

# Optimal-complexity and robust multigrid methods for high-order FEM



Pablo D. Brubeck  
Balliol College  
University of Oxford

DPhil thesis

Michelmas 2022

# Acknowledgements

This thesis is dedicated to my parents. Thank you for educating me with perseverance and feeding my scientific curiosity, and thanks for all the great love and sacrifice that have sustained me up to this point. I also thank my supervisor Patrick Farrell for his patient and insightful guidance and for all the useful help and motivation. It has been a great pleasure to work with him for the past three years. Thanks to my colleagues Francis Aznaran, Gonzalo Gonzalez De Diego, and Fabian Laakmann for their comradery and helpful discussions. A special thanks to the Firedrake developers team, in particular to Lawrence Mitchell and Jack Betteridge, who laid the groundwork and kindly provided continued support for the software implementation, ultimately making this work available to the wider scientific computing community.

# Abstract

The numerical solution of elliptic PDEs is often the most computationally intensive task in large-scale continuum mechanics simulations. High-order finite element methods can efficiently exploit modern parallel hardware while offering very rapid convergence properties. As the polynomial degree is increased, the efficient solution of such PDEs becomes difficult.

This thesis develops preconditioners for high-order discretizations. We build upon the pioneering work of Pavarino, who proved in 1993 that the additive Schwarz method with vertex patches and a low-order coarse space gives a solver for symmetric and coercive problems that is robust to the polynomial degree. However, for very high polynomial degrees it is not feasible to assemble or factorize the matrices for each vertex patch, as the patch matrices contain dense blocks, which couple together all degrees of freedom within a cell. The central novelty of the preconditioners we develop is that they have optimal time and space complexity on unstructured meshes of tensor-product cells.

Our solver relies on new finite elements for the de Rham complex that enable the blocks in the stiffness matrix corresponding to the cell interiors to become diagonal for scalar PDEs or block diagonal for vector-valued PDEs. With these new elements, the patch problems are as sparse as a low-order finite difference discretization, while having a sparser Cholesky factorization. In the non-separable case, the method can be applied as a preconditioner by approximating the problem with a separable surrogate. Through the careful use of incomplete factorizations and choice of space decomposition we achieve optimal fill-in in the patch factors, ultimately allowing for optimal-complexity storage and computational cost across the setup and solution stages.

We demonstrate the approach by solving a variety of symmetric and coercive problems, including the Poisson equation, the Riesz maps of  $H(\text{curl})$  and  $H(\text{div})$ , and a  $H(\text{div})$ -conforming interior penalty discretization of linear elasticity in three dimensions at  $p = 15$ .

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Outline of the thesis . . . . .	10
<b>2</b>	<b>Multigrid solvers for the Poisson equation</b>	<b>12</b>
2.1	Continuous Galerkin formulation . . . . .	12
2.2	Fast Poisson solvers on Cartesian meshes . . . . .	14
2.2.1	The fast diagonalization method . . . . .	15
2.2.2	Sparsity-promoting discretization . . . . .	16
2.2.3	Sparse vertex-star relaxation . . . . .	19
2.2.4	Computational complexity . . . . .	19
2.3	Sparse preconditioners on unstructured meshes . . . . .	21
2.4	Numerical results . . . . .	24
<b>3</b>	<b>Multigrid solvers for linear elasticity</b>	<b>28</b>
3.1	Primal formulation of linear elasticity . . . . .	28
3.2	Mixed FEM formulations of linear elasticity . . . . .	30
3.3	Extension to interior penalty DG methods . . . . .	33
3.4	Results for mixed formulations of linear elasticity . . . . .	37
<b>4</b>	<b>Multigrid solvers for the de Rham complex</b>	<b>40</b>
4.1	Introduction . . . . .	40
4.1.1	Related work . . . . .	45
4.2	Sparsity-promoting discretization . . . . .	46
4.2.1	Exterior calculus notation and weak formulation . . . . .	46
4.2.2	Orthogonal bases for $H(\text{grad})$ and $L^2$ on the interval . . . . .	47
4.2.3	Orthogonal bases for the de Rham complex . . . . .	48
4.3	Auxiliary sparse preconditioning . . . . .	49
4.3.1	Pullbacks and finite element assembly . . . . .	50
4.3.2	Construction of sparse preconditioners . . . . .	51
4.4	Multigrid relaxation by subspace correction . . . . .	53

4.4.1	Notation . . . . .	53
4.4.2	Designing space decompositions . . . . .	54
4.4.3	Choice of space decompositions . . . . .	56
4.4.4	Statically-condensed space decompositions . . . . .	58
4.4.5	Achieving optimal fill-in . . . . .	60
4.5	Numerical experiments . . . . .	60
4.5.1	Riesz maps: robust iteration counts and optimal complexity	62
4.5.2	Piecewise-constant coefficients . . . . .	63
4.5.3	Mixed formulation of Hodge Laplacians . . . . .	65
<b>5</b>	<b>Conclusions and future work</b>	<b>68</b>
5.1	Conclusions . . . . .	68
5.2	Future work . . . . .	69
5.2.1	Krylov methods on statically-condensed problems . . . . .	69
5.2.2	DG – CG multigrid solver for $H(\text{div}) \times L^2$ -conforming elasticity on unstructured meshes . . . . .	69
5.2.3	Application to nonlinear problems . . . . .	70
<b>Bibliography</b>		<b>71</b>

# 1

## Introduction

High-order finite element discretizations are well suited to exploit modern parallel hardware architectures. They converge exponentially fast to smooth solutions and allow for matrix-free solvers that balance the ratio of floating point operations (flops) to energy-intensive data movement [55]. Unfortunately, the conditioning of the stiffness matrix is severely affected by the polynomial degree of the approximation [67]. In order to obtain practical iterative solvers, we require good preconditioners.

Optimal iterative solvers for linear systems involve work that is proportional to the application of the matrix-vector product. Thus it is desirable that the number of iterations does not grow as the number of degrees of freedom (DOFs) is increased. Solvers with this property may be obtained from a multiplicative multigrid V-cycle where the smoother consists of a domain decomposition method, such as additive Schwarz with a particular space decomposition. The multigrid algorithm is then accelerated by a Krylov subspace method, such as preconditioned conjugate gradients (PCG). The choice of space decomposition in the relaxation is crucial for robustness with respect to the cell size  $h$ , the polynomial degree  $p$ , and any parameters in the equation.

One of the cheapest relaxations, with  $\mathcal{O}(p^d)$  computational cost, is diagonal scaling, also known as point-Jacobi. This offers a multigrid solver with optimal complexity at  $p = 1$ , but not for  $p > 1$ . The condition number of the Laplacian stiffness matrix is  $\mathcal{O}(p^{4(d-1)})$ , and point-Jacobi improves the conditioning to  $\mathcal{O}(p^{2(d-1)})$  [67]. This implies that the number of PCG iterations, and therefore the number of residual evaluations, is  $\mathcal{O}(p^{d-1})$ , incurring a total cost of  $\mathcal{O}(p^{2d})$ . In order to minimize the time to solution, it is reasonable to consider more expensive relaxation

methods that converge in fewer iterations<sup>1</sup>. Ideally, we wish to balance the cost of applying the relaxation with that of updating the residual. On tensor-product elements, the latter operation can be done quickly in  $\mathcal{O}(p^{d+1})$  operations via the sum-factorization introduced by Orszag in 1980 [73]. Sum-factorization breaks down the residual evaluation into products of one-dimensional operators and diagonal scalings. This thesis only considers unstructured meshes of tensor product cells.

In 1993, Pavarino proved that the additive Schwarz method with a vertex-centered space decomposition and an additive coarse space of lowest-order ( $p = 1$ ) gives a robust solver with respect to  $h$  and  $p$  for symmetric and coercive problems [75]. This type of space decomposition is often referred to as generous overlap and is illustrated in Figure 1.1(a). We use the terminology of [38] and refer to the subdomains in this space decomposition as vertex-star patches, as they are constructed by taking all DOFs on the topological entities in the *star* of each vertex<sup>2</sup>.

The most straightforward implementation of a vertex-star solver involves the assembly and direct factorization of the  $\mathcal{O}(p^d) \times \mathcal{O}(p^d)$  patch matrices, which are dense for Lagrange basis functions. This becomes prohibitively expensive at very high polynomial degrees, with the Cholesky factorization of such a matrix requiring  $\mathcal{O}(p^{3d})$  operations. However, there exist bases for which the element matrices are sparse on affine cells, such as the hierarchical Lobatto basis functions [94]. In this basis, the stiffness matrix has a 5-point stencil in 2D, and a much larger 13-point stencil in 3D. The spectral element method (SEM) [73] employs Lagrange basis functions and a quadrature rule collocated at the DOFs, that approximates the FEM mass matrix with a diagonal lumped mass matrix whose entries are equal to the row-sum of the FEM mass matrix. On rectangular cells the SEM stiffness matrix has a  $(dp + 1)$ -point stencil. Nevertheless, the sparsity that arises in these special cases is lost when the cells are distorted or when the PDE coefficients are spatially varying.

Efficient relaxation methods that are  $p$ -robust may arise from the discretization of an auxiliary problem for which fast inversion techniques are available. For a more general approach to auxiliary space techniques, we refer to the work of Xu [98]. In our context, the underlying PDE and/or the domain can be replaced by those of a problem which is solvable by the method of separation of variables. The fast diagonalization method (FDM) [65] is a  $\mathcal{O}(p^{d+1})$  direct factorization that breaks the problem down into a sequence of one-dimensional subproblems. This

---

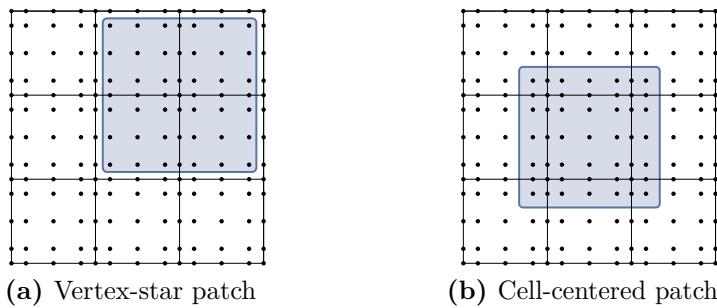
<sup>1</sup>Arguably, the cost of setting up the solver should not be the main concern in an iterative context, as it is often amortized by the cost of applying the relaxation multiple times.

<sup>2</sup>The vertex-star patch  $V_j$  includes the degrees of freedom associated with vertex  $\mathbf{v}_j$  of  $\mathcal{T}_h$  and all cells, facets, and edges adjacent to  $\mathbf{v}_j$  (the topological entities in the *star* of the vertex, a standard concept in algebraic topology [71, §2]).

approach applies to a very limited set of symmetric coercive PDEs, such as the Poisson and Helmholtz equations, and the two-dimensional  $H(\text{curl})$  and  $H(\text{div})$  Riesz maps, under very restrictive constraints on the geometry of the domain and the spatial dependence of the PDE coefficients [84, 35].

For the Poisson equation discretized on meshes with all cells Cartesian (all internal angles are right angles), the vertex-star problems can be solved directly with the FDM [96]. Huisman et al. [49] introduced a remarkably fast relaxation with  $\mathcal{O}(p^d)$  scaling on such meshes. The linear system is statically-condensed by elimination of the cell DOFs, and the reduced system on the interface is solved with  $p$ -multigrid and a restricted variant of the FDM onto the interface DOFs of a vertex-star<sup>3</sup>. Since the statically-condensed operator requires the exact inversion of the cell matrices, their approach has no obvious extension to the unstructured, non-Cartesian case.

The FDM can be applied as a relaxation by means of an auxiliary problem that is separable, but this requires a tensor-product grid discretization of the patch, which is only possible when the cells are laid out in a tensor-product grid [96]. When the cells are not Cartesian, the method of Witte et al. [96] approximates the whole patch as a single Cartesian domain and converges slowly even when the cells are slightly distorted. On general meshes, the patches may not have this structure, thus the FDM cannot be directly applied on such patches. An example of a vertex-star patch to which the FDM cannot be applied as a relaxation is shown in Figure 1.2(b).



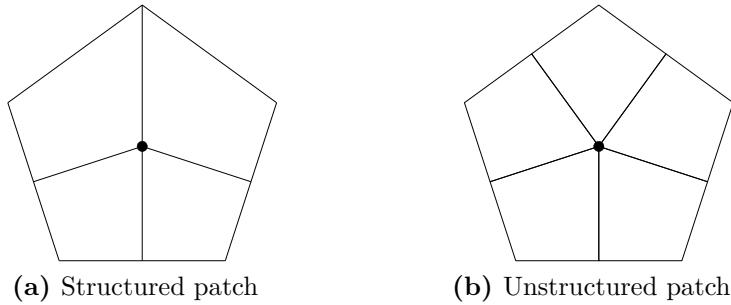
**Figure 1.1:** Subdomains for the additive Schwarz method on a regular mesh ( $p = 4$ ). In combination with a low-order coarse grid, the vertex-star patch gives a  $p$ -robust method for symmetric and coercive problems, while the cell-centered patch does not.

A popular alternative in the literature has been to use cell-centered patches with minimal overlap by including a few layers of DOFs of the neighboring cells

---

<sup>3</sup>In three dimensions the forward elimination of the cell interiors and backwards substitution steps incur a setup cost of  $12p^4$  flops, and one application of the relaxation involves  $592p^3$  flops. Given the magnitude of the  $\mathcal{O}(p^3)$  coefficient, the relaxation cost dominates over the setup and the perceived overall scaling is  $\mathcal{O}(p^3)$  for  $p \leq 48$ .

[39, 83, 93]. This can be done in such a way that every patch remains structured. This kind of space decomposition is more amenable to fast implementation, but does not give a  $p$ -robust solver. If the number of layers is fixed, then the measure of the overlap region will decrease as  $p$  is increased. Pavarino also proved that when the overlap is not generous, the rate of convergence of the additive Schwarz method will depend inversely on the overlap size [76]. To overcome this, Fischer and Lottes [64] applied a hybrid  $p$ -multigrid/Schwarz method, in the context of a Poisson problem. They implemented several levels of  $p$ -multigrid to overcome the lack of  $p$ -robustness of the cell-centered patches with minimal overlap. The use of multiple levels increases the overlap at the coarser levels with a minor impact on the overall computational cost. Cell-centered patches without overlap have also been employed for non-symmetric problems [79, 80, 30].



**Figure 1.2:** The FDM may be applied as a relaxation only on vertex-star patches that are structured, i.e., where the cells are laid out in a tensor-product grid.

Instead of replacing the vertex-star patches with cell-centered ones, the alternative low-order-refined (LOR) preconditioner [73, 28, 29, 23, 21] rediscretizes the problem on each vertex-star patch with  $p = 1$  on a GLL grid, a mesh with vertices at the DOFs of the high-order space. The theory behind this guarantees the spectral equivalence between the differential operator discretized on the two spaces [22]. Since low-order methods are naturally sparse, this approach is not constrained to Cartesian cells and can deal properly with mixed first derivatives that the FDM cannot handle. A downside of this approach is that the Cholesky factors of the patch matrices are quite dense, limiting its scalability to very high polynomial degree. Computationally advantageous approaches involve incomplete factorizations of the patch LOR matrices [77], or the use of algebraic multigrid on the global LOR operator [14].

## 1.1 Outline of the thesis

In Chapter 2 we first consider the Poisson equation. We develop a solver for the vertex-star patches arising in Pavarino's method that scales to very high polynomial degree. Our approach does not rely on a particular structure of the patch. In particular, it applies to both patches shown in Figure 1.2. The key idea to define a new finite element for  $H(\text{grad})$  on the interval, inspired by the FDM, for which the interior blocks of *both* the stiffness and mass matrices are diagonal. In multiple dimensions, we construct a  $H(\text{grad})$ -conforming element by tensor-products, obtaining sparse stiffness and mass matrices, in the Cartesian case. In particular, the total number of nonzeros is the same as that of a low-order finite difference discretization of the Laplacian. Moreover, fill-in in the Cholesky factorization is only introduced for the interface DOFs, resulting in very sparse Cholesky factors. The factorization requires  $\mathcal{O}(p^{3(d-1)})$  operations, while forward and back-substitution steps have a cost of  $\mathcal{O}(p^{2(d-1)})$  operations that is optimal for  $d \in \{2, 3\}$ , in contrast with the  $\mathcal{O}(p^{3d})$  and  $\mathcal{O}(p^{2d})$  costs of the naïve approach. In the non-Cartesian case, we approximate the form with one that is separable in the reference problem. A disadvantage is that the memory required scales like  $\mathcal{O}(p^{2(d-1)})$ , instead of the optimal  $\mathcal{O}(p^d)$  required for storing the solution. Robustness with respect to  $h$  and  $p$  follows from the spectral equivalence between the forms, and numerical experiments indicate that the approach is effective when the cells are moderately deformed.

In Chapter 3 we consider the application of our solver to linear elasticity. We demonstrate the effectiveness of our approach by applying it to a  $H(\text{div}) \times L^2$ -conforming discretization of a mixed formulation of incompressible linear elasticity. In the primal formulation, although our approach can be applied to patch problems for the individual components of displacement, we explain why it cannot be applied to the coupled vector-valued problem, which is necessary for parameter robustness in the incompressible regime. We therefore consider a mixed formulation instead. Developing a  $p$ -robust solver requires both a  $p$ -robust preconditioner and a  $p$ -robust discretization, and for the latter we choose a  $H(\text{div}) \times L^2$  conforming approach. We then extend the method to symmetric interior penalty discontinuous Galerkin discretizations, required for the displacement block of the mixed problem. We apply our relaxation to the displacement block of the incompressible elasticity system in conjunction with block-preconditioned Krylov methods.

In Chapter 4 we extend our multigrid approach to high-order FEM discretizations of the Riesz maps of the  $L^2$  de Rham complex, the canonical problems in  $H(\text{grad})$ ,  $H(\text{curl})$ ,  $H(\text{div})$ , and  $L^2$ , which frequently arise as subproblems in the construction

of fast preconditioners for more complicated problems. Here we specifically address one of the major drawbacks from the approach introduced in Chapter 2, which is the lack of optimality in the storage and setup costs. Building on the finite element introduced in Chapter 2, we devise new finite elements for each space in the de Rham complex with orthogonality properties in both the  $L^2$ - and  $H(d)$ -inner products ( $d \in \{\text{grad}, \text{curl}, \text{div}\}$ ) on the reference hexahedron. The resulting sparsity enables the fast solution of the patch problems arising in the Pavarino, Arnold–Falk–Winther, and Hiptmair space decompositions, in the separable case. In the non-separable case, the method can be applied to an auxiliary operator that is sparse by construction. With exact Cholesky factorizations of the sparse patch problems, the application complexity is optimal but the setup costs and storage are not, as in Chapter 2. We overcome this with the finer Hiptmair space decomposition for  $H(\text{curl})$  and  $H(\text{div})$ , and the use of incomplete Cholesky factorizations imposing the sparsity pattern arising from static condensation, which applies whether static condensation is used for the solver or not. This yields multigrid relaxations with time and space complexity that are both optimal in the polynomial degree, i.e., with  $\mathcal{O}(p^d)$  storage and  $\mathcal{O}(p^{d+1})$  computational cost for setup and application. We then apply our solvers on mixed formulations of the Hodge Laplacians of the  $L^2$  de Rham complex.

We end with conclusions in Chapter 5.

# 2

## Multigrid solvers for the Poisson equation<sup>1</sup>

### 2.1 Continuous Galerkin formulation

We start from the standard weak formulation of the Poisson equation. Consider a bounded domain  $\Omega \subset \mathbb{R}^d$ ,  $d \in \{1, 2, 3\}$ , and let  $\Gamma_D \subseteq \partial\Omega$  be the part of the boundary where the Dirichlet boundary condition  $u|_{\Gamma_D} = u_0$  is prescribed. The problem is to find  $u - u_0$  in  $V := H_D^1(\Omega) = \{v \in H^1(\Omega), v|_{\Gamma_D} = 0\}$  such that

$$a(v, u) = L(v) \quad \forall v \in V. \quad (2.1)$$

where after integration by parts, the bilinear form becomes

$$a(v, u) = a_\Omega(v, u) := (\nabla v, \nabla u)_\Omega, \quad L(v) := (v, f)_\Omega. \quad (2.2)$$

where  $(\cdot, \cdot)_\Omega := (\cdot, \cdot)_{L^2(\Omega)}$  is the standard  $L^2(\Omega)$ -inner product.

The standard FEM discretization employs a mesh  $\mathcal{T}_h = \{K\}$  of  $\Omega$ . In this work we consider quadrilateral and hexahedral cells, so that each cell  $K$  can be mapped with a diffeomorphism  $F_K : \hat{K} \rightarrow K$  from the reference hypercube  $\hat{K} = \hat{\mathcal{I}}^d$ , where  $\hat{\mathcal{I}} = [-1, 1]$  is the reference interval. The approximate solution  $u_h \in V_h$  is sought in the space of piecewise continuous tensor-product polynomials on each cell, i.e.,  $V_h = Q_p(\mathcal{T}_h) \cap V$ . We first define the space of basis functions on  $\hat{K}$

$$Q_p(\hat{K}) := \bigotimes_{j=1}^d P_p(\hat{\mathcal{I}}), \quad P_p(\hat{\mathcal{I}}) := \text{span} \left\{ \hat{x}^j, 0 \leq j \leq p \right\}, \quad (2.3)$$

---

<sup>1</sup>This chapter is extracted from [20], published in SIAM J. Sci. Comput.

and via composition with  $F_K^{-1}$ , we define

$$V_h := \left\{ v \in H_D^1(\Omega) : \forall K \in \mathcal{T}_h \exists \hat{v} \in Q_p(\hat{K}) \text{ s.t. } v|_K = \hat{v} \circ F_K^{-1} \right\}. \quad (2.4)$$

The discrete problem is to find  $u_h - u_0 \in V_h$  such that

$$a(v_h, u_h) = L(v_h) \quad \forall v_h \in V_h. \quad (2.5)$$

Once we fix a basis  $\{\phi_j\}_{j=1}^n$  for  $V_h$ , the approximate solution is expanded as  $u_h = \sum_{j=1}^n u_j \phi_j$ . The resulting  $n \times n$  system of linear equations is

$$A\underline{u} = \underline{f}, \quad (2.6)$$

where  $[A]_{ij} = a(\phi_i, \phi_j)$  is the stiffness matrix,  $\underline{u} = (u_1, \dots, u_n)^\top$  is the vector of coefficients, and  $\underline{f} = (L(\phi_1), \dots, L(\phi_n))^\top$  is the load vector.

We recall the standard construction of the basis  $\{\phi_j\}$  [50]. The basis is defined in terms of basis functions  $\{\hat{\phi}_j\}$  on  $\hat{K}$ . Given basis functions  $\{\hat{\phi}_j^{1D}\}_{j=0}^p$  for  $P_p(\hat{I})$ , a tensor-product basis  $\{\hat{\phi}_j\}$  for  $Q_p(\hat{K})$  can be constructed as

$$\hat{\phi}_j(\hat{\mathbf{x}}) = \prod_{k=1}^d \hat{\phi}_{j_k}^{1D}(\hat{x}_k), \quad (2.7)$$

where we have expanded  $j = (j_1, \dots, j_d) \in (0 : p)^d$  as a multi-index.

The interval basis functions are decomposed into interface and interior modes. The interface modes have nonzero support on either endpoint of  $\hat{I}$ , while the interior modes vanish at the boundary of  $\hat{I}$ . In multiple dimensions, the basis functions decompose into interior, facet, edge, and vertex modes, depending on how many 1D interface functions are multiplied together. To generate a  $C^0$  basis, we simply match the shape of individual interface modes. Hence,  $A$  will be block sparse, since  $[A]_{ij} = 0$  when  $i$  and  $j$  correspond to interior modes supported on different cells.

For the interval basis functions, one standard choice is the set of Lagrange polynomials on the Gauß–Lobatto–Legendre (GLL) nodes  $\{\hat{\xi}_i\}_{i=0}^p \subset [-1, 1]$ . These nodes are the roots of  $(1 - \hat{\xi}^2)P'_p(\hat{\xi})$ , where  $P_k(\hat{\xi})$  is the Legendre polynomial of degree  $k$ . The Lagrange polynomials  $\{\ell_j(\hat{x})\}$  satisfy  $\ell_j(\hat{\xi}_i) = \delta_{ij}$  by construction,

$$\ell_j(\hat{x}) = \prod_{k=0, k \neq j}^p \frac{\hat{x} - \hat{\xi}_k}{\hat{\xi}_j - \hat{\xi}_k}, \quad j = 0, \dots, p. \quad (2.8)$$

Another useful basis is formed by the hierarchical Lobatto polynomials  $\{l_j\}$ , which are constructed by augmenting the so-called bubble functions (integrated Legendre polynomials) with linear Lagrange functions,

$$l_j(\hat{x}) = \begin{cases} (1 - \hat{x})/2 & \text{for } j = 0, \\ (1 + \hat{x})/2 & \text{for } j = p, \\ \int_{-1}^{\hat{x}} P_j(z) dz = P_{j+1}(\hat{x}) - P_{j-1}(\hat{x}) & \text{for } j = 1, \dots, p-1. \end{cases} \quad (2.9)$$

These two choices of basis functions are plotted in Figure 2.1.

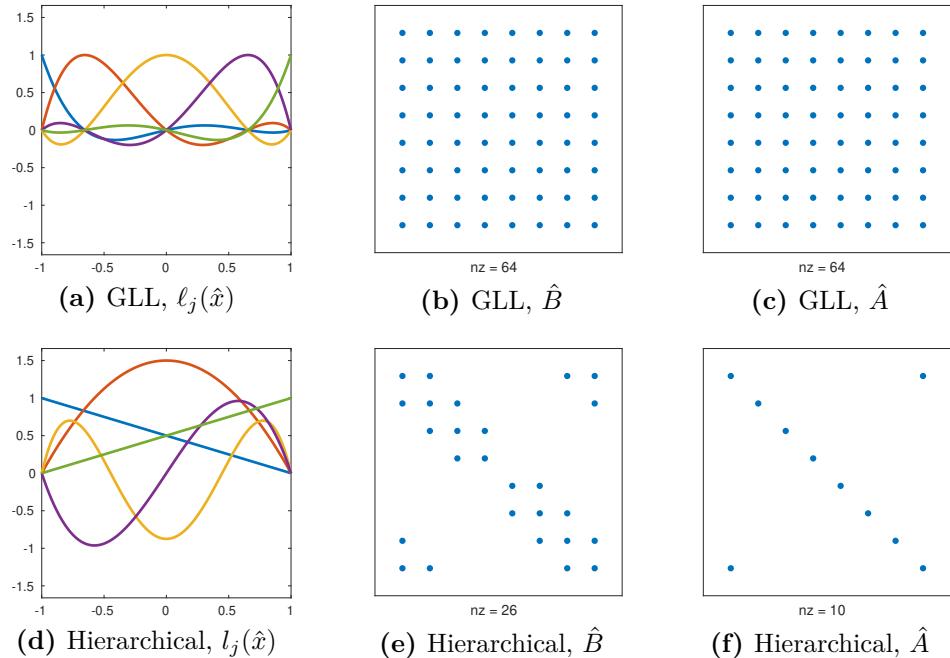
The assembly of the stiffness matrix  $A$  is described as follows. On each cell  $K \in \mathcal{T}_h$  we define the cell stiffness matrix  $A^K \in \mathbb{R}^{(p+1)^d \times (p+1)^d}$  in terms of the basis functions  $\{\phi_j^K\}$  that are supported on  $K$ , which are obtained from the reference basis functions  $\{\hat{\phi}_j\}$  via  $\phi_j^K = \hat{\phi}_j \circ F_K^{-1}$ . Then, the cell stiffness matrices are

$$[A^K]_{ij} = (\nabla \phi_i^K, \nabla \phi_j^K)_K. \quad (2.10)$$

The global stiffness matrix is then assembled via direct stiffness summation:

$$A = \sum_{K \in \mathcal{T}_h} R_K^\top A^K R_K, \quad (2.11)$$

where  $R_K \in \mathbb{R}^{(p+1)^d \times n}$  is the Boolean restriction matrix from the global DOFs to those local to cell  $K$ .



**Figure 2.1:** Plots of the interval basis functions ( $p = 4$ ) and nonzero structure of the mass and stiffness matrices on the reference interval ( $p = 7$ ).

## 2.2 Fast Poisson solvers on Cartesian meshes

If  $d = 1$  and  $F_K$  is an affine mapping, then the cell stiffness matrices are

$$[A^K]_{ij} = \frac{1}{L^K} (\hat{\phi}'_i, \hat{\phi}'_j)_{\hat{\mathcal{I}}} = \frac{1}{L^K} [\hat{A}]_{ij}. \quad (2.12)$$

Here  $\hat{A} \in \mathbb{R}^{(p+1) \times (p+1)}$  is the stiffness matrix on the reference interval, and  $L^K = |K|/|\hat{\mathcal{I}}|$ , where  $|K|$  denotes the measure of the cell  $K$ .

For  $d = \{2, 3\}$ , we will first consider the case where  $\Omega$  can be tessellated with a mesh  $\mathcal{T}_h$  consisting of Cartesian cells, i.e., the cells are rectangular quadrilaterals or hexahedra (all internal angles are right angles). In this setting, the cell stiffness matrices have the Kronecker product structure:

$$A^K = \begin{cases} \mu_1^K \hat{B} \otimes \hat{A} + \mu_2^K \hat{A} \otimes \hat{B} & d = 2, \\ \mu_1^K \hat{B} \otimes \hat{B} \otimes \hat{A} + \mu_2^K \hat{B} \otimes \hat{A} \otimes \hat{B} + \mu_3^K \hat{A} \otimes \hat{B} \otimes \hat{B} & d = 3, \end{cases} \quad (2.13)$$

where

$$[\hat{B}]_{ij} = (\hat{\phi}_i, \hat{\phi}_j)_{\hat{\mathcal{I}}}, \quad (2.14)$$

is the mass matrix on the reference interval,  $\mu_j^K = (L_j^K)^{-2} \prod_{i=1}^d L_i^K$ , and  $L_j^K$  is the length of  $K$  along the  $j$ -th axis divided by  $|\hat{\mathcal{I}}|$ . The symbol  $\otimes$  denotes the Kronecker product, which for matrices  $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{r \times s}$ , is defined as the block matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \in \mathbb{R}^{rm \times sn}. \quad (2.15)$$

It follows that if  $A$  and  $B$  are sparse, then  $A \otimes B$  is also sparse.

For the GLL basis  $\{\ell_j\}$ , both  $\hat{A}$  and  $\hat{B}$  are dense, but these are sparse in the hierarchical basis  $\{l_j\}$ . This is illustrated in Figure 2.1. On affine cells, the hierarchical basis yields a sparse stiffness matrix. The bubble functions  $\{l_j\}_{j=1}^{p-1}$ , satisfy  $l'_j(\hat{x}) = P_j(\hat{x})$ , and due to the orthogonality of the Legendre polynomials, the interior block of  $\hat{A}$  is diagonal. The only off-diagonal nonzeros in  $\hat{A}$  are due to the coupling between the interface modes  $l_0, l_p$ . Nevertheless, in order for this sparsity to propagate to higher dimensions, we would additionally wish that  $\hat{B}$  is also as sparse as possible. This is not quite the case for  $\{l_j\}$ , as  $\hat{B}$  has two interior blocks with tri-diagonal structure, in the even-odd decomposition, as  $l_j = P_{j+1} - P_{j-1}$  for  $j = 1, \dots, p-1$ . Therefore, on a typical row,  $A$  will have the structure of the 5-point stencil for  $d = 2$  and that of a 13-point stencil for  $d = 3$ .

### 2.2.1 The fast diagonalization method

Linear systems involving structured matrices such as that defined in (2.13) can be solved efficiently using the fast diagonalization method (FDM) [65]. This method reduces the computation into a sequence of eigenvalue problems on the interval in a similar fashion as the method of separation of variables. It requires a separable

PDE and a tensor-product basis; therefore it can only be applied on meshes or mesh patches with tensor product structure.

We introduce the index sets  $I$  and  $\Gamma$  that decompose the DOFs into interior and interface, respectively. For the space  $P_p(\hat{\mathcal{I}})$ ,  $I = \{1, \dots, p-1\}$  and  $\Gamma = \{0, p\}$ .

To illustrate the FDM, we consider solving a problem on the interior of a Cartesian cell,  $A_{II}^K \underline{u}_I^K = \underline{r}_I^K$ , where  $A_{II}^K = R_I^K A R_I^{K\top}$  and  $R_I^K \in \mathbb{R}^{(p-1)^d \times n}$  is the Boolean restriction matrix onto the interior DOFs of  $K$ . We may first solve the generalized eigenvalue problem on the interior of the reference interval

$$\hat{A}_{II} \hat{S}_{II} = \hat{B}_{II} \hat{S}_{II} \hat{\Lambda}_{II}, \quad (2.16)$$

in conjunction with the normalization condition  $\hat{S}_{II}^\top \hat{B}_{II} \hat{S}_{II} = \mathbb{I}$ , with  $\mathbb{I}$  the identity matrix. Here  $\hat{A}_{II}, \hat{B}_{II} \in \mathbb{R}^{(p-1) \times (p-1)}$  are the interior blocks of  $\hat{A}$  and  $\hat{B}$ , respectively,  $\hat{\Lambda}_{II} \in \mathbb{R}^{(p-1) \times (p-1)}$  is the diagonal matrix of eigenvalues, and  $\hat{S}_{II} \in \mathbb{R}^{(p-1) \times (p-1)}$  is the matrix of eigenvectors. The generalized eigenproblem (2.16) may be equivalently rewritten as

$$\hat{S}_{II}^\top \hat{A}_{II} \hat{S}_{II} = \hat{\Lambda}_{II}, \quad \hat{S}_{II}^\top \hat{B}_{II} \hat{S}_{II} = \mathbb{I}. \quad (2.17)$$

If  $A^K$  is given by (2.13), then its inverse has the following diagonal factorization

$$(A_{II}^K)^{-1} = \left\{ \bigotimes_{k=1}^d \hat{S}_{II} \right\} (\Lambda_{II}^K)^{-1} \left\{ \bigotimes_{k=1}^d \hat{S}_{II}^\top \right\}, \quad (2.18)$$

where

$$\Lambda_{II}^K = \begin{cases} \mu_1^K \mathbb{I} \otimes \hat{\Lambda}_{II} + \mu_2^K \hat{\Lambda}_{II} \otimes \mathbb{I} & d = 2, \\ \mu_1^K \mathbb{I} \otimes \mathbb{I} \otimes \hat{\Lambda}_{II} + \mu_2^K \mathbb{I} \otimes \hat{\Lambda}_{II} \otimes \mathbb{I} + \mu_3^K \hat{\Lambda}_{II} \otimes \mathbb{I} \otimes \mathbb{I} & d = 3. \end{cases} \quad (2.19)$$

Therefore, the solution of a system  $A_{II}^K \underline{u}_I^K = \underline{r}_I^K$  can be obtained with  $\mathcal{O}(p^{d+1})$  computational work.

The main limitation of this approach is that it does not generalize to terms that contain first derivatives, ruling out the possible extension to advection problems. Mixed first derivative terms are also very common in symmetric coercive problems, for instance, when the cells have non-orthotropic deformations, or for vector-valued operators that mix first derivatives of distinct vector components, such as  $\nabla(\nabla \cdot \mathbf{u})$ .

## 2.2.2 Sparsity-promoting discretization

We construct a new finite element basis on the interval, inspired by the FDM, which yields a sparse stiffness matrix. The basis functions can then be extended to

tensor-product cells in arbitrary dimensions by tensor-products. The essential idea is to solve the one-dimensional eigenproblem: find  $\{\hat{s}_i\}_{i=1}^{p-1} \subset P_p(\hat{\mathcal{I}})$  such that

$$(\hat{s}'_i, \hat{s}'_j)_{\hat{\mathcal{I}}} = \lambda_i \delta_{ij}, \quad (\hat{s}_i, \hat{s}_j)_{\hat{\mathcal{I}}} = \delta_{ij}, \quad \hat{s}_i(-1) = \hat{s}_i(1) = 0, \quad i, j \in 1 : (p-1), \quad (2.20)$$

where  $P_p(\hat{\mathcal{I}})$  is the set of polynomials of degree  $p$  on  $\hat{\mathcal{I}}$ ,  $a : b := [a, b] \cap \mathbb{Z}$ , and where summation is not implied. This eigenproblem (2.20) is solved once for a given  $p$ , offline. With these functions, we define the degrees of freedom  $\{\hat{s}_i^*\}_{i=0}^p$  as

$$\hat{s}_i^*(v) := \begin{cases} v(-1), & i = 0, \\ (\hat{s}_i, v)_{\hat{\mathcal{I}}}, & i \in 1 : (p-1), \\ v(1), & i = p. \end{cases} \quad (2.21)$$

The Ciarlet triple [25] for our element is  $(\hat{\mathcal{I}}, P_p(\hat{\mathcal{I}}), \{\hat{s}_i^*\}_{i=0}^p)$ . The point evaluations at the vertices guarantee  $C^0(\hat{\mathcal{I}})$  continuity, and hence  $H(\text{grad}, \hat{\mathcal{I}})$ -conformity.

The finite element induces a reference nodal basis dual to  $\{\hat{s}_i^*\}_{i=0}^p$  in the usual way. The basis functions associated with  $i \in 1 : (p-1)$  are the eigenfunctions  $\hat{s}_i$ , by construction (cf. (2.20)). It remains to determine the interface basis functions  $\hat{s}_0, \hat{s}_p$ . These two functions are defined via the duality condition  $\hat{s}_i^*(\hat{s}_j) = \delta_{ij}$ , which reads

$$\begin{bmatrix} \hat{s}_0(-1) & \hat{s}_j(-1) & \hat{s}_p(-1) \\ (\hat{s}_i, \hat{s}_0)_{\hat{\mathcal{I}}} & (\hat{s}_i, \hat{s}_j)_{\hat{\mathcal{I}}} & (\hat{s}_i, \hat{s}_p)_{\hat{\mathcal{I}}} \\ \hat{s}_0(1) & \hat{s}_j(1) & \hat{s}_p(1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \delta_{ij} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad i, j \in 1 : (p-1). \quad (2.22)$$

As a direct consequence, the reference mass matrix  $\hat{B}_{ij} = (\hat{s}_i, \hat{s}_j)_{\hat{\mathcal{I}}}$  for  $i, j \in 0 : p$  will become almost diagonal, with the only nonzero off-diagonal entries being  $\hat{B}_{0p} = \hat{B}_{p0}$ . This is crucial for maintaining sparsity in higher dimensions, as the stiffness matrix on Cartesian cells in higher dimensions is the Kronecker product of reference mass and stiffness matrices.

We obtain  $\{\hat{s}_j\}$  numerically via Lagrange interpolants. We denote by  $\hat{S} \in \mathbb{R}^{(p+1) \times (p+1)}$  the tabulation of the basis functions onto the GLL points, i.e.,  $\hat{S}_{ij} = \hat{s}_j(\hat{\xi}_i)$ , such that  $\hat{s}_j = \ell_i \hat{S}_{ij}$ , where  $\{\ell_j\}$  are the Lagrange polynomials associated with the GLL points  $\{\hat{\xi}_i\}_{i=0}^p$ . The matrix of coefficients  $\hat{S}$  is determined as follows. From the first and last rows of (2.22) we deduce that  $\hat{S}_{\Gamma I} = 0$  and  $\hat{S}_{\Gamma\Gamma} = \mathbb{I}$ . To determine the tabulation of the interior basis functions onto  $\{\hat{\xi}_i\}_{i \in I}$ , we note that (2.20) is equivalent to the generalized eigenvalue problem: find  $\hat{S}_{II} \in \mathbb{R}^{(p-1) \times (p-1)}$  and  $\{\lambda_j\}_{j \in I}$  such that

$$\hat{S}_{II}^\top \hat{A}_{II}^{\text{GLL}} \hat{S}_{II} = \Lambda_{II}, \quad \hat{S}_{II}^\top \hat{B}_{II}^{\text{GLL}} \hat{S}_{II} = \mathbb{I}. \quad (2.23)$$

Here  $[\hat{A}^{\text{GLL}}]_{ij} = (\ell'_i, \ell'_j)_{\hat{\mathcal{I}}}$ ,  $[\hat{B}^{\text{GLL}}]_{ij} = (\ell_i, \ell_j)_{\hat{\mathcal{I}}}$  are the stiffness and mass matrices discretized in the GLL basis<sup>2</sup>, and  $\Lambda_{II} = \text{diag}(\lambda_1, \dots, \lambda_{p-1})$  is the diagonal matrix of eigenvalues. We solve this problem numerically with the LAPACK routine `dsgv` [3], which uses a Cholesky factorization of  $\hat{B}_{II}^{\text{GLL}}$  and the QR algorithm on a standard eigenvalue problem.

To determine  $\hat{S}_{I\Gamma}$ , we employ the duality condition  $(\hat{s}_i, \hat{s}_j)_{\hat{\mathcal{I}}} = 0$  for  $i \in I, j \in \Gamma$  to obtain

$$\hat{S}_{II}^\top (\hat{B}_{II}^{\text{GLL}} \hat{S}_{I\Gamma} + \hat{B}_{I\Gamma}^{\text{GLL}} \hat{S}_{\Gamma\Gamma}) = 0. \quad (2.24)$$

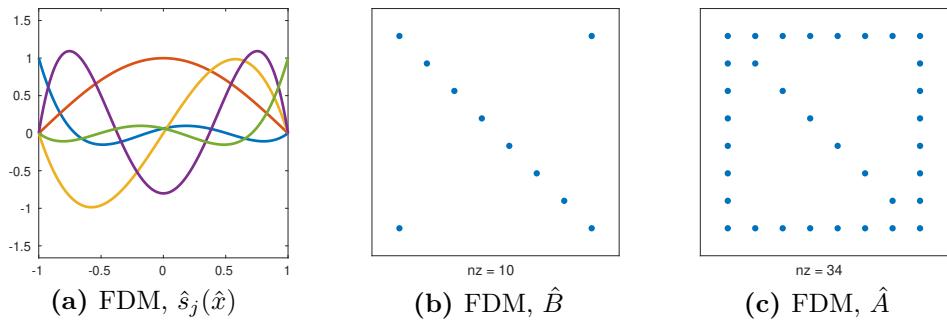
Using (2.23) and  $\hat{S}_{\Gamma\Gamma} = \mathbb{I}$ , we obtain

$$\hat{S}_{I\Gamma} = -\hat{S}_{II} \hat{S}_{II}^\top \hat{B}_{I\Gamma}^{\text{GLL}}. \quad (2.25)$$

In the new basis, the stiffness and mass matrices can be computed as

$$\hat{A} = \hat{S}^\top \hat{A}^{\text{GLL}} \hat{S}, \quad \hat{B} = \hat{S}^\top \hat{B}^{\text{GLL}} \hat{S}. \quad (2.26)$$

Figure 2.2 shows the computed shape functions and the nonzero structure of  $\hat{A}$  and  $\hat{B}$  for the FDM basis.



**Figure 2.2:** Plots of the interval FDM basis functions ( $p = 4$ ) and nonzero structure of the mass and stiffness matrices in the FDM basis on the reference interval ( $p = 7$ ).

This new element will yield a stiffness matrix that is much sparser than with the elements in Figure 2.1. We will introduce a multigrid solver using a vertex-star relaxation that exploits the sparsity.

---

<sup>2</sup>Our approach can be extended to the spectral element method by replacing  $\hat{B}^{\text{GLL}}$  with a lumped mass matrix. The resulting FDM basis preserves the diagonal structure of the lumped mass matrix. Because the vertex-star contains a single vertex, the sparsity pattern for the vertex-star patch matrices is the same for both the high-order FEM and SEM discretizations.

### 2.2.3 Sparse vertex-star relaxation

The solver of Pavarino is fully additive, across both the coarse grid and the vertex-star patches. In our work we consider a small variation of this, with the solver multiplicative between the two levels while remaining additive among the vertex-star patches. This improves the convergence at essentially no cost. The method can be interpreted as a hybrid multiplicative two-level V(1, 1)-cycle with the additive Schwarz method [33] with vertex-star patches as the fine grid relaxation and the lowest-order discretization on the same mesh as the coarse space. The sparse matrix for the coarse space may be assembled and factorized, or other preconditioners such as geometric or algebraic multigrid may be applied instead.

We may write the multigrid relaxation as

$$P_{\text{ASM}}^{-1} = \sum_{j=1}^J R_j^\top A_j^{-1} R_j, \quad (2.27)$$

with  $R_j$  the Boolean restriction matrix onto  $V_j$ , and  $A_j = R_j A R_j^\top$  are the sparse patch matrices for which we may explicitly compute a Cholesky decomposition. The relaxation is scaled by the damping coefficient

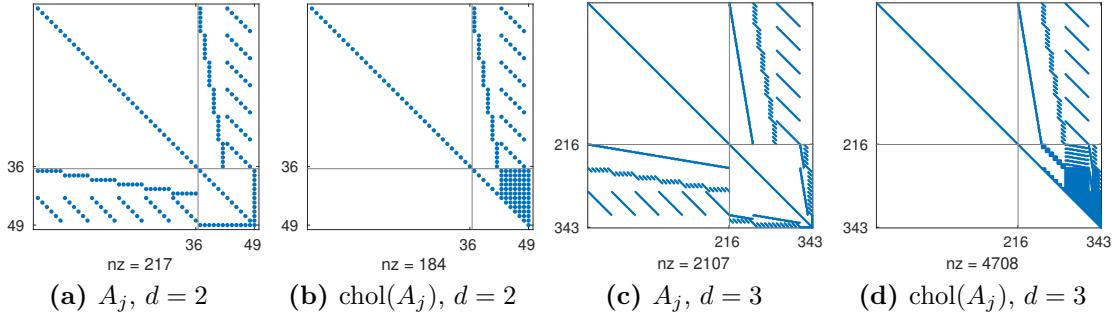
$$\omega = 2[(1 + \alpha)\lambda_{\max} + (1 - \alpha)\lambda_{\min}]^{-1}, \quad (2.28)$$

where  $\lambda_{\min}, \lambda_{\max}$  are the extremal eigenvalues of  $P_{\text{ASM}}^{-1} A$  estimated via the CG-Lanczos procedure [59], and  $\alpha = 0.25$  is chosen to tackle the high frequency error, also ensuring that the error iteration matrix  $\mathbb{I} - \omega P_{\text{ASM}}^{-1} A$  is contractive.

To illustrate the direct solver on the Cartesian vertex-star patch shown in Figure 1.1a, we show in Figure 2.3 the nonzero structure for the patch matrix  $A_j$  and its Cholesky factor. The sparsity pattern of the global matrix  $A$  connects the interior DOFs to their projections onto the facets, hence a typical interior row of  $A$  will have  $2d+1$  nonzeros. For the patch matrix  $A_j$ , an interior row will only have  $d+1$  nonzeros, as the patch only includes one facet per dimension on each cell. Moreover, the total number of nonzeros of  $A_j$  is the same as that of a low-order finite difference or finite element discretization with the 5-point or 7-point stencil on the same grid.

### 2.2.4 Computational complexity

Here we discuss the computational cost of the solution of the patch problem using the Cholesky factorization. Once the factorization has been computed, it may be applied in  $\mathcal{O}(p^{2(d-1)})$  cost, which is optimal for  $d \in \{2, 3\}$ . Unfortunately, the factorization phase is suboptimal, requiring  $\mathcal{O}(p^{3(d-1)})$  operations to compute. The



**Figure 2.3:** Nonzero structure of the stiffness matrix in the FDM basis  $A_j = L_j L_j^\top$  and its upper Cholesky factor  $L_j^\top$  for the Poisson problem on a Cartesian vertex-star patch with  $p = 4$ . With the nested dissection reordering, the factor matrix has minimal fill-in, occurring only on the bottom-right block.

memory required to store the Cholesky factor is  $\mathcal{O}(p^{2(d-1)})$ , which for  $d = 3$  is one factor of  $p$  higher than that required to store the solution.

Consider a stiffness matrix  $A$  discretized with the FDM basis on any mesh with all cells Cartesian. The number of flops needed to solve a linear system using a sparse Cholesky factorization  $A = LL^\top$  is roughly four times the number of nonzero entries in  $L$  [24]. To maximize the sparsity in  $L$ , it is crucial to reorder the DOFs, such that interior DOFs are followed by the interface DOFs. This ensures that the fill-in is introduced only on the bottom-left block. To analyze the cost of factorization, we first introduce the block  $LDL^\top$  decomposition

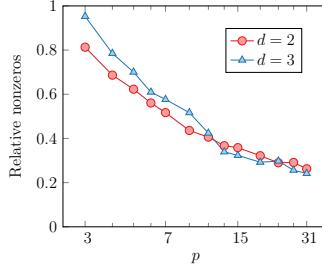
$$A = \begin{bmatrix} A_{II} & A_{I\Gamma} \\ A_{\Gamma I} & A_{\Gamma\Gamma} \end{bmatrix} = \begin{bmatrix} \mathbb{I} & 0 \\ A_{\Gamma I} A_{II}^{-1} & \mathbb{I} \end{bmatrix} \begin{bmatrix} A_{II} & 0 \\ 0 & S_\Gamma \end{bmatrix} \begin{bmatrix} \mathbb{I} & A_{II}^{-1} A_{I\Gamma} \\ 0 & \mathbb{I} \end{bmatrix}, \quad (2.29)$$

where  $S_\Gamma = A_{\Gamma\Gamma} - A_{\Gamma I} A_{II}^{-1} A_{I\Gamma}$  is the interface Schur complement. By construction, the top-left block  $A_{II}$  is diagonal with positive entries, with Cholesky factor  $A_{II}^{1/2}$ . If we decompose  $A_{II}$  and  $S_\Gamma$  in the second matrix on the RHS of (2.29) into their Cholesky factors, and distribute each factor onto the other two matrices, we obtain the Cholesky decomposition of  $A$ :

$$A = LL^\top = \begin{bmatrix} A_{II}^{1/2} & 0 \\ A_{\Gamma I} A_{II}^{-1/2} & L_\Gamma \end{bmatrix} \begin{bmatrix} A_{II}^{1/2} & A_{II}^{-1/2} A_{I\Gamma} \\ 0 & L_\Gamma^\top \end{bmatrix}, \quad (2.30)$$

where the Schur complement is factorized as  $S_\Gamma = L_\Gamma L_\Gamma^\top$ . Since  $A_{II}$  is diagonal, the off-diagonal block  $A_{\Gamma I} A_{II}^{-1/2}$  will preserve the nonzero structure of  $A_{\Gamma I}$ , and similarly for its transpose. Thus, fill-in is only introduced on the interface block through  $L_\Gamma$ .

An ordering strategy that minimizes fill-in consists of applying nested dissection [40] on the adjacency graph that connects topological entities. Each node in this



**Figure 2.4:** Relative number of nonzero entries in the Cholesky factors of the stiffness matrix with the FDM approach and the LOR preconditioner, for a Cartesian vertex-star patch of  $2^d$  cells. Both approaches use a nested dissection ordering; the FDM approach orders blocks of DOFs associated to topological entities, and LOR approach applies nested dissection in the usual sense. The FDM approach is sparser, with substantial gains at higher degrees.

graph represents a cell, face, edge or vertex. The ordering of the entities is then used to permute the corresponding blocks in  $A$ .

Assuming for the worst case that  $L_\Gamma^\top$  is dense, the memory required to store the Cholesky factor is  $\mathcal{O}(p^{2(d-1)})$ . This represents a significant increase from the traditional FDM, which is kept at the optimal  $\mathcal{O}(p^d)$ . However, the DOF ordering does lead to some structured sparsity in  $L_\Gamma^\top$ , as can be seen in Figure 2.3. Nevertheless, we still observe dense  $\mathcal{O}(p^{d-1}) \times \mathcal{O}(p^{d-1})$  blocks. The fact that  $L_\Gamma^\top$  contains these dense blocks indicates that  $\mathcal{O}(p^{3(d-1)})$  operations are required in the factorization phase. However, the forward and back-substitution steps have a computational cost of  $\mathcal{O}(p^{2(d-1)})$  operations, which is optimal for  $d \leq 3$ .

Compared to the LOR approach, our approach with the FDM basis has a sparser Cholesky factorization. In Figure 2.4 we present the ratio of the number of nonzeros in the Cholesky factors of our approach and the LOR preconditioner ordered with nested dissection for  $d \in \{2, 3\}$ . The fact that the ratio is always below 1 confirms that our approach is sparser, with a substantial gain at higher degrees. Practical LOR solvers use algebraic multigrid [14] or patchwise multigrid with incomplete Cholesky smoothers [77] to avoid the cost of the Cholesky factorizations of the patch matrices.

## 2.3 Sparse preconditioners on unstructured meshes

For arbitrarily deformed cells, the local stiffness matrices  $A^K$  cannot be expressed in terms of tensor-products of  $\hat{A}$  and  $\hat{B}$  as in (2.13), and  $A^K$  is not sparse in the FDM basis. The preconditioning techniques found in [27, 39, 96] introduce an auxiliary Cartesian domain to construct a separable problem for which the FDM is a direct solver. The method described by Fischer [39] constructs a preconditioner

by replacing  $K$  with its nearest rectangular approximation, whose dimensions are computed as the mean separation between the mapped GLL nodes from opposite facets of  $K$ . Witte et al. [96] obtain the lengths from the average arclength of opposite sides of  $K$ , but it is not clear how this extends to the 3D case. To the best of our knowledge, no theory underpins these choices.

Our approach to construct the separable surrogate is based on the theory of equivalent operator preconditioning [11]. We work with the bilinear form  $a(\cdot, \cdot)$  in terms of the reference coordinates. We discard the mixed derivative terms that prevent separability, and we replace the coefficients with piecewise constants in the reference coordinates<sup>3</sup>. We will prove that this choice yields a spectrally equivalent operator.

The bilinear form  $a(\cdot, \cdot)$  is numerically computed via a quadrature rule on the reference cell. The integration domain  $\Omega$  is decomposed into the cells, so that  $a(\cdot, \cdot)$  can be expressed as a sum of cell contributions  $a_K(\cdot, \cdot)$  where integration and differentiation are with respect to  $\hat{\mathbf{x}} \in \hat{K}$ ,

$$a(v, u) = \sum_{K \in \mathcal{T}_h} a_K(v, u). \quad (2.31)$$

The integration measure  $d\mathbf{x}$  is replaced with  $|DF_K|d\hat{\mathbf{x}}$ , thus  $a_K(v, u) = a_{\hat{K}}(v \circ F_K, |DF_K|u \circ F_K)$ . As the form arguments are composed by the mapping, their gradient is most conveniently computed by the chain rule, i.e.,  $\nabla = DF_K^{-\top} \hat{\nabla}$  where  $\hat{\nabla}$  is the gradient with respect to  $\hat{\mathbf{x}}$ . Therefore

$$a_K(v, u) = (\nabla v, \nabla u)_K = (\hat{\nabla} v \circ F_K, \hat{G}_K \hat{\nabla} u \circ F_K)_{\hat{K}}, \quad (2.32)$$

where  $\hat{G}_K : K \rightarrow \mathbb{R}^{d \times d}$  is the inverse metric of the coordinate transformation weighted by the Jacobian determinant,

$$\hat{G}_K = |DF_K|DF_K^{-1}DF_K^{-\top}. \quad (2.33)$$

This tensor encapsulates all the geometry-dependent information of the form; it is spatially dependent for generally-deformed elements, and constant in the case of affine transformations. For a separable geometry,  $\hat{G}_K$  is diagonal and constant.

To construct an auxiliary problem that is separable by the FDM in the reference coordinates, we replace  $\hat{G}^K$  in  $a_K(\cdot, \cdot)$  with a constant diagonal approximation  $\text{diag}(\mu_j^K)$ . Each  $\mu_j^K$  is given by the cell-wise average of the diagonal entry  $\hat{G}_{jj}^K$ ,

$$\mu_j^K := \frac{1}{|\hat{K}|} \int_{\hat{K}} \hat{G}_{jj}^K d\hat{\mathbf{x}}, \quad (2.34)$$

---

<sup>3</sup>Recall that piecewise constant coefficients in the physical coordinates will not yield piecewise constant coefficients in the reference coordinates.

where summation over the index  $j$  is not implied. As the approximation is local to each cell, it is still possible to assemble a sparse stiffness matrix discretizing the auxiliary problem on meshes where cells are not structured in a tensor-product grid.

We now establish the spectral equivalence between the original bilinear form (2.2) and the auxiliary separable one.

**Theorem 2.1.** *Let  $\hat{\mu}_K := \text{diag}(\mu_j^K)$  be the constant diagonal approximation of  $\hat{G}^K$ , and define the auxiliary bilinear form*

$$\tilde{a}(v, u) := \sum_{K \in \mathcal{T}_h} \tilde{a}_K(v, u) := \sum_{K \in \mathcal{T}_h} (\hat{\nabla} v \circ F_K, \hat{\mu}_K \hat{\nabla} u \circ F_K)_{\hat{K}}. \quad (2.35)$$

*Then, there exist  $p$ -independent constants  $c, C > 0$  that depend on  $\mathcal{T}_h$  through  $\hat{G}^K$  such that*

$$c \leq \frac{a(v, v)}{\tilde{a}(v, v)} \leq C \quad \forall v \in V \setminus \{0\}. \quad (2.36)$$

*Proof.* Let  $c_K, C_K$  be lower and upper bounds for the spectrum of the diagonally scaled metric, so that  $\sigma(\hat{\mu}_K^{-1/2} \hat{G}^K \hat{\mu}_K^{-1/2}) \in [c_K, C_K]$  for all  $\hat{x} \in \hat{K}$ . We claim that

$$c_K \leq \frac{a_K(v, v)}{\tilde{a}_K(v, v)} \leq C_K \quad \forall v \in \{v \in V : v|_K \neq 0\}. \quad (2.37)$$

This result is obtained by first rewriting  $a_K(v, v)$  with  $\hat{\mu}_K^{1/2} \hat{\mu}_K^{-1/2} \hat{G}^K \hat{\mu}_K^{-1/2} \hat{\mu}_K^{1/2}$  instead of  $\hat{G}^K$ , and then replacing  $\hat{\mu}_K^{-1/2} \hat{G}^K \hat{\mu}_K^{-1/2}$  with  $c_K \mathbb{I}$  or  $C_K \mathbb{I}$  to find the lower or upper bounds, respectively. It then follows that

$$c := \min_{K \in \mathcal{T}_h} c_K \leq \frac{a(v, v)}{\tilde{a}(v, v)} \leq \max_{K \in \mathcal{T}_h} C_K =: C \quad \forall v \in V \setminus \{0\}. \quad (2.38)$$

□

Let  $\tilde{A}$  be the stiffness matrix associated with the auxiliary form  $\tilde{a}(\cdot, \cdot)$ . By Theorem 2.1, the condition number  $\kappa(\tilde{A}^{-1} A)$  is bounded by  $C/c$  independently of  $p$ . Numerical experiments also indicate that  $\kappa(\tilde{A}^{-1} A)$  is independent of  $h$  under uniform refinements. Now consider a preconditioner  $P$  where the auxiliary form  $\tilde{a}(\cdot, \cdot)$  is used additively in both the coarse solve and the vertex-star patches. In this case, Theorem 1 of [75] guarantees that  $\kappa(P^{-1} \tilde{A})$  is bounded from above independently of  $h$  and  $p$ . Hence we may conclude that  $\kappa(P^{-1} A) \leq \kappa(P^{-1} \tilde{A}) \kappa(\tilde{A}^{-1} A)$  is bounded independently of  $h$  and  $p$ . In practice, we expect that using multiplicative coarse grid correction with the original form  $a(\cdot, \cdot)$  can only improve the preconditioner.

To gain useful insight, we consider the case where  $d = 2$  and  $F_K$  is an affine transformation, that is when  $K$  is a parallelogram. Without loss of generality,

suppose that one of the sides of  $K$  has length  $2L_1$  and is aligned with the first reference coordinate axis, and the other side of length  $2L_2$  is at an angle  $\theta$  with respect to the same axis. The Jacobian of the coordinate transformation is

$$DF_K = \begin{bmatrix} L_1 & L_2 \cos \theta \\ 0 & L_2 \sin \theta \end{bmatrix}, \quad (2.39)$$

to which corresponds the Jacobian-weighted inverse metric

$$\hat{G}^K = \frac{1}{L_1 L_2 |\sin \theta|} \begin{bmatrix} L_2^2 & -L_1 L_2 \cos \theta \\ -L_1 L_2 \cos \theta & L_1^2 \end{bmatrix}. \quad (2.40)$$

Since  $\hat{G}^K$  is constant,  $\hat{\mu}_K$  is simply the diagonal part of  $\hat{G}^K$ . The spectrum of the diagonally scaled metric will be independent of  $L_1$  and  $L_2$ , but still depend on  $\theta$ ,

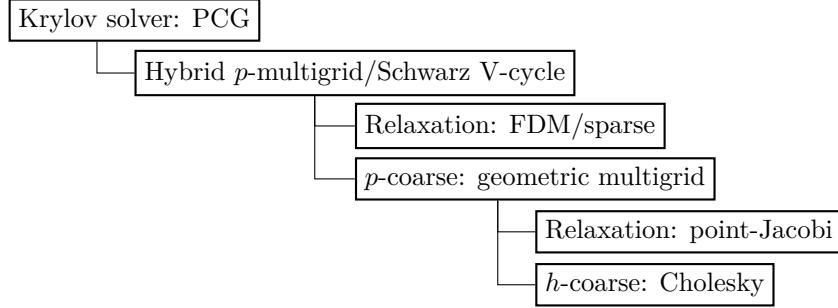
$$\sigma(\hat{\mu}_K^{-1/2} \hat{G}^K \hat{\mu}_K^{-1/2}) = [1 - |\cos \theta|, 1 + |\cos \theta|]. \quad (2.41)$$

This spectrum is desirable because it is centered at 1 and bounded above for all  $\theta$ . If we follow the geometric approaches of [39, 96], we would have to choose a rectangle with side lengths  $2L_1$  and  $2L_2$  as the auxiliary domain for the Poisson problem. Then, the previous bounds (2.41) would become scaled by  $|\sin \theta|^{-1}$ . In this case, the spectrum is unbounded from above in the limit  $\theta \rightarrow 0$ .

## 2.4 Numerical results

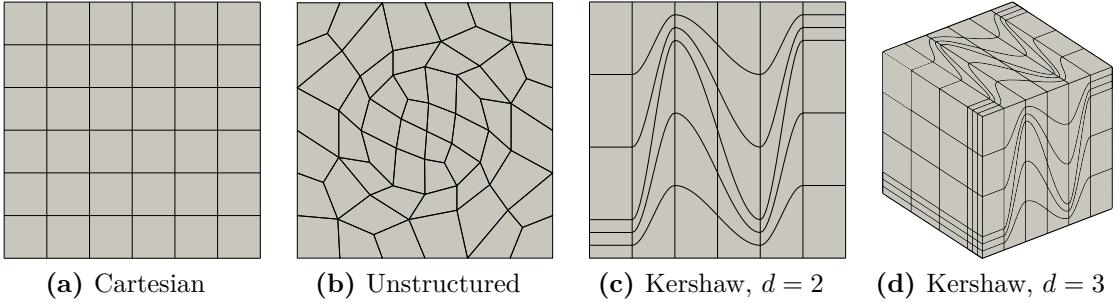
We provide an implementation of the new element with the FDM shape functions on the interval in the FIAT [52] package. The extension to quadrilaterals and hexahedra is achieved by taking tensor-products of the one-dimensional element with FIAT [47]. Code for the sum-factorized evaluation of the residual is automatically generated by Firedrake [82, 48], implementing a Gauß–Lobatto quadrature rule with  $3(p+1)/2$  points along each direction. The sparse preconditioner discretizing the auxiliary form is implemented in `firedrake.FDMPC` using PETSc [13]. The Cholesky factorization of the patch matrices is computed using CHOLMOD [24]. Most of our computations were performed using an Intel Xeon CPU E5-4627 v2 @ 3.30GHz with 32 cores and 67.6 GB of RAM storage.

The hybrid  $p$ -multigrid/Schwarz solver employing the FDM/sparse relaxation is illustrated in Figure 2.5. To achieve scalability with respect to the mesh parameter  $h$ , on the  $p$ -coarse problem we employ geometric multigrid with damped point-Jacobi relaxation and a Cholesky factorization on the coarsest level using MUMPS [2]. We test the effectiveness of this approach on a hierarchy of meshes obtained by  $l \geq 0$  uniform refinements of the base meshes shown in Figure 2.6.



**Figure 2.5:** Solver diagram for the Poisson problem.

We present results for the Poisson equation in  $\Omega = (0, 1)^d$  discretized on the three hierarchies of Cartesian, unstructured, and structured but deformed (Kershaw) [51] meshes. The coordinate field of the Kershaw mesh is in  $[Q_3(\Omega)]^d \cap C^1(\Omega)$ , with a cell aspect ratio of  $\varepsilon_y = \varepsilon_z = 0.3$  near the corners of the domain. We impose homogeneous Dirichlet BCs on  $\Gamma_D = \partial\Omega$  and a constant forcing  $f = 1$ . In Table 2.1 we present PCG iteration counts required to reduce the Euclidean norm of the residual by a factor of  $10^8$  starting from a zero initial guess. In Table 2.2 we show the condition number  $\kappa(P^{-1}A)$  estimated by CG-Lanczos. The results show almost complete  $p$ - and  $h$ -robustness in the Cartesian case, and very slow growth of iteration counts in the unstructured case. Given the lack of shape regularity, the Kershaw mesh is significantly more challenging; even with exact patch solvers, we do not expect  $h$ -robustness<sup>4</sup>.



**Figure 2.6:** Base meshes for the Poisson problem. The Cartesian and unstructured base meshes used for  $d = 3$  are the extrusion with six layers of the two-dimensional meshes shown here.

To assess the computational performance of our approach, we solve the three-dimensional Poisson equation on a Cartesian mesh with  $3 \times 3 \times 3$  cells with a single core of an Intel Core i7-10875H CPU @ 2.30GHz. We plot in Figure 2.7 the runtimes,

<sup>4</sup>Unreported numerical experiments with exact patch solvers on the Kershaw mesh show  $p$ -robust iteration counts, which increase with the level of  $h$ -refinement.

**Table 2.1:** PCG iteration counts for the hybrid  $p$ -multigrid/Schwarz solver with the FDM/sparse relaxation for varying polynomial degree  $p$  and level of uniform  $h$ -refinement  $l$ . The patch problems are solved exactly on the Cartesian mesh.

$d$	$p \setminus l$	Cartesian			Unstructured			Kershaw		
		0	1	2	0	1	2	0	1	2
2	3	7	8	9	12	13	14	27	35	54
	7	8	8	9	16	16	17	44	56	78
	15	8	9	9	19	19	19	58	69	90
	31	8	9	9	21	20	21	67	80	97
3	3	12	12	12	17	17	18	54	66	102
	7	12	12	12	22	21	21	98	106	158
	15	12	13		25	24		131	132	

**Table 2.2:** Estimated condition numbers for the preconditioned operator  $\kappa(P^{-1}A)$  using the hybrid  $p$ -multigrid/Schwarz solver with the FDM/sparse relaxation.

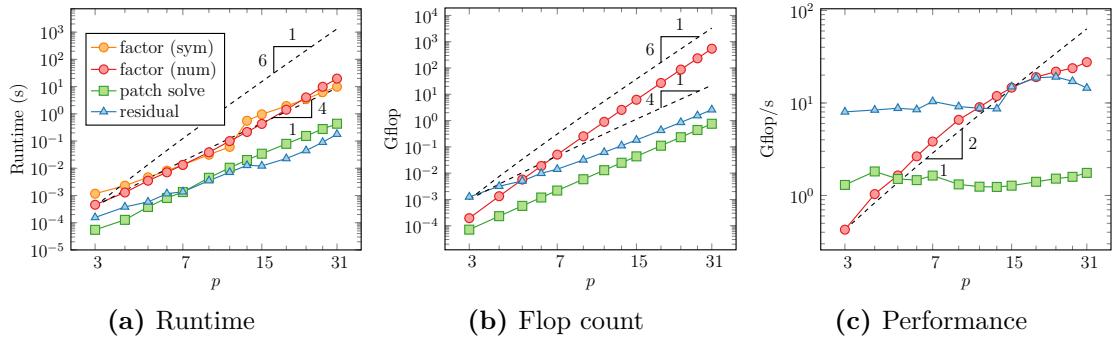
$d$	$p \setminus l$	Cartesian			Unstructured			Kershaw		
		0	1	2	0	1	2	0	1	2
2	3	1.44	1.49	1.50	2.14	2.37	2.81	9.34	15.6	34.5
	7	1.48	1.48	1.50	3.23	3.27	3.79	19.6	30.3	57.6
	15	1.51	1.51	1.52	4.06	3.78	4.13	30.5	45.8	69.0
	31	1.54	1.52	1.52	4.45	4.06	4.36	40.4	57.1	73.3
3	3	2.87	2.49	2.45	4.16	4.21	4.55	34.8	46.1	117
	7	2.79	2.70	2.67	5.88	5.54	5.47	100	110	266
	15	2.83	2.79		7.12	6.44		165	151	

flop counts, and achieved arithmetic performance for the Cholesky factorization of the patch matrices, the solution of the patch problems using this factorization (per application of the relaxation), and the matrix-free evaluation of the residual (per Krylov iteration, excluding the application of the global to local map) as functions of  $p$ . The dotted lines are to indicate powers of  $2p - 1$ , which is the number of DOFs along each side of typical vertex-star patch not intersecting the mesh boundary.

Despite the  $\mathcal{O}(p^{3(d-1)})$  computational cost of the Cholesky factorization, these results show  $\mathcal{O}(p^{2(d-1)})$  scaling for runtime up to  $p = 15$ . This speedup can be explained mainly by data locality. The sparse Cholesky factorization is obtained by recursively applying the block  $LDL^\top$  decomposition up to the point where the Schur complement is sufficiently dense. The computation of this Schur complement via dense matrix-matrix multiplication (BLAS-3 `dgemm`) dominates the computational cost. As  $p$  is increased, the utilization of arithmetic units increases in proportion to

the dimension of the Schur complement, which explains the  $\mathcal{O}(p^{d-1})$  scaling of the achieved arithmetic performance. As the arithmetic capabilities become saturated for  $p > 15$ , the  $\mathcal{O}(p^{3(d-1)})$  scaling in the factorization runtime should become apparent.

Most of the time in the relaxation step is spent in accessing the factor matrix from memory, given the  $\mathcal{O}(p^{2(d-1)})$  sub-optimal storage per patch. The relaxation is therefore limited by memory bandwidth and not arithmetically intense, which explains the poor arithmetic performance. This is in contrast to the sum-factorized residual evaluation, which has a  $\mathcal{O}(p^d)$  memory footprint and presents better arithmetic utilization [58]. Nevertheless, the results indicate that the runtime for the solution of the patch problems with the sparse Cholesky factorization remains very close to that of the matrix-free residual evaluation for moderate  $p$ , being slightly faster for  $p \leq 7$ , mainly due to lower operation count.



**Figure 2.7:** Runtimes, flop counts, and achieved arithmetic performance for the Cholesky factorization (symbolic and numeric), solution of the patch problems within a single application of the relaxation, and a single residual evaluation for a Cartesian mesh with  $3 \times 3 \times 3$  cells on a single CPU core. We observe that the factorization runtime scales better than expected, close to the optimal  $\mathcal{O}(p^{d+1})$  complexity.

# 3

## Multigrid solvers for linear elasticity<sup>1</sup>

### 3.1 Primal formulation of linear elasticity

We now consider how these ideas may be applied in the more complex setting of a nonseparable, vector-valued PDE. The equations of linear elasticity describe the displacement  $\mathbf{u} : \Omega \rightarrow \mathbb{R}^d$  of a solid body with a reference configuration  $\Omega \subset \mathbb{R}^d$ . The primal formulation is to find  $\mathbf{u} - \mathbf{u}_0 \in V := [H_D^1(\Omega)]^d$  such that

$$a(\mathbf{v}, \mathbf{u}) = L(\mathbf{v}) \quad \forall \mathbf{v} \in V, \quad (3.1)$$

where

$$a(\mathbf{v}, \mathbf{u}) = \int_{\Omega} 2\mu\varepsilon(\mathbf{v}) : \varepsilon(\mathbf{u}) + \lambda\nabla \cdot \mathbf{v} \nabla \cdot \mathbf{u} \, dx, \quad L(\mathbf{v}) = \int_{\Omega} \mathbf{v} \cdot \mathbf{B} \, dx. \quad (3.2)$$

Here, we assume that the material is homogeneous and isotropic, and can thus be described by the Lamé parameters  $\mu, \lambda > 0$ ;  $\varepsilon(\mathbf{u}) = (\nabla \mathbf{u} + \nabla \mathbf{u}^\top)/2$  is the linearized strain tensor;  $\mathbf{u}_0$  is Dirichlet data prescribed on  $\Gamma_D \subseteq \partial\Omega$ ; and  $\mathbf{B} \in [L^2(\Omega)]^d$  is a body force. The Poisson ratio  $\nu = \lambda/(2\mu + 2\lambda)$  measures the compressibility of the material. In the incompressible limit  $\lambda \rightarrow \infty$  (i.e.,  $\nu \rightarrow 1/2$ ), the problem becomes ill-conditioned, as  $a(\cdot, \cdot)$  becomes insensitive to divergence-free perturbations in the arguments.

Consider the partitioning of the stiffness matrix  $A$  into blocks that act on each displacement component,

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1d} \\ \vdots & \ddots & \vdots \\ A_{d1} & \cdots & A_{dd} \end{bmatrix}. \quad (3.3)$$

---

<sup>1</sup>This chapter is extracted from [20], published in SIAM J. Sci. Comput.

The diagonal block  $A_{jj}$  discretizes the bilinear form

$$\int_{\Omega} \mu \nabla v_j \cdot \nabla u_j + (\mu + \lambda) \frac{\partial v_j}{\partial x_j} \frac{\partial u_j}{\partial x_j} \, d\mathbf{x}, \quad (3.4)$$

where summation is not implied, and  $u_j$  and  $v_j$  are components of  $\mathbf{u}$  and  $\mathbf{v}$ , respectively. The off-diagonal blocks  $A_{ij}$ ,  $i \neq j$ , discretize

$$\int_{\Omega} \mu \frac{\partial v_i}{\partial x_j} \frac{\partial u_j}{\partial x_i} + \lambda \frac{\partial v_i}{\partial x_i} \frac{\partial u_j}{\partial x_j} \, d\mathbf{x}. \quad (3.5)$$

The diagonal blocks can be diagonalized by the FDM on the interior of a Cartesian cell when the reference axes are aligned with the physical coordinates. The same statement does not hold true for the off-diagonal blocks, as they couple together different displacement components. This is because they discretize products of different first derivatives on the different components and hence are not separable.

The separate displacement components (SDC) preconditioner [17, 41] is defined as the block diagonal matrix  $A_{\text{SDC}} = \text{diag}(A_{11}, \dots, A_{dd})$ . In other words, this approach is also described as block-Jacobi in the displacement components. The SDC preconditioner discretized with the FDM basis is sparse for Cartesian cells aligned with the coordinate axes. On arbitrary cells, for each separate component, we obtain an auxiliary form that is separable in the reference coordinates by selecting constant diagonal coefficients  $\hat{\mu}_K$ .

It is shown in [17] that for a homogeneous isotropic material with principal axes parallel to the axes of the reference coordinate system, the condition number of the preconditioned matrix will depend on the Poisson ratio:

$$\kappa(A_{\text{SDC}}^{-1} A) \leq \frac{d-1}{\gamma} \frac{1-\nu}{1-2\nu}, \quad (3.6)$$

where  $\gamma$  is the constant appearing in Korn's inequality,

$$\|\mathbf{u}\|_{H^1(\Omega)^d}^2 \leq \gamma \int_{\Omega} \mathbf{u} \cdot \mathbf{u} + \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{u}) \, d\mathbf{x} \quad \forall \mathbf{u} \in V. \quad (3.7)$$

Thus, the convergence rate of the SDC preconditioner will deteriorate for  $\nu$  approaching  $1/2$ , the so-called nearly incompressible case.

We consider the reference configuration  $\Omega = (0, 1)^d$  discretized on a Cartesian mesh with 8 cells along each direction. We specify  $\mu = 1$ , a uniform downwards body force  $\mathbf{B} = -0.02\mathbf{e}_2$ , and homogeneous Dirichlet BCs on  $\Gamma_D = \{\mathbf{x} \in \partial\Omega, x_1 = 0\}$ . In Table 3.1 we present the PCG iteration counts required to reduce the Euclidean norm of the residual by a factor of  $10^8$  starting from a zero initial guess. As the preconditioner, we employ the hybrid  $p$ -multigrid/Schwarz method

with vertex-star patches and the SDC/FDM/sparse relaxation and a coarse space with  $p = 1$ . As expected from (3.6), the results confirm that the approach is reasonably  $p$ -robust, but that robustness with respect to  $\nu$  cannot be achieved with SDC relaxation on vertex-star patches.

**Table 3.1:** PCG iteration counts for the primal formulation of the linear elasticity problem using the SDC/FDM/sparse relaxation.

$d$	$p \setminus \lambda$	0	$10^0$	$10^1$	$10^2$	$10^3$
2	3	13	14	24	70	199
	7	17	17	28	76	236
	15	18	19	30	81	249
	31	20	20	32	84	258
3	3	20	22	39	114	362
	7	25	28	48	123	381
	15	27	29	51	125	373

## 3.2 Mixed FEM formulations of linear elasticity

In order to avoid locking in nearly incompressible continua, or impose the incompressibility constraint, the standard approach is to introduce a pressure-like variable and discretize with a mixed FEM. This is expressed by the weak formulation: find  $(\mathbf{u} - \mathbf{u}_0, p) \in V \times Q$  such that

$$a(\mathbf{v}, \mathbf{u}) + b(p, \mathbf{v}) = L(\mathbf{v}) \quad \forall \mathbf{v} \in V, \quad (3.8)$$

$$b(q, \mathbf{u}) - c(q, p) = 0 \quad \forall q \in Q, \quad (3.9)$$

where

$$a(\mathbf{v}, \mathbf{u}) = \int_{\Omega} 2\mu \varepsilon(\mathbf{v}) : \varepsilon(\mathbf{u}) \, d\mathbf{x}, \quad b(q, \mathbf{u}) = \int_{\Omega} q \operatorname{div}(\mathbf{u}) \, d\mathbf{x}, \quad c(q, p) = \int_{\Omega} \lambda^{-1} qp \, d\mathbf{x}, \quad (3.10)$$

and  $Q = L_0^2(\Omega)$  for  $\lambda = \infty$  and  $\Gamma_D = \partial\Omega$ , or  $Q = L^2(\Omega)$  otherwise.

In order for this problem to have a unique solution, we require the well-known Brezzi conditions: the solution for  $\mathbf{u}$  is unique if  $a(\cdot, \cdot)$  is coercive on the kernel of  $b(\cdot, \cdot)$ , and the solution for  $p$  is unique if there exists a right inverse for  $b(\cdot, \cdot)$ . This is expressed in the so-called inf-sup condition or LBB condition [12, 18]: there exists  $\beta$ , which might depend on  $\Omega$ , such that

$$0 < \beta := \inf_{q \in Q} \sup_{\mathbf{v} \in V} \frac{b(q, \mathbf{v})}{a(\mathbf{v}, \mathbf{v})^{1/2} \|q\|_Q}. \quad (3.11)$$

After selecting suitable finite dimensional subspaces  $V_h \subset V$ ,  $Q_h \subset Q$ , we obtain a system of linear equations with the saddle point structure

$$\begin{bmatrix} A & B^\top \\ B & -C \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}} \\ p \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{f}} \\ g \end{bmatrix}. \quad (3.12)$$

We require the analogous Brezzi conditions for the discrete problem: that  $a(\cdot, \cdot)$  is coercive on the discrete kernel of  $b(\cdot, \cdot)$ , and that there exists a discrete inf-sup constant  $\tilde{\beta}$  independent of the mesh but possibly depending on  $p$  such that

$$0 < \tilde{\beta} := \inf_{q_h \in Q_h} \sup_{\mathbf{v}_h \in V_h} \frac{b(q_h, \mathbf{v}_h)}{a(\mathbf{v}_h, \mathbf{v}_h)^{1/2} \|q_h\|_{Q_h}}. \quad (3.13)$$

The discretization  $V_h \times Q_h$  must be chosen carefully to satisfy these conditions; the discrete inf-sup condition will not be satisfied by arbitrary discretizations. If they are, we have the well-known error estimates

$$\|\mathbf{u}_h - \mathbf{u}\|_V \leq C_1 \left\{ \inf_{\mathbf{v}_h \in V_h} \|\mathbf{v}_h - \mathbf{u}\|_V + \inf_{q_h \in Q_h} \|q_h - p\|_Q \right\}, \quad (3.14a)$$

$$\|p_h - p\|_Q \leq \tilde{\beta}^{-1} C_2 \left\{ \inf_{\mathbf{v}_h \in V_h} \|\mathbf{v}_h - \mathbf{u}\|_V + \inf_{q_h \in Q_h} \|q_h - p\|_Q \right\}, \quad (3.14b)$$

where  $C_1, C_2 > 0$  are generic constants independent of the mesh parameter  $h$ . For the use of high-order discretizations, it is desirable to choose an element pair where  $\tilde{\beta}$  does not decrease as the polynomial degree of the approximation is increased. Such a discretization is referred to as  $p$ -stable.

In fact,  $p$ -stability is important for solvers also. Approaches based on block-Gaußian elimination, such as the Uzawa algorithm [10] and block-preconditioned MINRES [74], require preconditioners for the negative pressure Schur complement  $S = C + BA^{-1}B^\top$ . It is well known that for the Stokes system, the continuous analogue of  $S$ ,  $\nabla \cdot (-\nabla^2)^{-1}\nabla$ , is well approximated by the identity operator [92]. It follows that  $S$  is spectrally equivalent to the pressure mass matrix,  $M_p$ ,

$$\beta_0^2 \leq \frac{\underline{q}^\top S \underline{q}}{\underline{q}^\top M_p \underline{q}} \leq \beta_1^2 \quad \forall \underline{q} \in \mathbb{R}^{\dim(Q_h)} \setminus \{0\}. \quad (3.15)$$

The rate of convergence of block-preconditioned MINRES will be determined by the ratio  $\beta_1/\beta_0$ . For Stokes flows with pure Dirichlet BCs,  $\beta_1 = 1$ , and  $\beta_1 = \sqrt{d}$  otherwise. In general, we have  $\beta_0 = \tilde{\beta}$ . Since  $A$  is spectrally equivalent to the vector Laplacian, these results also hold for linear elasticity. We may expect solvers based on such techniques to degrade with  $p$ -refinement if the discretization is not  $p$ -stable.

If we choose to work with the  $[H^1(\Omega)]^d$ -conforming space  $V_h = [Q_p]^d$ , some standard inf-sup stable choices for  $Q_h$  are  $Q_{p-1}$ ,  $DQ_{p-2}$  and  $DP_{p-1}$ .  $DQ_{p-2}$  denotes

discontinuous piecewise polynomials of degree at most  $p - 2$  in each direction, while  $\text{DP}_{p-1}$  denotes discontinuous piecewise polynomials of total degree at most  $p - 1$ . The choice  $Q_h = \text{Q}_{p-1}$  gives rise to the high-order generalization of the Taylor–Hood mixed element [95]. Here,  $M_p$  will not be block diagonal, and hence more expensive preconditioning techniques will be required. Moreover, it is shown numerically in [1] that the Taylor–Hood element is not  $p$ -stable. The choice  $Q_h = \text{DQ}_{p-2}$  exhibits an asymptotic decay of  $\tilde{\beta} \leq Cp^{(1-d)/2}$  as  $p \rightarrow \infty$  [16], and thus is not  $p$ -stable. In practice, it is observed that this is quite a pessimistic bound for moderate  $p$  [66]. The choice  $Q_h = \text{DP}_{p-1}$  is  $p$ -stable, but numerical experiments reveal that the stability is severely affected by the cell aspect ratio, unlike the previous two choices [89]. Moreover this last space does not have tensor-product shape functions, so its efficient implementation becomes challenging. Another choice,  $Q_h = \text{DQ}_{[\lambda p]}$  for fixed  $\lambda < 1$ , is  $p$ -stable for  $p \geq 2/(1-\lambda)$ , and leads to optimal error estimates on the pressure [15].

To construct  $p$ -stable discretizations that are also robust to cell aspect ratio, we turn to nonconforming schemes with  $V_h \subset H(\text{div}, \Omega)$  [26, 60]. In particular we consider the use of Raviart–Thomas elements [7] of degree  $p$  for  $V_h$  for the displacement, paired with  $Q_h = \text{DQ}_{p-1}$ . This pair satisfies  $\text{div}(V_h) = Q_h$ , which enforces the incompressibility constraint (3.9) exactly in the numerical approximation for  $\lambda = \infty$ . The Raviart–Thomas elements are defined on the reference quadrilateral as

$$\text{RT}_p(\hat{K}) = \text{HDiv}(\text{P}_p(\hat{\mathcal{I}}) \otimes \text{DP}_{p-1}(\hat{\mathcal{I}})) \oplus \text{HDiv}(\text{DP}_{p-1}(\hat{\mathcal{I}}) \otimes \text{P}_p(\hat{\mathcal{I}})). \quad (3.16)$$

Here  $\text{HDiv}$  transforms a scalar-valued function into a vector-valued function in the coordinate direction of the continuous axis. The analogous element in three dimensions is referred to as the Nédélec face element [72]. The definition can be extended to curvilinear cells via the contravariant Piola transform: for a function  $\hat{\mathbf{u}} : \hat{K} \rightarrow \mathbb{R}^d$ , we define  $\mathbf{u} : K \rightarrow \mathbb{R}^d$  as

$$\mathbf{u} = \mathcal{F}_K^{\text{div}}(\hat{\mathbf{u}}) := \frac{1}{|\text{DF}_K|} \text{DF}_K \left( \hat{\mathbf{u}} \circ F_K^{-1} \right), \quad (3.17)$$

and set

$$\text{RT}_p(K) = \mathcal{F}_K^{\text{div}}(\text{RT}_p(\hat{K})). \quad (3.18)$$

These elements have superb properties, but their nonconforming nature must be suitably addressed in the discretization. They only impose continuity of the normal components of  $\mathbf{u}$  across cell facets, and we therefore weakly enforce the tangential continuity via the symmetric interior penalty (SIPG) method [4]. The use of SIPG for the displacement requires further extension of the FDM/sparse relaxation; in particular, we must consider the additional facet integrals arising in the method, and show that the stiffness remains sparse.

### 3.3 Extension to interior penalty DG methods

Interior penalty discontinuous Galerkin (IP-DG) methods relax the continuity requirement of the discretization space. For instance, instead of  $[H^1(\Omega)]^d$ , we consider a larger function space with weaker continuity, such as  $[L^2(\Omega)]^d$  or  $H(\text{div}, \Omega)$ . As previously mentioned, in order to deal with the non-conformity,  $C^0$ -continuity is weakly enforced via the introduction of a penalty term on the set of interior facets  $\Gamma_I$  of the mesh  $\mathcal{T}_h$  that vanishes for  $C^0$ -continuous functions. Similarly, the weak prescription of the Dirichlet BC  $\mathbf{u} = \mathbf{u}_0$  on  $\Gamma_D$  is achieved by introducing a penalty term on  $\Gamma_D$ .

We consider the following SIPG formulation:

$$\begin{aligned} a(\mathbf{v}, \mathbf{u}) &= \sum_{K \in \mathcal{T}_h} \int_K \nabla \mathbf{v} : \mathcal{F}^v(\nabla \mathbf{u}) \, dx \\ &+ \sum_{e \in \Gamma_I \cup \Gamma_D} \int_e \eta h_e^{-1} \{G^\top\} [\![\mathbf{v}]\!] : [\![\mathbf{u}]\!] - [\![\mathbf{v}]\!] : \{\mathcal{F}^v(\nabla \mathbf{u})\} - \{G^\top \nabla \mathbf{v}\} : [\![\mathbf{u}]\!] \, ds, \end{aligned} \quad (3.19)$$

$$L(\mathbf{v}) = \int_\Omega \mathbf{v} \cdot \mathbf{B} \, dx + \int_{\Gamma_D} \eta h_e^{-1} G^\top (\mathbf{v} \otimes \mathbf{n}) : (\mathbf{u}_0 \otimes \mathbf{n}) - G^\top \nabla \mathbf{v} : (\mathbf{u}_0 \otimes \mathbf{n}) \, ds. \quad (3.20)$$

Here  $\mathbf{n}$  is the outward-facing unit normal on  $\Gamma_D$  and  $\mathcal{F}^v(\nabla \mathbf{u})$  is a linear viscous flux. For the vector Poisson equation, the viscous flux is given by  $\mathcal{F}^v(\nabla \mathbf{u}) = \nabla \mathbf{u}$ . For the primal formulation of linear elasticity, the viscous flux corresponds to the stress tensor  $\mathcal{F}^v(\nabla \mathbf{u}) = \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^\top) + \lambda \nabla \cdot \mathbf{u} \mathbb{I}$ . For the mixed formulation of linear elasticity, the  $(1, 1)$ -block of the system has viscous flux  $\mathcal{F}^v(\nabla \mathbf{u}) = \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^\top)$ .

From left to right, the terms in the surface integral in (3.19) are referred to as the penalty, consistency, and adjoint consistency terms. The quantity  $G$  is known as the homogeneity tensor,

$$G_{ijkl} = \frac{\partial}{\partial u_{k,l}} [\mathcal{F}^v(\nabla \mathbf{u})]_{ij}. \quad (3.21)$$

For the vector Poisson equation,  $G$  is the identity tensor, and in the case of linear elasticity,  $G$  corresponds to the compatibility tensor. We also define the adjoint product of  $G$  with  $\nabla \mathbf{v}$

$$[G^\top \nabla \mathbf{v}]_{kl} = G_{ijkl} v_{i,j}. \quad (3.22)$$

The average  $\{\cdot\}$  and jump  $[\![\cdot]\!]$  operators are defined for scalar, vector, and tensor arguments as follows. Let  $e$  be a facet of the mesh. For an interior facet, let  $K^-$

and  $K^+$  be the two mesh cells that share it, and let  $\mathbf{w}^-$  and  $\mathbf{w}^+$  be the traces of a function  $\mathbf{w}$  on  $e$  from  $K^-$  and  $K^+$ , respectively. On each facet we define

$$\{\mathbf{w}\}_e = \begin{cases} \frac{1}{2}(\mathbf{w}^- + \mathbf{w}^+) & e \in \Gamma_I, \\ \mathbf{w} & \text{otherwise,} \end{cases} \quad [\mathbf{w}]_e = \begin{cases} \mathbf{w}^- \otimes \mathbf{n}^- + \mathbf{w}^+ \otimes \mathbf{n}^+ & e \in \Gamma_I, \\ \mathbf{w} \otimes \mathbf{n} & \text{otherwise,} \end{cases} \quad (3.23)$$

where  $\mathbf{n}^-$ ,  $\mathbf{n}^+$  are the outward-facing unit normals on  $\Gamma_I$  from  $K^-$  and  $K^+$ , respectively. In order to ensure coercivity of  $a(\cdot, \cdot)$  as we do  $h$ - or  $p$ -refinement, the penalty term must be sufficiently large. The penalty coefficient  $\eta h_e^{-1}$  must be chosen inversely proportional to the normal spacing of GLL nodes near the facet, i.e.,  $\eta = \mathcal{O}(p(p+1))$  [91, 69]. For the reciprocal length scale in the direction normal to facet  $e$  we use

$$h_e^{-1} := |e| \left\{ |K|^{-1} \right\}_e. \quad (3.24)$$

The stiffness matrix that corresponds to  $a(\cdot, \cdot)$  in the SIPG formulation is obtained via direct stiffness summation over the cells and facets:

$$A = \sum_{K \in \mathcal{T}_h} R_K^\top A^K R_K + \sum_{e \in \Gamma_I \cup \Gamma_D} R_e^\top A^e R_e, \quad (3.25)$$

where  $A^K$  is the cell matrix discretizing the volume integral in  $K$ ,  $R_K$  is the Boolean restriction onto the DOFs of  $K$ ,  $A^e$  is the facet matrix discretizing the surface integral on  $e$ , and  $R_e$  is Boolean restriction onto the DOFs of the cells sharing facet  $e$ .

To illustrate the extension of our approach to the SIPG discretization, we consider again the scalar Poisson equation. The discrete problem is to find  $u_h \in V_h = \text{DQ}_p(\Omega) \subset L^2(\Omega)$ . On Cartesian cells, both  $A^K$  and  $A^e$  have a tensor-product structure of the form (2.13), with matrices of operators on the interval that can be sparsified by the FDM. To illustrate this, suppose that, for  $e \in \Gamma_I$ ,  $R_e$  reorders the DOFs such that the cells  $K^-$  and  $K^+$  share  $e$  along the  $d$ -th reference coordinate axis, while leaving the other axes consistently oriented on both cells. The facet matrices are

$$A^e = \begin{cases} E^e \otimes \hat{B} & \text{if } d = 2, \\ E^e \otimes \hat{B} \otimes \hat{B} & \text{if } d = 3, \end{cases} \quad (3.26)$$

where the interval facet matrix  $E^e$  is defined in terms of the coefficients  $\mu_j^K$  appearing in (2.13), the 1D basis functions  $\{\hat{\phi}_j\}$ , and their normal derivatives  $\frac{\partial}{\partial n} \hat{\phi}_j$  on  $\partial \hat{\mathcal{I}}$  (the usual derivative with a sign). When  $e \in \Gamma_D$ ,  $E^e \in \mathbb{R}^{(p+1) \times (p+1)}$  is given by

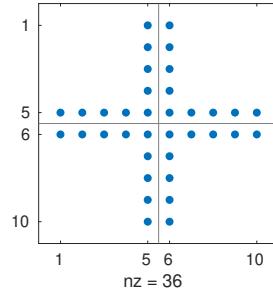
$$[E^e]_{ij} = \mu^e \left( \eta \hat{\phi}_i(\hat{x}^e) \hat{\phi}_j(\hat{x}^e) - \hat{\phi}_i(\hat{x}^e) \frac{\partial}{\partial n} \hat{\phi}_j(\hat{x}^e) - \frac{\partial}{\partial n} \hat{\phi}_i(\hat{x}^e) \hat{\phi}_j(\hat{x}^e) \right). \quad (3.27)$$

Here  $\mu^e = \mu_l^K$ , where  $\hat{x}_l$  is the reference coordinate normal to  $e$ , and  $\hat{x}^e \in \partial\hat{\mathcal{I}}$  describes the facet  $e$  as the image of the plane  $\hat{x}_l = \hat{x}^e$  under  $F_K$ . When  $e \in \Gamma_I$ ,  $E^e$  is a  $2 \times 2$  block matrix with blocks  $E_{rs}^e \in \mathbb{R}^{(p+1) \times (p+1)}$ ,  $r, s \in \{0, 1\}$ , given by

$$[E_{rs}^e]_{ij} = \frac{(-1)^{r-s}}{2} \left( \eta(\mu_0^e + \mu_1^e) \hat{\phi}_i(\hat{x}_r^e) \hat{\phi}_j(\hat{x}_s^e) - \mu_s^e \hat{\phi}_i(\hat{x}_r^e) \frac{\partial}{\partial n} \hat{\phi}_j(\hat{x}_s^e) - \mu_r^e \frac{\partial}{\partial n} \hat{\phi}_i(\hat{x}_r^e) \hat{\phi}_j(\hat{x}_s^e) \right). \quad (3.28)$$

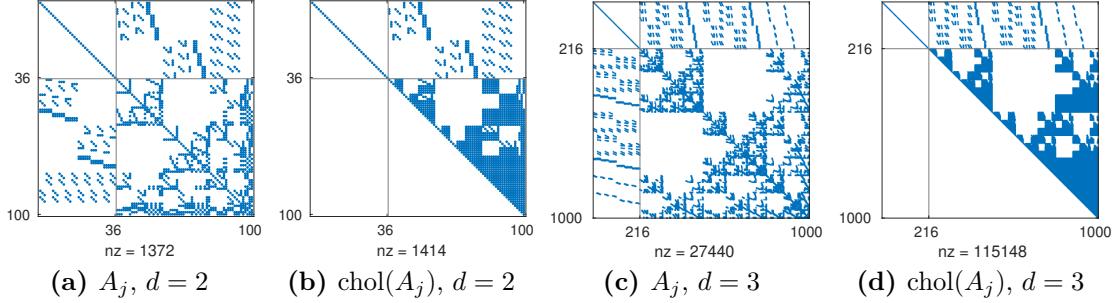
Here  $\mu_0^e = \mu_l^{K^-}$ ,  $\mu_1^e = \mu_m^{K^+}$ , where  $\hat{x}_l$  and  $\hat{x}_m$  are the reference directions normal to  $e$  on  $K^-$  and  $K^+$ , respectively. Similarly, the facet  $e$  is the image of  $\hat{x}_l = \hat{x}_0^e$  under  $F_{K^-}$  and that of  $\hat{x}_m = \hat{x}_1^e$  under  $F_{K^+}$ .

Some implementations of DQ <sub>$p$</sub>  do not feature an interior-interface decomposition and use the Gauß-Legendre (GL) nodal basis functions. The GL nodes do not include the endpoints, thus all basis functions have nonzero support at the facets, causing  $E^e$  to be dense. The matrices  $E^e$  are sparse for a basis with an interior-interface decomposition, such as the GLL Lagrange polynomials  $\{\ell_j\}$ , the hierarchical Lobatto polynomials  $\{l_j\}$ , and the FDM polynomials  $\{\hat{s}_j\}$ . Since  $\hat{\phi}_j(\pm 1)$  is nonzero for a single  $j \in \{0, p\}$ , each term in (3.28) and (3.27) corresponds to a nonzero entry, a nonzero row, and a nonzero column of  $E^e$ , respectively, as seen in Figure 3.1.



**Figure 3.1:** Nonzero structure for the interior facet matrix  $E^e$  on the interval ( $p = 4$ ).

Instead of diagonalizing the SIPG patch matrix as in [96], our FDM-based approach produces a sparse matrix with diagonal interior blocks on (possibly) unstructured vertex-star patches. Figure 3.2 shows the sparsity pattern of the matrix for the SIPG formulation of the Poisson equation on a Cartesian vertex-star patch in the FDM basis, along with its Cholesky factor. Here the matrix size is increased from  $(2p - 1)^d$  DOFs in the CG case to  $(2p + 2)^d$ . At low polynomial degrees, the interface DOFs form a large fraction of the total number, but the proportion decreases as  $p$  increases. The computational complexity analysis of Section 2.2.4 carries over to the SIPG case.



**Figure 3.2:** Nonzero structure of the SIPG stiffness matrix in the FDM basis  $A_j = L_j L_j^\top$  and its upper Cholesky factor  $L_j^\top$  for the Poisson problem on a Cartesian vertex-star patch with  $p = 4$ . Since the space is discontinuous, the number of DOFs in a patch is increased. The number of nonzeros in an interior row is  $3d + 1$ , since the interior DOFs are connected to each of the  $2d$  facets of their corresponding cell, plus  $d$  more facets from the adjacent cells.

For a general viscous flux, we construct an auxiliary separable form by expressing the cell integrals in terms of the reference coordinates

$$a_K(\mathbf{v}, \mathbf{u}) = \int_K G_{ijkl}^K \frac{\partial v_i}{\partial x_j} \frac{\partial u_k}{\partial x_l} \, d\mathbf{x} = \int_{\hat{K}} \hat{G}_{ijkl}^K \frac{\partial}{\partial \hat{x}_j} (v_i \circ F_K) \frac{\partial}{\partial \hat{x}_l} (u_k \circ F_K) \, d\hat{\mathbf{x}}, \quad (3.29)$$

where  $\hat{G}^K$  is the homogeneity tensor in the reference coordinates,

$$\hat{G}_{ijkl}^K = |\text{DF}_K| [\text{DF}_K^{-1}]_{jm} [\text{DF}_K^{-1}]_{ln} G_{imkn}^K. \quad (3.30)$$

The auxiliary form  $\tilde{a}(\cdot, \cdot)$  is constructed by approximating  $\hat{G}_{ijkl}^K$  with a piecewise constant tensor that discards the entries where  $i \neq k$  or  $j \neq l$ . The corresponding cell stiffness matrices become sparse in the FDM basis, and have a similar form as (2.13), except that the coefficients  $\mu_j^K$  are diagonal matrices that multiply each term through an additional Kronecker product. Hence, we expect that preconditioners based on the auxiliary form to be limited by the coupling between vector components, by the mesh geometry, and by how  $G^K$  varies within  $K$ .

This approach carries over to non-Cartesian cells for DG discretizations of the Poisson equation in the same way as the CG case. Unfortunately, the extension of our approach to vector-valued problems in  $H(\text{div})$  on non-Cartesian cells does not yield a good preconditioner. In this setting, we construct a block diagonal preconditioner separating the components of the DOFs, which are in the reference coordinates. For the vector Poisson problem on non-Cartesian cells, the Piola transform introduces off-diagonal contributions from the volume and surface terms, which does not occur on non-Piola-mapped elements, such as  $[\mathbf{Q}_p]^d$  and  $[\mathbf{DQ}_p]^d$ . The excluded terms are required for the surface integral terms to vanish for arguments with  $C^0$  continuity, and without them the preconditioner might become indefinite on non-Cartesian cells.

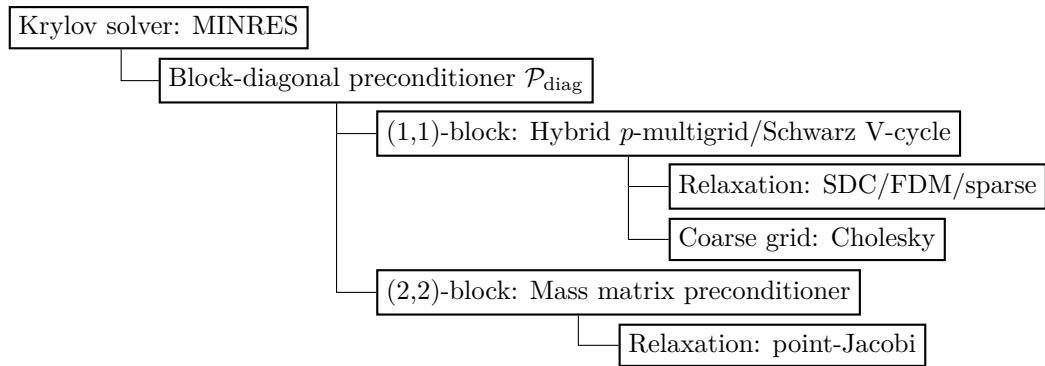
### 3.4 Results for mixed formulations of linear elasticity

We consider the same problem as in Table 3.1, with both a conforming  $[Q_p]^d \times DQ_{p-2}$  discretization and a non-conforming  $RT_p \times DQ_{p-1}$  discretization. For the  $H(\text{div})$ -conforming discretization, the normal components of the Dirichlet BCs are enforced strongly, while the tangential components of the BCs are weakly enforced with SIPG. Enforcing the normal conditions strongly is crucial for achieving a divergence-free solution in the Stokes limit  $\lambda = \infty$ . Following [69], we use  $\eta = p(p + 1)$  for the penalty coefficient. We restrict our experiment to Cartesian cells, so that the FDM/sparse relaxation is applicable to the  $H(\text{div})$ -conforming discretization.

We iteratively solve the discrete system (3.12) via MINRES with a symmetric positive definite block diagonal preconditioner,

$$\mathcal{P}_{\text{diag}} = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix}. \quad (3.31)$$

Here  $P_1$  is a preconditioner for the displacement block  $A$ , and  $P_2$  is a preconditioner for the scaled pressure mass matrix  $(\mu^{-1} + \lambda^{-1})M_p$ . For  $P_1$  we employ the hybrid  $p$ -multigrid/Schwarz method with the SDC/FDM/sparse relaxation and  $[Q_1]^d$  as the coarse space. In our tests, we discretize the pressure space with the GL basis, and employ point-Jacobi on the pressure mass matrix, i.e.,  $P_2 = (\mu^{-1} + \lambda^{-1}) \text{diag}(M_p)$ . When  $\mathcal{T}_h$  consists of Cartesian cells,  $M_p = \text{diag}(M_p)$  in the GL basis. The solver is illustrated in Figure 3.3.



**Figure 3.3:** Solver diagram for the mixed linear elasticity problem.

In Table 3.2 we present MINRES iteration counts for the same configuration considered in Table 3.1 in Section 3.1, using the  $[Q_p]^d \times DQ_{p-2}$  and  $RT_p \times DQ_{p-1}$  elements, respectively. Both discretizations yield robust iteration counts with

**Table 3.2:** MINRES iteration counts for the mixed linear elasticity problem, using the solver in Figure 3.3.

$d$	$p \setminus \lambda$	$[Q_p]^d \times DQ_{p-2}$					$RT_p \times DQ_{p-1}$				
		$10^0$	$10^1$	$10^2$	$10^3$	$\infty$	$10^0$	$10^1$	$10^2$	$10^3$	$\infty$
2	3	28	40	43	43	43	25	36	39	40	40
	7	31	45	50	51	51	28	40	43	45	45
	15	34	50	57	57	57	30	43	48	48	48
	31	36	53	64	65	65	31	45	51	51	51
3	3	44	67	75	76	76	34	50	55	56	56
	7	50	83	96	97	98	39	58	63	65	65
	15	53	88	111	118	119	41	63	70	70	70

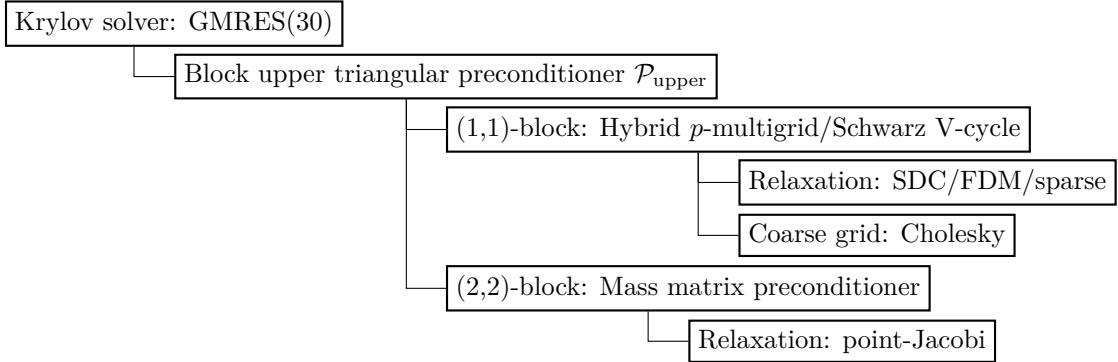
respect to  $\lambda$ ; the iterations grow with the former discretization much more quickly than the latter, especially in 3D.

The solver configuration shown in Figure 3.3 is optimized for memory usage, employing a block diagonal preconditioner so that the short-term recurrences of MINRES may be exploited. If one is willing to trade memory for time, one may consider an alternative configuration shown in Figure 3.4 employing right-preconditioned GMRES [85] with a block upper triangular preconditioner,

$$\mathcal{P}_{\text{upper}} = \begin{bmatrix} P_1 & B^\top \\ 0 & -P_2 \end{bmatrix}, \quad (3.32)$$

which requires a single application of each  $P_1^{-1}$ ,  $P_2^{-1}$ , and  $B^\top$  per GMRES iteration. Here  $P_1, P_2$  refer to the same positive definite preconditioners used for  $\mathcal{P}_{\text{diag}}$  in (3.31), therefore a minus sign is needed in front of  $P_2$ , since the pressure Schur complement is negative definite. The GMRES iteration counts are presented in Table 3.3. For 3D, in the incompressible regime, we observe that the GMRES solver requires fewer than half of the iterations required by the MINRES solver.

In Table 3.4 we study the performance of our solver on an unstructured mesh. We consider the  $[Q_p]^d \times DQ_{p-2}$  discretization of incompressible linear elasticity ( $\lambda = \infty$ ). We prescribe  $\mu = 1$ , a uniform downwards body force  $\mathbf{B} = -0.02\mathbf{e}_2$ , and homogeneous Dirichlet BCs on the displacement on the holes of the domain. The three-dimensional mesh is obtained via extrusion by 16 layers of the two-dimensional mesh. The iteration counts follow the same pattern as before for this element: they are not  $p$ -robust as expected, but they remain modest even at very high degrees.



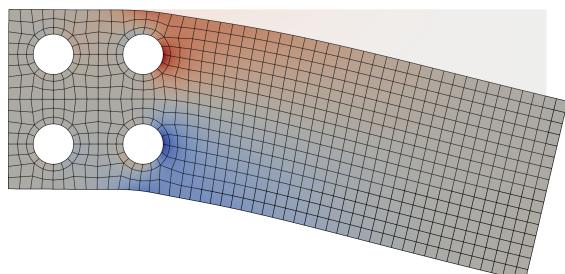
**Figure 3.4:** Solver diagram for the mixed linear elasticity problem that trades memory for iteration counts.

**Table 3.3:** GMRES iteration counts for the mixed linear elasticity problem, using the solver in Figure 3.4.

$d$	$p \setminus \lambda$	$[\mathbf{Q}_p]^d \times \mathbf{D}\mathbf{Q}_{p-2}$					$\mathbf{R}\mathbf{T}_p \times \mathbf{D}\mathbf{Q}_{p-1}$				
		$10^0$	$10^1$	$10^2$	$10^3$	$\infty$	$10^0$	$10^1$	$10^2$	$10^3$	$\infty$
2	3	17	23	25	26	26	14	20	22	22	22
	7	18	25	27	28	28	16	22	24	24	24
	15	20	27	33	34	34	17	23	26	26	26
	31	21	30	38	38	39	18	24	28	28	28
3	3	24	32	33	34	34	17	23	25	25	25
	7	27	35	38	38	38	21	26	29	29	29
	15	28	38	44	46	46	22	28	31	31	32

**Table 3.4:** GMRES iteration counts for the mixed formulation of the incompressible linear elasticity problem on the unstructured mesh shown here, using the solver in Figure 3.4.

$d$	$p$	# DOFs	Iter.
2	3	17 466	26
	7	104 250	29
	15	499 002	35
	31	2 173 242	44
3	3	588 927	39
	7	7 876 575	47
	11	31 236 927	51



# 4

## Optimal complexity multigrid solvers for the de Rham complex<sup>1</sup>

### 4.1 Introduction

In this chapter we introduce solvers for high-order finite element discretizations of the following boundary value problems posed on a bounded Lipschitz domain  $\Omega \subset \mathbb{R}^d$  in  $d = 3$  dimensions<sup>2</sup>:

$$\beta u - \nabla \cdot (\alpha \nabla u) = f \text{ in } \Omega, \quad u = 0 \text{ on } \Gamma_D, \quad \alpha \nabla u \cdot \mathbf{n} = 0 \text{ on } \Gamma_N; \quad (4.1)$$

$$\beta \mathbf{u} + \nabla \times (\alpha \nabla \times \mathbf{u}) = \mathbf{f} \text{ in } \Omega, \quad \mathbf{u} \times \mathbf{n} = 0 \text{ on } \Gamma_D, \quad \alpha \nabla \times \mathbf{u} \times \mathbf{n} = 0 \text{ on } \Gamma_N; \quad (4.2)$$

$$\beta \mathbf{u} - \nabla (\alpha \nabla \cdot \mathbf{u}) = \mathbf{f} \text{ in } \Omega, \quad \mathbf{u} \cdot \mathbf{n} = 0 \text{ on } \Gamma_D, \quad \alpha \nabla \cdot \mathbf{u} = 0 \text{ on } \Gamma_N; \quad (4.3)$$

where  $\alpha, \beta : \Omega \rightarrow \mathbb{R}_+$  are problem parameters,  $\Gamma_D \subseteq \partial\Omega$ , and  $\Gamma_N = \partial\Omega \setminus \Gamma_D$ . For  $\alpha = \beta = 1$ , these equations are the so-called *Riesz maps* associated with subsets of the spaces  $H(\text{grad}, \Omega) = H^1(\Omega)$ ,  $H(\text{div}, \Omega)$  and  $H(\text{curl}, \Omega)$  respectively. These function spaces are defined as:

$$H(\text{grad}, \Omega) := \left\{ v \in L^2(\Omega) : \text{grad } v \in [L^2(\Omega)]^3 \right\}, \quad (4.4)$$

$$H(\text{curl}, \Omega) := \left\{ \mathbf{v} \in [L^2(\Omega)]^3 : \text{curl } \mathbf{v} \in [L^2(\Omega)]^3 \right\}, \quad (4.5)$$

$$H(\text{div}, \Omega) := \left\{ \mathbf{v} \in [L^2(\Omega)]^3 : \text{div } \mathbf{v} \in L^2(\Omega) \right\}. \quad (4.6)$$

For brevity we shall write  $H(\text{grad}) = H(\text{grad}, \Omega)$  etc. where there is no potential confusion. Our problems of interest (4.1)-(4.3) often arise as subproblems in the

<sup>1</sup>This chapter is extracted from [19], submitted to SIAM J. Sci. Comput.

<sup>2</sup>Our solver strategy extends to  $d \in \mathbb{N}_+$ , but we describe the case  $d = 3$  for concreteness.

construction of fast preconditioners for more complex systems involving solution variables in (4.4)-(4.6) [45, 70], and are the canonical maps for transforming derivatives to gradients in optimization problems posed in these spaces [90].

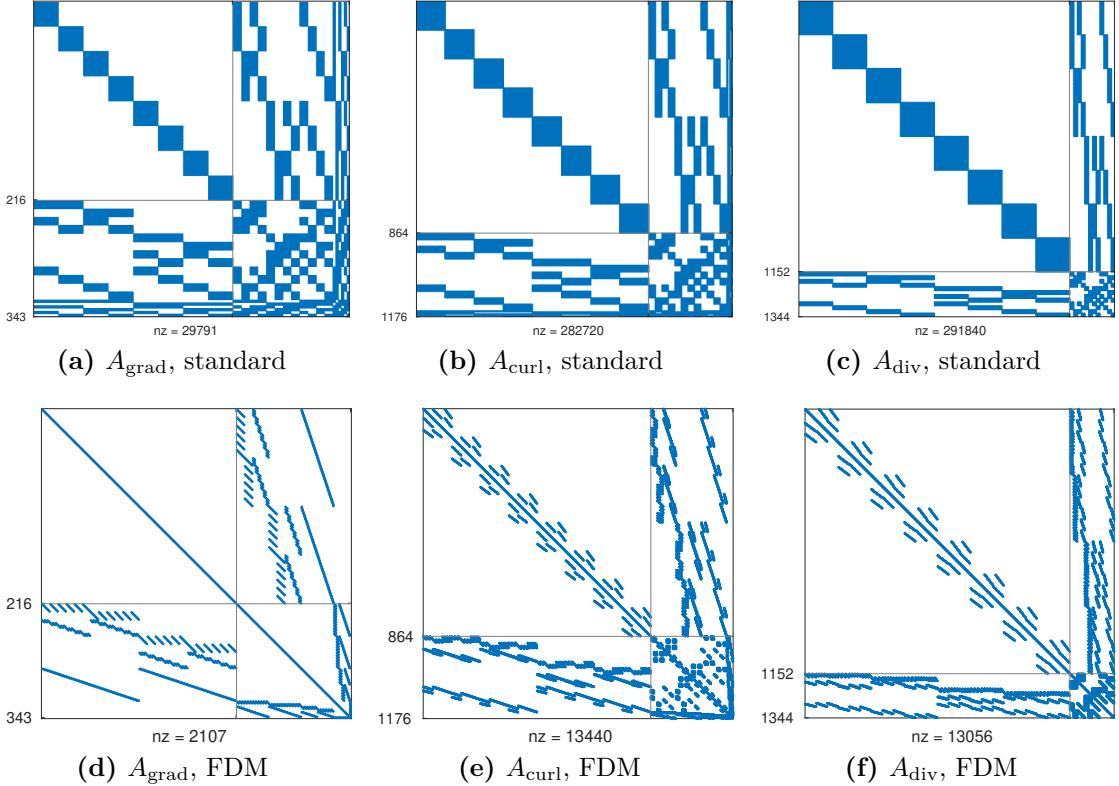
The spaces (4.4)-(4.6) and their discretizations are organized in the  $L^2$  de Rham complex

$$\begin{array}{ccccccc} H(\text{grad}) & \xrightarrow{\text{grad}} & H(\text{curl}) & \xrightarrow{\text{curl}} & H(\text{div}) & \xrightarrow{\text{div}} & L^2 \\ \downarrow & & \downarrow & & \downarrow & & \downarrow \\ Q_p & \xrightarrow{\text{grad}} & \text{NCE}_p & \xrightarrow{\text{curl}} & \text{NCF}_p & \xrightarrow{\text{div}} & \text{DQ}_{p-1} \end{array}, \quad (4.7)$$

where the complex property means that the image of one map (grad, curl, or div) is contained in the kernel of the next, e.g.,  $\text{grad}(H(\text{grad})) \subset \ker(\text{curl}, H(\text{curl}))$ . Here  $Q_p \subset H(\text{grad})$ ,  $\text{NCE}_p \subset H(\text{curl})$ ,  $\text{NCF}_p \subset H(\text{div})$ , and  $\text{DQ}_p \subset L^2$  are piecewise polynomial spaces of maximum polynomial degree  $p$  on a mesh  $\mathcal{T}_h$  of tensor-product cells (hexahedra) used for the finite element discretization of (4.1)-(4.3).  $\text{NCE}_p$  and  $\text{NCF}_p$  are the discrete function spaces induced by the Nédélec edge elements and face elements [72] respectively.

Standard multigrid relaxation schemes such as point-Jacobi and Gauß-Seidel are not effective for high-order discretizations of these problems; these relaxations are only effective for (4.1) at low-order, and are never effective for (4.2) and (4.3). In particular, the convergence of the multigrid scheme is not robust with respect to  $\alpha$ ,  $\beta$ , or  $p$ . However, space decompositions that experimentally exhibit convergence robust to  $\alpha$ ,  $\beta$  and  $p$  are known, proposed by Pavarino [75], Arnold, Falk & Winther (AFW) [9], and Hiptmair [44]. The relaxation schemes these space decompositions induce require the solution of patchwise problems e.g., gathering all DOFs around each vertex, edge, or face.

Solving these patch problems becomes challenging as  $p$  increases. The storage and factorization of the patch matrices becomes prohibitively expensive, since standard basis functions for  $Q_p$ ,  $\text{NCE}_p$ , and  $\text{NCF}_p$  introduce coupling between all interior DOFs within a cell, causing  $\mathcal{O}(p^d) \times \mathcal{O}(p^d)$  dense blocks in the matrix. The Cholesky factorization of such matrices takes  $\mathcal{O}(p^{3d})$  flops and  $\mathcal{O}(p^{2d})$  storage. These complexity estimates severely limit the use of very high polynomial degrees. Here we will present an alternative strategy for solving these subproblems with  $\mathcal{O}(p^{d+1})$  flops and  $\mathcal{O}(p^d)$  storage. These complexity bounds are optimal in the context of Krylov methods: they match the computational complexity of applying the discretized operator via sum-factorization [73].



**Figure 4.1:** Sparsity patterns for the  $2 \times 2 \times 2$  Pavarino–Arnold–Falk–Winther patch problem around a vertex ( $p = 4$ ), for (a, d) (4.1) (b, e) (4.2) (c, f) (4.3). The top row (a)-(c) uses standard (GLL/GL) elements, while the bottom row (d)-(f) uses the FDM variants we propose.

Our strategy relies on three main components. First, we propose new finite elements (with different degrees of freedom and basis functions) for building  $\text{NCE}_p$ ,  $\text{NCF}_p$ , and  $\text{DQ}_p$  with useful orthogonality properties on the reference cell. The elements are derived from a finite element for  $\text{Q}_p$  introduced in Chapter 2 via tensor-product construction. The new finite elements are simple and convenient to implement; by their orthogonality properties, the patch matrices on Cartesian cells are sparse. For example, the sparsity patterns of a vertex-patch problem for (4.1)–(4.3) with  $p = 4$  are shown in Figure 4.1, for both the standard GLL/GL finite elements and our proposals (referred to as ‘FDM’ elements, as they are inspired by the fast diagonalization method [65]).

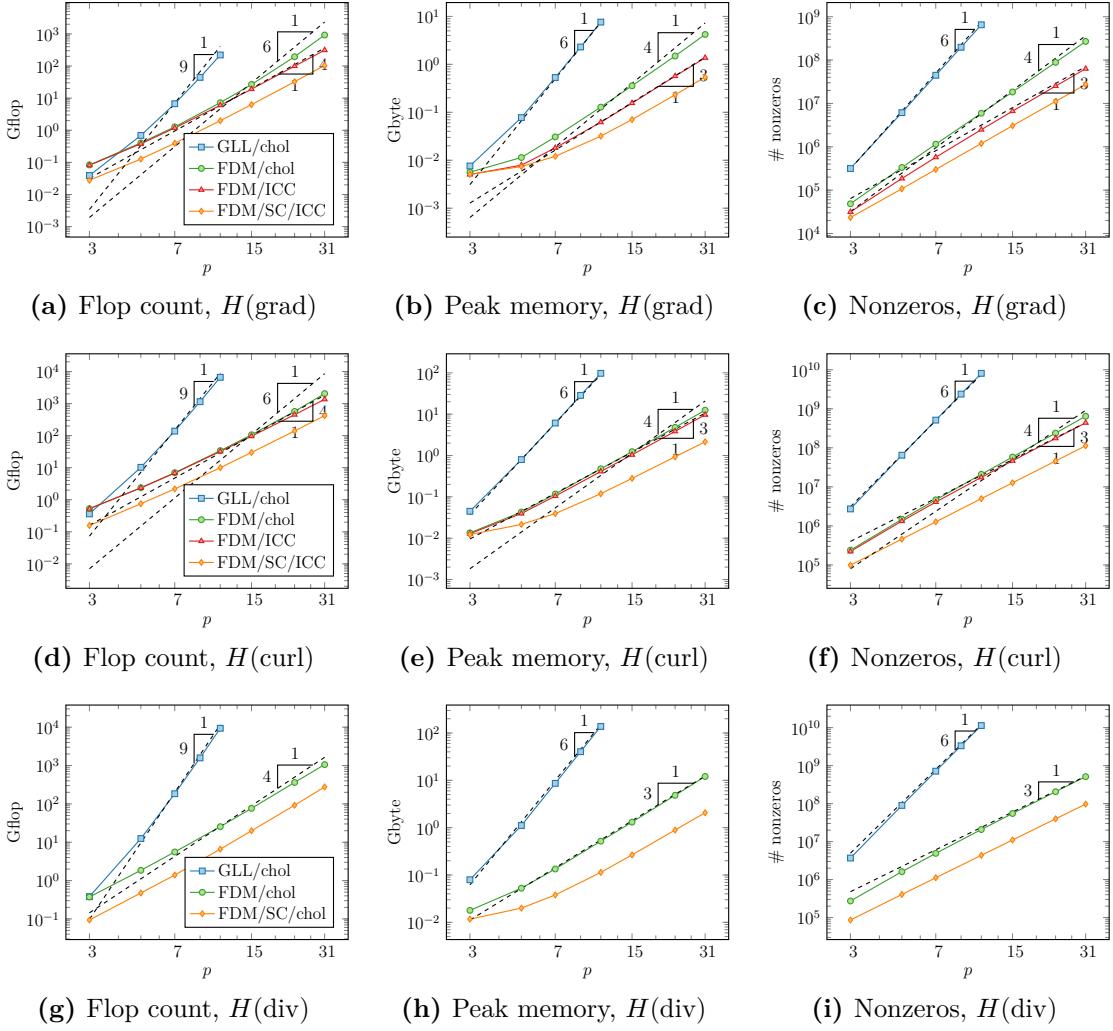
The second main component is to ensure optimal fill-in in the factorization of the patch problems. The Cholesky factorizations of the matrices shown in Figure 4.1(d-f) are sparse, even sparser than the Cholesky factorization of a low-order discretization on a grid with the same number of DOFs. However, this still incurs suboptimal setup and storage costs of  $\mathcal{O}(p^{2d})$  and  $\mathcal{O}(p^{d+1})$  respectively. We overcome this through the

choice of Hiptmair space decompositions, which require smaller patch solves around edges and faces, and through the careful use of incomplete factorizations of vertex patch problems. Choosing edge patches in  $H(\text{curl})$  and face patches in  $H(\text{div})$  (along with patches for scalar and vector potential fields, respectively) results in patch factors with fill-in of optimal space complexity of  $\mathcal{O}(p^d)$ . However, this does not address the case of  $H(\text{grad})$ . A natural strategy is to employ incomplete Cholesky (ICC) factorizations. The zero-fill-in ICC factorization does not work: the factorization may fail, and even when it is computed it may not offer an effective relaxation. Instead, we use a nested dissection ordering and impose the sparsity pattern associated with static condensation (i.e., when the interior DOFs are eliminated) by padding with zeros the interface block of the patch matrices, without carrying out the elimination. Computational experiments indicate that this still offers an excellent relaxation, while achieving optimality in both flops and storage.

The third main component is the use of auxiliary operators. The patch matrices assembled with the FDM elements are not sparse for distorted cells and/or spatially-varying  $\alpha$  or  $\beta$ . To overcome this, we apply our preconditioner to an auxiliary operator which is constructed so that the patch matrices are sparse. The auxiliary operator employed in this work is different to that in Chapter 2 for solving (4.1): here, we construct the auxiliary operator by taking diagonal approximations of mass matrices involved in the definition of the stiffness matrices. The implementation of this auxiliary operator is more convenient for  $H(\text{curl})$  and  $H(\text{div})$ . The quality of this approximation depends on the mesh distortion and the degree of any spatial variation in coefficients. Computational experiments suggest a slow growth of the equivalence constants with respect to  $p$ , but this growth is not fast enough to affect optimality of the solver.

With these components we achieve optimal complexity solvers. To illustrate this, we show in Figure 4.2 the number of flops and bytes required to solve the Riesz maps (4.1)–(4.3) with CG and  $\alpha = \beta = 1$  on an unstructured hexahedral mesh. The setup is described in more detail in Section 4.5.1.

For patch matrices assembled in the GLL basis (and hence with dense blocks) and factorized, the time complexity is  $\mathcal{O}(p^9)$ , and the space complexity is  $\mathcal{O}(p^6)$ , as expected. When assembled in the FDM basis in a sparse matrix format, the time complexity is reduced to approximately  $\mathcal{O}(p^5)$  (the empirical slope between  $p = 23$  and  $p = 31$  is  $\mathcal{O}(p^{5.22})$  in fig. 4.2(a)), and the space complexity is  $\mathcal{O}(p^4)$  in fig. 4.2(c). The peak memory used, as shown in fig. 4.2(b), is not yet scaling as  $\mathcal{O}(p^4)$ , indicating that the sparse Cholesky factors are not yet the dominant term in memory. These complexities are further reduced when the FDM elements



**Figure 4.2:** Flop counts, peak memory usage, and nonzeros in the sparse matrices and patch factors in the solution of the Riesz maps on a 3D unstructured mesh with 27 cells.

are combined with ICC factorizations: the time complexity becomes  $\mathcal{O}(p^4)$  and the space complexity  $\mathcal{O}(p^3)$ , as desired.

A fourth, optional ingredient is the use of static condensation on the sparse auxiliary operator, eliminating the interior DOFs to yield patch problems posed only on the interfaces between cells. As shown in Figure 4.2, the time and space complexities remain  $\mathcal{O}(p^4)$  and  $\mathcal{O}(p^3)$  respectively, but there are several advantages nevertheless. First, the peak memory required is substantially reduced, by a factor of 2.5–5.3 for  $p = 31$ , thus enabling the use of modern HPC hardware with limited RAM available per core. This reduction is caused by the fact that the patch matrices have fewer DOFs. Second, the volume of data shared between processes is reduced, to the same order as that required by operator application. This reduces parallel

communication. These advantages result in a substantially faster solver.

### 4.1.1 Related work

The fast diagonalization method (FDM) [65] is a matrix factorization that enables the direct solution of problems such as (4.1) in optimal complexity, whenever separation of variables is applicable (i.e., only on certain domains and for certain coefficients  $\alpha, \beta$ ). The FDM breaks down the  $d$ -dimensional problem into a sequence of one-dimensional generalized eigenvalue problems. Our construction of finite elements with orthogonality properties below in Section 4.2 was inspired by the FDM.

Hiptmair [42, 43] studied the extension of the FDM to the  $H(\text{curl})$  Riesz map (4.2). The algorithm relies on the elimination of one vector component. The reduced system for the remaining vector component can be solved directly with the FDM. With this approach, one has to solve one generalized eigenvalue problem for every cell of the mesh, whereas in the  $H(\text{grad})$  case (4.1) one solves a single generalized eigenvalue problem on the reference cell. Unfortunately, this strategy does not extend well to the case with three vector components, as the nested Schur complement is no longer a sum of three Kronecker products.

Low-order-refined (LOR) preconditioners [73, 29] are well known to be spectrally equivalent [21] to discrete high-order operators in  $H(\text{grad})$ . For problems in  $H(\text{curl})$  and  $H(\text{div})$ , LOR preconditioners have been studied in [32, 78]. The auxiliary low-order problem is sparse, but even for the  $H(\text{grad})$  case (4.1), devising efficient relaxations is challenging [77]. This is because the low-order-refined grid is anisotropic, and pointwise smoothers on the auxiliary low-order problem become ineffective as  $p$  increases. To overcome this, Pazner [77] applies patchwise multigrid with ICC relaxations to (4.1), relying on a good ordering of the DOFs. This method has not been studied for (4.2) or (4.3), to the best of our knowledge. Instead, Pazner, Kolev & Dohrmann apply the AMS and ADS algebraic multigrid solvers of Hypre [36, 56, 57] to the auxiliary low-order problem, which implement the strategy proposed by Hiptmair & Xu [46]. These LOR approaches do not naturally combine with static condensation, since the density of the Schur complement arising from the elimination of DOFs that are interior to the original grid counteracts the sparsity offered by the low-order problem.

In common with the work of Schöberl & Zaglmayr [88, 99], we obtain basis functions with local complete sequence properties, i.e., at the interior DOF level the discrete spaces form a subcomplex of the de Rham complex. Our construction of the basis functions for  $\text{NCE}_p$ ,  $\text{NCF}_p$ , and  $\text{DQ}_p$  from that of  $\text{Q}_p$  is the same, but

we start from a different basis for  $Q_p$ ; Schöberl & Zaglmayr start with integrated Lagrange polynomials on the reference interval, whereas we employ the FDM element proposed in Chapter 2. This choice yields greater sparsity, because the mass matrix on the reference interval decouples the interior DOFs.

## 4.2 Sparsity-promoting discretization

Our goal is to construct finite elements so that the discretizations of (4.1)–(4.3) are sparse even at high  $p$ . Specifically, we desire that the number of nonzeros in the stiffness matrix is of the same order as its number of rows or columns, in certain cases (Cartesian cells and cellwise-constant coefficients).

### 4.2.1 Exterior calculus notation and weak formulation

To unify the discussion of (4.1)–(4.3), we adopt the language of the finite element exterior calculus (FEEC) [5]. We recognize functions in  $H(\text{grad})$ ,  $H(\text{curl})$ ,  $H(\text{div})$ , and  $L^2$  as differential  $k$ -forms for  $k = 0, 1, 2, 3$  respectively, writing  $H(d^k, \Omega) = H\Lambda^k(\Omega)$ , with the exterior derivative  $d^k$  corresponding to  $d^0 = \text{grad}$ ,  $d^1 = \text{curl}$ ,  $d^2 = \text{div}$ ,  $d^3 = \text{null}$  (the zero map). We define the spaces

$$V^k := \{v \in H\Lambda^k(\Omega) : \text{tr } v = 0 \text{ on } \Gamma_D\}, \quad (4.8)$$

where the trace operator on  $\partial\Omega$  is  $\text{tr } v = v|_{\partial\Omega}$  for 0-forms,  $\text{tr } v = (v \times \mathbf{n})|_{\partial\Omega}$  for 1-forms,  $\text{tr } v = (v \cdot \mathbf{n})|_{\partial\Omega}$  for 2-forms, and  $\text{tr } v = 0$  for 3-forms [5], where  $\mathbf{n}$  denotes the outward-facing unit normal on  $\partial\Omega$ . In FEEC notation, the discrete spaces on the bottom row of (4.7) are denoted as  $Q_p^- \Lambda^k(\mathcal{T}_h)$ . We denote the discrete spaces employed as  $V_{h,p}^k := Q_p^- \Lambda^k(\mathcal{T}_h) \cap V^k$ .

With this notation, the common weak formulation of (4.1)–(4.3) is to find  $u \in V^k$  such that

$$a^k(v, u) := (v, \beta u)_\Omega + (d^k v, \alpha d^k u)_\Omega = F(v) \text{ for all } v \in V^k, \quad (4.9)$$

for  $k \in \{0, 1, 2\}$ , and where  $(\cdot, \cdot)_\Omega$  denotes the  $L^2(\Omega)$ -inner product. The discretization we consider is to find  $u_h \in V_{h,p}^k$  such that

$$a^k(v_h, u_h) = F(v_h) \text{ for all } v_h \in V_{h,p}^k. \quad (4.10)$$

### 4.2.2 Orthogonal bases for $H(\text{grad})$ and $L^2$ on the interval

In Section 2.2.2 we introduced new DOFs for  $P_p(\hat{\mathcal{I}}) \subset H(\text{grad}, \hat{\mathcal{I}})$ , the basis for the dual (2.21) consists of point evaluation at the vertices and integral moments against interior basis functions that are orthogonal in both the  $L^2(\hat{\mathcal{I}})$ - and  $H(\text{grad}, \hat{\mathcal{I}})$ -inner products (2.20). We recall our notation for this FDM-inspired basis  $\{\hat{s}_j\}_{j=0}^p$  and the interior and interface index sets  $I = 1 : p - 1$  and  $\Gamma = \{0, p\}$ . Through standard tensor-product construction we obtain an interior-orthogonal basis for  $V_{h,p}^0 = Q_p$  for which discretizations of (4.1) on Cartesian cells (rectangular hexahedra) are sparse, as sparse as a low-order discretization, with a sparser Cholesky factorization. By introduction of an additional basis for  $L^2(\hat{\mathcal{I}})$ , we will construct tensor-product bases for  $V_{h,p}^k$ ,  $k \in 1 : 3$  with interior-orthogonality properties that promote sparsity on Cartesian cells.

We first define a basis  $\{\hat{r}_j\}_{j=0}^{p-1}$  for  $\text{DP}_{p-1}(\hat{\mathcal{I}}) \subset L^2(\hat{\mathcal{I}})$  in one dimension, by exploiting the fact that  $d(P_p) = \text{DP}_{p-1}$  (where  $d$  is the one-dimensional derivative operator). We define the basis for  $\text{DP}_{p-1}(\hat{\mathcal{I}})$  as the derivatives of the interior basis functions for  $P_p$  defined above,  $\{\hat{s}'_j\}_{j \in I}$ , augmented with the constant function:

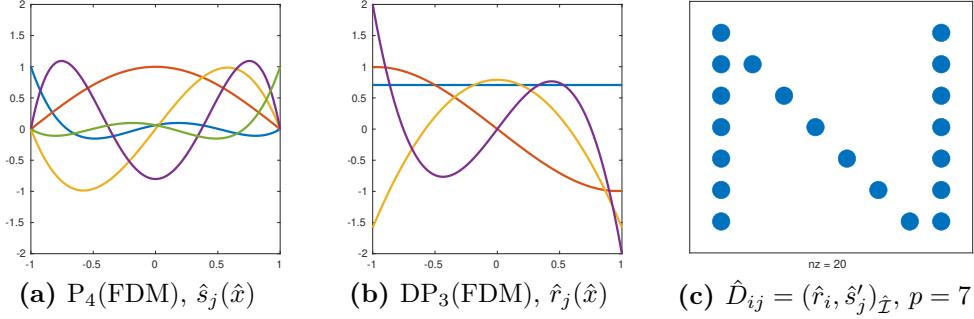
$$\hat{r}_j := \begin{cases} \lambda_0^{-1/2} & j = 0, \\ \lambda_j^{-1/2} \hat{s}'_j & j = 1, \dots, p-1. \end{cases} \quad (4.11)$$

Here  $\lambda_0 := |\hat{\mathcal{I}}|$ , and  $\lambda_j := (\hat{s}'_j, \hat{s}'_j)_{\hat{\mathcal{I}}}$  for  $j \in I$  are required to normalize the basis. By construction, the set  $\{\hat{s}'_j\}_{j \in I}$  is orthogonal in the  $L^2(\hat{\mathcal{I}})$ -inner product. In addition  $(\hat{r}_0, \hat{r}_j)_{\hat{\mathcal{I}}} = 0$  for  $j \in I$ , which follows from the fact that the interior basis functions  $\{\hat{s}_j\}_{j \in I}$  vanish at the endpoints of  $\hat{\mathcal{I}}$ :

$$(\hat{r}_0, \hat{r}_j)_{\hat{\mathcal{I}}} = (\lambda_0 \lambda_j)^{-1/2} \int_{\hat{\mathcal{I}}} \hat{s}'_j \, d\hat{x} = (\lambda_0 \lambda_j)^{-1/2} (\hat{s}_j(1) - \hat{s}_j(-1)) = 0. \quad (4.12)$$

This dependence of the basis of  $\text{DP}_{p-1}$  on that of  $P_p$  becomes useful in higher dimensions for enforcing interior-orthogonality in  $H(\text{curl})$  and  $H(\text{div})$ .

Figure 4.3 shows the FDM basis functions for  $P_p$  and  $\text{DP}_{p-1}$  and the nonzero structure of the one-dimensional differentiation matrix  $\hat{D} \in \mathbb{R}^{p \times (p+1)}$  that interpolates the derivatives of  $P_p$  onto  $\text{DP}_{p-1}$  in the FDM bases.



**Figure 4.3:** Plots of the FDM basis functions for  $p = 4$  on the reference interval  $\hat{\mathcal{I}}$ .

#### 4.2.3 Orthogonal bases for the de Rham complex

With bases for  $P_p(\hat{\mathcal{I}})$  and  $\text{DP}_{p-1}(\hat{\mathcal{I}})$ , we construct the basis functions for  $V_{h,p}^k(\hat{K})$ ,  $k \in 0 : 3$ , on the reference hexahedron  $\hat{K} := \hat{\mathcal{I}}^3$  in the usual tensor-product fashion [72, 6]:

$$V_{h,p}^0(\hat{K}) = Q_p(\hat{K}) = P_p(\hat{\mathcal{I}}) \otimes P_p(\hat{\mathcal{I}}) \otimes P_p(\hat{\mathcal{I}}), \quad (4.13)$$

$$V_{h,p}^1(\hat{K}) = \text{NCE}_p(\hat{K}) = \begin{bmatrix} \text{DP}_{p-1}(\hat{\mathcal{I}}) \otimes P_p(\hat{\mathcal{I}}) \otimes P_p(\hat{\mathcal{I}}) \\ P_p(\hat{\mathcal{I}}) \otimes \text{DP}_{p-1}(\hat{\mathcal{I}}) \otimes P_p(\hat{\mathcal{I}}) \\ P_p(\hat{\mathcal{I}}) \otimes P_p(\hat{\mathcal{I}}) \otimes \text{DP}_{p-1}(\hat{\mathcal{I}}) \end{bmatrix}, \quad (4.14)$$

$$V_{h,p}^2(\hat{K}) = \text{NCF}_p(\hat{K}) = \begin{bmatrix} P_p(\hat{\mathcal{I}}) \otimes \text{DP}_{p-1}(\hat{\mathcal{I}}) \otimes \text{DP}_{p-1}(\hat{\mathcal{I}}) \\ \text{DP}_{p-1}(\hat{\mathcal{I}}) \otimes P_p(\hat{\mathcal{I}}) \otimes \text{DP}_{p-1}(\hat{\mathcal{I}}) \\ \text{DP}_{p-1}(\hat{\mathcal{I}}) \otimes \text{DP}_{p-1}(\hat{\mathcal{I}}) \otimes P_p(\hat{\mathcal{I}}) \end{bmatrix}, \quad (4.15)$$

$$V_{h,p}^3(\hat{K}) = \text{DQ}_{p-1}(\hat{K}) = \text{DP}_{p-1}(\hat{\mathcal{I}}) \otimes \text{DP}_{p-1}(\hat{\mathcal{I}}) \otimes \text{DP}_{p-1}(\hat{\mathcal{I}}). \quad (4.16)$$

We introduce tensor-product bases for each finite element space in (4.7). For  $Q_p(\hat{K})$  we define  $\{\hat{\psi}_{ijl}\}$  as

$$\hat{\psi}_{ijl} := \hat{s}_i(\hat{x}_1)\hat{s}_j(\hat{x}_2)\hat{s}_l(\hat{x}_3), \quad (i, j, l) \in (0 : p)^3. \quad (4.17)$$

For  $\text{NCE}_p(\hat{K})$  we define  $\{\hat{\Psi}_{ijl}^{(m)}\}$  as

$$\begin{aligned} \hat{\Psi}_{ijl}^{(1)} &:= \hat{r}_i(\hat{x}_1)\hat{s}_j(\hat{x}_2)\hat{s}_l(\hat{x}_3)\mathbf{e}_1, \quad (i, j, l) \in (0 : p-1) \times (0 : p) \times (0 : p), \\ \hat{\Psi}_{ijl}^{(2)} &:= \hat{s}_i(\hat{x}_1)\hat{r}_j(\hat{x}_2)\hat{s}_l(\hat{x}_3)\mathbf{e}_2, \quad (i, j, l) \in (0 : p) \times (0 : p-1) \times (0 : p), \\ \hat{\Psi}_{ijl}^{(3)} &:= \hat{s}_i(\hat{x}_1)\hat{s}_j(\hat{x}_2)\hat{r}_l(\hat{x}_3)\mathbf{e}_3, \quad (i, j, l) \in (0 : p) \times (0 : p) \times (0 : p-1). \end{aligned} \quad (4.18)$$

For  $\text{NCF}_p(\hat{K})$  we define  $\{\hat{\Phi}_{ijl}^{(m)}\}$  as

$$\begin{aligned} \hat{\Phi}_{ijl}^{(1)} &:= \hat{s}_i(\hat{x}_1)\hat{r}_j(\hat{x}_2)\hat{r}_l(\hat{x}_3)\mathbf{e}_1, \quad (i, j, l) \in (0 : p) \times (0 : p-1) \times (0 : p-1), \\ \hat{\Phi}_{ijl}^{(2)} &:= \hat{r}_i(\hat{x}_1)\hat{s}_j(\hat{x}_2)\hat{r}_l(\hat{x}_3)\mathbf{e}_2, \quad (i, j, l) \in (0 : p-1) \times (0 : p) \times (0 : p-1), \\ \hat{\Phi}_{ijl}^{(3)} &:= \hat{r}_i(\hat{x}_1)\hat{r}_j(\hat{x}_2)\hat{s}_l(\hat{x}_3)\mathbf{e}_3, \quad (i, j, l) \in (0 : p-1) \times (0 : p-1) \times (0 : p). \end{aligned} \quad (4.19)$$

For  $\mathrm{DQ}_{p-1}(\hat{K})$  we define  $\{\hat{\phi}_{ijl}\}$  as

$$\hat{\phi}_{ijl} := \hat{r}_i(\hat{x}_1)\hat{r}_j(\hat{x}_2)\hat{r}_l(\hat{x}_3), \quad (i, j, l) \in (0 : p - 1)^3. \quad (4.20)$$

By construction, the interior basis functions of these four bases are orthonormal in the  $L^2(\hat{K})$ -inner product. Moreover, each horizontal arrow in (4.7) gives rise to the following relations between the interior basis functions, where  $(i, j, l) \in (1 : p - 1)^3$ :

$$\mathrm{grad} \hat{\psi}_{ijl} = \lambda_i^{1/2} \hat{\Psi}_{ijl}^{(1)} + \lambda_j^{1/2} \hat{\Psi}_{ijl}^{(2)} + \lambda_l^{1/2} \hat{\Psi}_{ijl}^{(3)}, \quad (4.21)$$

$$\begin{aligned} \mathrm{curl} \hat{\Psi}_{ijl}^{(1)} &= \lambda_j^{1/2} \hat{\Phi}_{ijl}^{(3)} - \lambda_l^{1/2} \hat{\Phi}_{ijl}^{(2)}, \\ \mathrm{curl} \hat{\Psi}_{ijl}^{(2)} &= \lambda_l^{1/2} \hat{\Phi}_{ijl}^{(1)} - \lambda_i^{1/2} \hat{\Phi}_{ijl}^{(3)}, \\ \mathrm{curl} \hat{\Psi}_{ijl}^{(3)} &= \lambda_i^{1/2} \hat{\Phi}_{ijl}^{(2)} - \lambda_j^{1/2} \hat{\Phi}_{ijl}^{(1)}, \end{aligned} \quad (4.22)$$

$$\mathrm{div} \hat{\Phi}_{ijl}^{(1)} = \lambda_i^{1/2} \hat{\phi}_{ijl}, \quad \mathrm{div} \hat{\Phi}_{ijl}^{(2)} = \lambda_j^{1/2} \hat{\phi}_{ijl}, \quad \mathrm{div} \hat{\Phi}_{ijl}^{(3)} = \lambda_l^{1/2} \hat{\phi}_{ijl}. \quad (4.23)$$

Therefore, the FDM bases form a local complete sequence at the interior DOF level, i.e., for a fixed  $(i, j, l) \in (1 : p - 1)^3$  we can establish a subcomplex of the discrete de Rham complex on the reference cube

$$\begin{array}{ccccccc} \mathrm{Q}_p(\hat{K}) & \xrightarrow{\mathrm{grad}} & \mathrm{NCE}_p(\hat{K}) & \xrightarrow{\mathrm{curl}} & \mathrm{NCF}_p(\hat{K}) & \xrightarrow{\mathrm{div}} & \mathrm{DQ}_{p-1}(\hat{K}) \\ \downarrow & & \downarrow & & \downarrow & & \downarrow \\ \mathrm{span}\{\hat{\psi}_{ijl}\} & \xrightarrow{\mathrm{grad}} & \mathrm{span}\{\hat{\Psi}_{ijl}^{(m)}\}_{m=1}^3 & \xrightarrow{\mathrm{curl}} & \mathrm{span}\{\hat{\Phi}_{ijl}^{(m)}\}_{m=1}^3 & \xrightarrow{\mathrm{div}} & \mathrm{span}\{\hat{\phi}_{ijl}\} \end{array} \quad (4.24)$$

Taking into account the  $L^2(\hat{K})$ -orthogonality of the interior basis functions, (4.24) implies that on Cartesian cells, the sparsity pattern of the stiffness matrices  $A^k$  discretizing the bilinear form for the Riesz map  $a^k(\cdot, \cdot)$  in (4.9) connects each interior DOF only to interior DOFs that share  $(i, j, l)$ . Thus the interior block of  $A^k$  has at most  $d$  nonzeros per row (for  $k \in \{1, 2\}$ ) or one nonzero per row (for  $k \in \{0, 3\}$ ), as depicted in Figure 4.1.

### 4.3 Auxiliary sparse preconditioning

The orthogonality on the reference cell, and the subsequent sparsity of the mass and stiffness matrices, will only carry over to cells that are Cartesian and when  $\alpha, \beta$  are cellwise constant. In this section we construct a preconditioner that extends the sparsity we would have in the Cartesian case to the case of practical interest, with distorted cells and spatially varying coefficients. The essential idea is to build an auxiliary operator which is sparse in the FDM basis, by construction. To explain this, we must first introduce some notions of finite element assembly.

### 4.3.1 Pullbacks and finite element assembly

The discrete spaces  $V_{h,p}^k$  are defined in such way that the trace is continuous across facets. This is achieved through the *pullback*  $\mathcal{F}_K^k : V^k(\hat{K}) \rightarrow V^k(K)$  that maps functions on the reference cell  $\hat{K}$  to functions on the physical cell  $K$ . The discrete spaces are defined in terms of the pullback,

$$V_{h,p}^k(\mathcal{T}_h) := \left\{ v_h \in V^k : \forall K \in \mathcal{T}_h \exists \hat{v} \in V_{h,p}^k(\hat{K}) \text{ s.t. } v_h|_K = \mathcal{F}_K^k(\hat{v}) \right\}. \quad (4.25)$$

The application of the pullback to a reference function can be described as the composition of the inverse of the coordinate mapping  $F_K : \hat{K} \rightarrow K$  with multiplication by a factor  $\mathcal{R}_K^k$  depending on the Jacobian of the coordinate transformation  $J_K := DF_K$ . Let  $u$  be a  $k$ -form on  $K$  mapped from  $\hat{u}$  in  $\hat{K}$ . Then

$$u(\mathbf{x}) = \mathcal{F}_K^k(\hat{u}(\hat{\mathbf{x}})) = \mathcal{R}_K^k \hat{u}(F_K^{-1}(\mathbf{x})) = \begin{cases} \hat{u}(F_K^{-1}(\mathbf{x})) & k = 0, \\ J_K^{-\top} \hat{u}(F_K^{-1}(\mathbf{x})) & k = 1, \\ (\det J_K)^{-1} J_K \hat{u}(F_K^{-1}(\mathbf{x})) & k = 2, \\ (\det J_K)^{-1} \hat{u}(F_K^{-1}(\mathbf{x})) & k = 3, \end{cases} \quad (4.26)$$

for  $\mathbf{x} \in K$  mapped from  $\hat{\mathbf{x}} \in \hat{K}$  via  $\mathbf{x} = F_K(\hat{\mathbf{x}})$ . The pullback preserves continuity of the traces of a  $k$ -form across cell facets, which is the natural continuity requirement for  $d^k$ .

Another key property of the pullback is that it commutes with  $d^k$ . The exterior derivative  $d^k u$  can be mapped from that of the reference value  $\hat{d}^k \hat{u}$ ,

$$d^k \mathcal{F}_K^k(\hat{u}) = \mathcal{F}_K^{k+1}(\hat{d}^k \hat{u}), \quad (4.27)$$

where  $\hat{d}^k$  is the exterior derivative with respect to the reference coordinates  $\hat{\mathbf{x}}$ . The pullback is incorporated in FEM by storing reference values as the DOFs in the vector of coefficients  $\underline{u} = (\hat{u}_1, \dots, \hat{u}_N)^\top$  representing a discrete function on a cell  $K$  as

$$u_h|_K = \sum_{j=1}^N \hat{u}_j \mathcal{F}_K^k(\hat{\psi}_j^k), \quad (4.28)$$

where  $\hat{\psi}_j^k$  indexes the basis functions for  $V_{h,p}^k(\hat{K})$  defined in (4.17)-(4.20). The assembly of a bilinear form involves the cell matrices

$$[A_K^k]_{ij} = a^k(\mathcal{F}_K^k(\hat{\psi}_i^k), \mathcal{F}_K^k(\hat{\psi}_j^k)). \quad (4.29)$$

### 4.3.2 Construction of sparse preconditioners

We rewrite the bilinear form  $a^k(\cdot, \cdot)$  in terms of reference arguments and use the property (4.27), to obtain

$$a^k(v_h, u_h) = (\mathcal{F}_K^k(\hat{v}), \beta \mathcal{F}_K^k(\hat{u}))_K + (\mathcal{F}_K^{k+1}(\hat{d}^k \hat{v}), \alpha \mathcal{F}_K^{k+1}(\hat{d}^k \hat{u}))_K, \quad (4.30)$$

for  $v_h, u_h \in V_{h,p}^k(K)$ . From (4.30) we see that the second term is an inner product of arguments in  $V_{h,p}^{k+1}(K)$ . This means that the cell matrices can be sum-factorized in terms of the differentiation matrix  $\hat{D}$  acting on reference values, and weighted mass matrices on  $V_{h,p}^k$  and  $V_{h,p}^{k+1}$ ,

$$A_K^k = M_{\beta,K}^k + \hat{D}^\top M_{\alpha,K}^{k+1} \hat{D}. \quad (4.31)$$

Intuitively, we want each of the matrices in the sum-factorization (4.31) to be sparse in order to achieve sparsity in  $A_K^k$ . In higher dimensions the matrix  $\hat{D}$  inherits the sparsity of the one-dimensional differentiation matrix depicted in Figure 4.3(c). On Cartesian cells and for cellwise constant  $\alpha, \beta$ , the matrices  $M_{\alpha,K}^{k+1}, M_{\beta,K}^k$  are sparse, but they are not sparse when these conditions do not hold.

We rewrite (4.30) as

$$a_K^k(v, u) = (\hat{v}, \hat{\beta}_K \hat{u})_{\hat{K}} + (\hat{d}\hat{v}, \hat{\alpha}_K \hat{d}\hat{u})_{\hat{K}}, \quad (4.32)$$

where  $\hat{\beta}_K := (\det J_K) \mathcal{R}_K^{k+1} \beta(F_K(\hat{\mathbf{x}})) \mathcal{R}_K^k$  and  $\hat{\alpha}_K := (\det J_K) \mathcal{R}_K^{k+1} \alpha(F_K(\hat{\mathbf{x}})) \mathcal{R}_K^{k+1}$  are Jacobian-weighted push-forwards of the bilinear form coefficients.

For the matrices  $M_{\beta,K}^k, M_{\alpha,K}^{k+1}$  to be sparse,  $\hat{\alpha}_K, \hat{\beta}_K$  must be constant and diagonal. We propose to precondition  $A_K^k$  with a sparse stiffness matrix  $\tilde{A}_K^k$  discretizing a spectrally equivalent form

$$\tilde{a}_K^k(v, u) := (\hat{v}, \tilde{\beta}_K \hat{u})_{\hat{K}} + (\hat{d}\hat{v}, \tilde{\alpha}_K \hat{d}\hat{u})_{\hat{K}}, \quad (4.33)$$

where  $\tilde{\beta}_K, \tilde{\alpha}_K$  are constant diagonal approximations to  $\hat{\beta}_K, \hat{\alpha}_K$ , respectively.

One convenient way to obtain such approximations is to assemble the diagonal of the scaled mass matrices with cell-wise constant spaces  $\text{DP}_0$  with the appropriate pullbacks for  $V_{h,p}^k$  and  $V_{h,p}^{k+1}$ .

We now state the spectral equivalence result between the auxiliary form  $\tilde{a}^k(\cdot, \cdot)$  and the original form  $a^k(\cdot, \cdot)$ .

**Theorem 4.1.** *There exist constants  $C \geq c > 0$  independent of the polynomial degree of the discretization, that depend on  $\alpha|K|/\int_K \alpha \, d\mathbf{x}$ ,  $\beta|K|/\int_K \beta \, d\mathbf{x}$ , and on  $\mathcal{T}_h$  through  $DF_K$  such that*

$$c \leq \frac{a^k(v, v)}{\tilde{a}^k(v, v)} \leq C \quad \forall v \in V_{h,p}^k \setminus \{0\}. \quad (4.34)$$

*Proof.* The proof is similar to that of Theorem 2.1. Let  $C_K^\beta, c_K^\beta, C_K^\alpha, c_K^\alpha$  be the bounds for the spectra of the diagonally scaled coefficients on a given cell, such that  $\sigma(\tilde{\beta}_K^{-1/2} \hat{\beta}_K \tilde{\beta}_K^{-1/2}) \in [c_K^\beta, C_K^\beta]$  and  $\sigma(\tilde{\alpha}_K^{-1/2} \hat{\alpha}_K \tilde{\alpha}_K^{-1/2}) \in [c_K^\alpha, C_K^\alpha]$ . Then

$$c_K := \min\{c_K^\beta, c_K^\alpha\} \leq \frac{a_K^k(v, v)}{\tilde{a}_K^k(v, v)} \leq \max\{C_K^\beta, C_K^\alpha\} =: C_K \quad \forall v \in V_{h,p}^k(K) \setminus \{0\}. \quad (4.35)$$

The spectral equivalence constants on  $V_{h,p}^k$  are  $c := \min_K\{c_K\}$ ,  $C := \max_K\{C_K\}$ .  $\square$

We therefore expect the convergence rate of Krylov methods using preconditioners based on  $\tilde{A}^k$  to be robust to the polynomial degree when the problem parameters are piecewise constant. We also expect the performance to be affected by how much the cells are distorted with respect to the Cartesian case.

We describe an alternative strategy to obtain a sparse approximation to  $A_K^k$ , that does involve the mapped  $\text{DP}_0$  spaces, that we implement for our numerical experiments. We consider a sparse approximation to  $M_{\beta,K}^k$  first. As  $M_{\beta,K}^k$  discretizes a weighted  $L^2(K)$ -inner product, we propose to assemble the matrix in terms of a *broken* space  $\bar{V}_{h,p}^k$  with a fully  $L^2$ -orthonormal basis. In this broken space, the mass matrix to approximate will be diagonal in the Cartesian, constant-coefficient case. The basis for  $\text{DP}_{p-1}$  was already  $L^2$ -orthogonal; we define a new basis for  $\bar{P}_p$ , the broken variant of  $P_p$ , simply by orthogonalizing the interface basis functions with respect to each other. The interface functions were already orthogonal to the interior ones, which follows from the definition of the interior degrees of freedom (2.21) and the duality condition (2.22).

Let  $\bar{M}_{\beta,K}^k$  be the weighted mass matrix in the basis for  $\bar{V}_{h,p}^k(K)$ . Then

$$M_{\beta,K}^k = \bar{G}^\top \bar{M}_{\beta,K}^k \bar{G}, \quad (4.36)$$

where  $\bar{G}$  is a sparse basis transformation matrix from  $V_{h,p}^k(\hat{K})$  to  $\bar{V}_{h,p}^k(\hat{K})$ . This matrix  $\bar{G}$  is block diagonal with one block per vector component, where each block is a Kronecker product of identity matrices and (sparse) basis transformation matrices from  $P_p(\hat{I})$  to  $\bar{P}_p(\hat{I})$ . The matrix  $\bar{M}_{\beta,K}^k$  is diagonal when  $K$  is Cartesian and  $\beta$  is constant, unlike  $M_{\beta,K}^k$  (which is sparse, but not diagonal in this case). To obtain a sparse approximation to  $M_{\beta,K}^k$ , we simply take the diagonal of  $\bar{M}_{\beta,K}^k$  in (4.36). This contrasts with taking the diagonal directly of  $M_{\beta,K}^k$ , which would alter the operator even when  $K$  is Cartesian and  $\beta$  is constant.

Applying the same idea to  $M_{\alpha,K}^{k+1}$ , we approximate the stiffness matrix  $A_K^k$  with an auxiliary matrix that is sparse on any given cell, for any spatial variation

of problem coefficients:

$$A_K^k \approx P_K^k := \overline{G}^\top \operatorname{diag}(\overline{M}_{\beta,K}^k) \overline{G} + \overline{D}^\top \operatorname{diag}(\overline{M}_{\alpha,K}^{k+1}) \overline{D}, \quad (4.37)$$

where  $\overline{D} := \overline{G}\hat{D}$ .

Using this auxiliary operator ensures that the patchwise problems that we solve in our multigrid relaxation are sparse. We describe these patchwise problems next.

## 4.4 Multigrid relaxation by subspace correction

### 4.4.1 Notation

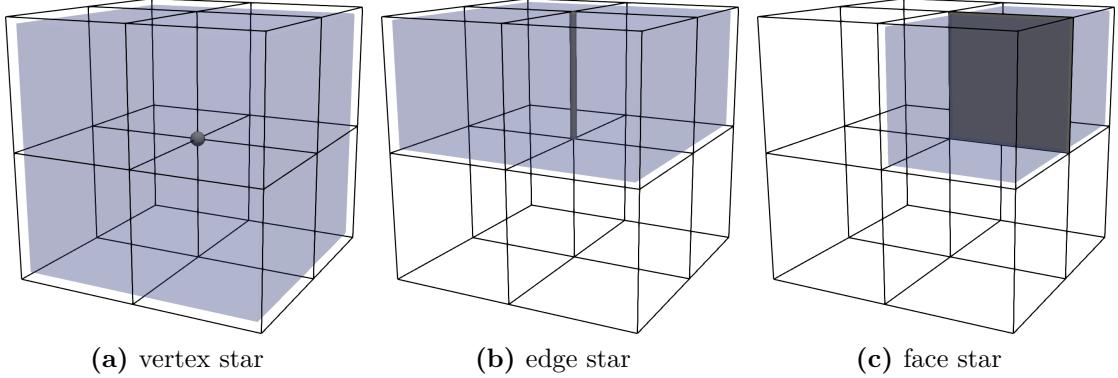
We now introduce the preconditioners we use to solve (4.9). We express the solvers in terms of *space decompositions* [97], which we summarize briefly here. Given a discrete space  $V_{h,p}^k$ , the preconditioner is induced by a particular choice of how to write it as a sum of (smaller) function spaces:

$$V_{h,p}^k = \sum_i V_i. \quad (4.38)$$

This notation for the sum of vector spaces means that for any  $v_h \in V_{h,p}^k$ , there exist  $\{v_i \in V_i\}_i$  such that  $v_h = \sum_i v_i$ . The decomposition is not typically unique. Given an initial guess for the solution to a variational problem posed over  $V_{h,p}^k$ , the Galerkin projection of the equation for the error is solved over each  $V_i$  (additively or multiplicatively). This gives an approximation to the error in each subspace  $V_i$ , which are combined. A cycle over each subspace constitutes one step of a subspace correction method. For more details, see Xu [97].

In order to describe the space decompositions we will use, we require some concepts from algebraic topology. We conceive of the mesh  $\mathcal{T}_h$  as a *regular cell complex* [81, 54, 63]. This represents the mesh as a set of entities of different dimensions, with incidence relations between them. For  $d = 3$ , the entities are vertices, edges, faces, and cells, of dimensions  $k = 0, 1, 2, 3$  respectively. The incidence relations encode the boundary operator, relating an entity of dimension  $k \geq 1$  to its bounding sub-entities of dimension  $k - 1$ . For example, they encode that a cell has as its sub-entities certain faces, while a face has as its sub-entities certain edges. Let  $E^k(\mathcal{T}_h)$  denote the set of entities of dimension  $k$  in  $\mathcal{T}_h$ . We also define  $E^{-1}(\mathcal{T}_h) := \emptyset$  for notational convenience.

The *star* operation on an entity  $e$  of dimension  $k$ , denoted  $\star e$ , returns the union of the interiors of all entities of dimension at least  $k$  that recursively contain  $e$  as a sub-entity [71, §2][34]. For example, the star of a cell is simply its interior, since



**Figure 4.4:** The stars of (a) a vertex (b) an edge (c) a face. Solving a discrete problem restricted to a star involves solving for all degrees of freedom contained in the star.

there are no entities of higher dimension. The star of an internal face  $\star f$  returns the patch of cells formed of the two cells that share  $f$ , excluding the boundary of the patch. Similarly, the star of a vertex  $\star v$  returns the union of the interiors of all edges, faces, and cells sharing  $v$ , as well as the vertex itself; geometrically, this forms the patch of cells sharing  $v$ , again excluding the boundary of the patch. The stars of a vertex, edge, and face are shown in Figure 4.4.

Given a function space  $V_h$  and an entity  $e$ , we define

$$V_h|_{\star e} := \{v_h \in V_h : \text{support}(v_h) \subseteq \bar{\star e}\}. \quad (4.39)$$

This gives a local function space around an entity upon which a variational problem may be solved. Informally, it defines a block employed in a block Jacobi method, taking all the DOFs in the patch of cells around  $e$ , excluding those on the boundary.

#### 4.4.2 Designing space decompositions

The framework (4.38) offers a great deal of freedom in designing solvers. Since the bilinear form  $a^k(\cdot, \cdot)$  is symmetric and coercive, powerful and general theories are available to guide the choice of space decomposition [97, 86, 61]; for a summary, see [38]. We remark on some general principles here.

First, the cost of each iteration of subspace correction will depend on the dimensions of the subspaces  $\{V_i\}$ . It is therefore desirable that the space decomposition be as fine as possible, i.e.,  $\dim(V_i)$  be as small as possible for each  $i$ . If  $V_{h,p}^k = \text{span}\{\psi_1, \dots, \psi_J\}$ , then choosing  $V_i := \text{span}\{\psi_i\}$  for  $i = 1, \dots, J$  gives the finest possible space decomposition. The subspace correction methods induced by this space decomposition are (point) Jacobi or Gauß–Seidel iterations. However, the

convergence of these schemes on their own (without a coarse space) is unacceptably slow, even for the Poisson problem (4.1) with  $k = 0$ ,  $p = 1$ ,  $\beta = 0$ ,  $\Gamma_D = \partial\Omega$ .

To discuss the convergence of the scheme, we introduce some notation; we draw this discussion from [38]. Define the operator  $\mathcal{A}^k : V_{h,p}^k \rightarrow (V_{h,p}^k)^*$  associated with the bilinear form  $a^k(\cdot, \cdot)$  via

$$\langle v, \mathcal{A}^k u \rangle := a^k(v, u) \text{ for all } v, u \in V_{h,p}^k, \quad (4.40)$$

where  $\langle \cdot, \cdot \rangle$  denotes the duality pairing. For each subspace we denote the inclusion  $I_i : V_i \rightarrow V_{h,p}^k$  and its  $L^2$  adjoint  $I_i^* : (V_{h,p}^k)^* \rightarrow V_i^*$ , and we define the restriction of  $\mathcal{A}^k$  to  $V_i$  by

$$\langle v_i, \mathcal{A}_i u_i \rangle = \langle I_i v_i, \mathcal{A}^k I_i u_i \rangle \text{ for all } v_i, u_i \in V_i, \quad (4.41)$$

i.e.,  $\mathcal{A}_i = I_i^* \mathcal{A}^k I_i$ . The additive subspace correction preconditioner associated with the decomposition  $\{V_i\}$  is then given by

$$D^{-1} = \sum_i I_i \mathcal{A}_i^{-1} I_i^*. \quad (4.42)$$

Let  $T = D^{-1} \mathcal{A}^k$  be the preconditioned operator. Our goal is to estimate the condition number  $\kappa(T)$  to bound the convergence of the conjugate gradient method. The condition number is bounded by two parameters.

The first,  $N_O$ , is the maximum overlap among subspaces. For each subspace  $V_i$ , consider its set of overlapping subspaces  $\text{overlap}(V_i) := \{V_j : \exists v_i \in V_i, v_j \in V_j \text{ s.t. } a(I_i v_i, I_j v_j) \neq 0\}$ ;  $N_O$  is the maximum over  $i$  of  $|\text{overlap}(V_i)|$ . This bounds the maximal eigenvalue of  $T$ . It is straightforward to analyze  $N_O$  by inspection on a particular family of meshes, and  $N_O$  is naturally independent of  $h$ ,  $p$ ,  $\alpha$ , and  $\beta$ . The second measures the stability of the space decomposition. Assume that there exists  $c_1$  such that

$$\inf_{\substack{v_i \in V_i \\ \sum_i v_i = v_h}} \sum_i \|v_i\|_{\mathcal{A}_i}^2 \leq c_1 \|v_h\|_{\mathcal{A}}^2 \text{ for all } v_h \in V_{h,p}^k. \quad (4.43)$$

Then the minimum eigenvalue of  $T$  is bounded below by  $c_1^{-1}$ .

In order for the convergence of our solver to be robust, we require that  $c_1$  is independent of  $h$ ,  $p$ ,  $\alpha$ , and  $\beta$ . For example, the slow convergence of the Jacobi iteration on the Poisson problem is because the space decomposition is not stable in the mesh parameter  $h$ . This may be addressed by incorporating a global coarse space (on a coarser mesh with  $H > h$ ) into the space decomposition (as in a two-level domain decomposition method, or a multigrid method), which gives a

stability constant independent of  $h$  for the Poisson equation. However, this space decomposition is still not stable in  $p$  for (4.1), nor is it stable in  $h, \alpha, \beta$  for (4.2) or (4.3). Essentially, this is because for  $k = 0$ , eigenfunctions associated with small eigenvalues of the operator are smooth and can be well-represented on a coarse grid, but this is not true for  $k \in \{1, 2\}$ . Building on work by Pavarino [75], Arnold, Falk & Winther [8, 9], and Hiptmair [44], we now discuss the space decompositions proposed by Schöberl and Zaglmayr [88, 99] that are robust in  $h, p, \alpha$ , and  $\beta$  in numerical experiments.

#### 4.4.3 Choice of space decompositions

The Pavarino–Arnold–Falk–Winther (PAFW) decomposition is

$$V_{h,p}^k = V_{h,1}^k + \sum_{v \in E^0(\mathcal{T}_h)} V_{h,p}^k|_{\star v}. \quad (4.44)$$

This combines solving local problems on overlapping patches of cells sharing a vertex with a coarse solve at lowest order ( $p = 1$ ). The Pavarino–Hiptmair (PH) decomposition is

$$V_{h,p}^k = V_{h,1}^k + \sum_{e \in E^k(\mathcal{T}_h)} V_{h,p}^k|_{\star e} + \sum_{e' \in E^{k-1}(\mathcal{T}_h)} d^{k-1} V_{h,p}^{k-1}|_{\star e'}. \quad (4.45)$$

These decompositions (4.44) and (4.45) coincide for  $k = 0$ . For example, for  $H(\text{curl})$  ( $k = 1$ ), the PH decomposition iterates over all edges  $e$  of the mesh, solving patch problems on the cells sharing each edge, while  $e'$  iterates over all vertices. The PH decomposition involves an auxiliary problem on the local potential space  $V_{h,p}^{k-1}|_{\star e'}$ : find  $\psi \in V_{h,p}^{k-1}|_{\star e'}$  such that

$$a^k(d^{k-1}\phi, d^{k-1}\psi) = (d^{k-1}\phi, \beta d^{k-1}\psi)_{\star e'} = L(d^{k-1}\phi) \quad \text{for all } \phi \in V_{h,p}^{k-1}|_{\star e'}, \quad (4.46)$$

since  $d^k \circ d^{k-1} = 0$ . For example, for  $H(\text{curl})$ , this becomes: for each vertex  $v$ , find  $\psi \in V_{h,p}^0|_{\star v}$  such that

$$(\text{grad } \phi, \beta \text{ grad } \psi)_{\star v} = L(\text{grad } \phi) \quad \text{for all } \phi \in V_{h,p}^0|_{\star v},$$

a local scalar-valued Poisson-type problem.

*Remark 4.1.* In the case of  $H(\text{div})$  ( $k = 2$ ), the auxiliary problem is singular, with a kernel consisting of the curl-free functions. One alternative is to define an auxiliary space with this kernel removed, i.e., posed on the space  $V_{h,p}^{k-1}/d^{k-2}V_{h,p}^{k-2}$ . Another approach is to add a symmetric and positive-definite term on the kernel, such

as  $(\operatorname{div} \phi, \beta \operatorname{div} \psi)$ , as done by the Hiptmair–Xu decomposition [46], but with the conforming auxiliary space  $[H(\operatorname{grad})]^d$ . Our implementation deals with this problem in a pragmatic way by simply adding a small multiple of the mass matrix for  $V_{h,p}^{k-1}$  to the patch matrix.

*Remark 4.2.* The  $p$ -robustness of the PAFW decomposition was proven for  $k = 0$  in the tensor-product case by Pavarino [75] and in the simplicial case by Schöberl et al. [87]. A similar decomposition (with geometric multigrid coarsening, not coarsening in polynomial degree) was proposed by Arnold, Falk, & Winther for  $k = 1$  and  $k = 2$  [8, 9], and proven to be robust to mesh size  $h$  and variations in constant  $\alpha$  and  $\beta$ . To the best of our knowledge the  $p$ -robustness of (4.44) and (4.45) have not been proven for  $k = 1$  or  $k = 2$ , although numerical experiments indicate that they are  $p$ -robust for  $p \leq 31$ .

To simplify notation, we drop the superscript  $k$  from the stiffness matrices  $A^k$ . The algebraic realization of the PAFW relaxation (combined additively) reads

$$P_{\text{PAFW}}^{-1} = R_0^\top A_0^{-1} R_0 + \sum_{v \in E^0(\mathcal{T}_h)} R_v^\top A_v^{-1} R_v. \quad (4.47)$$

Here  $R_0$  is the restriction matrix from  $(V_{h,p}^k)^*$  to  $(V_{h,1}^k)^*$ ,  $A_0$  is the stiffness matrix for the original bilinear form  $a^k(\cdot, \cdot)$  rediscretized with the lowest-order element ( $p = 1$ )<sup>3</sup>,  $R_v$  are Boolean restriction matrices onto the DOFs of each vertex-star patch  $\star v$ , and  $A_v = R_v A R_v^\top$  are sub-matrices of  $A$  corresponding to the rows and columns of DOFs of the patch.

Similarly, the additive PH relaxation is implemented as

$$P_{\text{PH}}^{-1} = R_0^\top A_0^{-1} R_0 + \sum_{e \in E^k(\mathcal{T}_h)} R_e^\top A_e^{-1} R_e + D \left( \sum_{e' \in E^{k-1}(\mathcal{T}_h)} R_{e'}^\top B_{e'}^{-1} R_{e'} \right) D^\top. \quad (4.48)$$

where  $R_0, A_0$  have the same meaning as in (4.47),  $D$  is the matrix tabulating the exterior derivative  $d^{k-1} : V_{h,p}^{k-1} \rightarrow V_{h,p}^k$ ,  $R_e, R_{e'}$  are Boolean restriction matrices onto star patches on entities of dimension  $k$  and  $k - 1$ , respectively,  $A_e = R_e A R_e^\top$  are patch matrices, and  $B_{e'}$  are patch matrices extracted from  $B$ , which is obtained as the discretization of  $a^k(d^{k-1}\phi, d^{k-1}\psi)$ . With the FDM basis it is feasible to compute and store the  $D$  matrix, as it is sparse and applies to reference values.

---

<sup>3</sup>We apologize for this notation; the use of a subscript  $_0$  to indicate the coarse grid is widely used in the domain decomposition literature. In our case the coarse grid is formed with  $p = 1$ , not  $p = 0$ .

#### 4.4.4 Statically-condensed space decompositions

As seen from Figure 4.1, the minimal coupling between interior DOFs that arises from the orthogonality of the FDM elements invites the use of static condensation. Static condensation yields a finer space decomposition with smaller subspaces by eliminating the interior DOFs. The overlapping subspaces only involve interface DOFs.

The statically-condensed Pavarino–Arnold–Falk–Winther (SC-PAFW) decomposition is

$$V_{h,p}^k = V_{h,1}^k + \sum_{c \in \mathcal{I}} V_{h,p}^k|_c + \sum_{v \in E^0(\mathcal{T}_h)} \tilde{V}_{h,p}^k|_{\star v}. \quad (4.49)$$

Here  $\mathcal{I} := E^d(\mathcal{T}_h)$  is the set of cell interiors of  $\mathcal{T}_h$ . The cell-interior problems do not overlap with each other and can be solved independently. We denote by  $\tilde{V}_{h,p}^k := (V_{h,p}^k|_{\mathcal{I}})^\perp$  the space of discrete harmonic functions of  $V_{h,p}^k$ ,

$$\tilde{V}_{h,p}^k = \left\{ \tilde{v} \in V_{h,p}^k : a^k(w, \tilde{v}) = 0 \forall w \in V_{h,p}^k|_{\mathcal{I}} \right\}. \quad (4.50)$$

By definition  $V_{h,p}^k = V_{h,p}^k|_{\mathcal{I}} \oplus \tilde{V}_{h,p}^k$ ; this orthogonality leads to reduced overlap of the star patches, compared to the non-statically-condensed case.

The statically-condensed Pavarino–Hiptmair (SC-PH) decomposition is

$$V_{h,p}^k = V_{h,1}^k + \sum_{c \in \mathcal{I}} V_{h,p}^k|_c + \sum_{e \in E^k(\mathcal{T}_h)} \tilde{V}_{h,p}^k|_{\star e} + \sum_{e' \in E^{k-1}(\mathcal{T}_h)} d^{k-1} \tilde{V}_{h,p}^{k-1}|_{\star e'}. \quad (4.51)$$

These decompositions again coincide for  $k = 0$ .

Denote by  $I, \Gamma$  the sets of interior and interface (vertex, edge, and face) DOFs, respectively. Reordering the DOFs of  $A$  yields a  $2 \times 2$  block matrix, with inverse obtained from its block  $LDL^\top$  decomposition

$$\begin{bmatrix} A_{II} & A_{I\Gamma} \\ A_{\Gamma I} & A_{\Gamma\Gamma} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbb{I} & -A_{II}^{-1}A_{I\Gamma} \\ 0 & \mathbb{I} \end{bmatrix} \begin{bmatrix} A_{II}^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} \mathbb{I} & 0 \\ -A_{\Gamma I}A_{II}^{-1} & \mathbb{I} \end{bmatrix}, \quad (4.52)$$

where  $S$  denotes the interface Schur complement

$$S = A_{\Gamma\Gamma} - A_{\Gamma I}A_{II}^{-1}A_{I\Gamma}. \quad (4.53)$$

When solvers for  $A_{II}$  and  $S$  are available, the application of  $A^{-1}$  times a residual vector can be performed with a single application of  $S^{-1}$  and only two applications of  $A_{II}^{-1}$ . This is because the second and third instances of  $A_{II}^{-1}$  in the RHS of (4.52) act on the same interior DOFs of the incoming residual vector.

Another way to write (4.52) gives rise to the additive interpretation of the harmonic subspace correction step

$$A^{-1} = R_I^\top A_{II}^{-1} R_I + R_\Gamma^\top S^{-1} R_\Gamma, \quad (4.54)$$

where  $R_I$  is a Boolean restriction onto the interior DOFs, and

$$R_\Gamma = \begin{bmatrix} -A_{\Gamma I} A_{II}^{-1} & \mathbb{I} \end{bmatrix} \quad (4.55)$$

is the ideal restriction operator onto the discrete harmonic subspace.  $R_\Gamma^\top$  maps vectors of coefficients in  $\tilde{V}_{h,p}^k$  to  $V_{h,p}^k$ . The orthogonality between  $\tilde{V}_{h,p}^k$  and  $V_{h,p}^k|_{\mathcal{I}}$  is reflected by the identity  $R_\Gamma A R_I^\top = 0$ .

*Remark 4.3.* In  $H(\text{grad})$  ( $k = 0$ ), the interior DOFs are fully decoupled and  $A_{II}$  is diagonal, for the sparse auxiliary operator constructed in Section 4.3. In general, the cell-interior problems only couple at most  $d$  DOFs, as shown in (4.24). There exists a reordering of the interior DOFs for which  $A_{II}$  becomes block diagonal with diagonal blocks of dimension at most  $d \times d$ , implying that  $A_{II}$  shares its sparsity pattern with its inverse. Therefore  $A_{II}$  coincides with its zero-fill-in incomplete Cholesky factorization. Hence we may assemble and store the matrix  $S$ , even for very high  $p$ .

*Remark 4.4.* We choose to use a Krylov method on  $A$ , as opposed to implementing one on the condensed system involving  $S$ . The action of the true  $A_{II}^{-1}$  involves the iterative solution of local problems on cell-interiors, inducing  $\mathcal{O}(p^{d+1})$  computational cost in the application of  $S$ . Although the sum-factorized application of  $A_{\Gamma\Gamma}, A_{I\Gamma}, A_{\Gamma I}$  only involves  $\mathcal{O}(p^d)$  flops, and the conditioning of the preconditioned operator is generally better, this results in a longer runtime when compared to using a Krylov method on  $A$  with a statically-condensed preconditioner, especially for the case  $k = 1$ .

The SC-PAFW relaxation approximates

$$S^{-1} \approx S_{\text{PAFW}}^{-1} := \tilde{R}_0^\top A_0^{-1} \tilde{R}_0 + \sum_{v \in E^0(\mathcal{T}_h)} \tilde{R}_v^\top S_v^{-1} \tilde{R}_v, \quad (4.56)$$

where  $\tilde{R}_0$  is the restriction matrix from  $(\tilde{V}_{h,p}^k)^*$  to  $(V_{h,1}^k)^*$ , and  $\tilde{R}_v$  are the Boolean restriction matrices onto the interface DOFs of the vertex-star patch  $\star v$ , and  $S_v = \tilde{R}_v S \tilde{R}_v^\top$ .

Similarly, the SC-PH relaxation is implemented as

$$S_{\text{PH}}^{-1} := \tilde{R}_0^\top A_0^{-1} \tilde{R}_0 + \sum_{e \in E^k(\mathcal{T}_h)} \tilde{R}_e^\top S_e^{-1} \tilde{R}_e + \tilde{D} \left( \sum_{e' \in E^{k-1}(\mathcal{T}_h)} \tilde{R}_{e'}^\top \tilde{B}_{e'}^{-1} \tilde{R}_{e'} \right) \tilde{D}^\top, \quad (4.57)$$

where  $\tilde{R}_0, A_0$  have the same meaning as in (4.56),  $\tilde{D} = D_{\Gamma\Gamma}$  is the matrix tabulating the exterior derivate restricted onto the interface,  $\tilde{R}_e, \tilde{R}_{e'}$  are Boolean restriction matrices onto the interface DOFs of star patches on entities of dimension  $k$  and  $k-1$ , respectively,  $S_e = \tilde{R}_e S \tilde{R}_e^\top$  are patch matrices, and  $\tilde{B}_{e'}$  are patch matrices extracted from the interface Schur complement of the discretization of  $a^k(d^{k-1}\phi, d^{k-1}\psi)$ .

#### 4.4.5 Achieving optimal fill-in

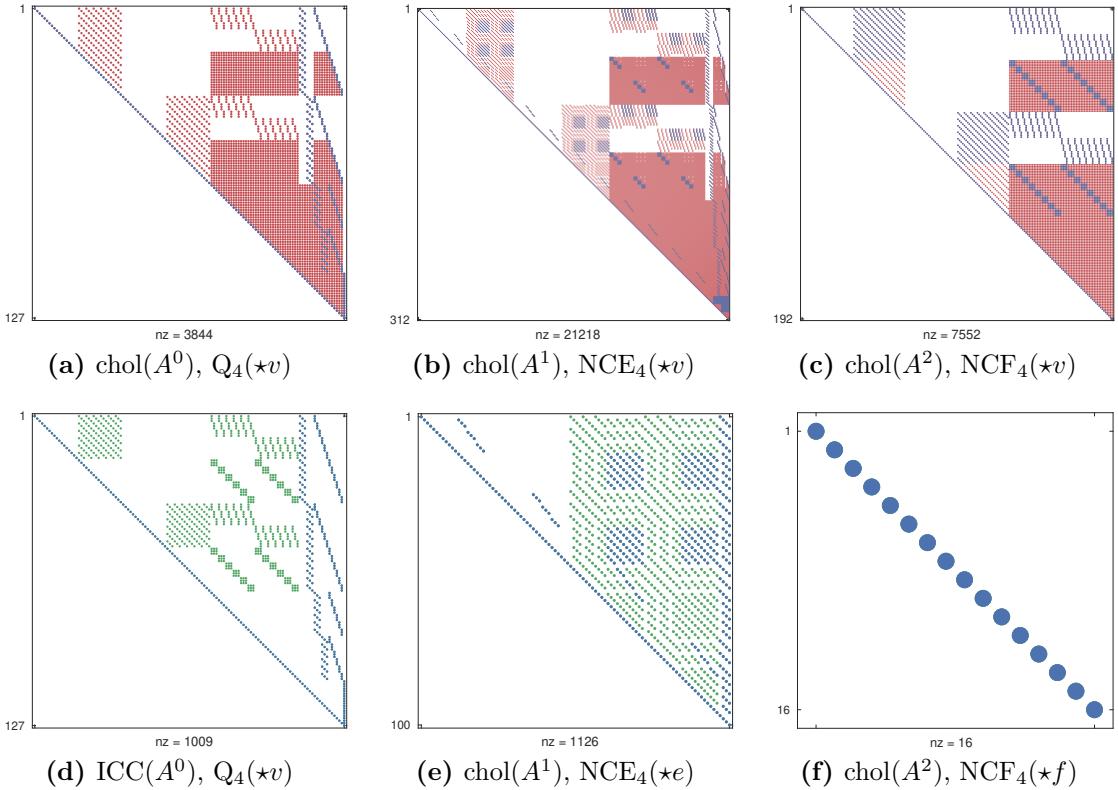
To achieve optimal complexity of our solver, we require that the factorizations of the patch matrices arising in the space decomposition be optimal in storage. The number of nonzeros in the factorizations also relates to the number of flops required to compute them. For the PAFW space decomposition (4.44), in Figure 4.5(a-c) we observe fill-in of  $\mathcal{O}(p^4)$  nonzeros in the Cholesky factorization of the vertex patch problems in  $H(\text{grad})$ ,  $H(\text{curl})$ , and  $H(\text{div})$ , even with a nested dissection ordering. Moreover, computing this factorization incurs  $\mathcal{O}(p^6)$  flops. These costs compare unfavorably with the  $\mathcal{O}(p^3)$  storage and  $\mathcal{O}(p^4)$  flops required by sum-factorized operator application.

To overcome this, for  $H(\text{grad})$  we employ an incomplete Cholesky factorization. Incomplete Cholesky factorization with zero fill-in (ICC(0), depicted in blue in Figure 4.5(a) and (d)) does not yield an effective relaxation method, even when it can be computed. In contrast, the incomplete Cholesky factorization on the statically-condensed sparsity pattern does yield an effective relaxation. This factorization has  $\mathcal{O}(p^3)$  fill-in ( $\mathcal{O}(p)$  nonzeros on  $\mathcal{O}(p^2)$  rows). It appears that this sparsity pattern (depicted in green in Figure 4.5(d)) offers a suitable intermediate between the zero-fill-in pattern and the full Cholesky factorization (depicted in red in Figure 4.5(a)).

For  $H(\text{curl})$  and  $H(\text{div})$ , the PH (4.45) space decomposition is finer than PAFW: it requires the solution of smaller vector-valued subproblems, in the stars of edges or faces, instead of in the stars of vertices (cf. Figure 4.4). For the edge-star problems solved for  $H(\text{curl})$ , in Figure 4.5(e) we observe  $\mathcal{O}(p^3)$  nonzeros in the Cholesky factorization, and so the Cholesky factorization for the PH patch has optimal storage without requiring the use of incomplete factorizations. On the auxiliary scalar-valued problem posed on the vertex star, we employ the incomplete Cholesky factorization described above. For the face-star problems solved in  $H(\text{div})$ , there is no coupling at all between the face degrees of freedom in the FDM basis, and ICC(0) offers a direct solver (Figure 4.5(f)). In fact, the SC-PH relaxation for  $H(\text{div})$  is equivalent to point-Jacobi applied to the interface Schur complement.

### 4.5 Numerical experiments

We provide an implementation of the  $P_p$  and  $DP_{p-1}$  elements with the FDM basis functions on the interval in FIAT [52]. The extension to quadrilaterals and hexahedra is achieved by taking tensor-products of the one-dimensional elements with FInAT [47]. Code for the sum-factorized evaluation of the residual is automatically



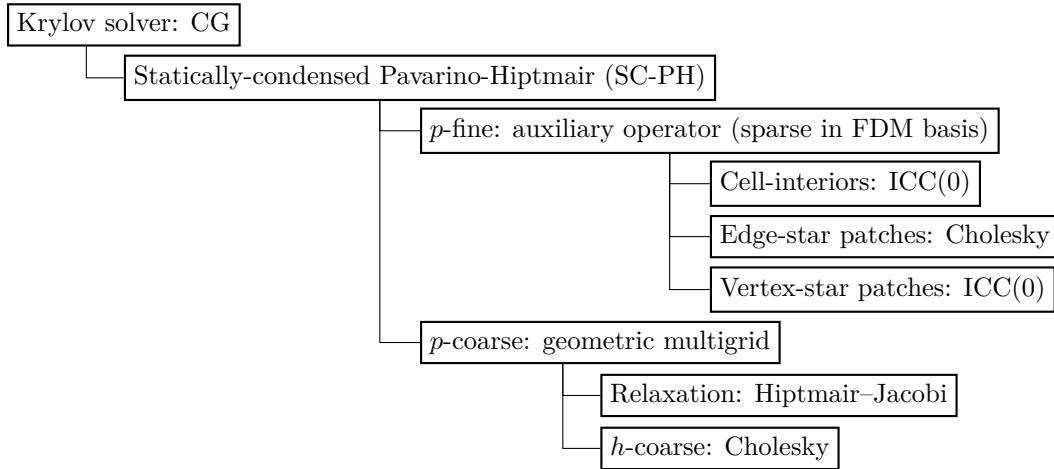
**Figure 4.5:** The interface-interface block of the patch problems arising in the PAFW space decomposition (a-c) and the PH space decomposition (d-f) on a regular Cartesian mesh with FDM elements. The sparsity pattern of the problem posed in the FDM basis is coloured in blue. The additional nonzeros required by the Cholesky factorization are colored in red or green, depending on whether they require suboptimal storage (as in (a-c)) or optimal storage (as in (e-f)). In (d) we plot the sparsity pattern of the interface Schur complement arising in static condensation, which offers a sparsity pattern intermediate between the zero fill-in and the full Cholesky factorization.

generated by Firedrake [82, 48], implementing a Gauß–Lobatto quadrature rule with  $3(p+1)/2$  points along each direction. The sparse preconditioner discretizing the auxiliary operator is implemented as a PETSc [13] preconditioner as `firedrake.FDMPC`. The preconditioners are additive on each level and multiplicative between levels. The Cholesky factorization of the patch matrices is computed using CHOLMOD [24] and the ICC factorization is done with PETSc’s own implementation. Most of our computations were performed on a single node of the ARCHER2 system, with two 64-core AMD EPYC 7742 CPUs (2.25 GHz) and 512 GiB of RAM.

Code for all examples has been archived and is available at [100].

### 4.5.1 Riesz maps: robust iteration counts and optimal complexity

We first present numerical evidence demonstrating that our preconditioner for (4.1)–(4.3) yields CG iteration counts that are robust to mesh size  $h$ , polynomial degree  $p$ , and the coefficients  $\alpha$  and  $\beta$ . For concreteness, we present a solver diagram for the  $H(\text{curl})$  problem (4.2) with the SC-PH space decomposition in Figure 4.6; the solver diagrams for the other Riesz maps and space decompositions are analogous.



**Figure 4.6:** Solver diagram for the  $H(\text{curl})$  Riesz map using static condensation and incomplete Cholesky factorization on vertex patches (SC-PH/FDM/ICC).

We consider two mesh hierarchies of  $\Omega = (0, 1)^3$ , one structured and one unstructured. The coarse meshes are the extrusions with six cells in the vertical of the two-dimensional meshes depicted in Figure 2.6(a)–(b). We then uniformly refine these  $l \geq 0$  times. Each run is terminated when the (natural)  $P^{-1}$ -norm of the residual is reduced by a factor of  $10^8$  starting from a zero initial guess. Each problem has homogeneous Dirichlet boundary conditions on  $\Gamma_D = \partial\Omega$  and a randomized right-hand side that is prescribed independently of  $\alpha, \beta$ , through its Riesz representative

$$F(v) = (v, w_{h,p}^k)_\Omega + (\mathbf{d}^k v, \mathbf{d}^k w_{h,p}^k)_\Omega, \quad \text{for } w_{h,p}^k \in V_{h,p}^k. \quad (4.58)$$

We first consider the robustness of our solvers with respect to mesh refinement  $l$  and polynomial degree  $p$ , in Tables 4.1–4.3. In these experiments we fix  $\alpha = 1$  and  $\beta = 10^{-8}$ . The results show almost complete  $p$ - and  $h$ -robustness in the Cartesian case, and very slow growth of iteration counts in the unstructured case.

We next consider the robustness with respect to  $\alpha$  and  $\beta$  for fixed  $p = 7$ ,  $l = 2$ , for the  $H(\text{curl})$  and  $H(\text{div})$  problems (4.2)–(4.3), in Tables 4.4–4.5. Again, iteration counts do not vary substantially as the coefficients are varied, indicating

the desired parameter-robustness. As expected, we observe that the iteration counts only depend on the ratio  $\alpha/\beta$ .

We next record the flop counts, peak memory usage, and matrix nonzeros for the PH and SC-PH solvers (with either Cholesky or ICC for the  $H(\text{grad})$  vertex-star patches) while varying  $p$  with  $\alpha = 1$ ,  $\beta = 1$  with the mesh shown in Figure 4.7. This smaller mesh was used so that we could solve the problem at higher  $p$  using the GLL basis. The results were reported above in Figure 4.2.

**Table 4.1:** CG iteration counts for the  $H(\text{grad})$  Riesz map solved with the {chol/SC-chol; ICC/SC-ICC} preconditioners.

$p \setminus l$	Cartesian						Unstructured					
	0	1	2	0	1	2	0	1	2	0	1	2
1	1/1	1/1	8/8	8/8	8/8	8/8	1/1	1/1	12/12	12/12	14/14	14/14
3	14/11	14/11	15/11	15/11	15/11	15/11	18/18	18/18	18/19	18/19	18/21	18/21
7	12/9	13/10	12/9	13/10	12/9	13/10	19/21	20/21	19/23	20/23	19/24	20/24
11	11/9	12/10	11/9	12/10	-/9	-/10	20/23	22/24	19/24	22/24	-/-	-/25
15	10/8	13/11	10/8	13/11			21/24	24/25	20/25	23/25		

**Table 4.2:** CG iteration counts for the  $H(\text{curl})$  Riesz map solved with the {PAFW/SC-PAFW; PH/SC-PH} preconditioners.

$p \setminus l$	Cartesian						Unstructured					
	0	1	2	0	1	2	0	1	2	0	1	2
1	2/2	2/2	13/13	11/11	14/14	12/12	2/2	2/2	18/18	16/16	20/20	19/19
3	14/11	20/14	15/12	22/16	15/12	22/16	18/20	26/23	19/22	27/24	19/23	27/25
7	13/10	20/15	14/11	21/16	14/11	21/16	21/23	33/29	21/25	32/28	21/25	32/28
11	13/10	20/15	14/11	21/16			23/25	36/31	22/26	34/30		
15	13/10	20/16	14/12	21/17			24/26	37/32	-/-	-/31		

### 4.5.2 Piecewise-constant coefficients

As a test case for our  $H(\text{curl})$  multigrid solver, we consider a definite Maxwell problem proposed by Kolev & Vassilevski [56, §6.2] and adapted by Pazner et al. [78, §6.4]. The problem models electromagnetic diffusion in an annular copper wire in air, with a piecewise-constant diffusion coefficient  $\beta$  in (4.2). As in Pazner et al., we employ a  $Q_3$  coordinate field. We set  $\Gamma_D = \emptyset$ ,  $\alpha = 1$ ,  $\beta_{\text{copper}} = 1$ , and vary the diffusion constant of air  $\beta_{\text{air}}$ . Since the SC-PH space decomposition exhibits the best performance in the experiments of Section 4.5.1, we only consider this

**Table 4.3:** CG iteration counts for the  $H(\text{div})$  Riesz map solved with the {PAFW/SC-PAFW; PH/SC-PH} preconditioners.

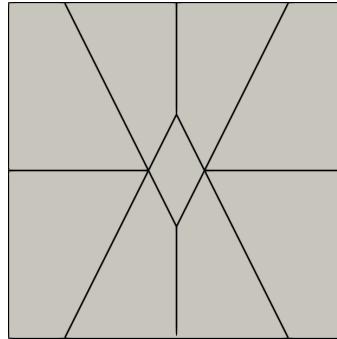
$p \setminus l$	Cartesian						Unstructured					
	0	1	2	0	1	2	0	1	2	0	1	2
1	2/2	2/2	11/11	14/14	12/12	15/15	2/2	2/2	17/17	22/22	19/19	22/22
3	9/8	15/11	9/8	16/13	10/9	16/13	15/21	24/22	16/23	25/25	18/25	26/27
7	9/7	17/13	9/8	17/13	9/9	17/14	18/25	31/27	18/26	30/28	19/27	30/29
11	9/7	17/13	9/8	18/14			19/27	33/29	19/28	31/29		
15	9/8	17/13	9/8	18/14			19/28	34/30	-/28	-/30		

**Table 4.4:** CG iteration counts for the  $H(\text{curl})$  Riesz map discretized with  $p = 7$ ,  $l = 2$  solved using the {PAFW/SC-PAFW; PH/SC-PH} hybrid preconditioners.

$\beta \setminus \alpha$	Cartesian						Unstructured					
	$10^{-3}$	$10^0$	$10^3$	$10^{-3}$	$10^0$	$10^3$	$10^{-3}$	$10^0$	$10^3$	$10^{-3}$	$10^0$	$10^3$
$10^{-6}$	12/9	20/15	14/11	22/16	-/24	22/18	19/27	31/28	21/27	32/28	20/25	-/32
$10^{-3}$	10/8	18/14	12/9	20/15	14/11	22/16	17/27	30/28	19/27	31/28	21/27	32/28
$10^0$	12/10	20/15	10/8	18/14	12/9	20/15	19/27	31/28	17/27	30/28	19/27	31/28
$10^3$	11/8	21/13	12/10	20/15	10/8	18/14	24/28	42/32	19/27	31/28	17/27	30/28
$10^6$	11/8	23/14	11/8	21/13	12/9	20/15	29/30	51/38	24/28	42/32	19/27	31/28

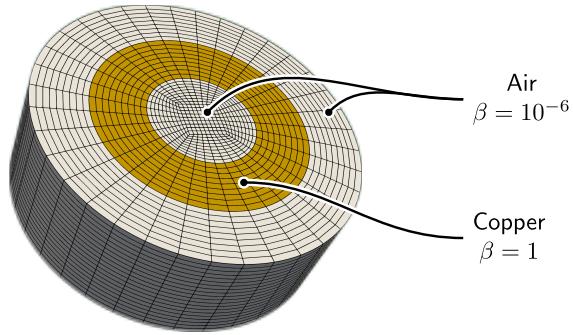
**Table 4.5:** CG iteration counts for the  $H(\text{div})$  Riesz map discretized with  $p = 7$ ,  $l = 2$  solved using the {PAFW/SC-PAFW; PH/SC-PH} hybrid preconditioners.

$\beta \setminus \alpha$	Cartesian						Unstructured					
	$10^{-3}$	$10^0$	$10^3$	$10^{-3}$	$10^0$	$10^3$	$10^{-3}$	$10^0$	$10^3$	$10^{-3}$	$10^0$	$10^3$
$10^{-6}$	8/8	15/11	9/9	17/14	10/11	17/15	16/23	26/25	19/27	30/29	19/27	30/29
$10^{-3}$	6/6	12/9	8/8	15/11	9/9	17/14	12/18	20/19	16/23	26/25	19/27	30/29
$10^0$	8/7	15/11	6/6	12/9	8/8	15/11	16/22	26/23	12/18	20/19	16/23	26/25
$10^3$	7/5	21/11	8/7	15/11	6/6	12/9	18/28	38/28	16/22	26/23	12/18	20/19
$10^6$	7/4	23/10	7/5	21/11	8/7	15/11	22/31	48/33	18/28	38/28	16/22	26/23



**Figure 4.7:** The mesh employed for the complexity plots in Figure 4.2 is the extrusion with three cells in the vertical of the two-dimensional meshes shown here.

solver here, and as the  $p$ -coarse solver we apply the Hypre AMS [56] algebraic multigrid cycle, and set a relative tolerance of  $10^{-8}$ .



**Figure 4.8:** Schematic of the definite Maxwell  $H(\text{curl})$  problem with piecewise-constant coefficients, obtained from [78]. The conductivity coefficient  $\beta$  is a piecewise constant coefficient determined by the material of the elements.

We first consider robustness of CG iteration counts to the magnitude of the jump in the coefficients, in Table 4.6. The results exhibit almost perfect robustness across twelve orders of magnitude for  $\beta_{\text{air}}$  and across polynomial degrees between 2 and 14. This contrasts with [78, Table 2], where the iteration counts for the LOR-AMS solver roughly double from  $p = 2$  to  $p = 6$ . We also tabulate the memory and solve times required as a function of  $p$  in Table 4.7 for fixed  $\beta_{\text{air}} = 10^{-6}$ .

### 4.5.3 Mixed formulation of Hodge Laplacians

The Riesz maps provide building blocks for developing preconditioners for more complex systems [45, 53, 70, 68]. In this final example, we demonstrate this by constructing preconditioners for the mixed formulation of the Hodge Laplacians associated with the  $L^2$  de Rham complex. For  $\Gamma_D = \emptyset$ , the problem is to find

**Table 4.6:** CG iteration counts for the  $H(\text{curl})$  definite Maxwell problem with piecewise-constant coefficients, solved with the SC-PH preconditioner.

$p$	#DOFs	$\beta_{\text{air}}$				
		$10^{-6}$	$10^{-3}$	$10^0$	$10^3$	$10^6$
2	516,820	25	23	21	29	33
3	1,731,408	24	22	21	26	31
4	4,088,888	25	23	21	25	31
5	7,968,340	26	24	21	28	30
6	13,748,844	27	24	21	26	29
10	63,462,980	28	25	21	27	29
14	173,920,348	28	25	22	28	25

**Table 4.7:** Memory usage and runtimes for the definite Maxwell problem in  $H(\text{curl})$  with piecewise-constant coefficients ( $\beta_{\text{air}} = 10^{-6}$ ) solved with the SC-PH preconditioner. NNZ Mat includes the number of nonzeros of the symmetric part of  $H(\text{curl})$  and  $H(\text{grad})$ . Schur complements, the ideal restriction matrix onto the interface of  $H(\text{curl})$ , and the tabulation of the gradient of the interface basis functions. Memory records the storage of all matrices across the solver. The runtime has been broken down into assembly of the sparse Schur complements, setup of the subspace problems, and solve times.

$p$	NNZ Mat	NNZ Fact	Memory (GB)	Assembly (s)	Setup (s)	Solve (s)
2	$1.73 \times 10^7$	$1.47 \times 10^7$	1.03	0.42	1.31	1.11
3	$5.40 \times 10^7$	$6.53 \times 10^7$	2.53	0.59	1.58	1.46
4	$1.25 \times 10^8$	$1.73 \times 10^8$	5.45	0.90	3.47	2.73
5	$2.42 \times 10^8$	$3.61 \times 10^8$	10.22	1.42	6.46	4.48
6	$4.16 \times 10^8$	$6.49 \times 10^8$	17.28	2.22	8.49	7.64
10	$1.91 \times 10^9$	$3.24 \times 10^9$	76.04	9.60	31.67	34.15
14	$5.23 \times 10^9$	$9.16 \times 10^9$	209.86	27.08	75.10	94.90

$(\sigma, u) \in V^{k-1} \times V^k$  such that

$$-(\tau, \sigma)_\Omega + (\mathbf{d}^{k-1} \tau, u)_\Omega = 0 \quad \forall \tau \in V^{k-1}, \quad (4.59)$$

$$(v, \mathbf{d}^{k-1} \sigma)_\Omega + (\mathbf{d}^k v, \mathbf{d}^k u)_\Omega = F(v) \quad \forall v \in V^k, \quad (4.60)$$

where  $F(v)$  is a random right-hand side given by (4.58).

To solve this saddle point problem, we follow the operator preconditioning framework of Hiptmair and Mardal & Winther [45, 70], employing a block diagonal preconditioner with the Riesz maps ( $\alpha = \beta = 1$ ) for  $V^{k-1}$  and  $V^k$ . We use 4 Chebyshev iterations preconditioned by the SC-PH solver for each block. For  $L^2$ , we use 4 Chebyshev iterations preconditioned by point-Jacobi in the FDM basis for  $DQ_{p-1}$ . As outer Krylov solver we employ the minimum residual method

(MINRES) [74], as this allows for the solution of indefinite problems using a symmetric coercive preconditioner. The convergence criterion for the iteration is a relative reduction of the (natural)  $P^{-1}$ -norm of the residual by a factor of  $10^8$ , starting from a zero initial guess. The iteration counts for the cases  $k = 1, 2, 3$  are reported in Table 4.8. As for the Riesz maps, we observe robustness with respect to both  $h$  and  $p$ .

**Table 4.8:** MINRES iteration counts for the mixed formulation of the Hodge Laplacians ( $k = 1, 2, 3$ ) preconditioned with the Riesz maps using 4 Chebyshev iterations of the SC-PH solvers on each block.

$p \setminus l$	$H(\text{grad}) \times H(\text{curl})$			$H(\text{curl}) \times H(\text{div})$			$H(\text{div}) \times L^2$					
	Cartesian		Unstructured	Cartesian		Unstructured	Cartesian		Unstructured	Cartesian		
0	1	2	0	1	2	0	1	2	0	1	2	
3	9	9	9	16	16	17	8	7	7	15	16	18
7	9	9	8	20	19	19	8	8	6	19	18	18
11	9	9		23	21		8	8		20	18	
15	9	9		23	21		8	6		20	18	

# 5

## Conclusions and future work

### 5.1 Conclusions

We have developed multigrid solvers for the Riesz maps associated with the  $L^2$  de Rham complex whose space and time complexities in polynomial degree are the same as that required for operator application. Numerical experiments demonstrate that the solvers are robust to mesh refinement, polynomial degree, and problem coefficients, and that they remain effective on unstructured grids. The approach relies on developing new finite elements with desirable interior-orthogonality properties, auxiliary operators that are sparse by construction, the careful use of incomplete factorizations, and the choice of space decomposition. The resulting solvers can be employed in the operator preconditioning framework to develop preconditioners for more complex problems with solution variables in  $H(\text{grad})$ ,  $H(\text{curl})$ ,  $H(\text{div})$ , and  $L^2$ .

A downside of our approach is its narrow applicability; it will not be effective on more general problems, especially for nonsymmetric problems where the dominant terms include first order derivatives. In addition, our method relies on having a good quality mesh, with its performance depending on the minimal angle; however, mesh generators with guarantees on the minimal angle are available in two dimensions [62]. So far, we have only experimented with piecewise constant-coefficient problems, but the theory of [11] suggests that our approach would remain effective for spatially varying coefficients.

## 5.2 Future work

### 5.2.1 Krylov methods on statically-condensed problems

As we pointed out in Remark 4.4, our statically-condensed solvers do not eliminate the interior DOFs upfront, but implement a statically-condensed space decomposition on an auxiliary operator. Traditionally, solvers for the statically-condensed problem rely on the availability of fast solvers for the interior DOFs. In general, a preconditioned Krylov method on the condensed system will improve the conditioning with respect to a non-statically-condensed solver, and thus reduce the number of Krylov iterations. However, this comes at the cost of evaluating the Schur complement at each iteration, which involves the solution of linear systems for each cell interior (this can be done approximately up to a tolerance). Our preliminary implementation of a statically-condensed Krylov method for  $H(\text{grad})$  with point-Jacobi/CG interior solvers was not competitive with the Krylov method on the full set of DOFs with the statically-condensed preconditioner, as the former involves increased work per iteration. Pursuing an efficient implementation of local solvers based on point-block Jacobi for the de Rham complex is part of our future research.

### 5.2.2 DG – CG multigrid solver for $H(\text{div}) \times L^2$ -conforming elasticity on unstructured meshes

In Chapter 3 we describe an issue with the FDM/sparse relaxation applied to the displacement block discretized with  $H(\text{div})$ -conforming elements and the symmetric IP-DG method on deformed cells. The main difficulty is that, in order to ensure consistency and coercivity, the additional surface integrals must vanish for arguments in  $C^0$ . On generally-deformed cells, these surface integrals introduce off-diagonal couplings between distinct vector components in the stiffness matrix, which will not be sparse in the FDM basis.

Inspired by two level DG-CG multigrid schemes such as [31], an approach that we found effective is the introduction of an additional  $H(\text{grad})$ -conforming level in the  $p$ -multigrid hierarchy, with sufficiently high polynomial degree on which we implement the vertex-star relaxation. On the finest level we propose to apply a cheap relaxation, such as point-Jacobi. We have promising preliminary results, although these are not robust with respect to the penalty coefficient.

### 5.2.3 Application to nonlinear problems

With regards to Chapter 3, we are interested in the application of the linear elasticity solvers to the linearization of nonlinear elasticity models, such as incompressible neo-Hookean hyperelasticity. For such nonlinear problems the Jacobian stiffness matrix is solution-dependent, hence the coefficients in the bilinear forms are spatially varying.

We are particularly interested in computing multiple solutions of nonlinear problems through deflated continuation [37]. This framework applies continuation on the parameters of the problem, e.g., the magnitude of the load applied to a structure or the viscosity of a fluid, to discover unknown branches of solutions. Such analyses can involve thousands or tens of thousands of Newton iterations, and thus a good preconditioner will become crucial to save valuable computational resources.

These extensions appear to be feasible, and would greatly expand the scope of applicability of the preconditioners developed here. In this way I hope that we will be able to solve challenging problems at much higher polynomial degrees than are currently affordable.

## Bibliography

- [1] M. Ainsworth, P. Coggins, and B. Senior. Mixed  $hp$ -finite element methods for incompressible flow. *Chapman and Hall CRC Research Notes in Mathematics*, pages 1–20, 2000.
- [2] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Anal. Appl.*, 23(1):15–41, 2001.
- [3] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. SIAM, Philadelphia, PA, third edition, 1999.
- [4] D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.*, 19(4):742–760, 1982.
- [5] D. N. Arnold. *Finite Element Exterior Calculus*. SIAM, 2018.
- [6] D. N. Arnold, D. Boffi, and F. Bonizzoni. Finite element differential forms on curvilinear cubic meshes and their approximation properties. *Numer. Math.*, 129(1):1–20, 2015.
- [7] D. N. Arnold, D. Boffi, and R. S. Falk. Quadrilateral  $H(\text{div})$  finite elements. *SIAM J. Numer. Anal.*, 42(6):2429–2451, 2005.
- [8] D. N. Arnold, R. Falk, and R. Winther. Preconditioning in  $H(\text{div})$  and applications. *Math. Comput.*, 66(219):957–984, 1997.
- [9] D. N. Arnold, R. S. Falk, and R. Winther. Multigrid in  $H(\text{div})$  and  $H(\text{curl})$ . *Numer. Math.*, 85(2):197–217, 2000.
- [10] K. J. Arrow, L. Hurwicz, and H. Uzawa. *Studies in Non-linear Programming*. Stanford University Press, 1958.
- [11] O. Axelsson and J. Karátson. Equivalent operator preconditioning for elliptic problems. *Numer. Algorithms*, 50(3):297–380, 2009.

- [12] I. Babuška. The finite element method with Lagrangian multipliers. *Numer. Math.*, 20(3):179–192, 1973.
- [13] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Karpeyev, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.11, Argonne National Laboratory, 2019.
- [14] P. D. Bello-Maldonado and P. F. Fischer. Scalable low-order finite element preconditioners for high-order spectral element Poisson solvers. *SIAM J. Sci. Comput.*, 41(5):S2–S18, 2019.
- [15] C. Bernardi and Y. Maday. Uniform inf–sup conditions for the spectral discretization of the Stokes problem. *Math. Models Methods Appl. Sci.*, 9(03):395–414, 1999.
- [16] C. Bernardi, Y. Maday, and B. Métivet. Spectral approximation of the periodic-nonperiodic Navier-Stokes equations. *Numer. Math.*, 51(6):655–700, 1987.
- [17] R. Blaheta. Displacement decomposition—Incomplete factorization preconditioning techniques for linear elasticity problems. *Numer. Lin. Algebra Appl.*, 1(2):107–128, 1994.
- [18] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. *ESAIM Math. Model. Num.*, 8(R2):129–151, 1974.
- [19] P. D. Brubeck and P. E. Farrell. Multigrid solvers for the de Rham complex with optimal complexity in polynomial degree. *arXiv preprint arXiv:2211.14284*, 2022.
- [20] P. D. Brubeck and P. E. Farrell. A scalable and robust vertex-star relaxation for high-order FEM. *SIAM J. Sci. Comput.*, 44(5):A2991–A3017, 2022.
- [21] C. Canuto. Stabilization of spectral methods by finite element bubble functions. *Comput. Methods Appl. Mech. Engrg.*, 116(1):13–26, 1994.
- [22] C. Canuto, P. Gervasio, and A. Quarteroni. Finite-element preconditioning of G-NI spectral methods. *SIAM J. Sci. Comput.*, 31:4422–4451, 2010.

- [23] C. Canuto and A. Quarteroni. Preconditioned minimal residual methods for Chebyshev spectral calculations. *J. Comput. Phys.*, 60(2):315–337, 1985.
- [24] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam. Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Down-date. *ACM Trans. Math. Softw.*, 35(3), 2008.
- [25] P. G. Ciarlet. *The finite element method for elliptic problems*. SIAM, 2002.
- [26] B. Cockburn, G. Kanschat, and D. Schötzau. A note on discontinuous Galerkin divergence-free solutions of the Navier–Stokes equations. *J. Sci. Comput.*, 31(1-2):61–73, 2007.
- [27] W. Couzy and M. O. Deville. A fast Schur complement method for the spectral element discretization of the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 116(1):135 – 142, 1995.
- [28] M. O. Deville and E. H. Mund. Chebyshev pseudospectral solution of second-order elliptic equations with finite element preconditioning. *J. Comput. Phys.*, 60(3):517–533, 1985.
- [29] M. O. Deville and E. H. Mund. Finite-element preconditioning for pseudospectral solutions of elliptic problems. *SIAM J. Sci. Stat. Comput.*, 11(2):311–342, 1990.
- [30] L. T. Diosady and S. M. Murman. Scalable tensor-product preconditioners for high-order finite-element methods: Scalar equations. *J. Comput. Phys.*, 394:759–776, 2019.
- [31] V. A. Dobrev, R. D. Lazarov, P. S. Vassilevski, and L. T. Zikatanov. Two-level preconditioning of discontinuous Galerkin approximations of second-order elliptic equations. *Numer. Linear Algebra Appl.*, 13(9):753–770, 2006.
- [32] C. R. Dohrmann. Spectral Equivalence of Low-Order Discretizations for High-Order  $H(\text{curl})$  and  $H(\text{div})$  Spaces. *SIAM J. Sci. Comput.*, 43(6):A3992–A4014, 2021.
- [33] M. Dryja and O. Widlund. *An additive variant of the Schwarz alternating method for the case of many subregions*. Technical Report 339, Ultracomputer Note 131. Department of Computer Science, Courant Institute, 1987.

- [34] O. Egorova, M. Savchenko, V. Savchenko, and I. Hagiwara. Topology and geometry of hexahedral complex: combined approach for hexahedral meshing. *J. Comput. Sci. Technol.*, 3(1):171–182, 2009.
- [35] L. P. Eisenhart. Separable systems of Stäckel. *Ann. Math.*, 35(2):284–305, 1934.
- [36] R. D. Falgout and U. Meier Yang. Hypre: A Library of High Performance Preconditioners. In P. M. A. Sloot, A. G. Hoekstra, C. J. K. Tan, and J. J. Dongarra, editors, *Computational Science – ICCS 2002*, volume 2331 of *Lecture Notes in Computer Science*, pages 632–641. Springer Berlin Heidelberg, 2002.
- [37] P. E. Farrell, Á. Birkisson, and S. W. Funke. Deflation techniques for finding distinct solutions of nonlinear partial differential equations. *SIAM J. Sci. Comput.*, 37(4):A2026–A2045, 2015.
- [38] P. E. Farrell, M. G. Knepley, L. Mitchell, and F. Wechsung. PCPATCH: software for the topological construction of multigrid relaxation methods. *ACM Trans. Math. Softw.*, 47(3), 2021.
- [39] P. F. Fischer, H. M. Tufo, and N. I. Miller. An overlapping Schwarz method for spectral element simulation of three-dimensional incompressible flows. In *Parallel Solution of Partial Differential Equations*, pages 159–180. Springer, 2000.
- [40] A. George. Nested dissection of a regular finite element mesh. *SIAM J. Numer. Anal.*, 10(2):345–363, 1973.
- [41] I. Gustafsson and G. Lindskog. On parallel solution of linear elasticity problems: Part I: theory. *Numer. Lin. Algebra Appl.*, 5(2):123–139, 1998.
- [42] B. Hientzsch. *Fast solvers and domain decomposition preconditioners for spectral element discretizations of problems in  $H(\text{curl})$* . PhD thesis, New York University, 2001.
- [43] B. Hientzsch. Domain decomposition preconditioners for spectral Nédélec elements in two and three dimensions. In *Domain Decomposition Methods in Science and Engineering*, pages 597–604. Springer, 2005.
- [44] R. Hiptmair. Multigrid Method for Maxwell’s Equations. *SIAM J. Numer. Anal.*, 36(1):204–225, 1998.

- [45] R. Hiptmair. Operator preconditioning. *Comput. Math. Appl.*, 52(5):699–706, 2006.
- [46] R. Hiptmair and J. Xu. Nodal auxiliary space preconditioning in  $H(\text{curl})$  and  $H(\text{div})$  spaces. *SIAM J. Numer. Anal.*, 45(6):2483–2509, 2007.
- [47] M. Homolya, R. C. Kirby, and D. A. Ham. Exposing and exploiting structure: optimal code generation for high-order finite element methods. *arXiv preprint arXiv:1711.02473*, 2017.
- [48] M. Homolya, L. Mitchell, F. Luporini, and D. A. Ham. TSFC: a structure-preserving form compiler. *SIAM J. Sci. Comput.*, 40(3):C401–C428, 2018.
- [49] I. Huismann, J. Stiller, and J. Fröhlich. Scaling to the stars—a linearly scaling elliptic solver for  $p$ -multigrid. *J. Comput. Phys.*, 398:108868, 2019.
- [50] G. Karniadakis and S. Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2013.
- [51] D. S. Kershaw. Differencing of the diffusion equation in Lagrangian hydrodynamic codes. *J. Comput. Phys.*, 39(2):375–395, 1981.
- [52] R. C. Kirby. Algorithm 839: FIAT, a new paradigm for computing finite element basis functions. *ACM Trans. Math. Soft.*, 30(4):502–516, 2004.
- [53] R. C. Kirby. From functional analysis to iterative methods. *SIAM Rev.*, 52(2):269–293, 2010.
- [54] M. G. Knepley and D. A. Karpeev. Mesh algorithms for PDE with Sieve I: Mesh distribution. *Sci. Program.*, 17(3):215–230, 2009.
- [55] T. Kolev, P. Fischer, M. Min, J. Dongarra, J. Brown, V. Dobrev, T. Warburton, S. Tomov, M. S. Shephard, A. Abdelfattah, et al. Efficient exascale discretizations: High-order finite element methods. *Int. J. High Perform. Comput. Appl.*, 35(6):527–552, 2021.
- [56] T. V. Kolev and P. S. Vassilevski. Parallel auxiliary space AMG for  $H(\text{curl})$  problems. *J. Comput. Math.*, 27(5):604–623, 2009.
- [57] T. V. Kolev and P. S. Vassilevski. Parallel auxiliary space AMG solver for  $H(\text{div})$  problems. *SIAM J. Sci. Comput.*, 34(6):A3079–A3098, 2012.

- [58] M. Kronbichler and W. A. Wall. A performance comparison of continuous and discontinuous Galerkin methods with fast multigrid solvers. *SIAM J. Sci. Comput.*, 40(5):A3423–A3448, 2018.
- [59] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. 1950.
- [60] P. L. Lederer and J. Schöberl. Polynomial robust stability analysis for  $H(\text{div})$ -conforming finite elements for the Stokes equations. *IMA J. Numer. Anal.*, 38(4):1832–1860, 2018.
- [61] Y.-J. Lee, J. Wu, J. Xu, and L. Zikatanov. Robust subspace correction methods for nearly singular systems. *Math. Models Methods Appl. Sci.*, 17(11):1937–1963, 2007.
- [62] X. Liang and Y. Zhang. Hexagon-based all-quadrilateral mesh generation with guaranteed angle bounds. *Comput. Methods Appl. Mech. Engrg.*, 200(23):2005–2020, 2011.
- [63] A. Logg. Efficient representation of computational meshes. *Int. J. Comput. Sci. Eng.*, 4(4):283–295, 2009.
- [64] J. W. Lottes and P. F. Fischer. Hybrid multigrid/Schwarz algorithms for the spectral element method. *J. Sci. Comput.*, 24(1):45–78, 2005.
- [65] R. E. Lynch, J. R. Rice, and D. H. Thomas. Direct solution of partial difference equations by tensor product methods. *Numer. Math.*, 6:185–199, 1964.
- [66] Y. Maday, D. Meiron, A. T. Patera, and E. M. Rønquist. Analysis of iterative methods for the steady and unsteady Stokes problem: Application to spectral element discretizations. *SIAM J. Sci. Comput.*, 14(2):310–337, 1993.
- [67] J.-F. Maitre and O. Pourquier. Condition number and diagonal preconditioning: comparison of the  $p$ -version and the spectral element methods. *Numer. Math.*, 74(1):69–84, 1996.
- [68] J. Málek and Z. Strakoš. *Preconditioning and the Conjugate Gradient Method in the Context of Solving PDEs*, volume 1 of *SIAM Spotlights*. SIAM, 2014.
- [69] J. Manzanero, A. M. Rueda-Ramírez, G. Rubio, and E. Ferrer. The Bassi Rebay 1 scheme is a special case of the symmetric interior penalty formulation for discontinuous Galerkin discretisations with Gauss–Lobatto points. *J. Comput. Phys.*, 363:1–10, 2018.

- [70] K.-A. Mardal and R. Winther. Preconditioning discretizations of systems of partial differential equations. *Numer. Linear Algebra Appl.*, 18(1):1–40, 2011.
- [71] J. R. Munkres. *Elements of Algebraic Topology*. CRC Press, 1984.
- [72] J.-C. Nédélec. Mixed finite elements in  $\mathbb{R}^3$ . *Numer. Math.*, 35(3):315–341, 1980.
- [73] S. A. Orszag. Spectral methods for problems in complex geometries. *J. Comput. Phys.*, 37(1):70–92, 1980.
- [74] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 12(4):617–629, 1975.
- [75] L. F. Pavarino. Additive Schwarz methods for the  $p$ -version finite element method. *Numer. Math.*, 66(1):493–515, 1993.
- [76] L. F. Pavarino, E. Zampieri, R. Pasquetti, and F. Rapetti. Overlapping Schwarz methods for Fekete and Gauss–Lobatto spectral elements. *SIAM J. Sci. Comput.*, 29(3):1073–1092, 2007.
- [77] W. Pazner. Efficient low-order refined preconditioners for high-order matrix-free continuous and discontinuous Galerkin methods. *SIAM J. Sci. Comput.*, 42(5):A3055–A3083, 2020.
- [78] W. Pazner, T. Kolev, and C. Dohrmann. Low-order preconditioning for the high-order finite element de Rham complex. *arXiv preprint arXiv:2203.02465*, 2022.
- [79] W. Pazner and P.-O. Persson. Approximate tensor-product preconditioners for very high order discontinuous Galerkin methods. *J. Comput. Phys.*, 354:344–369, 2018.
- [80] W. Pazner and P.-O. Persson. Interior penalty tensor-product preconditioners for high-order discontinuous Galerkin discretizations. In *2018 AIAA Aerospace Sciences Meeting*, page 1093, 2018.
- [81] M. Pellikka, S. Suuriniemi, L. Kettunen, and C. Geuzaine. Homology and cohomology computation in finite element modeling. *SIAM J. Sci. Comput.*, 35(5):B1195–B1214, 2013.

- [82] F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. McRae, G.-T. Bercea, G. R. Markall, and P. H. J. Kelly. Firedrake: automating the finite element method by composing abstractions. *ACM Trans. Math. Soft.*, 43(3):24:1–24:27, 2016.
- [83] J.-F. Remacle, R. Gandham, and T. Warburton. GPU accelerated spectral finite elements on all-hex meshes. *J. Comput. Phys.*, 324:246–257, 2016.
- [84] H. P. Robertson. Bemerkung über separierbare Systeme in der Wellenmechanik. *Math. Ann.*, 98(1):749–752, 1928.
- [85] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, 1986.
- [86] J. Schöberl. *Robust multigrid methods for parameter dependent problems*. PhD thesis, Johannes Kepler Universität Linz, Linz, Austria, 1999.
- [87] J. Schöberl, J. M. Melenk, C. Pechstein, and S. Zaglmayr. Additive Schwarz preconditioning for p-version triangular and tetrahedral finite elements. *IMA J. Numer. Anal.*, 28(1):1–24, 2008.
- [88] J. Schöberl and S. Zaglmayr. High order Nédélec elements with local complete sequence properties. *COMPEL - Int. J. Comput. Math. Electr. Electron. Eng.*, 2005.
- [89] D. Schötzau and C. Schwab. Mixed  $hp$ -FEM on anisotropic meshes. *Math. Models Methods Appl. Sci.*, 8(05):787–820, 1998.
- [90] T. Schwedes, D. A. Ham, S. W. Funke, and M. D. Piggott. *Mesh Dependence in PDE-Constrained Optimisation*. Springer, 2017.
- [91] K. Shahbazi. An explicit expression for the penalty parameter of the interior penalty method. *J. Comput. Phys.*, 205(2):401–407, 2005.
- [92] D. Silvester and A. Wathen. Fast iterative solution of stabilised Stokes systems Part II: using general block preconditioners. *SIAM J. Numer. Anal.*, 31(5):1352–1367, 1994.
- [93] J. Stiller. Robust multigrid for Cartesian interior penalty DG formulations of the Poisson equation in 3D. In M. L. Bittencourt, N. A. Dumont, and

- J. S. Hesthaven, editors, *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016*, pages 189–201, Cham, 2017. Springer International Publishing.
- [94] B. Szabó and I. Babuška. *Finite element analysis*. John Wiley & Sons, 1991.
  - [95] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Comput. Fluids*, 1(1):73–100, 1973.
  - [96] J. Witte, D. Arndt, and G. Kanschat. Fast tensor product Schwarz smoothers for high-order discontinuous Galerkin methods. *J. Comput. Methods Appl. Math.*, 2020.
  - [97] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Rev.*, 34(4):581–613, 1992.
  - [98] J. Xu. The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids. *Computing*, 56(3):215–235, 1996.
  - [99] S. Zaglmayr. *High Order Finite Element Methods for Electromagnetic Field Computation*. PhD thesis, Johannes Kepler University Linz, 2006.
  - [100] Software used in ‘Multigrid solvers for the de Rham complex with optimal complexity in polynomial degree’. <https://doi.org/10.5281/zenodo.7358044>, Nov 2022.