LOW-ORDER PRECONDITIONING FOR THE HIGH-ORDER FINITE ELEMENT DE RHAM COMPLEX

WILL PAZNER¹, TZANIO KOLEV¹, AND CLARK DOHRMANN²

Abstract. In this paper we present a unified framework for constructing spectrally equivalent low-order-refined discretizations for the high-order finite element de Rham complex. This theory covers diffusion problems in $H^1$, $H(\text{curl})$, and $H(\text{div})$, and is based on combining a low-order discretization posed on a refined mesh with a high-order basis for Nédélec and Raviart–Thomas elements that makes use of the concept of polynomial histopolation (polynomial fitting using prescribed mean values over certain regions). This spectral equivalence, coupled with algebraic multigrid methods constructed using the low-order discretization, results in highly scalable matrix-free preconditioners for high-order finite element problems in the full de Rham complex. Additionally, a new lowest-order (piecewise constant) preconditioner is developed for high-order interior penalty discontinuous Galerkin (DG) discretizations, for which spectral equivalence results and convergence proofs for algebraic multigrid methods are provided. In all cases, the spectral equivalence results are independent of polynomial degree and mesh size; for DG methods, they are also independent of the penalty parameter. These new solvers are flexible and easy to use; any “black-box” preconditioner for low-order problems can be used to create an effective and efficient preconditioner for the corresponding high-order problem. A number of numerical experiments are presented, using the finite element library MFEM, with which the construction of such preconditioners requires only one or two lines of code. The theoretical properties of these preconditioners are corroborated, and the flexibility and scalability of the method are demonstrated on a range of challenging three-dimensional problems.

1. Introduction

High-order finite element methods posed on the discrete de Rham complex (cf. [5, 6]) are of increasing relevance for a wide range of computational applications [1, 7, 27, 42, 74]. Problems such as electromagnetic diffusion (Maxwell’s equations), radiation–diffusion transport, and porous media flow require the solution of finite element problems posed in $H(\text{curl})$ and $H(\text{div})$ spaces (using Nédélec and Raviart–Thomas elements, respectively) [16, 60, 68] and radiative transfer [41] require the solution of interior penalty-type DG discretizations of elliptic problems posed on $L^2$ finite element spaces. In addition to their attractive accuracy properties, high-order discretizations also lend themselves well to efficient implementations on modern computing architectures [35, 46, 47]; in particular, sum-factorization techniques allow for efficient operator evaluation on tensor-product (quadrilateral and hexahedral) meshes at high-orders [59, 62]. However, the efficient iterative solution of the linear systems that result from these discretizations remains a challenging problem. The condition number of these linear systems typically scales like $O(p^3/h^2)$ or $O(p^4/h^2)$, where $p$ is the polynomial degree, and $h$ is the mesh size, cf. [56, 58], necessitating the use of scalable preconditioners that are robust with respect to the polynomial degree. Furthermore, while sum factorization techniques give efficient operator evaluation, it is typically not feasible to assemble and store the associated system matrix, ruling out the direct use of matrix-based preconditioning techniques such as incomplete factorizations and algebraic multigrid. This motivates the development of matrix-free preconditioners, that can be constructed without explicit access to the matrix entries of the high-order operator. Matrix-free solvers for continuous Galerkin discretizations of elliptic problems have been well studied [10, 52, 53, 54, 66]. In this paper, we develop scalable matrix-free preconditioners for high-order finite element discretizations posed on $H(\text{curl})$, $H(\text{div})$, and $L^2$ finite element spaces using a low-order-refined methodology. Our work extends this well known approach (see below) from $H^1$ to the full de Rham complex, utilizing and complementing the results from [38] and [32].

Low-order preconditioning is a classical technique for preconditioning high-order and spectral discretizations of the Poisson problem. This approach was first proposed by Orszag in 1980, who studied the use of
standard finite difference methods as preconditioners for spectral methods. The idea was further developed, using low-order finite element methods as a preconditioner for Chebyshev spectral and pseudospectral methods, independently by Deville and Mund and Canuto and Quarteroni; see for a review of these techniques. Fischer and Lottes developed Schwarz solvers for the pressure solver of the incompressible Navier–Stokes equations using low-order preconditioning in Pazner and Kolev applied low-order preconditioning to high-order continuous and discontinuous Galerkin methods with (nonconforming) $hp$-refinement. The aforementioned works use tensor-product elements (mapped quadrilaterals and hexahedra); Chalmers and Warbuton considered the extension to simplex elements.

The key property that enables the low-order preconditioning of high-order methods is the spectral equivalence of the low-order and high-order operators. This property is often known as the finite element method–spectral element method (FEM–SEM) equivalence, and was shown in Pazner and Kolev applied this low-order equivalence results from $H^1$ to $H(\text{curl})$ and $H(\text{div})$ is more involved, in part because of the nontrivial nullspaces of the curl and divergence operators (consisting of irrotational and solenoidal vector fields, respectively). In contrast, in the case of $H^1$ finite element spaces, the kernel of the gradient operator consists only of constant functions. In this paper, we give a unified presentation of high-order–low-order spectral equivalence for all spaces in the de Rham complex. These spectral equivalences are based on properties of one-dimensional polynomial interpolation and histopolation operators (polynomial histopolation is the process of finding a polynomial with given integrals over a certain number of disjoint intervals). The interpolation and histopolation operators give rise to specially designed high-order bases for Nédélec and Raviart–Thomas elements. These bases were introduced and their spectral equivalence properties and applications to preconditioning were studied in.

The structure of the paper is as follows. High-order and low-order interpolation and histopolation in one spatial dimension are discussed in Section 2. The extension to differential operators in multiple spatial dimensions is then considered in Section 3. The application of these properties to the spectral equivalence of high-order and low-order-refined finite element discretizations is presented in Section 4. The performance of matrix-free preconditioners for high-order problems based on this spectral equivalence is studied numerically in Section 5. In this section, we consider scalable preconditioners using algebraic multigrid methods applied to large-scale three-dimensional problems. We end with conclusions in Section 6.

2. ONE-DIMENSIONAL INTERPOLATION AND HISTOPOLATION

The high-order–low-order-refined equivalence depends on the appropriate choice of basis for the high-order finite element spaces. In this work, we construct bases for the $H^1$, $H(\text{curl})$ (Nédélec), $H(\text{div})$ (Raviart–Thomas), and $L^2$ finite element spaces on meshes with tensor-product elements that are designed to satisfy this equivalence. These bases are defined using the concepts of polynomial interpolation and histopolation on the reference 1D interval $[-1, 1]$. Interpolation defines a nodal (Lagrange) basis for the space $Q_p$ of degree-$p$ polynomials in $[-1, 1]$ using a set of $p + 1$ distinct nodal points and values. Histopolation, on the other hand, is a procedure that defines a degree-$(p - 1)$ polynomial in terms of its integrals over $p$ distinct subintervals.

Definition 1 (Interpolation). Let $\{x_i\}_{i=1}^{p+1} \subseteq [-1, 1]$ be a set of $p + 1$ distinct nodes. Then, the associated interpolation operator $I_p: \mathbb{R}^{p+1} \to Q_p$ maps prescribed point values $u = \{u_i\}_{i=1}^{p+1}$ to the unique polynomial interpolant $u = I_p(u) \in Q_p$ such that $u(x_i) = u_i$ for all $i$.

Definition 2 (Histopolation). Let $\{I_i = (x_i^L, x_i^R)\}_{i=1}^p$ be a set of $p$ disjoint subintervals of $[-1, 1]$, and let $h_i$ denote the size of the $i$th interval, $h_i = x_i^R - x_i^L$. Then, the associated histopolation operator $H_{p-1}: \mathbb{R}^p \to Q_{p-1}$ maps prescribed average values $\tilde{u} = \{\tilde{u}_i\}_{i=1}^p$ to the unique polynomial histopolant $u = H_{p-1}(\tilde{u}) \in Q_{p-1}$ such that $\frac{1}{h_i} \int_{x_i^L}^{x_i^R} u(x) dx = \tilde{u}_i$ for all $i$.

Proposition 1. The interpolation and histopolation operators $I_p$ and $H_{p-1}$ are well-defined linear bijections.

Proof. The operator $I_p$ can be defined through the classical Lagrange interpolation procedure. For the case of $H_{p-1}$, it suffices to show that for a given $u \in Q_{p-1}$, if $\int_{x_i^L}^{x_i^R} u dx = 0$ for all $i$ then $u \equiv 0$. This is a well-known and simple result, which we repeat here (cf. [20], see also [45] for a generalization). On each of the disjoint intervals $I_i$, since $\int_{x_i^L}^{x_i^R} u dx = 0$, either $u \equiv 0$ or $u$ changes sign on $I_i$. In the latter case, $u \in Q_{p-1}$ has $p$ distinct zeros, and so $u \equiv 0$. \qed
2.1. Gauss–Lobatto points. In what follows, we choose the interpolation points and histopolation subintervals to be compatible, in the sense that the $p+1$ interpolation points \( \{x_i\}_{i=1}^{p+1} \) also define the $p$ subintervals by \( \{ (x_i, x_{i+1}) \}_{i=1}^{p} \). In particular, we choose $x_i$ to be the $p+1$ Gauss–Lobatto points on the reference interval, which are given by the zeros of \((1-x^2)P_p(x)\), where $P_p(x)$ is the degree-$p$ Legendre polynomial. Since these points include the interval endpoints, the associated subintervals form a partition of $[-1, 1]$. Additionally, these points, as well as others such as the Gauss–Legendre and Chebyshev points, are asymptotically distributed according to the Chebyshev density $\left(\frac{p+1}{\pi \sqrt{1-x^2}}\right)$ [13] [72].

Remark 1 (Notation). We will write $a \lesssim b$ to mean that $a \leq Cb$, where $C$ is independent of the polynomial degree $p$ (and other discretization parameters, such as the mesh size $h$, when relevant). $a \gtrsim b$ means $b \leq a$, and $a \approx b$ means both $a \lesssim b$ and $b \lesssim a$. Similarly, for two symmetric and positive-definite matrices $A$ and $B$, $A \approx B$ means that $A$ and $B$ are spectrally equivalent, and the constants of equivalence are independent of $p$ (and other discretization parameters).

2.2. Low-order equivalences. Analogous to the high-order interpolation and histopolation operators, we also define low-order interpolation and histopolation operators, $I_h$ and $H_h$. Given the interpolation points \( \{x_i\} \), let $V_h$ denote the space of piecewise linear functions with nodes at each point $x_i$. Similarly, given the $p$ disjoint subintervals \( \{(x_i, x_{i+1})\} \), we define $\hat{V}_h$ to be the space of piecewise constant functions spanned by the indicator functions of these subintervals. Let $I_h : \mathbb{R}^{p+1} \rightarrow V_h$ denote the piecewise linear interpolation operator, and let $H_h : \mathbb{R}^p \rightarrow \hat{V}_h$ denote the operator that maps average values to the piecewise constant function in $\hat{V}_h$ taking those values over each subinterval. Then, we have the following equivalences in the $L^2([-1,1])$-norm (cf. [19] [45]).

**Proposition 2.** It holds that

\[
\begin{align*}
\|I_h u\|_0^2 &\approx \|I_p u\|_0^2 \quad \text{for all } u \in \mathbb{R}^{p+1}, \\
\|H_h \hat{u}\|_0^2 &\approx \|H_{p-1} \hat{u}\|_0^2 \quad \text{for all } \hat{u} \in \mathbb{R}^p.
\end{align*}
\]

**Proof.** The proof of (1) is given in [19] Prop. 2.1. The proof of (2) is given in [45] Prop. 5. \qed

Remark 2. It additionally holds that $\|H_h u\|_0^2 \approx \|I_{p-1} u\|_0^2$ for all $u \in \mathbb{R}^p$. This equivalence directly compares the interpolation and histopolation operators applied to the same vector of values, and is a consequence of the asymptotic equivalence between the Gauss–Lobatto quadrature weights and subinterval lengths (cf. [45] Proposition 3).

Remark 3. The operators $I_p$ and $H_{p-1}$ induce bases for the polynomial spaces $Q_p$ and $Q_{p-1}$, where each basis vector is the image of a standard Cartesian basis vector under the aforementioned operators. Similarly, the operator $I_q$ induces the standard basis of “hat functions” on the piecewise linear space $V_h$, and $H_h$ induces the basis of piecewise constant indicator functions on the space $\hat{V}_h$.

The high-order interpolatory (nodal) basis functions $\ell_j \in Q_p$ are given by the Lagrange interpolating polynomials of the nodal points $x_i$. The high-order histopolatory basis functions $\vartheta_j \in Q_{p-1}$ satisfy $\int_{x_i}^{x_{i+1}} \vartheta_j(x) \, dx = h_i \delta_{ij}$. It is straightforward to see that these functions are given by the negative partial sum of the derivatives of the nodal basis functions,

\[
\vartheta_j(x) = -h_i \sum_{k=1}^{i} \ell_k(x).
\]

Graphical plots of these basis functions are shown in [32].

The high-order and low-order interpolants and histopolants are closely related through their derivatives. For any $f \in H^1$ that interpolates nodal values $(x_i, y_i)$, it is straightforward to see that $f'$ has mean value $m_i = (y_{i+1} - y_i)/(x_{i+1} - x_i)$ over the interval $[x_i, x_{i+1}]$. As a consequence, the derivative $I_p(y_i)'$ of the nodal interpolant of $f$ is equal to the histopant $H_{p-1}(m_i)$, and similarly $I_h(y_i)' = H_h(m_i)$. This gives the following natural relationship between the derivatives of the high-order and low-order interpolants. Let $I_h : Q_p \rightarrow V_h$ and $H_h : Q_{p-1} \rightarrow \hat{V}_h$ be the high-order to low-order interpolation and histopolation operators, defined by $I_h = I_h I_p^{-1}$ and $H_h = H_h H_{p-1}^{-1}$, respectively. Similarly, let $I_p = I^{-1}_h$ and $H_p = H_{p-1}^{-1}$ denote the low-order to high-order operators.
Proposition 3. Let \( u \in \mathbb{R}^{p+1} \) be given, and let \( u_p = I_p(u) \in Q_p \), and \( u_h = I_h(u) \in V_h \). Then,
\[
\mathcal{H}_p^{-1} u_p' = \mathcal{H}_h^{-1} u_h',
\]
and so \( u_h' = I_h(u_p)' = H_h(u_p') \) and \( u_p' = I_p(u_h)' = H_p(u_h') \).

Equivalently, the following diagram commutes, where \( \partial \) represents the derivative operator.

\[
\begin{array}{ccc}
Q_p & \xrightarrow{\partial} & Q_{p-1} \\
\downarrow I_h & & \downarrow H_h \\
V_h & \xrightarrow{\partial} & \hat{V}_h
\end{array}
\]

Proof. \( \int_{x_i}^{x_{i+1}} u_h'(x) \, dx = u_h(x_{i+1}) - u_h(x_i) = u_{i+1} - u_i = u_p(x_{i+1}) - u_p(x_i) = \int_{x_i}^{x_{i+1}} u_p'(x) \, dx. \)

Remark 4. A restatement of the above proposition is that interpolating and then differentiating is equivalent to differentiating and then histopolating, using both the high-order and low-order operators. Consequently, from the perspective of high-order–low-order-refined equivalence, it is natural to represent functions by histopolating average values of subintervals defined by the same nodal points, and it is natural to represent their derivatives by histopolating their point values at nodal points, and it is natural to represent their derivatives by histopolating.

Corollary 1. \( \| (I_h u)' \|_0^2 \approx \| (I_p u)' \|_0^2 \) for all \( u \in \mathbb{R}^{p+1} \).

Proof. Let \( \hat{u} = \mathcal{H}_p^{-1} u_p' = \mathcal{H}_h^{-1} u_h' \) (by Proposition 3). Then, \( (I_h u)' = \mathcal{H}_h \hat{u} \) and \( (I_p u)' = \mathcal{H}_p^{-1} \hat{u} \), and so the result follows from Proposition 3.

3. Equivalences in multiple dimensions

In Section 2.2, the norm equivalence for the one-dimensional high-order and low-order interpolation and differentiation operators was established. In this section, we extend the construction of interpolation and histopolation operators to multiple dimensions using a tensor-product construction. Additionally, we prove the analogous norm equivalence properties for the high-order and low-order interpolation, gradient, curl, and divergence operators.

The interpolation operator on the \( d \)-dimensional reference element \([-1, 1]^d\) is defined by
\[
I_p^d : \mathbb{R}^{(p+1)^d} \to Q_p([-1, 1]^d), \quad I_p^d = I_p \otimes \cdots \otimes I_p.
\]

This operator maps point values defined on the Cartesian product of nodal points to their unique multivariate interpolating polynomial. The low-order piecewise multilinear interpolation operator \( I_p^d \) can be defined analogously, and similarly for the histopolation operators \( \mathcal{H}_{p-1} \) and \( \mathcal{H}_h \).

Proposition 4. Let \( u \in \mathbb{R}^{(p+1)^d} \) and \( \hat{u} \in \mathbb{R}^{p^d} \). Then, \( \| I_p^d u \|_0^2 \approx \| I_h^d u \|_0^2 \) and \( \| \mathcal{H}_h^d \hat{u} \|_0^2 \approx \| \mathcal{H}_{p-1}^d \hat{u} \|_0^2 \).

Proof. The proof is an immediate consequence of Proposition 2 and properties of the tensor product.
3.1. **Gradient operators.** These interpolation operators naturally give rise to high-order and low-order gradient operators,

\[ G_p^d = \nabla I_p^d \quad \text{and} \quad G_h^d = \nabla I_h^d. \]

The norm equivalences of the previous section can be extended to show the equivalence of the high-order and low-order gradient operators.

**Proposition 5.** Let \( u \in \mathbb{R}^{(p+1)^d} \). Then, \( \|G_p^d u\|_0^2 \approx \|G_p u\|_0^2 \).

**Proof.** For ease of notation, we write the proof here for \( d = 3 \), although the extension to arbitrary \( d \) is straightforward. Note that \( G_p^3 \) and \( G_h^3 \) are given by

\[ G_p^3 = \nabla I_p^3 = \begin{pmatrix} (\partial I_p) \otimes I_p \otimes I_p \\ I_p \otimes (\partial I_p) \otimes I_p \\ I_p \otimes I_p \otimes (\partial I_p) \end{pmatrix}, \quad G_h^3 = \nabla I_h^3 = \begin{pmatrix} (\partial I_h) \otimes I_h \otimes I_h \\ I_h \otimes (\partial I_h) \otimes I_h \\ I_h \otimes I_h \otimes (\partial I_h) \end{pmatrix}. \]

By Proposition 3, we have \( \partial I_h = H_h H_p^{-1} \partial I_p \), and therefore can write

\[ G_h^3 = \begin{pmatrix} H_h H_p^{-1} \otimes I_h I_p^{-1} \otimes I_h I_p^{-1} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} G_p^3. \]

From Proposition 2, we have that \( \|I_h I_p^{-1} u\|^2 \approx \|u\|_0^2 \) for all \( u \in Q_p \) and \( \|H_h H_p^{-1} u\|^2 \approx \|v\|_0^2 \) for all \( v \in Q_p^{-1} \). The result then follows from properties of the tensor product. \( \square \)

3.2. **Curl operators.** In light of Remark 4 and the mapping \( H^1 \overset{\text{grad}}{\rightarrow} H(\text{curl}) \), it is natural to represent discrete functions in \( H(\text{curl}) \) using the histopolation and interpolation operators

\[ I_p^{\text{curl}} = \begin{pmatrix} H_p^{-1} \otimes I_p \otimes I_p \\ 0 \\ 0 \end{pmatrix}, \quad I_h^{\text{curl}} = \begin{pmatrix} H_h \otimes I_h \otimes I_h \\ 0 \\ 0 \end{pmatrix}. \]

The high-order and low-order curl operators are then defined naturally as

\[ C_p = \nabla \times I_p^{\text{curl}}, \quad \text{and} \quad C_h = \nabla \times I_h^{\text{curl}}. \]

**Proposition 6.** Let \( u \in \mathbb{R}^{p(p+1)^2} \). Then, \( \|I_p^{\text{curl}} u\|_0^2 \approx \|I_h^{\text{curl}} u\|_0^2 \) and \( \|C_h u\|_0^2 \approx \|C_p u\|_0^2 \).

**Proof.** The first equivalence is a simple consequence of Proposition 2. To show the second equivalence, we write the operators \( C_p \) and \( C_h \) explicitly as

\[ C_p = \begin{pmatrix} 0 \\ -I_p \otimes H_p^{-1} \otimes (\partial I_h) \\ -H_p^{-1} \otimes (\partial I_p) \otimes I_p \\ 0 \\ (\partial I_h) \otimes H_h \otimes (\partial I_h) \\ -H_h \otimes (\partial I_h) \otimes I_h \\ 0 \end{pmatrix}, \]

\[ C_h = \begin{pmatrix} H_h \otimes I_h \otimes (\partial I_h) \\ -H_h \otimes (\partial I_h) \otimes I_h \\ 0 \end{pmatrix}, \]

from which the relation

\[ C_h = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} C_p \]

is clear. The conclusion then follows as in the proof of Proposition 5. \( \square \)
3.3. Divergence operators. Given the mapping $H(\text{curl}) \xrightarrow{\text{curl}} H(\text{div})$, the interpolation operators in $H(\text{div})$ are naturally given by

$$T_p^{\text{div}} = \begin{pmatrix} I_p \otimes \mathcal{H}_{p-1} \otimes \mathcal{H}_{p-1} & 0 & 0 \\ 0 & \mathcal{H}_{p-1} \otimes I_p \otimes \mathcal{H}_{p-1} & 0 \\ 0 & 0 & \mathcal{H}_{p-1} \otimes \mathcal{H}_{p-1} \otimes I_p \end{pmatrix},$$

and the divergence operators by

$$D_p = \nabla \cdot T_p^{\text{div}}, \quad \text{and} \quad D_h = \nabla \cdot I_h^{\text{div}}.$$

**Proposition 7.** Let $u \in \mathbb{R}^{p^2(p+1)}$. Then, $\|T_p^{\text{div}}u\|^2_0 \approx \|T_h^{\text{div}}u\|^2_0$ and $\|D_h u\|^2_0 \approx \|D_p u\|^2_0$.

**Proof.** The first equivalence is a simple consequence of Proposition 2. To show the second equivalence, we write the operators $D_p$ and $D_h$ as

$$D_p = \left( (\partial I_p) \otimes \mathcal{H}_{p-1} \otimes \mathcal{H}_{p-1} \mathcal{H}_{p-1} \otimes (\partial I_p) \right);$$

$$D_h = \left( (\partial I_h) \otimes \mathcal{H}_h \otimes \mathcal{H}_h \mathcal{H}_h \otimes (\partial I_h) \right),$$

from which it can be seen that $D_h = H_h^- (H_h^p)^{-1} D_p$, and the conclusion follows.

**Remark 6.** As an immediate consequence of the above propositions, the high-order and low-order gradient, curl, and divergence operators have identical nullspaces. This is particularly important for the construction of preconditioners for discretizations in $H(\text{curl})$ and $H(\text{div})$, for which treating the nontrivial nullspaces of the curl and divergence operators is a key challenge.

**Remark 7.** The spectral equivalence results of Propositions 4-7 were first proven using explicit calculations in [5, 6]. In this section, we have provided alternative, systematic proofs of these results based only on the one-dimensional interpolation and histopolation operators.

4. Finite element spaces

The results of the previous sections can be used to define spectral equivalences of operators defined on $H^1$, $H(\text{curl})$, $H(\text{div})$, and $L^2$ finite element spaces, which make up the discrete de Rham complex in 3D (cf. [3, 6]):

$$H^1(\Omega) \xrightarrow{\text{grad}} H(\text{curl}, \Omega) \xrightarrow{\text{curl}} H(\text{div}, \Omega) \xrightarrow{\text{div}} L^2(\Omega).$$

We begin by considering a spatial domain $\Omega \subseteq \mathbb{R}^3$ (analogous results also apply in a straightforward manner to domains in one and two spatial dimensions). The domain is discretized using a hexahedral mesh $T_p$. Each element of the mesh $\kappa \in T_p$ is the image of the reference element $\kappa = [-1, 1]^3$ under a smooth mapping, $\kappa = T_\kappa(\hat{\kappa})$. Given the mesh $T_p$, we define the following high-order and low-order-refined finite element spaces.

4.1. High-order spaces. Recall the standard $H^1$, $H(\text{curl})$ (Nédélec), $H(\text{div})$ (Raviart-Thomas), and $L^2$ finite element spaces, defined as follows. Note that in addition to the natural $L^2$ finite element space in the de Rham complex, we also define a discontinuous Galerkin spaces that can be considered as a “broken $H^1$” space.

- The $H^1$ finite element space
  $$V_p = \{ v \in H^1(\Omega) : v|_\kappa \circ T_\kappa \in Q_p(\kappa) \}.$$  

- The $H(\text{curl})$ finite element space with Nédélec elements
  $$W_p = \{ w \in H(\text{curl}) : w|_\kappa \circ T_\kappa \in W_p(\kappa) \}.$$  

The local space $W_p(\kappa)$ is the image of the reference space $W_p(\kappa) = Q_{p-1,p,p} \times Q_{p,p-1,p} \times Q_{p,p,p}$ under the $H(\text{curl})$ Piola transformation $W_p(\kappa) = J_\kappa^{-T} W_p(\kappa)$, where $J_\kappa$ is the Jacobian matrix of the element transformation $T_\kappa$. 
The $H(\text{div})$ finite element space with Raviart–Thomas elements

$$X_p = \{ x \in H(\text{div}) : w|_\kappa \circ \tau_\kappa \in X_p(\kappa) \}.$$  

The local space $X_p(\kappa)$ is the image of the reference space $X_p(\tilde{\kappa}) = \mathbb{Q}_{p,p-1,p-1} \times \mathbb{Q}_{p-1,p,p-1} \times \mathbb{Q}_{p-1,p-1,p}$ under the $H(\text{div})$ Piola transformation $X_p(\kappa) = \det(J_\kappa)^{-1}J_\kappa X_p(\tilde{\kappa})$, where $J_\kappa$ is the Jacobian matrix of the element transformation $\tau_\kappa$.

- The $L^2$ finite element space

$$Y_{p-1} = \{ y \in L^2(\Omega) : y|_\kappa \circ \tau_\kappa \in Y_{p-1}(\kappa) \}.$$  

The local space $Y_{p-1}(\kappa)$ is the image of the reference space $Y_{p-1}(\tilde{\kappa}) = \mathbb{Q}_{p-1}$ under the integral-preserving mapping, $Y_{p-1}(\kappa) = \det(J_\kappa)^{-1}Y_{p-1}(\tilde{\kappa})$, where $J_\kappa$ is the Jacobian matrix of the element transformation $\tau_\kappa$.

- The discontinuous Galerkin space

$$Z_p = \{ z \in L^2(\Omega) : z|_\kappa \circ \tau_\kappa \in \mathbb{Q}_p(\tilde{\kappa}) \}.$$  

In contrast to the $L^2$ space $Y_p$, the local DG space does not incorporate the integral-preserving mapping. For our purposes, it is more natural to consider the DG space $Z_p$ as a “broken $H^1$” finite element space rather than an $L^2$ space.

The spaces $V_p$, $W_p$, $X_p$, $Y_{p-1}$ provide a discrete analogue of the $L^2$ de Rham complex, in the sense that

$$V_p \xrightarrow{\text{grad}} W_p \xrightarrow{\text{curl}} X_p \xrightarrow{\text{div}} Y_{p-1}$$

is a complete sequence, where the range of each operator is exactly the kernel of the next one, e.g. $\text{curl} \ w = 0$ for $w \in W_p$ if and only if $w = \text{grad} \ v$ for some $v \in V_p$.

4.2. Low-order-refined spaces. For each of the spaces $V_p$, $W_p$, $X_p$, $Y_{p-1}$, $Z_p$, the corresponding low-order-refined spaces $V_h$, $W_h$, $X_h$, $Y_h$, $Z_p$ are given by the lowest-order finite element spaces, defined on a Gauss–Lobatto refined mesh. This refined mesh, denoted $\mathcal{T}_h$, is obtained by refining each element $\kappa \in \mathcal{T}_p$ as follows. Let $\{x_i\}_{i=1}^{p+1}$ denote the $p+1$ Gauss–Lobatto points in $[-1,1]$, and let $\{x_i\}_{i=1}^{(p+1)^3}$ denote their 3-fold Cartesian product in $[-1,1]^3$. The points $x_i$ define a submesh of the element $\kappa$, consisting of $p^3$ subelements. This submesh is structured (Cartesian), but nonuniform because of the clustering of the Gauss–Lobatto points at the endpoints of the interval. The $H^1$ space $V_h$ is the space of piecewise trilinear functions defined on $\mathcal{T}_h$, with degrees of freedom given by vertex values. The $H(\text{curl})$ space $W_h$ is the space of lowest-order edge elements, and the $H(\text{div})$ space $X_h$ is the space of lowest-order face elements defined on $\mathcal{T}_h$. The $L^2$ space $Y_h$ is the space of piecewise constants defined on $\mathcal{T}_h$. Similarly, the low-order DG space $Z_h$ (corresponding to the degree-$p$ DG space $Z_p$) is the space of piecewise constants defined on the mesh $\mathcal{T}_h$ obtained by refining $\mathcal{T}_p$ using $p+2$ Gauss–Lobatto points. An illustration of a low-order-refined hexahedral mesh is shown in Figure 1.
4.3. Choice of basis for the high-order spaces. To obtain the high-order–low-order-refined equivalences between the finite element spaces, we make use of bases built using the interpolation and histopolation operators from Sections 2 and 3. These bases are closely related to the so-called mimetic bases introduced by Gerritsma and colleagues in [38, 51, 75], and studied in the context of high-order–low-order equivalence in [32].

Recall the multi-dimensional interpolation and histopolation operators defined in Section 3.

\[ I_p^3 = I_p \otimes I_p \otimes I_p, \]
\[ I_p^{\text{curl}} = \begin{pmatrix} \mathcal{H}_{p-1} \otimes I_p \otimes I_p & 0 & 0 \\ 0 & I_p \otimes \mathcal{H}_{p-1} \otimes I_p & 0 \\ 0 & 0 & I_p \otimes I_p \otimes \mathcal{H}_{p-1} \end{pmatrix}, \]
\[ I_p^{\text{div}} = \begin{pmatrix} I_p \otimes \mathcal{H}_{p-1} \otimes I_p & 0 & 0 \\ 0 & \mathcal{H}_{p-1} \otimes I_p \otimes \mathcal{H}_{p-1} & 0 \\ 0 & 0 & \mathcal{H}_{p-1} \otimes \mathcal{H}_{p-1} \otimes I_p \end{pmatrix}, \]
\[ \mathcal{H}_{p-1}^3 = \mathcal{H}_{p-1} \otimes \mathcal{H}_{p-1} \otimes \mathcal{H}_{p-1}. \]

The images of the Cartesian basis vectors \( e^i \) (where \( e^i_j = \delta_{ij} \)) under each of the above mappings naturally define basis functions for the corresponding space of polynomials.

- The basis functions defined by \( I_p^3 : \mathbb{R}^{(p+1)^3} \rightarrow \mathcal{Q}_p \) are the standard nodal (Lagrange) basis functions corresponding to the Gauss–Lobatto points. By identifying coincident nodal points at element interfaces, the resulting functions are continuous across element interfaces, giving a basis for \( V_p \). In this case, the degrees of freedom are point values at the nodes.

- The basis functions defined by \( I_p^{\text{curl}} : \mathbb{R}^{(p+1)^3} \rightarrow \mathcal{Q}_{p-1,p,p} \times \mathcal{Q}_{p,p-1,p} \times \mathcal{Q}_{p,p,p-1} \) are used to define \( H(\text{curl}) \) basis functions. In this case, the degrees of freedom of the ith vector component are integrals over the segment connecting two neighboring nodes in the ith coordinate direction. This naturally results in tangential continuity, ensuring that the resulting piecewise polynomial functions form a basis for \( W_p \).

- The basis functions defined by \( I_p^{\text{div}} : \mathbb{R}^{(p+1)^3} \rightarrow \mathcal{Q}_{p,p-1,p-1} \times \mathcal{Q}_{p,p-1,p-1,p} \times \mathcal{Q}_{p,p,p-1,p} \) are used to define \( H(\text{div}) \) basis functions. The resulting degrees of freedom for the ith vector component are integrals over the two-dimensional surface defined by neighboring nodes in the two orthogonal coordinate directions. This naturally results in normal continuity, ensuring that the resulting piecewise polynomial functions form a basis for \( X_p \).

- The basis functions defined by \( \mathcal{H}_{p-1}^3 : \mathbb{R}^3 \rightarrow \mathcal{Q}_{p-1} \) are used for the \( L^2 \) space \( Y_{p-1} \). The resulting degrees of freedom represent integrals over subvolumes defined by the nodal points, enforcing no continuity between elements.

- For the discontinuous Galerkin space \( Z_p \), the basis functions defined by \( I_{p-1}^3 \) are used (as in the case of \( H^1 \) elements). The case of DG spaces is discussed in further detail in Section 4.5.

Remark 8. The degrees of freedom for the high-order finite element spaces described above coincide exactly with the standard lowest-order degrees of freedom for the low-order-refined spaces. Consequently, in the lowest-order cases (\( p = 1 \) for \( H^1 \), \( H(\text{curl}) \), and \( H(\text{div}) \) finite elements, \( p = 0 \) for \( L^2 \) finite elements), the basis functions described above reduce to the standard basis functions used for the lowest-order finite element spaces, and we recover the standard low-order vertex, edge, face, and element basis functions.

4.4. High-order–low-order-refined spectral equivalence. In this section, we consider the spectral equivalence of the mass and stiffness matrices defined on the finite element spaces \( V_p, W_p, X_p, \) and \( Y_{p-1} \).

For simplicity, we restrict the analysis in this section to the case where the mesh element transformations \( T_\kappa \) have constant Jacobians (in other words, \( T_\kappa \) is an affine transformation, and \( \kappa \) is a parallelepiped). The case of more general meshes (and variable coefficients) is studied numerically in Section 5.
We define the following transfer operators between the high-order and low-order spaces.

\[ P_V : V_p \rightarrow V_h \quad P_V(v)|_\kappa = T_h^3 \left( I_p^3 \right)^{-1} (u|_\kappa) \]
\[ P_W : W_p \rightarrow W_h \quad P_W(w)|_\kappa = T_p^{-1} \left( \mathcal{H}^1_{\text{curl}} \right)^{-1} (w|_\kappa) \]
\[ P_X : X_p \rightarrow X_h \quad P_X(x)|_\kappa = T_h^{\text{div}} \left( \mathcal{H}^1_{\text{div}} \right)^{-1} (x|_\kappa) \]
\[ P_Y : Y_{p-1} \rightarrow Y_h \quad P_Y(y)|_\kappa = \mathcal{H}^3_{\text{curl}} \left( \mathcal{H}^3_{p-1} \right)^{-1} (y|_\kappa) \]
\[ P_Z : Z_p \rightarrow Z_h \quad P_Z(z)|_\kappa = T_h^3 \left( I_p^3 \right)^{-1} (z|_\kappa) \]

The definition of these transfer operators (together with the standard commuting projection operators \( \Pi_\kappa \)) ensure that the following diagram commutes:

\[
\begin{array}{cccc}
H^1(\Omega) \xrightarrow{\text{grad}} H(\text{curl}, \Omega) \xrightarrow{\text{curl}} H(\text{div}, \Omega) \xrightarrow{\text{div}} L^2(\Omega) \\
\Pi_V \downarrow \quad \Pi_W \downarrow \quad \Pi_X \downarrow \quad \Pi_Y \\
V_p \xrightarrow{\text{grad}} W_p \xrightarrow{\text{curl}} X_p \xrightarrow{\text{div}} Y_{p-1} \\
\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\
V_h \xrightarrow{\text{grad}} W_h \xrightarrow{\text{curl}} X_h \xrightarrow{\text{div}} Y_h
\end{array}
\]

**Remark 9.** Note that because of the choice of basis laid out in Section 4.3, the matrix representation of each of the above \( P \) operators is the identity matrix. In other words, the same vector of degrees of freedom represents both element of the high-order finite element space, and its image under the transfer operator in the low-order-refined space.

These transfer operators result in the following norm and seminorm equivalences, which immediately give the spectral equivalences of the high-order and low-order-refined mass and stiffness matrices.

**Theorem 1.** It holds that

\[
(3) \quad \|v\|_0^2 \approx \|P_V(v)\|_0^2, \quad |v|_1^2 \approx |P_V(v)|_1^2 \quad \text{for all } v \in V_p,
\]
\[
(4) \quad \|w\|_0^2 \approx \|P_W(w)\|_0^2, \quad \|w\|_{\text{curl}}^2 \approx \|P_W(w)\|_{\text{curl}}^2 \quad \text{for all } w \in W_p,
\]
\[
(5) \quad \|x\|_0^2 \approx \|P_X(x)\|_0^2, \quad |x|_1^2 \approx |P_X(x)|_1^2 \quad \text{for all } x \in X_p,
\]
\[
(6) \quad \|y\|_0^2 \approx \|P_Y(y)\|_0^2 \quad \text{for all } y \in Y_{p-1},
\]
\[
(7) \quad \|z\|_0^2 \approx \|P_Z(z)\|_0^2, \quad \|z\|_h^2 \approx \|P_Z(z)\|_h^2 \quad \text{for all } z \in Z_p,
\]

where \( |w|_{\text{curl}} = \|\nabla \times w\|_0 \) and \( |x|_{\text{div}} = \|\nabla \cdot x\|_0 \) denote the curl and divergence seminorms, respectively, and \( \| \cdot \|_p \) and \( \| \cdot \|_h \) denote the high-order and low-order mesh-dependent DG norms (defined in Section 4.5).

**Proof.** The proof proceeds easily by summing over each element \( \kappa \in T_p \), using the assumption of constant Jacobians, and the norm equivalences established in Propositions 4.7. The equivalence in the DG norms \( \| \cdot \|_p \) and \( \| \cdot \|_h \) is deferred to Section 4.5.

**Theorem 2.** Let \( M_\bullet \) and \( K_\bullet \) denote the mass and stiffness matrices, respectively, where \( \bullet \) represents one of the above-defined finite element spaces with basis as in Section 4.3. Then we have the following spectral equivalences, independent of mesh size \( h \) and polynomial degree \( p \).

\[
M_{V_h} \sim M_{V_p}, \quad K_{V_h} \sim K_{V_p},
\]
\[
M_{W_h} \sim M_{W_p}, \quad K_{W_h} \sim K_{W_p},
\]
\[
M_{X_h} \sim M_{X_p}, \quad K_{X_h} \sim K_{X_p},
\]
\[
M_{Y_{h-p+1}} \sim M_{Y_{p-1}}, \quad K_{Z_h} \sim K_{Z_p}.
\]

**Proof.** These spectral equivalences follow immediately from the norm equivalences of Theorem 1.
4.5. Discontinuous Galerkin discretizations. In the context of the DG space $Z_p$, since no continuity is enforced between elements, it is possible to use either the interpolatory basis induced by $I_p^d$ or the histopolation basis induced by $H_p^d$. Both of these basis give rise of $L^2$ norm equivalence for the high-order and low-order spaces. However, the histopolation basis does not give a straightforward low-order equivalence for interior penalty discretizations of the diffusion operator. For this reason, we prefer to use the interpolatory (nodal) basis induced by $I_p^d$ for DG spaces. This is the same basis that is used for $H^1$ spaces, which is natural if we interpret the DG space as a “broken $H^1$” space rather than an $L^2$ finite element space.

The DG low-order-refined mesh $T_p^h$ is obtained by subdividing each element $\kappa \in T_p$ into $(p+1)^d$ subelements, defined by the Cartesian product of the $p+2$ Gauss– Lobatto points $x_i$. The nodal basis of $Z_p$ is defined using the $p+1$ Gauss– Lobatto points $x_i$. Note that the interfacing property of the Gauss– Lobatto quadrature implies that in every interval $[x_{i}, x'_{i+1}]$ there lies exactly one point $x_i$. The transfer operator $P_T$ then maps piezewise polynomials $u_p$ to piecewise constant functions whose constant value over each subcell is given by the value of $u_p$ at the unique nodal point lying in that subcell.

Consider the symmetric interior penalty (IP) discretization of the Poisson problem \[ (8) \]
\[ A_{Z_p}(u, v) = (\nabla_h u, \nabla_h v) - \langle \{\nabla_h u\}, \{v\} \rangle - \langle \{u\}, \{\nabla_h v\} \rangle + \langle \sigma_p [u], [v] \rangle = (f, v), \]
where $\nabla_h$ denotes the broken gradient operator and $\langle \cdot, \cdot \rangle$ denotes integration over element interfaces (i.e. over the mesh skeleton $\Gamma_p$ of $T_p^h$). The notation $\{ \cdot \}$ and $\langle \cdot \rangle$ is used to denote the average and jump of a function at element interfaces, respectively. In the above, $\sigma_p = \eta p^2/h$ is the penalty parameter, which must be chosen sufficiently large to obtain a stable method. The norm induced by IP discretization is equivalent to mesh dependent norm $\| \cdot \|_p$, defined by
\[ \|u\|_p^2 = \|\nabla_h u\|_0^2 + \|\sigma_p^{1/2} [u]\|_0^2 \|\Gamma_p \| \|v\|_p^2. \]

Proposition 8 (Cf. [2, 43]). For all $u, v \in Z_p$, it holds that
\[ A_{Z_p}(u, v) \lesssim \|u\|_p \|v\|_p, \]
\[ A_{Z_p}(u, u) \gtrsim \|u\|_p^2. \]

This norm equivalence allows for the construction of a spectrally equivalent low-order ($p = 0$) discretization defined on the refined mesh $T_h$. We first note that $A_{Z_p}$, restricted to the piecewise constant space $Z_h$ reduces to only the penalty term $\langle \sigma_p [u], [v] \rangle$, since $\nabla_h u = 0$ for all $u \in Z_h$. Therefore, we define the low-order interior penalty form
\[ (9) \]
\[ A_{Z_h} = \langle \sigma_h [u], [v] \rangle, \]
where the integrals are performed over $\Gamma_h$, the mesh skeleton of $T_h$. The choice of penalty parameter $\sigma_h$, defined as a piecewise constant field on each face $e \in \Gamma_h$, is of critical importance. The form $A_{Z_h}$ induces the low-order DG norm $\|u\|_h = \|\sigma_h^{1/2} [u]\|_0 \|\Gamma_h \|$.

We first write the low-order mesh skeleton $\Gamma_h$ as the disjoint union $\Gamma_h = \Gamma_o \cup \Gamma_d$. The set $\Gamma_d$ consists of faces that are subsets of coarse mesh faces, i.e. $e_d \in \Gamma_d$ satisfies $e_d \subseteq e_C$ for some $e_C \in \Gamma_p$. On the other hand, the set $\Gamma_o$ denotes those faces that lie in the interior of the coarse high-order macro-elements. A face $e_o \in \Gamma_o$ satisfies $e_o \not\subseteq e_C$ for all $e_C \in \Gamma_p$. The piecewise constant penalty parameter $\sigma_h(e)$ will be defined separately for $e \in \Gamma_o$ and $e \in \Gamma_d$.

Consider the reference macro-element $\kappa = [-1, 1]^3$, which has been decomposed into $(p+1)^3$ subelements. On general meshes, a scaling factor $\alpha$ is defined by multiplying by the ratio of the reference and physical element sizes, $\alpha = h/h$. The element size $h$ at the face is computed as the average of the sizes $h = \frac{1}{2} (h^+ + h^-)$ of the adjacent elements $\kappa^\pm$, where $h^\pm$ is computed as the perpendicular length of the element, $h^\pm = \mu (k^\pm) / \mu(e)$, where $\mu$ denotes measure.

First we consider an interior face $e_o \in \Gamma_o$, and, without loss of generality, we assume that $e_o$ is normal to the $z$ coordinate direction. In this case, $e_o = [x_{i}, x'_{i+1}] \times [x'_{j}, x'_{j+1}] \times [x'_{k}, x'_{k+1}]$ for some $(i, j, k)$. Since $e_o$ is an interior face, we have $1 < k < p + 2$. Let $x_i$ and $x_j$ denote the unique nodal points lying in $[x'_{i}, x'_{i+1}]$ and $[x'_{j}, x'_{j+1}]$, respectively, and let $w_i$ and $w_j$ denote the corresponding Gauss– Lobatto weights. Additionally, let $x_k$ and $x_{k+1}$ denote the unique nodal points lying in $[x'_{k-1}, x'_{k}]$ and $[x'_{k}, x'_{k+1}]$, respectively. Then, define
\[ (10) \]
\[ \sigma_p(e_o) = \frac{\alpha w_i w_j}{(x'_{i+1} - x'_{i})(x'_{j+1} - x'_{j})(x_{k+1} - x_k)}. \]
Now, consider the case of \( e_0 \in \Gamma_0 \). Without loss of generality, write \( e_0 = [x'_i, x'_{i+1}] \times [x'_j, x'_{j+1}] \times \{0\} \). As before, let \( w_i \) and \( w_j \) denote the Gauss–Lobatto weights corresponding to the unique nodes lying in the intervals \([x'_i, x'_{i+1}]\) and \([x'_j, x'_{j+1}]\). Then, define
\[
\sigma_h(e_0) = \alpha w_i w_j.
\]

Remark 10. The discretization defined by (9) is equivalent to the weighted graph Laplacian defined on the connectivity graph of the mesh \( T_h \). Let \( G = (V, E) \) be the graph defined by \( T_h \), such that each element \( \kappa_i \in T_h \) corresponds to a vertex \( i \in V \), and the edge \((i, j) \in E\) exists whenever elements \( \kappa_i, \kappa_j \in T_h \) share a common face \( e \in \Gamma_h \), in which case we also write \( i \sim j \).

For each graph edge \((i, j) \in E\), define the weight \( w_{ij} \) by \( w_{ij} = \sigma_h(e) \) (given by (10) and (11)), where \( e \in \Gamma_h \) is the interface between elements \( \kappa_i \) and \( \kappa_j \). For each graph vertex \( i \), the weight \( w_i \) is defined by \( w_i = \sum_{i \sim j} w_{ij} \). Then, the weighted graph Laplacian \( L \) of \( G \) is the matrix defined by
\[
L_{ij} = \begin{cases} 
-w_{ij} & \text{if } i \sim j, \\
w_i & \text{if } i = j, \\
0 & \text{otherwise.}
\end{cases}
\]

It is straightforward to see that the matrix \( L \) is identical to the stiffness matrix \( K_{Z_h} \) corresponding to the bilinear form \( A_{Z_p} \) in the case of Neumann boundary conditions.

**Theorem 3.** The low-order DG discretization defined by (9) with \( \sigma_h \) given by (10) and (11) is spectrally equivalent to the high-order DG discretization (8).

**Proof.** Let \( u_p \in Z_p \) be given, and let \( u_h = P_{Z_p} u_p \). Then, by Proposition [8]
\[
A_{Z_p}(u_p, u_p) \approx \|u_p\|^2 = \|\nabla_h u_p\|^2_0 + \|\sigma_h^{1/2}u_p\|^2_{0, \Gamma_p}.
\]

We first consider the term \( \|\sigma_h^{1/2}u_p\|^2_{0, \Gamma_p} \). For a given face \( e_p \in \Gamma_p \), consider the set \( \mathcal{E}(e_p) = \{e \in \Gamma_0 : e \subseteq e_p\} \). Each such subelement face \( e_i \in \mathcal{E}(e_p) \) corresponds to a nodal point \( x_i \in e_p \) and Gauss–Lobatto weight \( w_i \). Given definition (11), we have \( \sigma_h(e_i) = \alpha w_i^2 \). Using the property of Gauss–Lobatto quadrature that \( \|f\|^2_{0, e} \approx \sum_i w_i f(x_i)^2 \) (cf. [18]), we have
\[
\sum_{e_i \in \mathcal{E}(e_p)} \|\sigma_h^{1/2}u_{e_i}\|^2_{e_i} \approx \sum_{e_i \in \mathcal{E}(e_p)} \alpha w_i^2 \mu(e_i) \|u_h(x_i)\|^2 \approx \alpha h^{-1} \sum_i w_i \mu(e_i) \|u_p(x_i)\|^2 \approx \|\sigma_h^{1/2}u_p\|^2_{0, \Gamma_p},
\]

using that \( \alpha \approx h^{-1} \). Therefore, \( \|\sigma_h^{1/2}u_p\|^2_{0, \Gamma_p} \approx \|\sigma_h^{1/2}u_h\|^2_{0, \Gamma_p} \).

Now we consider the term \( \|\nabla_h u_p\|^2_{0, \kappa} \) on a given element \( \kappa \in T_p \). By Proposition [8], we have that \( \|\nabla_h u_p\|^2_{0, \kappa} \approx \|\nabla_h \tilde{u}_h\|^2_{0, \kappa} \), where \( \tilde{u}_h \) is the piecewise linear interpolant of \( u_p \). Now, consider an interior face \( e_o \in \Gamma_o \). Without loss of generality, assume that \( e_o \) is the image under the element transformation mapping \( T_\kappa \) of \( e = [x'_i, x'_{i+1}] \times [x'_j, x'_{j+1}] \times \{x\} \) for some \((i, j, k)\); the cases of faces normal to the \( x \) and \( y \) coordinate directions in the reference element follow analogously. On the face \( e_o \) we have \( \frac{\partial \tilde{u}_h}{\partial z} \approx h^{-1} \frac{[u_h]}{[x_{k+1} - x_k]} \) and so
\[
\int_\kappa \left( \frac{\partial \tilde{u}_h}{\partial z} \right)^2 \ dx \approx \sum_{ijk} h^3 w_i w_j w_k \left( \frac{\partial \tilde{u}_h}{\partial z} \right)^2 \approx \sum_{ijk} h w_i w_j w_k \left( \frac{[u_h]}{[x_{k+1} - x_k]} \right)^2.
\]

We compute
\[
\int_{e_o} \sigma_h \|u_h\|^2 \ ds = \int_{e_o} \frac{\alpha w_i w_j \mu(e_o)}{(x'_{i+1} - x'_i)(x'_{j+1} - x'_j)(x_{k+1} - x_k)} \|u_h\|^2 \approx \frac{h w_i w_j}{x_{k+1} - x_k} \|u_h\|^2 = h w_i w_j (x_{k+1} - x_k) \left( \frac{[u_h]}{[x_{k+1} - x_k]} \right)^2.
\]
Recalling that \( w_k \approx x_{k+1} - x_k \), and summing over all interior faces (including the \( x \)- and \( y \)-normal faces), we obtain
\[
\sum_{e \in E} \sigma_e \|u_b\|^2 \, ds \approx \sum_{e \in E} \|\nabla_h \tilde{u}_b\|_{0,e}^2 \approx \|\nabla_h u_p\|_0^2,
\]
and the result follows. \( \square \)

5. Algebraic Multigrid Preconditioning

Let \( A_h \) denote a convex combination of the low-order-refined stiffness and mass matrices \( K_h \) and \( M_h \) (where the subscript \( h \) is shorthand for one of \( V_h, W_h, X_h, Y_h, \) or \( Z_h \)). Let \( A_p \) denote the associated high-order operator. The spectral equivalence results of Theorem 2 (i.e. \( A_h \sim A_p \)) imply that any good preconditioner for the low-order and sparse system \( A_h \) will also be a good preconditioner for the corresponding high-order system \( A_p \). In principle, there are a number of multigrid, domain decomposition, and incomplete factorization preconditioners that will result in well-conditioned systems. In this work, we focus on algebraic multigrid (AMG) methods: these methods give essentially black-box highly scalable preconditioners for \( A_h \) requiring minimal discretization information. AMG convergence for lowest-order \( H^1 \) and DG finite element discretizations for elliptic problems has been extensively studied in the literature \([14, 23, 57, 70]\) and has further been extended to definite \( H(\text{curl}) \) and \( H(\text{div}) \) problems. Additionally, several high-performance massively parallel and GPU-accelerated implementations such as the BoomerAMG, AMS and ADS preconditioners in the hypre library \([34]\) are available.

Although not studied here, domain decomposition algorithms can also be used as preconditioners for the low-order matrix \( A_h \). Both iterative substructuring and overlapping Schwarz algorithms were analyzed in \([24]\) using the FEM–SEM equivalence of \( H^1 \) discretizations. That analysis was motivated by earlier numerical experiments in \([64]\). Overlapping Schwarz preconditioners using the FEM–SEM equivalence were studied recently in \([32]\) for both \( H(\text{curl}) \) and \( H(\text{div}) \) discretizations. The numerical results in that study were promising, but the analysis of domain decomposition preconditioners for \( A_h \) for problems in these two function spaces remains an open problem.

5.1. Mass matrix preconditioning. It is well known that the high-order mass matrix (using either nodal Gauss–Lobatto or Gauss–Legendre basis) is spectrally equivalent to its diagonal, independent of the polynomial degree \( p \) \([18, 20]\) (on parallelepiped elements, the Gauss–Legendre matrix is equal to its diagonal). In fact, it can be shown that on the reference interval \([-1, 1]\), the fully integrated mass matrix with Gauss–Lobatto basis is given by a rank-one update to the diagonal matrix of Gauss–Lobatto weights \([73]\). In this case, the matrix \( D^{-1} M \), where \( D = \text{diag}(M) \), has only two distinct eigenvalues, and its condition number decreases with increasing \( p \).

It is straightforward to show that the mass matrix using the interpolation–histopolation basis defined in this paper is also spectrally equivalent to its diagonal. A comparison of diagonal preconditioners for the high-order mass matrix is included in Section 6.3.

**Proposition 9.** Let \( M_p \) denote the high-order mass matrix defined on one of the spaces \( V_p, W_p, X_p, Y_{p-1}, \) or \( Z_p \). Let \( D_p = \text{diag}(M_p) \). Then, \( M_p \sim D_p \), independent of \( p \).

**Proof.** Let \( M_h \) denote the mass matrix defined on the corresponding low-order space. By Theorem 2 \( M_p \sim M_h \). But \( M_h \) is the standard mass matrix using the lowest-order basis, which is spectrally equivalent to its diagonal, \( D_h = \text{diag}(M_h) \). Since \( M_p \sim M_h \), we also have \( D_p \sim D_h \), and so \( M_p \sim M_h \sim D_h \sim D_p \). \( \square \)

5.2. Discontinuous Galerkin discretizations. We consider classical AMG methods applied to the low-order-refined DG discretization described in Section 4.5. Note that the graph Laplacian \( L \) defined by (12) is an M-matrix, for which the convergence of classical algebraic multigrid methods is well-studied \([70]\). As in the preceding sections, it is then expected that AMG applied to \( L \) will result in convergence that is independent of \( h \) and \( p \) (modulo the low-order mesh anisotropy). In this section, we show that classical AMG applied to \( L \) also converges independently of the penalty parameter \( \eta \). We relate this result to the family of preconditioners that use an associated \( H^1 \)-conforming discretization to precondition discontinuous Galerkin methods \([3, 31, 66]\).

Recalling the language of Remark 10, we consider the graph \( G = (V, E) \), and its associated weighted graph Laplacian \( L \). The edges \( e \in E \) can be categorized as either interior edges \( e \in E^\circ \), in which case the associated weight is given by \([10]\), or as boundary edges \( e \in E^\partial \), in which case the associated weight is given
The weights associated with interior edges are independent of \( \eta \), whereas the weights associated with boundary edges scale linearly with \( \eta \). In the following, we assume that \( \eta \gg 1 \), i.e. the boundary weights dominate the interior weights, and so we will say that two vertices are strongly connected if there exists a boundary edge connecting them.

We partition the graph \( G \) into a set of disjoint strongly connected components
\[
G = G^{(1)} \cup G^{(2)} \cup \cdots \cup G^{(n)} \cup G^o,
\]
where any two vertices of \( G^{(i)} = (V^{(i)}, E^{(i)}) \) must be connected by a path consisting of boundary edges. The strongly connected components of \( G \) correspond to groups of degrees of freedom lying on distinct mesh entities of the coarse mesh \( \mathcal{T}_p \). For example, all the degrees of freedom that are coincident with a mesh vertex belong to the same strongly connected component. Likewise, the two coincident degrees of freedom lying in the interior of a mesh element \( \kappa \in \mathcal{T}_p \) has no strong connections, and such vertices are included in the interior component \( G^o \). The number of vertices in a given strongly connected component of \( G \) is bounded by the valence of the coarse mesh \( \mathcal{T}_p \), which we assume to be \( \mathcal{O}(1) \).

Classical algebraic multigrid methods partition the vertices \( V \) of the graph into coarse (C) points and fine (F) points. Let \( P = \begin{bmatrix} W & I \end{bmatrix} \) denote the C-to-F interpolation operator. Let \( Q = P(P^T P)^{-1} P^T \) denote the orthogonal projection onto the range of \( P \). Because of the assumption that \( \eta \gg 1 \) the operator \( Q \) is decoupled across the strongly connected components of \( G \). Let \( Q^{(i)} \) denote the projections corresponding to the subgraphs \( G^{(i)} \) and \( G^o \), respectively. \( Q \) is then given as the product of these operators (since the projections are decoupled, this product is commutative). The operators \( Q^{(i)} \) possess two important properties:

(P1) Since the AMG interpolation \( P \) preserves constants, \( Q^{(i)} \) has row-sum equal to one.

(P2) Since \( Q^{(i)} \) is an orthogonal projection, \( |Q^{(i)}_{jk}| \leq 1 \) for all \( j,k \).

We proceed to show that the coarse grid defined by the C-points, together with Jacobi relaxation, results in a stable decomposition independent of \( \eta \), and hence uniform AMG convergence. In what follows, let \( D \) denote the diagonal of \( L \).

**Lemma 1.** Let \( u \) be given. The decomposition \( u = v^o + w^o \), where \( w^o = Q^o u \), is stable in the sense that
\[
\langle v^o \rangle^T D v^o + \langle w^o \rangle^T Lw^o \lesssim u^T L u,
\]
where the implied constant is independent of the penalty parameter \( \eta \).

**Proof.** Note that for any interior edge \( (i,j) \in E^o \), the associated weight \( w_{ij} \) is independent of \( \eta \), and so
\[
\sum_{(i,j) \in E^o} w_{ij}(w_i - w_j)^2 \lesssim \sum_{(i,j) \in E^o} w_{ij}(u_i - u_j)^2,
\]
where the above expressions have no dependence of \( \eta \). Since the subgraph \( G^o \subseteq G \) consists of those vertices that belong to no boundary edges, we have that \( w_i = u_i \) for any \( i \in V^o = V \setminus V^\partial \). Therefore, writing \( E = E^o \cup E^\partial \),
\[
\langle w^o \rangle^T Lw^o = \sum_{c_{ij} \in E} w_{ij}(w_i^o - w_j^o)^2 = \sum_{c_{ij} \in E^o} w_{ij}(u_i - u_j)^2 + \sum_{c_{ij} \in E^\partial} w_{ij}(w_i^o - w_j^o)^2 \lesssim \sum_{c_{ij} \in E} w_{ij}(u_i - u_j)^2 = u^T L u.
\]
Similarly, the diagonal entries \( w_{ii} \) associated with the vertices \( V^o \) are independent of \( \eta \), and \( v_i = 0 \) for any \( i \in V^\partial \), so \( \langle v^o \rangle^T L v^o \lesssim u^T L u \), and the conclusion follows. \( \square \)

**Remark 11.** The matrix associated with the subgraph \( G^o \) is an M-matrix whose entries do not depend on the penalty parameter \( \eta \). Therefore, the standard algebraic multigrid theory for M-matrices applies, and so the decomposition of Lemma 1 is expected to be stable not only with respect to the penalty parameter \( \eta \), but also other relevant discretization parameters such as mesh size and coefficients.

**Lemma 2.** Let \( u \) be given, and let \( u^{(i)} = Q^{(i)} u \). Then, for all \( j \),
\[
(w_j^{(i)} - u_j)^2 \lesssim \sum_{k \in V^{(i)}} (u_k - u_j)^2.
\]
Proof. Recalling the properties of the operator $Q^{(i)}$,

$$
(w_j^{(i)} - u_j)^2 = \left( \sum_{k \in V^{(i)}} Q_{jk}^{(i)} u_k - u_j \right)^2 = \left( \sum_{k \in V^{(i)}} Q_{jk}^{(i)} (u_k - u_j) \right)^2 \quad \text{by \[P1\]}
$$

$$
\lesssim \sum_{k \in V^{(i)}} (Q_{jk}^{(i)} (u_k - u_j))^2 \leq \sum_{k \in V^{(i)}} (u_k - u_j)^2 \quad \text{by \[P2\]}
$$

$\square$

Lemma 3. Let $Q^\vartheta = \prod_i Q^{(i)}$, and let $u$ be given. Then, the decomposition $u = v + w$, where $w = Q^\vartheta u$ is stable in the sense that

$$
v^T Dv + w^T Lw \lesssim u^T Lu,
$$

where the implied constant is independent of the penalty parameter $\eta$.

Proof. Using the result of Lemma 2

$$
w^T Lw = \sum_{(i,j) \in E} w_{ij} (w_i - w_j)^2 \lesssim \sum_{(i,j) \in E} w_{ij} \left( (u_i - u_j)^2 + (w_i - u_i)^2 + (w_j - u_j)^2 \right)
$$

$$
\lesssim \sum_{(i,j) \in E} w_{ij} \left( (u_i - u_j)^2 + \sum_{k \in V^{(i)}} (u_k - u_i)^2 + \sum_{k \in V^{(i)}} (u_k - u_j)^2 \right)
$$

The number of vertices in each of the subgraphs $G^{(i)}$ is $O(1)$, so

$$
\sum_{(i,j) \in E} w_{ij} \sum_{k \in V^{(i)}} (u_k - u_i)^2 \lesssim \sum_{(i,j) \in E} w_{ij} (u_i - u_j)^2,
$$

and we can conclude that

$$
w^T Lw \lesssim \sum_{(i,j) \in E} w_{ij} (u_i - u_j)^2 = u^T Lu.
$$

We now turn our attention to the term $v^T Dv$. Note that

$$
v^T Dv = \sum_i w_i v_i^2 = \sum_i w_i (w_i - u_i)^2.
$$

By Lemma 2

$$
\sum_i w_i (w_i - u_i)^2 \lesssim \sum_i w_i \sum_{k \in V^{(i)}} (u_k - u_i)^2.
$$

$\square$

Theorem 4. Let $u$ be given. Then, the decomposition

$$
u = v + w \quad \text{where} \quad w = Qu
$$

is stable in the sense that

$$
v^T Dv + w^T Lw \lesssim u^T Lu,
$$

where the implied constant is independent of $\eta$.

Proof. The stability of this decomposition follows from the stability of the interior and boundary decompositions, demonstrated above. First, consider the decomposition $u = w^\circ + v^\circ$, where $w^\circ = Q^\circ u$. By Lemma 1, this decomposition is stable, i.e.

$$
v^\circ^T Dv^\circ + w^\circ^T Lw^\circ \lesssim u^T Lu.
$$

Then, further decompose $w^\circ = w + v^\vartheta$, where $w = Q^\vartheta w^\circ$, and define $v$ by $v = v^\vartheta + v^\circ$. By Lemma 3, this decomposition is stable in the sense that

$$
v^\vartheta^T Dv^\vartheta + wLw \lesssim w^\vartheta^T Lw^\vartheta.
$$
Therefore,
\[ v^T Dv + w^T Lw = (v^\theta + v^\sigma)^T D (v^\theta + v^\sigma) + w^T Lw \]
\[ \lesssim v^\theta v^\theta + v^\sigma v^\sigma + w^T Lw \]
\[ \lesssim v^\sigma v^\sigma + w^T Lw^\sigma \]
\[ \lesssim u^T Lu \]
and the result follows. □

As a consequence of Theorem 4 classical algebraic multigrid applied to the low-order DG discretization described above will result in uniform convergence, independent of the DG penalty parameter \( \eta \). Standard AMG theory regarding the robustness of the convergence for M-matrices with respect to discretization parameters such as mesh size and variation in the coefficients also carries over to this case. This convergence theory is verified numerically in Section 6.6.

**Remark 12** (CG preconditioning for DG methods). The use of continuous Galerkin discretizations (together with a smoothing operation, such as Jacobi or Gauss–Seidel) as preconditioners for discontinuous Galerkin methods has been studied extensively in the literature \([3, 31, 66, 67]\). Theorem 4 gives an alternative, elementary, proof of the optimality of CG preconditioning for DG methods. Each subgraph \( G^{(i)} \) corresponds to a “duplicated” DG degree of freedom in the high-order problem, and hence also maps to a single \( H^1 \) degree of freedom. Note that defining the operators \( Q^{(i)} \) as the Oswald averaging operators (mapping \( u_i \) to \( (\# V^{(i)})^{-1} \sum u_i \), cf. \([17, 63]\)), properties \([P_2]\) and \([P_1]\) are satisfied. This can be viewed as using the \( H^1 \)-conforming subspace of the DG finite element space as an AMG coarse space. The conclusions of Theorem 4 hold for this case, showing that the resulting two-level method with Jacobi smoothing results in uniform convergence.

### 6. Numerical results

The high-order and low-order-refined discretizations described in this work were implemented in the MFEM open-source finite element software library (cf. [1], \url{https://mfem.org}). The only modification made to MFEM’s existing \( H(\text{curl}) \) and \( H(\text{div}) \) discretizations was the implementation of the histopolation basis; these basis functions can be computed in a straightforward manner using partial sums of the derivatives of the standard Lagrange basis functions, as described in Remark 3. The high-order finite element operators were constructed using \emph{partial assembly}, such that the action of the operator is applied without assembling the corresponding matrix, using geometric factors and coefficients that are precomputed at quadrature points. Basis function evaluation and numerical integration are performed using sum factorization. Per degree of freedom, this technique requires \( O(1) \) storage and \( O(p) \) operations.

In the MFEM library, the high-order operator can be assembled using the interpolation–histopolation basis by choosing the basis types \emph{BasisType::GaussLobatto} (interpolation at the Gauss–Lobatto nodes) and \emph{BasisType::IntegratedGLL} (histopolation using the Gauss–Lobatto subcells). The low-order-refined versions of the high-order operators can be constructed in one line of code using the LORDiscretization class. Preconditioners for the high-order discretization based on the low-order-refined matrices can similarly be constructed in one line using the LORSolver class. The \emph{lor_solvers} miniapp, and its parallel counterpart \emph{plor_solvers}, illustrate the construction of low-order-refined discretizations and solvers, and come distributed with MFEM’s source code, available at \url{https://github.com/mfem/mfem}.

The low-order refined mass and stiffness matrices \( M_h \) and \( K_h \) in \( H(\text{curl}) \) and \( H(\text{div}) \) correspond to the standard finite element discretizations using lowest-order Nédélec and Raviart–Thomas elements, posed on a refined mesh. While any effective preconditioner can be used for the resulting low-order system, in this work we mainly make use of the algebraic multigrid preconditioners available in \emph{hypre} [34]. In particular, the Auxiliary-Space Maxwell (AMS) solver (cf. [48]) is used for \( H(\text{curl}) \) problems, and the Auxiliary-Space Divergence (ADS) solver (cf. [49]) is used for \( H(\text{div}) \) problems. Classical algebraic multigrid is used for the DG discretizations. A key feature of the LOR preconditioning approach is that the resulting high-order solvers inherit performance benefits and scalability from the traditional low-order solver implementations. For example, since \emph{hypre}’s AMG solvers are highly scalable on massively parallel supercomputers, and also feature GPU acceleration, the LOR-based preconditioners also enjoy favorable scalability and GPU acceleration.
6.1. Interpolation and histopolation equivalences. The spectral equivalence results in this paper are consequences of the one-dimensional norm equivalences of the interpolation and histopolation operators, cf. Proposition 2. In this section, we numerically estimate the constants of the $L^2$ norm equivalences of the one-dimensional interpolation and histopolation operators, $I_p$, $I_h$, $H_{p-1}$, and $H_h$ defined on Gauss–Lobatto nodes. For the interpolation operators, we will also consider the numerically integrated $L^2$ norm, $\| \cdot \|_{0,NI}$, which is computed using $p+1$ (collocated) Gauss–Lobatto quadrature points. Note that the exactly integrated $L^2$ norm $\| \cdot \|_0$ and its numerically integrated counterpart $\| \cdot \|_{0,NI}$ are equivalent, independent of polynomial degree $p$ [18]. The use of numerical integration (inexact quadrature) can decrease the condition number of the preconditioned system; this effect has been studied in [12, 20, 36]. We numerically evaluate the value of the constants in estimates of the form

$$c \| I_h(u) \|_0 \leq \| I_p(u) \|_0 \leq C \| I_h(u) \|_0.$$

The quantity $Cc^{-1}$ is shown for polynomial degrees $2 \leq p \leq 64$ in Figure 2. As expected given Proposition 2, the quantity $Cc^{-1}$ remains asymptotically bounded, independent of the polynomial degree $p$. Furthermore, the constants corresponding to the numerically integrated norms $\| \cdot \|_{0,NI}$ are smaller than those corresponding to the fully integrated $L^2$ norm. This indicates that it is beneficial to use collocated quadrature when assembling the low-order system. The use of collocated quadrature for the high-order system will also lead to a better conditioned systems, however, this will result in a modified discretization that may not be desired. These one-dimensional constants can be used to estimate the constants of the 2D and 3D equivalences, including for the gradient, curl, and divergence operators, using the results of Propositions 5, 6 and 7.

6.2. Single element condition numbers. In this section, we compute condition numbers of the preconditioned mass and stiffness matrices on the reference elements in 2D and 3D, $\Omega = [-1, 1]^d$. The linear system is given by the sum of the mass and stiffness matrices, $A_p = M_p + K_p$, where $M_p$ and $K_p$ are the mass and stiffness matrices corresponding to one of the finite element spaces $V_p, W_p, X_p, Z_p$. Let $A_h$ denote the corresponding low-order-refined system. Dirichlet boundary conditions are enforced at the domain boundary. The high-order system is integrated with $(p+1)^d$ Gauss–Lobatto quadrature points (collocated quadrature, or “numerical integration,” cf. [20]); this typically results in better conditioned systems, but the condition numbers still remain asymptotically bounded in the case of exact integration [12, 36]. The condition number of the matrix $A_h^{-1} A_p$ is reported in Figure 3. Note than in 2D, the $H(\text{curl})$ and $H(\text{div})$ spaces coincide, and so the reported condition numbers are identical. The low-order preconditioner for the $H^1$ system results in a condition number that is bounded by $\pi^2/4$ in all cases; the bound of $\pi^2/4$ was first established for the case of low-order preconditioning of spectral methods in [10].

It is also possible to bound the single-element condition numbers given the norm equivalence constants computed in Section 6.1. Let $\kappa_c$ denote the $L^2$ norm equivalence constant $Cc^{-1}$ associated with the interpolation operators, and similarly for $\kappa_H$. For example, in two dimensions, the high-order and low-order discrete gradient operators are connected through the relation

$$G_h^2 = \begin{pmatrix} H_h H_{p-1}^{-1} \otimes I_h I_p^{-1} & 0 \\ I_h I_p^{-1} \otimes H_h H_{p-1}^{-1} & 0 \end{pmatrix} G_p^2.$$
LOW-ORDER PRECONDITIONING FOR HIGH-ORDER DE RHAM

Figure 3. Condition numbers of the matrices $A_h^{-1} A_p$, where $A_p = M_p + K_p$ on the reference element $[-1, 1]^d$ in 2D and 3D.

Table 1. Comparison of estimated and computed condition numbers for the single element case.

| p | $M_{V_p} + K_{V_p}$ $\kappa_T \kappa_H$ | $M_{W_p} + K_{W_p}$ $\kappa_H^2$ |
|---|---|---|
| 2 | 1.41 | 2.67 | 1.76 | 1.78 |
| 4 | 1.63 | 3.18 | 2.84 | 2.86 |
| 6 | 1.82 | 3.49 | 3.51 | 3.52 |
| 8 | 1.95 | 3.68 | 3.95 | 3.96 |
| 10 | 2.04 | 3.82 | 4.27 | 4.28 |

| p | $M_{V_p} + K_{V_p}$ $\kappa_H^2$ | $M_{W_p} + K_{W_p}$ $\kappa_H^2$ | $M_{X_p} + K_{X_p}$ $\kappa_H^3$ |
|---|---|---|---|
| 2 | 2.37 | 5.33 | 2.37 | 2.37 |
| 4 | 1.99 | 5.98 | 3.10 | 5.37 | 4.82 | 4.83 |
| 6 | 2.03 | 6.48 | 3.66 | 6.54 | 6.60 | 6.61 |
| 8 | 2.08 | 6.81 | 4.05 | 7.33 | 7.88 | 7.89 |
| 10 | 2.13 | 7.05 | 4.34 | 7.89 | 8.84 | 8.84 |

Therefore, the resulting condition number can be bounded by $\kappa_T \kappa_H$. Estimates for the remaining spaces and operators can be derived similarly (with the exception of the DG interior penalty stiffness matrix, whose analysis, described in Section 4.5, requires a different framework). A comparison of the estimated and computed condition numbers is shown in Table 1. In all of the cases, the estimates computed using products of the one-dimensional constants give an upper bound for the computed condition numbers. In 2D and 3D, the estimates of the form $\kappa_H^2$ are quite sharp; the estimates for the $H^1$ and $H(\text{curl})$ cases are more pessimistic.

6.3. Mass matrix preconditioning. In this section, we numerically compare several options for preconditioning the high-order mass matrix. In particular, given their simplicity, efficiency, and effectiveness for mass matrix problems, we focus on diagonal preconditioning approaches. In light of Section 5.1, all of the diagonal preconditioners considered are spectrally equivalent to the high-order mass matrix, independent of $p$, and so the results in this section represent a numerical and empirical comparison of the constants of equivalence.

For the $H^1$ mass matrix $M_{V_p}$, we compare Jacobi preconditioning using the nodal Gauss–Lobatto basis, which we denote “Jacobi (Lobatto)” to “LOR Jacobi”, which indicates using the diagonal of the low-order mass matrix $M_V$ as a preconditioner. For the spaces $W_p$, $X_p$, and $Y_p$, we also consider Gauss–Legendre and histopolation bases for the components for which continuity is not enforced. The corresponding diagonal preconditioners are denoted “Jacobi (Legendre)” and “Jacobi (Integrated),” respectively. Note that the diagonal of the high-order mass matrix can be constructed efficiently in $O(p^d)$ operations, without assembling
We solve the definite Maxwell problem

\[ \nabla \times \nabla \times \mathbf{u} + \beta \mathbf{u} = \mathbf{f}, \]

where \( \beta \) represents the conductivity coefficient. This coefficient is given by a piecewise constant, with \( \beta_{\text{air}} = 10^{-6} \) and \( \beta_{\text{copper}} = 1 \). This problem is solved on a mesh with 21,060 curved \( Q_3 \) elements. A schematic of this problem and the computational mesh are shown in Figure 5. The right hand side is chosen to be \( \mathbf{f} = 1 \).

We compare the number of iterations and computational time required to solve this problem to a relative tolerance of \( 10^{-12} \) using the auxiliary space Maxwell algebraic multigrid solver, applied directly to the assembled high-order system (denoted “Matrix-Based AMS”), and applied to the low-order refined system (“LOR–AMS”). This problem is solved using 144 MPI ranks of LLNL’s Quartz supercomputer. The results are shown in Table 2. The iterations required for the LOR–AMS solver require at most 1.5× as many iterations as the matrix-based AMS solver. However, the assembly time (which for the LOR–AMS solver denotes the time required to assemble the LOR matrix, as well as the time required for the “partial assembly” of the high-order operator) is significantly reduced for the LOR–AMS solver. Additionally, the number of nonzero entries of the system matrix (and hence the memory requirements for the solver) are significantly reduced for the LOR–AMS solver. Speedup and memory reduction factors are reported in Table 3. For polynomial degree \( p = 6 \), the total runtime is reduced by a factor of 25, and the memory usage is reduced by a factor of 35.

### Table 2. Convergence results and runtimes for the copper wire Maxwell problem.

| \( p \) | It. | Assembly (s) | AMG Setup (s) | Solve (s) | # DOFs | # NNZ |
|-------|-----|-------------|---------------|-----------|--------|-------|
| 2     | 41  | 0.082       | 0.277         | 0.768     | 516,820| \( 1.65 \times 10^7 \) |
| 3     | 63  | 0.251       | 0.512         | 2.754     | 1,731,408| \( 5.64 \times 10^7 \) |
| 4     | 75  | 0.679       | 1.133         | 7.304     | 4,088,888| \( 1.34 \times 10^8 \) |
| 5     | 62  | 1.574       | 2.185         | 11.783    | 7,968,340| \( 2.61 \times 10^8 \) |
| 6     | 89  | 3.336       | 4.024         | 30.702    | 13,748,844| \( 4.51 \times 10^8 \) |

### Matrix-Based AMS

| \( p \) | It. | Assembly (s) | AMG Setup (s) | Solve (s) | # DOFs | # NNZ |
|-------|-----|-------------|---------------|-----------|--------|-------|
| 2     | 39  | 0.140       | 0.385         | 1.423     | 516,820| \( 5.24 \times 10^7 \) |
| 3     | 44  | 1.368       | 1.572         | 9.723     | 1,731,408| \( 4.01 \times 10^8 \) |
| 4     | 49  | 9.668       | 5.824         | 45.277    | 4,088,888| \( 1.80 \times 10^9 \) |
| 5     | 53  | 61.726      | 15.695        | 148.757   | 7,968,340| \( 5.92 \times 10^9 \) |
| 6     | 56  | 502.607     | 40.128        | 424.100   | 13,748,844| \( 1.59 \times 10^{10} \) |
6.5. Grad-div problem: crooked pipe. In this section we consider the “crooked pipe” grad-div problem, which is a benchmark problem related to radiation diffusion simulations [37, 39]. The problem is posed on a cylindrical sector, consisting of two material subdomains. The mesh elements near the interface between the subdomains are refined anisotropically, leading to highly stretched elements. The mesh used for this problem is shown in Figure 6. We solve the problem

$$\nabla (\alpha \nabla \cdot u) - \beta u = f,$$
Figure 5. Mesh of the copper wire Maxwell problem. The conductivity coefficient is a piecewise constant coefficient determined by the material of the elements.

Table 3. Speedup and memory reduction for the copper wire Maxwell problem. Runtime includes assembly, solver setup, and solve times. Memory indicates the size of the assembled system matrix in CSR format.

| p | LOR–AMS Runtime (s) Memory (GB) | Matrix-Based AMS Runtime (s) Memory (GB) | Speedup | Memory Reduction |
|---|-------------------------------|-----------------------------------------|---------|------------------|
| 2 | 1.13                          | 1.95                                    | 1.73×   | 3.16×            |
| 3 | 3.52                          | 12.66                                   | 3.60×   | 7.05×            |
| 4 | 9.11                          | 60.77                                   | 6.67×   | 13.31×           |
| 5 | 15.54                         | 226.18                                  | 14.55×  | 22.45×           |
| 6 | 38.06                         | 966.83                                  | 25.40×  | 35.00×           |

Figure 6. Crooked pipe grad-div problem. The domain is split into two materials. The mesh is refined in the vicinity of the material interface, resulting in highly stretched anisotropic elements.

Where the coefficients $\alpha$ and $\beta$ are given piecewise constant values according to the materials. In the larger subregion (colored green in Figure 6), we take $\alpha = 1.88 \times 10^{-3}$ and $\beta = 2000$. In the smaller subregion (colored blue in Figure 6), we take $\alpha = 1.641$ and $\beta = 0.2$. 
**Table 4.** Convergence results for the crooked-pipe grad-div problem.

| p  | Its. | Assembly (s) | AMG Setup (s) | Solve (s) | # DOFs  | # NNZ   |
|----|------|--------------|---------------|-----------|---------|---------|
| 2  | 115  | 0.027        | 0.299         | 1.782     | 356,500 | $3.81 \times 10^6$ |
| 3  | 168  | 0.075        | 0.583         | 5.986     | 1,190,115 | $1.28 \times 10^7$ |
| 4  | 197  | 0.218        | 1.245         | 15.932    | 2,805,520 | $3.04 \times 10^7$ |
| 5  | 243  | 0.456        | 2.428         | 40.814    | 5,461,375 | $5.93 \times 10^7$ |
| 6  | 276  | 0.907        | 4.969         | 86.882    | 9,416,340 | $1.03 \times 10^8$ |

**Matrix-Based ADS**

| p  | Its. | Assembly (s) | AMG Setup (s) | Solve (s) | # DOFs  | # NNZ   |
|----|------|--------------|---------------|-----------|---------|---------|
| 2  | 86   | 0.032        | 0.459         | 2.486     | 356,500 | $1.79 \times 10^7$ |
| 3  | 99   | 0.417        | 2.815         | 19.351    | 1,190,115 | $1.64 \times 10^8$ |
| 4  | 104  | 3.166        | 12.338        | 98.844    | 2,805,520 | $8.16 \times 10^8$ |
| 5  | 112  | 18.062       | 46.590        | 358.085   | 5,461,375 | $2.88 \times 10^9$ |
| 6  | 121  | 181.517      | 165.135       | 978.285   | 9,416,340 | $8.15 \times 10^9$ |

**Table 5.** Speedup and memory reduction for the crooked-pipe grad-div problem. Runtime includes assembly, solver setup, and solve times. Memory indicates the size of the assembled system matrix in CSR format.

| p  | LOR–ADS Runtime (s) Memory (GB) | Matrix-Based ADS Runtime (s) Memory (GB) | Speedup | Memory Reduction |
|----|---------------------------------|----------------------------------------|---------|------------------|
| 2  | 2.11                            | 0.04                                   | 1.41×  | 4.60×            |
| 3  | 6.64                            | 0.15                                   | 3.40×  | 12.44×           |
| 4  | 17.40                           | 0.35                                   | 6.57×  | 26.07×           |
| 5  | 43.70                           | 0.68                                   | 9.67×  | 47.11×           |
| 6  | 92.76                           | 1.18                                   | 14.28× | 77.13×           |

Analogously to the previous section, we compare the number of iterations and computational time required to solve this problem to a relative tolerance of $10^{-12}$ using the auxiliary space divergence algebraic multigrid solver, applied directly to the assembled high-order system (denoted “Matrix-Based ADS”), and applied to the low-order refined system (“LOR–ADS”). As in the previous section, this problem is solved in parallel using 144 MPI ranks. These results are shown in Table 4. For $p = 2$, the LOR–ADS solver requires about $1.3 \times$ as many iterations as the matrix-based ADS solver. This factor increases to about $2.3 \times$ for $p = 6$. This presymptotic increase is largely consistent with the conditioning results shown in Figures 3 and 4. Despite this increase in the number of iterations, the solve time alone is reduced by more than a factor of $10 \times$ by using the LOR–ADS preconditioner for $p = 6$. Total speedup and memory reduction results are shown in Table 5. For this test case, the reduction in memory usage is even more dramatic than for the $\mathbf{H}(\text{curl})$ problem; at $p = 6$, the memory required to store the system matrix is reduced by a factor of $77 \times$ using the LOR method.

### 6.6. Discontinuous Galerkin methods

In this section, we consider the low-order preconditioning for discontinuous Galerkin methods proposed in Section 4.5. As a test case, we use the solver benchmark problem proposed in [46], and solve the constant-coefficient Poisson problem

$$-\Delta u = f \quad \text{in } \Omega,$$

$$u = 0 \quad \text{on } \partial\Omega,$$

with homogeneous Dirichlet boundary conditions in the unit cube $\Omega = [0, 1]^3$. The right-hand side $f$ is determined by the prescribed exact solution $u$, which is given as the tensor-product of one-dimensional functions (parameterized by the so-called structure level $n \in \mathbb{N}$), $u(x, y, z) = w_n(x)w_n(y)w_n(z)$. The functions $w_n(x)$
Figure 7. Illustrations of the Kershaw mesh with $12^3$ elements and $\varepsilon = 1$ (left) and $\varepsilon = 0.3$ (right).

are given by

$$w_n(x) = \sum_{j=0}^{n-1} u_{3j}(x),$$

$$u_k(x) = \exp(-1/s_k^2(x)) \text{sign}(s_k(x)),$$

$$s_k(x) = \sin(2j\pi x).$$

We consider a family of Kershaw meshes (cf. [44]), parameterized by an anisotropy parameter $0 < \varepsilon \leq 1$. These meshes are obtained by distorting a Cartesian grid, such that layers of elements with aspect ratio $1/\varepsilon$ are placed in opposing corners of the cube. The mesh transitions through four intermediate layers in a “Z” pattern, giving rise to skewed and stretched elements. When $\varepsilon = 1$, the mesh is a uniform Cartesian grid. The geometric anisotropy induced by smaller values of $\varepsilon$ often proves challenging for linear solvers and preconditioners. Figure 7 illustrates examples of the Kershaw mesh for $\varepsilon = 1$ and $\varepsilon = 0.3$ on a mesh with $12^3$ elements.

We solve this problem using both $\varepsilon = 1$ (isotropic elements) and $\varepsilon = 0.3$ (anisotropic elements). We begin with polynomial degree $p = 1$ on a mesh with $36^3$ elements, corresponding to 373,248 degrees of freedom. For increasing polynomial degrees $p = 2, 3, 5, 7, 9$, the mesh is simultaneously coarsened to keep the total number of degrees of freedom fixed. In each case, we assemble the corresponding low-order-refined system $K_{Z_h}$ using the piecewise constant DG discretization defined in Section 4.5. As a preconditioner, we use hypre’s BoomerAMG algebraic multigrid with $\ell_1$–Jacobi smoothing [9]. In Table 6 we present the number of conjugate gradient iterations required to reduce the residual by a factor of $10^{12}$. In addition to the iteration counts required to solve the high-order problem $K_{Z_p}$, we also present the number of AMG-preconditioned CG iterations required to solve the low-order-refined problem $K_{Z_h}$. Furthermore, we compare these iteration counts to the number of iterations required to solve the high-order problem using “CG–DG preconditioning,” which is based on the idea of using the low-order-refined conforming (continuous Galerkin) problem as a preconditioner, together with a diagonal correction, cf. [3, 31, 66].

For $\varepsilon = 1$, the number of iterations required to solve the high-order problem using the low-order-refined problem (the column labeled $B_hK_{Z_p}$ in Table 6) as a preconditioner remains bounded, independent of $p$. For $\varepsilon = 0.3$, the problem is more challenging because of the mesh-induced anisotropy, however, after a mild presymptotic increase, the iteration counts appear to be uniform with respect to $p$. These results corroborate the spectral equivalence demonstrated in Theorem 3. In both of these cases, the iterations required to solve this problem using the low-order-refined preconditioner described in the present work are significantly less than those required to solve this problem using the CG–DG subspace preconditioning. We note that for both $\varepsilon = 1$ and $\varepsilon = 0.3$, BoomerAMG applied to the low-order-refined problem (the column labeled $B_hK_{Z_h}$), the number of iterations required to converge remains bounded, independent of $p$, corroborating the conclusions of Theorem 4.

We additionally study the dependence of the convergence properties of the low-order-refined AMG preconditioners on the magnitude of the DG interior penalty parameter. The condition number of the system $K_{Z_h}$ scales linearly with the penalty parameter $\eta$ (cf. Proposition 5), and geometric and algebraic multigrid preconditioners often give degraded convergence for large values of the penalty parameter. As an example,
Table 6. Convergence results for discontinuous Galerkin discretizations of the Poisson problem on Kershaw meshes with $\varepsilon = 1$ and $\varepsilon = 0.3$. Each column indicates the number of CG iterations required to converge to a relative tolerance of $10^{-12}$, where $K_{Z_p}$ and $K_{Z_h}$ denote the high-order and low-order-refined operators, respectively, $B_p$ denotes BoomerAMG formed using the high-order matrix, $B_h$ denotes BoomerAMG formed using the low-order-refined matrix, and CG–DG indicates the use of the conforming problem as a subspace correction preconditioner.

| $p$ | $B_hK_{Z_p}$ | $\varepsilon = 1$ | $B_hK_{Z_h}$ | CG–DG | $B_hK_{Z_p}$ | $B_hK_{Z_h}$ | CG–DG |
|-----|--------------|------------------|--------------|--------|--------------|--------------|--------|
| 1   | 51           | 20               | 87           |        | 88           | 25           | 385    |
| 2   | 48           | 22               | 68           |        | 83           | 25           | 311    |
| 3   | 49           | 23               | 61           |        | 97           | 27           | 300    |
| 5   | 47           | 23               | 60           |        | 115          | 29           | 319    |
| 7   | 48           | 23               | 66           |        | 121          | 32           | 308    |
| 9   | 49           | 23               | 71           |        | 111          | 29           | 285    |

Figure 8. Dependence of iteration counts on DG penalty parameter. No convergence in fewer than 2,000 iterations is indicated by a red “x” symbol.

we take the case of $p = 1$ with $36^3$ elements, and compute the number of CG iterations required to converge to a relative tolerance of $10^{-12}$ for increasing values of the penalty parameter. We compare BoomerAMG applied directly to $K_{Z_p}$ (this preconditioner is denoted $B_p$), BoomerAMG applied to $K_{Z_h}$ (this preconditioner is denoted $B_h$), and CG–DG preconditioning, and present the results in Figure 8. We note that the iteration counts for BoomerAMG applied directly to $K_{Z_p}$ increase substantially as $\eta$ increases; for the case of $\varepsilon = 0.3$, the convergence criterion was not met in under 2000 iterations for $\eta \geq 10^3$. The iteration counts for BoomerAMG formed using the low-order-refined system $K_{Z_h}$ and applied to both $K_{Z_p}$ and $K_{Z_h}$ remain bounded, independent of $\eta$, corroborating the results of Theorems 3 and 4. Similarly, the iteration counts for the CG–DG preconditioner are bounded independent of $\eta$, cf. [3, 66].

7. Conclusions

In this work, we have presented a framework for the construction of spectrally equivalent low-order-refined discretizations using interpolation and histopolation operators with Gauss–Lobatto points. Simple one-dimensional norm equivalence properties of these operators can be combined using tensor-product arguments to give natural norm and seminorm equivalences in all spaces of the de Rham complex. As an immediate consequence, we obtain spectral equivalence for the mass and stiffness matrices in $H^1$, $H(\text{curl})$, and $H(\text{div})$, using Lagrange, Nédélec, and Raviart–Thomas elements, independent of polynomial degree $p$ and mesh size $h$. We additionally present a novel piecewise constant discontinuous Galerkin discretization that is spectrally equivalent to the high-order interior penalty DG discretization, independent of $p$, $h$, and penalty parameter. This low-order discretization is equivalent to a certain weighted graph Laplacian, for which we demonstrate efficient algebraic multigrid convergence. We use the efficient and highly scalable algebraic multigrid methods from hypre, built using the low-order discretizations, to obtain matrix-free solvers for the high-order finite
element problems; for $H({\text{curl}})$ and $H({\text{div}})$ problems, we use the AMS and ADS algebraic solvers. The effectiveness of these preconditioners on a number of three-dimensional problems is studied. These problems possess challenging features such as coefficients with large contrasts and highly distorted geometries. The theoretical properties of the spectrally equivalent low-order discretizations are verified. Additionally, we demonstrate significant speedups and memory savings using the proposed solvers, in particular at higher orders. For discontinuous Galerkin discretization, the new method proposed compares favorably to techniques that make use of the conforming subspace as a preconditioner. Although not a focus of the present paper, these methods are highly amenable to GPU acceleration, which we anticipate to be the topic of future work.

8. Acknowledgments

The authors thank V. Dobrev for insightful comments and suggestions. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 and was supported by the LLNL-LDRD Program under Project No. 20-ERD-002 (LLNL-JRNL-831792). Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525.

References

[1] R. Anderson et al. “MFEM: a modular finite element methods library”. In: Computers & Mathematics with Applications (2020). DOI: 10.1016/j.camwa.2020.06.009
[2] P. F. Antonietti and P. Houston. “A class of domain decomposition preconditioners for $hp$-discontinuous Galerkin finite element methods”. In: Journal of Scientific Computing 46.1 (2010), pp. 124–149. DOI: 10.1007/s10915-010-9390-1
[3] P. F. Antonietti et al. “A uniform additive Schwarz preconditioner for high-order discontinuous Galerkin approximations of elliptic problems”. In: Journal of Scientific Computing 70.2 (2016), pp. 608–630. DOI: 10.1007/s10915-016-0259-9
[4] D. N. Arnold. “An interior penalty finite element method with discontinuous elements”. In: SIAM Journal on Numerical Analysis 19.4 (1982), pp. 742–760. DOI: 10.1137/0719052
[5] D. N. Arnold, R. S. Falk, and R. Winther. “Differential complexes and stability of finite element methods I. The de Rham complex”. In: Compatible Spatial Discretizations. Ed. by D. N. Arnold et al. Vol. 142. The IMA Volumes in Mathematics and its Applications. Springer New York, 2006, pp. 23–46. DOI: 10.1007/0-387-38034-5_2
[6] D. N. Arnold, R. S. Falk, and R. Winther. “Finite element exterior calculus, homological techniques, and applications”. In: Acta Numerica 15 (2006), pp. 1–155. DOI: 10.1017/s0962492906210018
[7] D. N. Arnold et al., eds. Compatible Spatial Discretizations. Vol. 142. The IMA Volumes in Mathematics and its Applications. Springer New York, 2006. DOI: 10.1007/0-387-38034-5.
[8] D. N. Arnold et al. “Unified analysis of discontinuous Galerkin methods for elliptic problems”. In: SIAM Journal on Numerical Analysis 39.5 (2002), pp. 1749–1779. DOI: 10.1137/S0036142901384162
[9] A. H. Baker et al. “Multigrid smoothers for ultrasparse computing”. In: SIAM Journal on Scientific Computing 33.5 (2011), pp. 2864–2887. DOI: 10.1137/100798806.
[10] P. Bastian et al. “Matrix-free multigrid block-preconditioners for higher order discontinuous Galerkin discretisations”. In: Journal of Computational Physics 394 (2019), pp. 417–439. DOI: 10.1016/j.jcp.2019.06.001
[11] B. Beckermann et al. “Gaussian, Lobatto and Radau positive quadrature rules with a prescribed abscissa”. In: Calcolo 51.2 (2013), pp. 319–328. DOI: 10.1007/s10092-013-0087-3
[12] P. D. Bello-Maldonado and P. F. Fischer. “Scalable low-order finite element preconditioners for high-order spectral element Poisson solvers”. In: SIAM Journal on Scientific Computing 41.5 (2019), S2–S18. DOI: 10.1137/18M1194997
[13] J.-P. Berrut and L. N. Trefethen. “Barycentric Lagrange interpolation”. In: SIAM Review 46.3 (2004), pp. 501–517. DOI: 10.1137/s0036144502417715
[14] A. Brandt. “Algebraic multigrid theory: The symmetric case”. In: Applied Mathematics and Computation 19.1 (1986), pp. 23–56. DOI: https://doi.org/10.1016/0098-3003(86)90095-0
[36] P. F. Fischer. “An overlapping Schwarz method for spectral element solution of the incompressible Navier–Stokes equations”. In: Journal of Computational Physics 133.1 (1997), pp. 84–101. doi: 10.1006/jcph.1997.5651

[37] N. Gentile. “Implicit Monte Carlo diffusion—an acceleration method for Monte Carlo time-dependent radiative transfer simulations”. In: Journal of Computational Physics 172.2 (2001), pp. 543–571. doi: 10.1006/jcph.2001.6836

[38] M. Gerritsma. “Edge functions for spectral element methods”. In: Lecture Notes in Computational Science and Engineering. Springer Berlin Heidelberg, 2010, pp. 199–207. doi: 10.1007/978-3-642-15337-2_17

[39] F. Graziani and J. LeBlanc. The crooked pipe test problem. Tech. rep. UCRL-MI-143393. Lawrence Livermore National Laboratory, 2000.

[40] P. Haldenwang et al. “Chebyshev 3-D spectral and 2-D pseudospectral solvers for the Helmholtz equation”. In: Journal of Computational Physics 55.1 (1984), pp. 115–128. doi: 10.1016/0021-9991(84)90018-4

[41] T. S. Haut et al. “Diffusion synthetic acceleration preconditioning for discontinuous Galerkin discretizations of \( S_N \) transport on high-order curved meshes”. In: SIAM Journal on Scientific Computing 42.5 (2020), B1271–B1301. doi: 10.1137/19m124993x

[42] M. Holec et al. “Arbitrary order energy and enstrophy conserving finite element methods for 2D incompressible fluid dynamics and drift-reduced magnetohydrodynamics”. Submitted for publication. 2022.

[43] P. Houston, C. Schwab, and E. Süli. “Discontinuous hp-finite element methods for advection-diffusion-reaction problems”. In: SIAM Journal on Numerical Analysis 39.6 (2002), pp. 2133–2163. doi: 10.1137/s0036142900374111

[44] T. V. Kolev and P. S. Vassilevski. “Parallel auxiliary space AMG solver for \( H(\text{curl}) \) problems”. In: SIAM Journal on Scientific Computing 34.6 (2012), A3079–A3098. doi: 10.1137/110859361

[45] T. V. Kolev and P. S. Vassilevski. “Parallel auxiliary space AMG for \( H(\text{div}) \) problems”. In: SIAM Journal on Scientific Computing 34.6 (2012), A3079–A3098. doi: 10.1137/110859361

[46] T. V. Kolev et al. “Efficient exascale discretizations: high-order finite element methods”. In: The International Journal of High Performance Computing Applications (2021), p. 109434202110208. doi: 10.1177/10943420211020803

[47] B. Krank et al. “A high-order semi-explicit discontinuous Galerkin solver for 3D incompressible flow with application to DNS and LES of turbulent channel flow”. In: Journal of Computational Physics 348 (2017), pp. 634–659. doi: 10.1016/j.jcp.2017.07.039

[48] J. Kreeft, A. Palha, and M. Gerritsma. “Mimetic framework on curvilinear quadrilaterals of arbitrary order”. arXiv:1111.4304. 2011.

[49] M. Kronbichler and K. Ljungkvist. “Multigrid for matrix-free high-order finite element computations on graphics processors”. In: ACM Transactions on Parallel Computing 6.1 (2019), pp. 1–32. doi: 10.1145/3322813

[50] M. Kronbichler and W. Wall. “A performance comparison of continuous and discontinuous Galerkin methods with fast multigrid solvers”. In: SIAM Journal on Scientific Computing 40.5 (2018), A3423–A3448. doi: 10.1137/16m110455x

[51] K. Ljungkvist. “Matrix-free finite-element computations on graphics processors with adaptively refined unstructured meshes”. In: Proceedings of the 25th High Performance Computing Symposium. HPC ’17. San Diego, CA, USA: Society for Computer Simulation International, 2017.

[52] J. W. Lottes and P. F. Fischer. “Hybrid multigrid/Schwarz algorithms for the spectral element method”. In: Journal of Scientific Computing 24.1 (2005), pp. 45–78. doi: 10.1007/s10915-004-4787-3
[56] J.-F. Maitre and O. Pourquier. “Condition number and diagonal preconditioning: comparison of the \( p \)-version and the spectral element methods”. In: Numerische Mathematik 74.1 (1996), pp. 69–84. DOI: 10.1007/s002110050208

[57] S. F. McCormick. “Multigrid methods for variational problems: general theory for the V-cycle”. In: SIAM Journal on Numerical Analysis 22.4 (1985), pp. 634–643. DOI: 10.1137/0722039

[58] J. Melenk. “On condition numbers in \( hp \)-FEM with Gauss–Lobatto-based shape functions”. In: Journal of Computational and Applied Mathematics 139.1 (2002), pp. 21–48. DOI: 10.1016/s0377-0427(01)00391-0

[59] J. Melenk, K. Gerdes, and C. Schwab. “Fully discrete \( hp \)-finite elements: fast quadrature”. In: Journal of Computational and Applied Mathematics 139.1 (2002), pp. 21–48. DOI: 10.1016/s0377-0427(01)00391-0

[60] P. Monk. Finite Element Methods for Maxwell’s Equations. Oxford University Press, 2003. DOI: 10.1093/acprof:oso/9780198508885.001.0001

[61] L. Olson. “Algebraic multigrid preconditioning of high-order spectral elements for elliptic problems on a simplicial mesh”. In: SIAM Journal on Scientific Computing 29.5 (2007), pp. 2189–2209. DOI: 10.1137/060663465

[62] S. S. Pahl. “Schwarz type domain decomposition methods for spectral element discretizations”. MA thesis. University of the Witwatersrand, 1993.

[63] S. V. Parter and E. E. Rothman. “Preconditioning Legendre spectral collocation approximations to elliptic problems”. In: SIAM Journal on Numerical Analysis 32.2 (1995), pp. 333–385. DOI: 10.1137/0732015

[64] W. Pazner. “Efficient low-order refined preconditioners for high-order matrix-free continuous and discontinuous Galerkin methods”. In: SIAM Journal on Scientific Computing 42.5 (2020), A3055–A3083. DOI: 10.10107/19m1282052

[65] W. Pazner and T. Kolev. “Uniform subspace correction preconditioners for discontinuous Galerkin methods with \( hp \)-refinement”. In: Communications on Applied Mathematics and Computation (2021). DOI: 10.1007/s42967-021-00136-3

[66] R. Rieben et al. “An arbitrary Lagrangian–Eulerian discretization of MHD on 3D unstructured grids”. In: Journal of Computational Physics 226.1 (2007), pp. 534–570. DOI: 10.1016/j.jcp.2007.04.051

[67] G. Szegő. Orthogonal Polynomials. Vol. 23. Colloquium Publications. American Mathematical Society, 1939. DOI: 10.1090/coll1023

[68] A. Teukolsky. “Short note on the mass matrix for Gauss–Lobatto grid points”. In: Journal of Computational Physics 283 (2015), pp. 408–413. DOI: 10.1016/j.jcp.2014.12.012

[69] D. A. White, J. M. Koning, and R. N. Rieben. “Development and application of compatible discretizations of Maxwell’s equations”. In: Compatible Spatial Discretizations. Ed. by D. N. Arnold et al. Vol. 142. The IMA Volumes in Mathematics and its Applications. Springer New York, 2006, pp. 209–234. DOI: 10.1007/978-3-8038-3034-5_11

[70] Y. Zhang et al. “A high order hybrid mimetic discretization on curvilinear quadrilateral meshes for complex geometries”. In: Proceedings of the 6th European Conference on Computational Mechanics (Solids, Structures and Coupled Problems) ECCM 6 and 7th European Conference on Computational Fluid Dynamics ECFD 7. Ed. by R. Owen et al. 2018, pp. 426–437.