effective solver for the (velocity) momentum block. For the latter, we already made a big step forward by demonstrating in Chapter 3 the effectiveness of AIR as a preconditioner for space-time HDG discretizations of time-dependent advection-diffusion equations on moving domains. The former, i.e., a good approximation to the (pressure) Schur complement of the discretization remains a major challenge. This is because it is not clear if the results from Chapter 2 for the stationary problem can directly be extended to the time-dependent problem.

Acknowledgments

This research was enabled in part by the support provided by Sharcnet (https://www.sharcnet.ca/) and Compute Canada (https://www.computecanada.ca). We are furthermore grateful for the computing resources provided by the Math Faculty Computing Facility at the University of Waterloo (https://uwaterloo.ca/math-faculty-computing-facility/).

Letter of copyright permission

RE: Permission to use material in doctoral thesis

Kelly Thomas <Thomas@siam.org>
Thu 2021-02-25 2:03 PM
To: Abdullah Ali Sivas <aasivas@uwaterloo.ca>

Dear Mr. Sivas:

SIAM is happy to give permission to reuse material from the articles mentioned below. Please acknowledge the original sources, using the complete bibliographic information for the published article.

Sincerely,

Kelly Thomas
Managing Editor
Society for Industrial and Applied Mathematics
3600 Market Street - 6th Floor
Philadelphia, PA 19104
thomas@siam.org / (215) 382-9800 ext. 387

From: Abdullah Ali Sivas [mailto:aasivas@uwaterloo.ca]
Sent: Wednesday, February 24, 2021 1:46 PM
To: Kelly Thomas <Thomas@siam.org>
Subject: Permission to use material in doctoral thesis

WARNING: This email originated from outside of the SIAM organization.

Dear Ms. Thomas,

I am writing to request permission to use material from the SIAM publication

Southworth, B. S., Sivas, A. A., & Rhebergen, S. (2020). On Fixed-Point, Krylov, and 2*2 Block Preconditioners for Nonsymmetric Problems. SIAM Journal on Matrix Analysis and Applications, 41(2), 871-900.

and the manuscript (under review with manuscript number M137510)

Sivas, A. A., Southworth, B. S., & Rhebergen, S. AIR algebraic multigrid for a space-time hybridizable discontinuous Galerkin discretization of advection(-diffusion)

in my doctoral thesis at the University of Waterloo (Canada). I am an author of both of the articles and SIAM holds the copyright for the first item. I am not sure about the copyright status of the second item, but I want to make sure. I plan to reuse text, figures and tables from the aforementioned papers.

At my university, theses are submitted to UWSpace (uwspace.uwaterloo.ca) which allows open access to the material.

If consent is given to use the above material, I will acknowledge the SIAM copyright for the above articles in my thesis according to my university guidelines (can be found under "Content previously published" at https://uwaterloo.ca/library/uwspace/thesis-submission-guide/third-party-content-use-and-specialized-content-submission and "Letter of copyright permission" at https://uwaterloo.ca/graduate-studies-postdoctoral-affairs/current-students/thesis/thesis-formatting).

Best regards,
Abdullah Ali Sivas

References

[1] *hypre: High Performance Preconditioners.* http://www.llnl.gov/casc/hypre, 2020.

[2] *MFEM: Modular finite element methods.* mfem.org, 2020.

[3] V. R. Ambati and O. Bokhove, *Space-time discontinuous Galerkin discretization of rotating shallow water equations*, J. Comput. Phys., 225 (2007), pp. 1233–1261.

[4] S. F. Ashby, T. A. Manteuffel, and P. E. Saylor, *A taxonomy for conjugate gradient methods*, SIAM J. Numer. Anal., 27 (1990), pp. 1542–1568.

[5] Z. Z. Bai, *Structured preconditioners for nonsingular matrices of block two-by-two structures*, Math. Comput., 75 (2006), pp. 791–815.

[6] Z. Z. Bai and M. K. Ng, *On inexact preconditioners for nonsymmetric matrices*, SIAM J. Sci. Comput., 26 (2005), pp. 1710–1724.

[7] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, *PETSc users manual*, Tech. Rep. ANL-95/11 - Revision 3.7, Argonne National Laboratory, 2016.

[8] R. E. Bank, J. W. Wan, and Z. Qu, *Kernel preserving multigrid methods for convection-diffusion equations*, SIAM Journal on Matrix Analysis and Applications, 27 (2006), pp. 1150–1171.

[9] B. Beckermann, S. A. Goreinov, and E. E. Tyrtyshnikov, *Some remarks on the Elman estimate for GMRES*, SIAM J. Matrix Anal. A., 27 (2005), pp. 772–778.

[10] M. Benzi, *Some uses of the field of values in numerical analysis*, B. Unione Mat. Ital., (2020), pp. 1–19.

[11] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), p. 1–137.

[12] M. Benzi and M. Olshanskii, *An augmented Lagrangian-based approach to the Oseen problem*, SIAM J. Sci. Comput., 28 (2006), pp. 2095–2113.

[13] M. Benzi and M. A. Olshanskii, *Field-of-values convergence analysis of augmented Lagrangian preconditioners for the linearized Navier–Stokes problem*, SIAM J. Numer. Anal., 49 (2011), pp. 770–788.

[14] M. Benzi, M. A. Olshanskii, and Z. Wang, *Modified augmented Lagrangian preconditioners for the incompressible Navier–Stokes equations*, Int. J. Numer Meth. Fl., 66 (2011), pp. 486–508.

[15] M. Benzi and Z. Wang, *Analysis of augmented Lagrangian-based preconditioners for the steady incompressible Navier–Stokes equations*, SIAM J. Sci. Comput., 33 (2011), pp. 2761–2784.

[16] ——, *A parallel implementation of the modified augmented Lagrangian preconditioner for the incompressible Navier–Stokes equations*, Numer. Algorithms, 64 (2013), pp. 73–84.

[17] M. Benzi and A. J. Wathen, *Some preconditioning techniques for saddle point problems*, in Model order reduction: theory, research aspects and applications, Springer, 2008, pp. 195–211.

[18] A. Bienz, W. D. Gropp, and L. N. Olson, *Node aware sparse matrix–vector multiplication*, J. Parallel Distr. Com., 130 (2019), pp. 166–178.

[19] ——, *Reducing communication in algebraic multigrid with multi-step node aware communication*, Int. J. High Perform. C., (2020), p. 1094342020925535.

[20] D. Boffi, F. Brezzi, M. Fortin, et al., *Mixed finite element methods and applications*, vol. 44, Springer, 2013.

[21] L. Botti and D. A. D. Pietro, *p-Multilevel preconditioners for HHO discretizations of the Stokes equations with static condensation*, 2020.

[22] S. Brenner and R. Scott, *The mathematical theory of finite element methods*, vol. 15, Springer Science & Business Media, 2007.

[23] M. Brezina, T. A. Manteuffel, S. F. McCormick, J. W. Ruge, and G. Sanders, *Towards Adaptive Smoothed Aggregation (αSA) for Nonsymmetric Problems*, SIAM J. Sci. Comput., 32 (2010), pp. 14–39.

[24] F. Brezzi, L. D. Marini, and E. Süli, *Discontinuous Galerkin methods for first-order hyperbolic problems*, Math. Mod. Meth. Appl. S., 14 (2004), pp. 1893–1903.

[25] E. Burman, *A posteriori error estimation for interior penalty finite element approximations of the advection-reaction equation*, SIAM J. Numer. Anal., 47 (2009), pp. 3584–3607.

[26] J. O. Campos, R. W. dos Santos, J. Sundnes, and B. M. Rocha, *Preconditioned augmented Lagrangian formulation for nearly incompressible cardiac mechanics*, Int. J. Numer. Meth. Bio., 34 (2018), p. e2948. e2948 cnm.2948.

[27] B. Cockburn, O. Dubois, J. Gopalakrishnan, and S. Tan, *Multigrid for an HDG method*, IMA J. Numer. Anal., 34 (2014), pp. 1386–1425.

[28] B. Cockburn, J. Gopalakrishnan, and R. Lazarov, *Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems*, SIAM J. Numer. Anal., 47 (2009), pp. 1319–1365.

[29] B. Cockburn and J. Shen, *A hybridizable discontinuous Galerkin method for the p-Laplacian*, SIAM J. Sci. Comput., 38 (2016), pp. A545–A566.

[30] M. Crouzeix and C. Palencia, *The numerical range is a $(1+\sqrt{2})$-spectral set*, SIAM J. Matrix Anal. A., 38 (2017), pp. 649–655.

[31] E. C. Cyr, J. N. Shadid, and R. S. Tuminaro, *Stabilization and scalable block preconditioning for the Navier–Stokes equations*, J. Comput. Phys., 231 (2012), pp. 345–363.

[32] X. Dai and Y. Maday, *Stable parareal in time method for first-and second-order hyperbolic systems*, SIAM J. Sci. Comput., 35 (2013), pp. A52–A78.

[33] J. de Frutos, B. García-Archilla, V. John, and J. Novo, *An adaptive SUPG method for evolutionary convection-diffusion equations*, Comput. Methods Appl. Mech. Engrg., 273 (2014), pp. 219–237.

[34] A. C. de Niet and F. W. Wubs, *Two preconditioners for saddle point problems in fluid flows*, Int. J. Numer Meth. Fl., 54 (2007), pp. 355–377.

[35] H. De Sterck, R. D. Falgout, S. Friedhoff, O. A. Krzysik, and S. P. MacLachlan, *Optimizing MGRIT and Parareal coarse-grid operators for linear advection*, arXiv preprint arXiv:1910.03726, (2019).

[36] H. De Sterck, S. Friedhoff, A. J. Howse, and S. P. MacLachlan, *Convergence analysis for parallel-in-time solution of hyperbolic systems*, Numer. Linear Algebra Appl., 27 (2020), p. e2271.

[37] B. M. DeBlois, *Linearizing convection terms in the navier-stokes equations*, Computer Methods in Applied Mechanics and Engineering, 143 (1997), pp. 289–297.

[38] L. T. Diosady, *Domain decomposition preconditioners for higher-order discontinuous Galerkin discretizations*, PhD thesis, Massachusetts Institute of Technology, 2011.

[39] V. Dobrev, T. Kolev, C. S. Lee, V. Tomov, and P. S. Vassilevski, *Algebraic hybridization and static condensation with application to scalable H(div) preconditioning*, SIAM J. Sci. Comput., 41 (2019), pp. B425–B447.

[40] H. S. Dollar, N. I. Gould, M. Stoll, and A. J. Wathen, *Preconditioning saddle-point systems with applications in optimization*, SIAM J. Sci. Comput., 32 (2010), pp. 249–270.

[41] M. Eiermann and O. G. Ernst, *Geometric aspects of the theory of Krylov subspace methods*, Acta Numer., 10 (2001), pp. 251–312.

[42] S. C. Eisenstat, H. C. Elman, and M. H. Schultz, *Variational iterative methods for nonsymmetric systems of linear equations*, SIAM J. Numer. Anal., 20 (1983), pp. 345–357.

[43] H. Elman, V. E. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro, *Block preconditioners based on approximate commutators*, SIAM J. Sci. Comput., 27 (2006), pp. 1651–1668.

[44] H. Elman, V. E. Howle, J. Shadid, D. Silvester, and R. Tuminaro, *Least squares preconditioners for stabilized discretizations of the Navier–Stokes equations*, SIAM J. Sci. Comput., 30 (2008), pp. 290–311.

[45] H. Elman and D. Silvester, *Fast nonsymmetric iterations and preconditioning for Navier–Stokes equations*, SIAM J. Sci. Comput., 17 (1996), pp. 33–46.

[46] H. Elman and D. Silvester, *Fast nonsymmetric iterations and preconditioning for Navier–Stokes equations*, SIAM J. Sci. Comput., 17 (1996), pp. 33–46.

[47] H. Elman and R. Tuminaro, *Boundary conditions in approximate commutator preconditioners for the Navier–Stokes equations*, Elec. Trans. Numer. Math, 35 (2009), pp. 257–280.

[48] H. C. Elman, *Iterative methods for large, sparse, nonsymmetric systems of linear equations*, PhD thesis, Yale University New Haven, Conn, 1982.

[49] H. C. Elman, *Preconditioning for the steady-state Navier–Stokes equations with low viscosity*, SIAM J. Sci. Comput., 20 (1999), pp. 1299–1316.

[50] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*, Oxford University Press, USA, 2nd ed., 2014.

[51] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*, Numerical Mathematics and Scientific Computation, 2014.

[52] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. Maclachlan, and J. B. Schroder, *Parallel time integration with multigrid*, SIAM J. Sci. Comput., 36 (2014), pp. C635–C661.

[53] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. MacLachlan, J. B. Schroder, and S. Vandewalle, *Multigrid methods with space-time concurrency*, Comput. Vis. Sci., 18 (2017), pp. 123–143.

[54] R. D. Falgout and U. M. Yang, *hypre: A library of high performance preconditioners*, European Conference on Parallel Processing, 2331 LNCS (2002), pp. 632–641.

[55] P. Farrell, P. A. G. Orozco, and E. Süli, *Finite element appoximation and augmented Lagrangian preconditioning for anisothermal implicitly-constituted non-Newtonian flow*, 2020.

[56] P. E. Farrell and P. A. Gazca-Orozco, *An augmented Lagrangian preconditioner for implicitly constituted non-Newtonian incompressible flow*, SIAM J. Sci. Comput., 42 (2020), pp. B1329–B1349.

[57] P. E. Farrell, L. Mitchell, L. R. Scott, and F. Wechsung, *A Reynolds-robust preconditioner for the Scott–Vogelius discretization of the stationary incompressible Navier–Stokes equations*, 2021.

[58] P. E. Farrell, L. Mitchell, and F. Wechsung, *An augmented Lagrangian preconditioner for the 3D stationary incompressible Navier–Stokes equations at high Reynolds number*, SIAM J. Sci. Comput., 41 (2019), pp. A3073–A3096.

[59] B. Fischer, A. Ramage, D. J. Silvester, and A. J. Wathen, *Minimum residual methods for augmented systems*, BIT, 38 (1998), pp. 527–543.

[60] M. Fortin and R. Glowinski, *Augmented Lagrangian methods in quadratic programming*, in Studies in Mathematics and its Applications, vol. 15, Elsevier, 1983, pp. 1–46.

[61] M. Franciolini, K. J. Fidkowski, and A. Crivellini, *Efficient discontinuous galerkin implementations and preconditioners for implicit unsteady compressible flow simulations*, Comput. Fluids, 203 (2020), p. 104542.

[62] S. Friedhoff and B. S. Southworth, *On "optimal" h-independent convergence of parareal and multigrid-reduction-in-time using Runge-Kutta time integration*, Numer. Linear Algebra Appl., (2020), p. e2301.

[63] G. Fu, *Uniform auxiliary space preconditioning for HDG methods for elliptic operators with a parameter dependent low order term*, 2020.

[64] M. J. Gander and S. Hajian, *Analysis of Schwarz methods for a hybridizable discontinuous Galerkin discretization*, SIAM J. Numer. Anal., 53 (2015), pp. 573–597.

[65] M. J. Gander and M. Neumuller, *Analysis of a new space-time parallel multigrid algorithm for parabolic problems*, SIAM J. Sci. Comput., 38 (2016), pp. A2173–A2208.

[66] G. Gantner, A. Haberl, D. Praetorius, and B. Stiftner, *Rate optimal adaptive fem with inexact solver for nonlinear operators*, IMA Journal of Numerical Analysis, 38 (2018), pp. 1797–1831.

[67] G. Golub and C. Greif, *On solving block-structured indefinite linear systems*, SIAM J. Sci. Comput., 24 (2003), pp. 2076–2092.

[68] J. Gopalakrishnan, M. Neumüller, and P. S. Vassilevski, *The auxiliary space preconditioner for the de Rham complex*, SIAM J. Numer. Anal., 56 (2018), pp. 3196–3218.

[69] A. Greenbaum, V. Pták, and Z. Strakoš, *Any nonincreasing convergence curve is possible for GMRES*, SIAM J. Matrix Anal. A., 17 (1996), pp. 465–469.

[70] A. Haberl, D. Praetorius, S. Schimanko, and M. Vohralík, *Convergence and quasi-optimal cost of adaptive algorithms for nonlinear operators including iterative linearization and algebraic solver*, Numerische Mathematik, 147 (2021), pp. 679–725.

[71] J. Hanophy, B. S. Southworth, R. Li, T. Manteuffel, and J. Morel, *Parallel approximate ideal restriction multigrid for solving the SN transport equations*, Nucl. Sci. Eng., (2020), pp. 1–20.

[72] T. Haut, P. Maginot, V. Tomov, B. Southworth, T. Brunner, and T. Bailey, *An efficient sweep-based solver for the SN equations on high-order meshes*, Nucl. Sci. Eng., 193 (2019), pp. 746–759.

[73] X. He and C. Vuik, *Efficient and robust Schur complement approximations in the augmented Lagrangian preconditioner for the incompressible laminar flows*, J. Comput. Phys., 408 (2020), p. 109286.

[74] X. He, C. Vuik, and C. Klaij, *Block-preconditioners for the incompressible Navier–Stokes equations discretized by a finite volume method*, J. Numer. Math., 25 (2017), pp. 89–105.

[75] X. He, C. Vuik, and C. Klaij, *Combining the augmented Lagrangian preconditioner with the simple Schur complement approximation*, SIAM J. Sci. Comput., 40 (2018), pp. A1362–A1385.

[76] Y. He, S. Rhebergen, and H. D. Sterck, *Local Fourier analysis of multigrid for hybridized and embedded discontinuous Galerkin methods*, 2020.

[77] T. Heister and G. Rapin, *Efficient augmented Lagrangian-type preconditioning for the Oseen problem using grad-div stabilization*, Int. J. Numer Meth. Fl., 71 (2013), pp. 118–134.

[78] R. A. Horn, R. A. Horn, and C. R. Johnson, *Topics in matrix analysis*, Cambridge university press, 1994.

[79] R. A. Horn and C. R. Johnson, *Matrix analysis*, Cambridge University Press, second ed., 2013.

[80] G. Horton and S. Vandewalle, *A space-time multigrid method for parabolic partial differential equations*, SIAM J. Sci. Comput., 16 (1995), pp. 848–864.

[81] T. L. Horváth and S. Rhebergen, *A locally conservative and energy-stable finite element method for the Navier–Stokes problem on time-dependent domains*, Int. J. Numer. Meth. Fluids, 89 (2019), pp. 519–532.

[82] ——, *An exactly mass conserving space-time embedded-hybridized discontinuous Galerkin method for the Navier–Stokes equations on moving domains*, J. Comput. Phys., 417 (2020).

[83] B. Hübner, E. Walhorn, and D. Dinkler, *A monolithic approach to fluid-structure interaction using space-time finite elements*, Comput. Methods Appl. Mech. Engrg., 193 (2004), pp. 2087–2104.

[84] I. Huismann, J. Stiller, and J. Fröhlich, *Linearizing the hybridizable discontinuous Galerkin method: A linearly scaling operator*, 2020.

[85] I. C. F. Ipsen, *A note on preconditioning nonsymmetric matrices*, SIAM J. Sci. Comput., 23 (2001), pp. 1050–1051.

[86] P. Jamet, *Galerkin-type approximations which are discontinuous in time for parabolic equations in a variable domain*, SIAM J. Numer. Anal., 15 (1978), pp. 912–928.

[87] D. Kay and D. Loghin, *A Green's function preconditioner for the steady-state Navier–Stokes equations*, 1999.

[88] D. Kay, D. Loghin, and A. Wathen, *A preconditioner for the steady-state Navier–Stokes equations*, SIAM J. Sci. Comput., 24 (2002), pp. 237–256.

[89] K. L. Kirk and S. Rhebergen, *Analysis of a pressure-robust hybridized discontinuous Galerkin method for the stationary Navier–Stokes equations*, J. Sci. Comput., 2 (2019), pp. 881–897.

[90] K. L. A. Kirk, T. L. Horvath, A. Cesmelioglu, and S. Rhebergen, *Analysis of a space-time hybridizable discontinuous Galerkin method for the advection-diffusion problem on time-dependent domains*, SIAM J. Numer. Anal., 57 (2029), pp. 1677–1696.

[91] A. Klawonn, *Block-triangular preconditioners for saddle point problems with a penalty term*, SIAM J. Sci. Comput., 19 (1998), pp. 172–184.

[92] A. Klawonn and L. F. Pavarino, *A comparison of overlapping Schwarz methods and block preconditioners for saddle point problems*, Numer. Linear. Algebr., 7 (2000), pp. 1–25.

[93] A. Klawonn and G. Starke, *Block triangular preconditioners for nonsymmetric saddle point problems: field-of-values analysis*, Numer. Math., 81 (1999), pp. 577–594.

[94] P. Krzyzanowski, *On block preconditioners for nonsymmetric saddle point problems*, SIAM J. Sci. Comput., 23 (2001), pp. 157–169.

[95] W. Layton and W. Lenferink, *Two-level picard and modified picard methods for the navier-stokes equations*, Applied mathematics and computation, 69 (1995), pp. 263–274.

[96] S. Le Borne and L. G. Rebholz, *Preconditioning sparse grad-div/augmented Lagrangian stabilized saddle point systems*, Comput. Vis. Sci., 16 (2013), p. 259–269.

[97] M. Lesoinne and C. Farhat, *Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations*, Comput. Methods. Appl. Mech. Engrg., 134 (1996), pp. 71–90.

[98] B. Li and X. Xie, *BPX preconditioner for nonstandard finite element methods for diffusion problems*, SIAM J. Numer. Anal., 54 (2016), pp. 1147–1168.

[99] J. Liesen and P. Tichy, *The field of values bound on ideal GMRES*, 2020.

[100] J. L. Lions, Y. Maday, and G. Turinici, *Résolution d'EDP par un schéma en temps pararéel*, C. R. Acad. Sci. Paris Sér. I Math, 332 (2001), pp. 661–668.

[101] D. Loghin and A. Wathen, *Analysis of preconditioners for saddle-point problems*, SIAM J. Sci. Comput., 25 (2004), pp. 2029–2049.

[102] P. Lu, A. Rupp, and G. Kanschat, *HMG – Homogeneous multigrid for HDG*, 2020.

[103] T. T. Lu and S. H. Shiou, *Inverses of 2 x 2 block matrices*, Comput. Math. Appl., 43 (2002), pp. 119–129.

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

[105] C. Makridakis and R. H. Nochetto, *A posteriori error analysis for higher order dissipative methods for evolution problems*, Numer. Math. (Heidelb), 104 (2006), pp. 489–514.

[106] T. A. Manteuffel, *The Tchebychev iteration for nonsymmetric linear systems*, Numer. Math., (1977), pp. 307–327.

[107] T. A. Manteuffel, S. Munzenmaier, J. Ruge, and B. S. Southworth, *Nonsymmetric reduction-based algebraic multigrid*, SIAM J. Sci. Comput., 41 (2019), pp. S242–S268.

[108] T. A. Manteuffel, L. N. Olson, J. B. Schroder, and B. S. Southworth, *A root-node based algebraic multigrid method*, SIAM J. Sci. Comput., 39 (2017), pp. S723–S756.

[109] T. A. Manteuffel, J. Ruge, and B. S. Southworth, *Nonsymmetric algebraic multigrid based on local approximate ideal restriction (ℓAIR)*, SIAM J. Sci. Comput., 40 (2018), pp. A4105–A4130.

[110] A. Masud and T. Hughes, *A space-time Galerkin/least-squares finite element formulation of the Navier–Stokes equations for moving domain problems*, Comput. Methods Appl. Mech. Engrg., 146 (1997), pp. 91–126.

[111] E. McDonald, S. Hon, J. Pestana, and A. J. Wathen, *Preconditioning for nonsymmetry and time-dependence*, in Domain decomposition methods in science and engineering XXIII, Springer, Cham, 2017, pp. 81–91.

[112] J. Moulin, P. Jolivet, and O. Marquet, *Augmented Lagrangian preconditioner for large-scale hydrodynamic stability analysis*, Comput. Method. Appl. M., 351 (2019), pp. 718–743.

[113] S. Muralikrishnan, T. Bui-Thanh, and J. N. Shadid, *A multilevel approach for trace system in hdg discretizations*, J. Comput. Phys., 407 (2020), p. 109240.

[114] M. F. Murphy, G. H. Golub, and A. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput., 21 (2000), pp. 1969–1972.

[115] Y. Notay, *Aggregation-based algebraic multigrid for convection-diffusion equations*, SIAM J. Sci. Comput., 34 (2012), pp. A2288–A2316.

[116] Y. Notay, *A new analysis of block preconditioners for saddle point problems*, SIAM J. Matrix Anal. A., 35 (2014), pp. 143–173.

[117] Y. Notay, *Convergence of some iterative methods for symmetric saddle point linear systems*, SIAM Journal on Matrix Analysis and Applications, 40 (2019), pp. 122–146.

[118] M. Olshanskii and A. Zhiliakov, *Recycling augmented Lagrangian preconditioner in an incompressible fluid solver*, 2020.

[119] M. A. Olshanskii and E. E. Tyrtyshnikov, *Iterative methods for linear systems: theory and applications*, SIAM, 2014.

[120] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.

[121] J. W. Pearson and A. J. Wathen, *A new approximation of the Schur complement in preconditioners for PDE-constrained optimization*, Numer. Linear. Algebr., 19 (2011), pp. 816–829.

[122] P.-O. Persson, J. Bonet, and J. Peraire, *Discontinuous Galerkin solution of the Navier–Stokes equations on deformable domains*, Comput. Methods. Appl. Mech. Engrg., 198 (2009), pp. 1585–1595.

[123] J. Pestana, *Nonstandard inner products and preconditioned iterative methods*, (2011).

[124] ——, *On the eigenvalues and eigenvectors of block triangular preconditioned block matrices*, SIAM J. Matrix Anal. A., 35 (2014), pp. 517–525.

[125] J. Pestana and A. J. Wathen, *Combination preconditioning of saddle point systems for positive definiteness*, Numer. Linear. Algebr., 20 (2012), pp. 785–808.

[126] J. Pestana and A. J. Wathen, *Natural preconditioning and iterative methods for saddle point systems*, SIAM Review, 57 (2015), pp. 71–91.

[127] T. Rees and M. Stoll, *Block-triangular preconditioners for PDE-constrained optimization*, Numer. Linear. Algebr., 17 (2010), pp. 977–996.

[128] S. Rhebergen and B. Cockburn, *A space-time hybridizable discontinuous Galerkin method for incompressible flows on deforming domains*, J. Comput. Phys., 231 (2012), pp. 4185–4204.

[129] ——, *Space-time hybridizable discontinuous Galerkin method for the advection-diffusion equation on moving and deforming meshes*, in The Courant–Friedrichs–Lewy (CFL) condition, 80 years after its discovery, C. A. de Moura and C. S. Kubrusly, eds., Birkhäuser Science, 2013, pp. 45–63.

[130] S. Rhebergen, B. Cockburn, and J. J. W. van der Vegt, *A space-time discontinuous Galerkin method for the incompressible Navier–Stokes equations*, J. Comput. Phys., 233 (2013), pp. 339–358.

[131] S. Rhebergen and G. Wells, *Analysis of a hybridized/interface stabilized finite element method for the Stokes equations*, SIAM J. Numer. Anal., 55 (2017), pp. 1982–2003.

[132] ——, *A hybridizable discontinuous Galerkin method for the Navier–Stokes equations with pointwise divergence-free velocity field*, J. Sci. Comput., 76 (2018), pp. 1484–1501.

[133] S. Rhebergen and G. N. Wells, *Preconditioning of a hybridized discontinuous Galerkin finite element method for the Stokes equations*, J. Sci. Comput., 77 (2018), pp. 1936–1952.

[134] B. Rivière, *Discontinuous Galerkin methods for solving elliptic and parabolic equations*, vol. 35 of Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, 2008.

[135] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid methods, SIAM, 1987, pp. 73–130.

[136] D. Ruprecht, *Wave propagation characteristics of Parareal*, Comput. Visual Sci., 19 (2018), pp. 1–17.

[137] Y. Saad, *A flexible inner-outer preconditioned gmres algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.

[138] Y. Saad, *Iterative methods for sparse linear systems*, vol. 82, siam, 2003.

[139] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 7 (1986), pp. 856–869.

[140] M. Sala and R. S. Tuminaro, *A new Petrov–Galerkin smoothed aggregation preconditioner for nonsymmetric linear systems*, SIAM J. Sci. Comput., 31 (2008), pp. 143–166.

[141] J. Schöberl and W. Zulehner, *Symmetric indefinite preconditioners for saddle point problems with applications to pde-constrained optimization problems*, SIAM J. Matrix Anal. A., 29 (2007), pp. 752–773.

[142] D. Schötzau and C. Schwab, *Time discretization of parabolic problems by the hp-version of the discontinuous Galerkin finite element method*, SIAM J. Numer. Anal., 38 (2000), pp. 837–875.

[143] J. B. Schroder, *Smoothed aggregation solvers for anisotropic diffusion*, Numer. Linear Algebra Appl., 19 (2012), pp. 296–312.

[144] C. Siefert and E. de Sturler, *Preconditioners for generalized saddle-point problems*, SIAM J. Numer. Anal., 44 (2006), pp. 1275–1296.

[145] D. Silvester, H. Elman, D. Kay, and A. Wathen, *Efficient preconditioning of the linearized navier–stokes equations for incompressible flow*, J. Comput. Appl. Math., 128 (2001), pp. 261 – 279. Numerical Analysis 2000. Vol. VII: Partial Differential Equations.

[146] A. A. Sivas, B. S. Southworth, and S. Rhebergen, *AIR algebraic multigrid for a space-time hybridizable discontinuous Galerkin discretization of advection(-diffusion)*, 2020.

[147] W. E. H. Sollie, O. Bokhove, and J. J. W. van der Vegt, *Space-time discontinuous Galerkin finite element method for two-fluid flows*, J. Comput. Phys., 230 (2011), pp. 789–817.

[148] B. S. Southworth, M. Holec, and T. S. Haut, *Diffusion synthetic acceleration for heterogeneous domains, compatible with voids*, Nucl. Sci. Eng., 195 (2021), pp. 119–136.

[149] B. S. Southworth and S. A. Olivier, *A note on 2×2 block-diagonal preconditioning*, (2020).

[150] B. S. Southworth, A. A. Sivas, and S. Rhebergen, *On fixed-point, Krylov, and 2×2 block preconditioners for nonsymmetric problems*, SIAM J. Matrix Anal. A., 41 (2020), pp. 871–900.

[151] G. Starke, *Field-of-values analysis of preconditioned iterative methods for nonsymmetric elliptic problems*, Numer. Math., 78 (1997), pp. 103–117.

[152] M. Tavelli and M. Dumbser, *A staggered space-time discontinuous Galerkin method for the incompressible Navier–Stokes equations on two-dimensional triangular meshes*, Comput. Fluids, 119 (2015), pp. 235–249.

[153] ——, *A staggered space-time discontinuous Galerkin method for the three-dimensional incompressible Navier–Stokes equations on unstructured tetrahedral meshes*, J. Comput. Phys., 319 (2016), pp. 294–323.

[154] C. Taylor and P. Hood, *A numerical solution of the Navier–Stokes equations using the finite element technique*, Comput. Fluids, 1 (1973), pp. 73–100.

[155] J. D. Tebbens and G. Meurant, *Prescribing the behavior of early terminating GMRES and Arnoldi iterations*, Numer. Algorithms, 65 (2013), pp. 69–90.

[156] T. E. Tezduyar, M. Behr, S. Mittal, and J. Liou, *A new strategy for finite element computations involving moving boundaries and interfaces. The deforming-spatial-domain/space-time procedure: II. Computation of free-surface flows, two-liquid flows, and flow with drifting cylinders*, Comput. Methods Appl. Mech. Engrg., 94 (1992b), pp. 353–371.

[157] T. E. Tezduyar, S. Sathe, and K. Stein, *Solution techniques for the fully discretized equations in computation of fluid-structure interactions with the space-time formulations*, Comput. Methods Appl. Mech. Engrg., 195 (2006), pp. 5743–5753.

[158] J. J. W. van der Vegt and J. J. Sudirham, *A space-time discontinuous Galerkin method for the time-dependent Oseen equations*, Appl. Numer. Math, 58 (2008), pp. 1892–1917.

[159] J. J. W. van der Vegt and H. van der Ven, *Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows: I. General formulation*, J. Comput. Phys., 182 (2002), pp. 546–585.

[160] H. van der Ven, *An adaptive multitime multigrid algorithm for time-periodic flow simulations*, J. Comput. Phys., 227 (2008), pp. 5286–5303.

[161] E. Walhorn, A. Kölke, B. Hübner, and D. Dinkler, *Fluid-structure coupling within a monolithic model involving free surface flows*, Comput. Struct., 83 (2005), pp. 2100–2111.

[162] L. Wang and P.-O. Persson, *A high-order discontinuous Galerkin method with unstructured space-time meshes for two-dimensional compressible flows on domains with large deformations*, Comput. Fluids, 118 (2015), pp. 53–68.

[163] A. Wathen, *Preconditioning and convergence in the right norm*, Int. J. Comput. Math., 84 (2007), pp. 1199–1209.

[164] T. Weinzierl and T. Köppl, *A geometric space-time multigrid algorithms for the heat equation*, Numer. Math. Theory Methods Appl., 5 (2012), pp. 110–130.

[165] G. N. Wells, *Analysis of an interface stabilized finite element method: the advection-diffusion-reaction equation*, SIAM J. Numer. Anal., 49 (2011), pp. 87–109.

[166] P. Wesseling and C. W. Oosterlee, *Geometric multigrid with applications to computational fluid dynamics*, J. Comput. Appl. Math., 128 (2001), pp. 311–334.

[167] J. A. White and R. I. Borja, *Block-preconditioned Newton–Krylov solvers for fully coupled flow and geomechanics*, Computat. Geosci., 15 (2011), p. 647.

[168] T. A. Wiesner, R. S. Tuminaro, W. A. Wall, and M. W. Gee, *Multigrid transfers for nonsymmetric systems based on Schur complements and Galerkin projections*, Numer. Linear Algebra Appl., 21 (2014), pp. 415–438.

[169] J. Xia, P. E. Farrell, and F. Wechsung, *Augmented Lagrangian preconditioners for the Oseen–Frank model of nematic and cholesteric liquid crystals*, 2020.

[170] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The finite element method: its basis and fundamentals*, Elsevier, 2013.

[171] O. C. Zienkiewicz and J. Z. Zhu, *A simple error estimator and adaptive procedure for practical engineering analysis*, Int. J. Numer. Methods Eng., 24 (1987), pp. 337–357.

[172] O. C. Zienkiewicz and J. Z. Zhu, *The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity*, Int. J. Numer. Methods Eng., 33 (1992), pp. 1365–1382.

APPENDICES

Appendix A

Approximate Ideal Restriction Algebraic Multigrid

In this chapter, we present a brief introduction to approximate ideal restriction (AIR) algebraic multigrid method as proposed in [109]. We particularly focus on the variant called local AIR (ℓAIR) as it is our choice of preconditioner. As discussed in Chapter 3, algebraic multigrid methods solve the problem $Ax = b$ by partitioning the DOFs into fine (F-) and coarse (C-) points, creating interpolation P and restriction R operators and then iterating using the formula

$$x^{(i+1)} = x^{(i)} + PA_c^{-1}Rr^{(i)},$$

where the coarse "grid" operator $A_c = RAP$, and the residual $r^{(i)} = b - Ax^{(i)}$. We can row-column permute the matrix A so that the F-points are ordered first

$$A = \begin{bmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{bmatrix}.$$

Furthermore, we can assume that C-points are interpolated and restricted using injection which simplifies the forms of P and R:

$$P = \begin{bmatrix} W \\ I_{n_c} \end{bmatrix}, \quad R = \begin{bmatrix} Z & I_{n_c} \end{bmatrix},$$

for some full-rank matrices $W \in \mathbb{R}^{n_f \times n_c}$ and $Z \in \mathbb{R}^{n_c \times n_f}$ where n_c and n_f are, respectively, the number of C- and F-points. Defining the error at the ith iteration as $e^{(i)} = x - x^{(i)}$,

we see that $r^{(i)} = Ae^{(i)}$ and that

$$\begin{aligned} e^{(i+1)} &= e^{(i)} + PA_c^{-1}RAe^{(i)} \\ &= \begin{bmatrix} e_f^{(i)} \\ e_c^{(i)} \end{bmatrix} + PA_c^{-1}RA \begin{bmatrix} e_f^{(i)} \\ e_c^{(i)} \end{bmatrix}. \end{aligned}$$

Since W is a full rank matrix, any error vector e can be decomposed as:

$$e = \begin{bmatrix} e_f \\ e_c \end{bmatrix} = \begin{bmatrix} We_c + \delta e_f \\ e_c \end{bmatrix} = \begin{bmatrix} W \\ I_{n_c} \end{bmatrix} e_c + \begin{bmatrix} \delta e_f \\ 0 \end{bmatrix} = Pe_c + \begin{bmatrix} \delta e_f \\ 0 \end{bmatrix},$$

where $e_f \in \mathbb{R}^{n_f}$, $e_c \in \mathbb{R}^{n_c}$ and $\delta e_f = e_f - We_c$ is that part of the F-point error which is not in the range of interpolation. From these relationships, we obtain:

$$\begin{aligned} e^{(i+1)} &= \begin{bmatrix} e_f^{(i)} \\ e_c^{(i)} \end{bmatrix} + PA_c^{-1}RA \begin{bmatrix} e_f^{(i)} \\ e_c^{(i)} \end{bmatrix} \\ &= \begin{bmatrix} e_f^{(i)} \\ e_c^{(i)} \end{bmatrix} + PA_c^{-1}RA \left(Pe_c^{(i)} + \begin{bmatrix} \delta e_f^{(i)} \\ 0 \end{bmatrix} \right) \\ &= \begin{bmatrix} \delta e_f^{(i)} \\ 0 \end{bmatrix} - PA_c^{-1}RA \begin{bmatrix} \delta e_f^{(i)} \\ 0 \end{bmatrix}. \end{aligned}$$

Now, we observe that if we choose $Z = -A_{cf}A_{ff}^{-1}$ then

$$RA \begin{bmatrix} \delta e_f^{(i)} \\ 0 \end{bmatrix} = 0,$$

for any $\delta e_f^{(i)}$, and the resulting restriction operator R is called the ideal restriction operator R_{ideal}. Using ideal restriction, we effectively prevent the error at F-points, $e_f^{(i)}$, from corrupting the coarse-grid problem: find v s.t. $A_c v = R_{ideal}Ae^{(i)}$, since $R_{ideal}Ae^{(i)} = R_{ideal}APe_c^{(i)}$. However, A_{ff}^{-1} is usually dense, hence it is not practical to construct R_{ideal} and an approximation is necessary. The algorithm we used in this thesis is called ℓAIR [107, 109], and the idea is to *locally* eliminate the contribution of all F-point error within a prescribed distance to the coarse-grid problem. To this end, for each i-th C-point a set of "nearby" F-points, $\mathcal{R}_i$, is chosen using a distance criterion (we used at most graph

distance 2) and a strength-of-connection criterion. Then, to determine the i-th row of Z, corresponding to the i-th C-point, we solve a set of equations

$$\sum_{k \in \mathcal{R}_i} z_{ik} a_{kj} = a_{ij} \qquad \forall j \in \mathcal{R}_i,$$

where z_{ij} and a_{ij} are the i-th row and j-th column entries of the matrices, respectively, Z and A. As a result, we need to solve a $|\mathcal{R}_i| \times |\mathcal{R}_i|$ linear system to construct each row of Z, hence the setup cost of the ℓAIR restriction operator is $\mathcal{O}(n_c)$ since the cardinality of the sets $\mathcal{R}_i$ is bounded from above by a constant independent of the C-point i and the mesh size [109]. We want to note that while choosing a higher graph distance improves the robustness and convergence of AIR when used as a preconditioner, the benefits are not worth the computational cost.

Now, we want to discuss why ℓAIR is a good preconditioner for our problem of interest. First of all, our HDG discretization guarantees that, in the absence of diffusion, each block of facet DOFs only has neighbours in the direction of the velocity field. This means that the cardinalities of the sets $\mathcal{R}_i$ are bounded from above and the bound is not large. Combinatorics of the problem becomes difficult depending on the strength of diffusion and the anisotropy of the problem; however, we have seen experimentally that ℓAIR can still be a good option for weakly diffusion dominated flows. Secondly, if the velocity field does not have cycles, then the *full* coefficient matrix can be permuted into a block-triangular form with small block-sizes, and static condensation does not change this property (Lemma 3.2.1). This property is more beneficial in a parallel setting, as each process can find such an ordering for their local problem (on-process) and use it as a part of block Gauss-Seidel relaxation, which amounts to a direct solve at the maximum cost $\mathcal{O}(N^2)$. Such a reordering can be found using an on-process depth-first-search at the cost $\mathcal{O}(N_p)$ where N_p is the number of DOFs local to the process.

A clear example of these benefits is given in Section 3.3.2. The moving internal layer problem has no diffusion, and the velocity field does not have any cycles. As a result, the resulting preconditioner is robust; looking at Figure 3.9, it can be seen that block-diagonal scaling (block-inv) combined with on-process solves has the least amount of iterations and the most robust performance since the advective component is inverted exactly under these conditions. We also see that applying block diagonal scaling is enough to improve the quality of forward Gauss-Seidel (FGS) and Jacobi relaxation techniques to a competitive level.