

EFFICIENT LOW-ORDER REFINED PRECONDITIONERS FOR HIGH-ORDER MATRIX-FREE CONTINUOUS AND DISCONTINUOUS GALERKIN METHODS*

WILL PAZNER[†]

Abstract. In this paper, we design preconditioners for the matrix-free solution of high-order continuous and discontinuous Galerkin discretizations of elliptic problems based on finite element method–spectral element method (FEM-SEM) equivalence and additive Schwarz methods. The high-order operators are applied without forming the system matrix, making use of sum factorization for efficient evaluation. The system is preconditioned using a spectrally equivalent low-order ($p = 1$) finite element operator discretization on a refined mesh. The low-order refined mesh is anisotropic and not shape regular in the polynomial degree of the high-order operator, requiring specialized solvers to treat the anisotropy. We make use of an element-structured, geometric multigrid V-cycle with ordered ILU(0) smoothing. The preconditioner is parallelized through an overlapping additive Schwarz method that is robust in h and p . The method is extended to interior penalty and Bassi and Rebay (BR2) discontinuous Galerkin discretizations, for which it is also robust in the size of the penalty parameter. Numerical results are presented on a variety of examples, verifying the uniformity of the preconditioner.

Key words. preconditioning, matrix-free, discontinuous Galerkin

AMS subject classifications. 65F08, 65N30, 65Y10

DOI. 10.1137/19M1282052

1. Introduction. High-order numerical methods have played an increasingly significant role in many areas of scientific computation in recent years [61, 58, 16]. These methods promise higher accuracy with fewer degrees of freedom, at the cost of more arithmetic operations performed per degree of freedom. Because of their high arithmetic intensity, high-order methods have been seen as promising candidates to run on GPUs and accelerator-based architectures [41, 60]. Due to the restrictive memory limitations on these architectures, much of the past research in this area has focused on problems with explicit time integration, thus avoiding the solution of large linear systems of equations. In this work, we study the iterative solution to the linear systems arising from high-order finite element discretizations in the matrix-free context, avoiding the restrictive memory cost of assembling the system matrix.

A standard finite element or discontinuous Galerkin method with polynomial degree p will result in coupling between all the degrees of freedom within a single element (and with a subset of the degrees of freedom of neighboring elements). The number of degrees of freedom per element scales like $\mathcal{O}(p^d)$ in d spatial dimensions, and as a result, the number of couplings (i.e., the number of nonzeros per row of the system matrix) will also scale like $\mathcal{O}(p^d)$. Consequently, the system matrix will have $\mathcal{O}(p^{2d})$

*Submitted to the journal's Methods and Algorithms for Scientific Computing section August 20, 2019; accepted for publication (in revised form) June 24, 2020; published electronically October 5, 2020. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

<https://doi.org/10.1137/19M1282052>

Funding: This work was supported by Lawrence Livermore National Laboratory. LLNL is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under contract DE-AC52-07NA27344 (LLNL-JRNL-787240).

[†]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94550 (pazner1@llnl.gov).

nonzero entries, and thus the memory required to form the matrix system matrix grows quadratically with the number of degrees of freedom under p -refinement. This is in contrast to h -refinement, in which the memory required for the system matrix grows only linearly with the number of degrees of freedom. This motivates the use of matrix-free operator evaluation, whereby the action of the operator applied to a vector is computed without forming the system matrix.

In the matrix-free context, the memory requirements for the high-order operator can be reduced to $\mathcal{O}(p^d)$. Naive implementations of the action of the operator will still require $\mathcal{O}(p^{2d})$ operations. However, making use of sum factorization techniques, this complexity can be reduced to $\mathcal{O}(dp^{d+1})$ [50, 52]. These techniques have been shown to be effective on modern many-core and accelerator-based architectures [15, 47]. In addition to efficient algorithms for the evaluation of the action of the operator, solving the resulting linear systems in a practical setting also requires effective preconditioners. The development of preconditioners in the matrix-free setting is particularly challenging, because traditional matrix-based methods such as Gauss–Seidel or algebraic multigrid require access to the entries of the matrix [55]. Furthermore, element-based domain decomposition methods such as additive and multiplicative Schwarz methods necessitate the solution of local problems, which, if performed by means of a direct solver, requires $\mathcal{O}(p^{3d})$ operations [59].

There has been significant past work on the development of preconditioners suitable for use in the high-order matrix-free context. Much of this work has made use of related low-order, sparse discretizations in order to construct preconditioners for the high-order system, an idea originally introduced by Orszag in 1980, and since built upon by numerous others [50, 28, 22, 19]. New low-order methods for preconditioning high-order spectral element discretizations were developed in [11]. The automatic construction of sparse preconditioners for high-order finite element discretizations was considered in [9]. Matrix-free multigrid methods using polynomial smoothers (cf. [1]) were considered in [42]. Matrix-free approximate block Jacobi methods using Kronecker product approximations were constructed for discontinuous Galerkin discretizations of conservation laws in [52] and extended to interior penalty discretizations in [53]. Overlapping Schwarz solvers for the spectral element discretization of the Navier–Stokes equations were proposed in [32]. Hybrid multigrid solvers with Schwarz smoothers for high-order spectral element discretizations were developed in [44], and extended to the Navier–Stokes equations in [33]. These Schwarz smoothers are based on solving structured subdomain problems using the fast diagonalization method.

In this work, we construct preconditioners for high-order finite element discretizations based on corresponding low-order ($p = 1$) finite element discretizations formed on a refined mesh. The refined mesh is generated using a structured grid of Gauss–Lobatto points within each element. These preconditioners make use of the spectral equivalence between the two discretizations, widely known as the finite element method–spectral element method (FEM–SEM) equivalence [23]. The resulting low-order discretization is sparse, and, because the polynomial degree is fixed, its system matrix can be formed using only constant memory per degree of freedom. In order to precondition the high-order system, it is necessary to solve the linear system corresponding to the low-order system. This is challenging because the low-order refined mesh is anisotropic and not shape-regular with respect to p . Thus, standard algebraic multigrid or geometric multigrid methods with pointwise smoothers do not converge uniformly with respect to p . To address this issue, we develop ordered ILU smoothers that perform the function of line relaxation in the context of unstructured meshes.

Distinguishing this solver from other Schwarz-based methods, the use of ordered ILU smoothing allows the solver to handle domains with highly anisotropic meshes. The method is parallelized using an additive Schwarz method based on overlapping sub-domains. The resulting preconditioner is robust in the polynomial degree p and mesh size h . The preconditioner can also be extended to discontinuous Galerkin methods, for which it is also robust in the size of the penalty parameter η .

The structure of this paper is as follows. In section 2 we introduce the model problem and finite element discretization. Then, in section 3 we design and analyze the matrix-free preconditioners considered in this paper. In section 4, we present a variety of numerical results using finite element and discontinuous Galerkin discretizations on structured and unstructured meshes. We end with conclusions in section 5.

2. Model problem and discretization. In this work, we will consider the solution to the model Poisson problem with homogeneous Dirichlet boundary conditions,

$$(2.1) \quad -\nabla \cdot (b\nabla u) = f \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega,$$

where the coefficient b and right-hand side f are sufficiently smooth, and b is bounded away from zero on the spatial domain $\Omega \subseteq \mathbb{R}^d$, $d \in \{2, 3\}$. To simplify the exposition, in what follows we will take $b \equiv 1$. However, numerical examples with variable coefficients are considered in section 4. The solution to this problem is approximated using a high-order finite element method. To begin, we consider standard H^1 conforming discretizations. Discontinuous Galerkin discretizations of this problem are presented in section 2.1. The spatial domain $\Omega \subseteq \mathbb{R}^d$ is tessellated using a mesh of tensor-product elements denoted \mathcal{T}_p . An element $D \in \mathcal{T}_p$ is given as the image of the unit cube $[0, 1]^d$ under a suitable transformation mapping. We introduce the usual continuous piecewise polynomial finite element space of degree at most p in each variable on the mesh \mathcal{T}_p , which we denote V_p .

The finite element problem corresponding to (2.1) is to find $u_p \in V_p$ such that

$$(2.2) \quad (\nabla u_p, \nabla v_p) = (f, v_p) \quad \text{for all } v_p \in V_p,$$

where (\cdot, \cdot) denotes the standard L^2 inner product on Ω , defined by

$$(2.3) \quad (u, v) = \int_{\Omega} u(\mathbf{x})v(\mathbf{x}) d\mathbf{x}.$$

Introducing a basis for the space V_p , we can write (2.2) as the algebraic system

$$(2.4) \quad K_p \mathbf{u} = M_p \mathbf{f},$$

where K_p is the stiffness matrix corresponding to the bilinear form $a(\cdot, \cdot) = (\nabla \cdot, \nabla \cdot)$, M_p is the mass matrix corresponding to the L^2 inner product (\cdot, \cdot) , and \mathbf{u} and \mathbf{f} are vectors of coefficients in the chosen basis. It is well known that the system of equations (2.4) quickly becomes ill-conditioned when the polynomial degree p is large or the mesh spacing h is small. For that reason, effective preconditioners are required for the efficient iterative solution of (2.4). Additionally, the number of nonzeros in the stiffness and mass matrices scales like $\mathcal{O}(p^{2d})$. Naive assembly of this matrix (using local dense matrix-matrix products) requires $\mathcal{O}(p^{3d})$ operations, and sum-factorized matrix assembly can be performed in $\mathcal{O}(p^{2d+1})$ operations [46]. This computational cost, in terms of both memory and arithmetic operations, is seen to be prohibitive

for high or even moderate p , especially in three dimensions. On the other hand, the action of the operators K_p and M_p may be computed using techniques such as sum factorization with $\mathcal{O}(p^{d+1})$ operations and $\mathcal{O}(p^d)$ memory [50]. The purpose of this paper is the development of *matrix-free* preconditioners for (2.4), which can be constructed without explicit knowledge of the entries of the matrices K_p and M_p , whose total memory cost scales like the number of degrees of freedom ($\mathcal{O}(p^d)$), and whose computational complexity is no more than that of applying the operator.

2.1. Discontinuous Galerkin methods. Also of interest are discontinuous Galerkin discretizations of (2.1). Let V_{DG} denote the degree- p discontinuous piecewise polynomial space defined on the mesh \mathcal{T}_p . No continuity is enforced between the elements of the mesh. There are a wide range of discontinuous Galerkin (DG) discretizations for elliptic problems [8]. We consider the symmetric interior penalty (IP) discretization [7], whose formulation is as follows: find $u_{DG} \in V_{DG}$ such that

$$(2.5) \quad \begin{aligned} a_{IP}(u_{DG}, v_{DG}) &= (\nabla u_{DG}, \nabla v_{DG}) - \langle \{\nabla u_{DG}\}, [\![v_{DG}]\!] \rangle - \langle [\![u_{DG}]\!], \{\nabla v_{DG}\} \rangle \\ &\quad + \langle \sigma [\![u_{DG}]\!], [\![v_{DG}]\!] \rangle = (f, v_{DG}) \end{aligned}$$

for all $v_{DG} \in V_{DG}$. Here, $\langle \cdot, \cdot \rangle$ denotes integration over the union of all faces of elements $D \in \mathcal{T}_p$, which we denote by Γ . Consider a common face shared by two mesh elements, D^- and D^+ . Then, $\{\cdot\}$ and $[\![\cdot]\!]$ denote the average and jump operators, respectively, defined by

$$(2.6) \quad \{\phi\} = \frac{1}{2}(\phi^- + \phi^+), \quad [\![\phi]\!] = \phi^- \mathbf{n}^- + \phi^+ \mathbf{n}^+,$$

where the superscripts \pm denote the traces from within the elements D^\pm . Similarly, \mathbf{n}^\pm denotes the outward facing normal vector from within D^\pm .

The parameter σ in (2.5) is a penalty parameter, which is required to be sufficiently large in order to stabilize the method. In particular, σ must be chosen to scale like p^2/h to ensure that the system is positive-definite [8], and so we write $\sigma = \eta p^2/h$. However, large values of the penalty parameter result in ill-conditioned systems [56]. In this paper, we will seek to design preconditioners for the DG system (2.5) whose convergence is independent of the value of the penalty parameter.

In addition to the symmetric interior penalty discretization (2.5), we are also interested in alternative DG discretizations, such as the second method of Bassi and Rebay (BR2) [10]. This method makes use of a different stabilization scheme based on lifting operators. In the BR2 method, the penalty term $\langle \sigma [\![u_{DG}]\!], [\![v_{DG}]\!] \rangle$ in (2.5) is replaced by an alternative penalty term $\langle \alpha(u_{DG}), [\![v_{DG}]\!] \rangle$. The term $\alpha(u_{DG})$ is defined on a face e by $\alpha(u_{DG}) = -\eta \{r_e([\![u_{DG}]\!])\}$, where the lifting operator r_e is given by

$$(2.7) \quad \int_{\Omega} r_e(\boldsymbol{\varphi}) \cdot \boldsymbol{\tau} dx = - \int_e \boldsymbol{\varphi} \cdot \{\boldsymbol{\tau}\} ds.$$

We can see that the stabilization term satisfies

$$\langle \alpha(u_{DG}), [\![v_{DG}]\!] \rangle = \sum_e (\eta r_e([\![u_{DG}]\!]), r_e([\![v_{DG}]\!])) ,$$

and thus the BR2 bilinear form $a_{BR2}(\cdot, \cdot)$ can be written as

$$(2.8) \quad \begin{aligned} a_{BR2}(u_{DG}, v_{DG}) &= (\nabla u_{DG}, \nabla v_{DG}) - \langle \{\nabla u_{DG}\}, [\![v_{DG}]\!] \rangle - \langle [\![u_{DG}]\!], \{\nabla v_{DG}\} \rangle \\ &\quad + \sum_e (\eta r_e([\![u_{DG}]\!]), r_e([\![v_{DG}]\!])) . \end{aligned}$$

The BR2 method has the advantage that the coefficient η of the stabilization term can be chosen to be $\mathcal{O}(1)$, and is not required to scale with the mesh spacing [13]. However, multigrid solvers often struggle with discretizations involving lifting operators [5, 48, 34].

3. Matrix-free preconditioning. We now describe the construction of a class of matrix-free preconditioners for the high-order Poisson problem (2.4). These preconditioners are based on a structured, geometric multigrid V-cycle applied to a low-order refined discretization described in section 3.1. Parallelization of the method is performed using an overlapping additive Schwarz method, described and analyzed in section 3.2. The low-order formulation gives rise to anisotropy that is treated by means of line smoothing. An algebraic alternative to line smoothing suitable for use on unstructured meshes is ordered incomplete LU (ILU), which is considered in section 3.4. The extension to discontinuous Galerkin discretizations, also by means of an additive Schwarz method, is described in 3.5.

Remark 1. Throughout what follows, we will use some notational conventions. We will use $x \lesssim y$ and $x \gtrsim y$ to mean $x \leq Cy$ and $x \geq Cy$, respectively, where C is a generic constant that is independent of the relevant discretization parameters (e.g., h and p , which represent the mesh spacing and polynomial degree of the high-order problem). When dealing with high-order objects (e.g., spaces, meshes, and operators) we will generally use a subscript p . Likewise, a subscript h will be used to indicate the corresponding low-order (i.e., multilinear) object.

3.1. Low-order refined equivalence. The main tool used in the construction of our preconditioners is the spectral equivalence between the high-order finite element operator K_p and a low-order finite element operator K_h defined on the *low-order refined* mesh. Essential to the construction of the refined mesh will be a structured subgrid of Gauss–Lobatto points within each element, which will give rise to the spectral equivalence between the high-order and low-order operators [20, 21]. This refined mesh, which, for simplicity of exposition, we assume in this section to be affine, is defined by subdividing each mesh element $D \in \mathcal{T}_p$ into mapped images of parallelepipeds with vertices at adjacent Gauss–Lobatto nodes in each dimension. The resulting refined mesh is denoted \mathcal{T}_h . The low-order stiffness matrix K_h corresponds to a standard multilinear ($p = 1$) finite-element discretization on the mesh \mathcal{T}_h . This low-order refined stiffness matrix has two important properties that make it a suitable preconditioner in our setting. First, K_h is much sparser than K_p : the number of nonzeros per row of this matrix is bounded independently of the polynomial degree p of the high-order operator. Second, the low-order operator K_h is spectrally equivalent to the high-order operator K_p , where the constants of equivalence are independent of p . This equivalence is often referred to as FEM-SEM equivalence [24, 23].

Following the work of Canuto et al., [23, 20] and Canuto, Gervasio, and Quarteroni [19], we give a very brief overview of the FEM-SEM equivalence. First, it is useful to establish the spectral equivalence in one spatial dimension between a Galerkin spectral method (G-NI) and piecewise linear finite elements on the Gauss–Lobatto points on the unit interval $[0, 1]$. Given a degree- p polynomial ϕ_p defined on $[0, 1]$, we associate to it its piecewise linear interpolant at the $p + 1$ Gauss–Lobatto points, denoted ϕ_h . We then have the following result due to Canuto.

PROPOSITION 1 (see [18, Propositions 2.1 and 2.2]). *There exist constants c and*

c' independent of p such that

$$\frac{1}{c} \|\phi_p\|_{L^2([0,1])} \leq \|\phi_h\|_{L^2([0,1])} \leq c \|\phi_p\|_{L^2([0,1])}$$

and

$$\frac{1}{c'} \|\phi'_p\|_{L^2([0,1])} \leq \|\phi'_h\|_{L^2([0,1])} \leq c' \|\phi'_p\|_{L^2([0,1])}.$$

This equivalence in the L^2 norm and H^1 seminorm immediately gives spectral equivalence of the constant-coefficient one-dimensional spectral and finite element mass and stiffness matrices, which we denote here by M_p^{1D} , M_h^{1D} , K_p^{1D} , and K_h^{1D} . The extension to multiple space dimensions is straightforward when using tensor-product elements. Define K_p^j by

$$K_p^j = \bigotimes_{i=1}^d G_i^j, \quad \text{where } G_i^j = \begin{cases} K_p^{1D} & \text{if } i = j, \\ M_p^{1D} & \text{if } i \neq j, \end{cases}$$

i.e., K_p^j is given by the Kronecker product of $(d - 1)$ one-dimensional mass matrices, and the one-dimensional stiffness matrix in the j th place. We define K_h^j analogously. The mass and stiffness matrices on the d -dimensional unit cube $[0, 1]^d$ are then given by

$$\begin{aligned} M_p &= \bigotimes_{i=1}^d M_p^{1D}, \\ M_h &= \bigotimes_{i=1}^d M_h^{1D}, \\ K_p &= \sum_{j=1}^d K_p^j, \\ K_h &= \sum_{j=1}^d K_h^j. \end{aligned}$$

Spectral equivalence of the d -dimensional operators follows from the product formula for eigenvalues of Kronecker products and the Courant–Fischer min–max theorem. Finally, a simple Rayleigh quotient argument allows us to conclude the following spectral equivalence of the low-order and high-order operators.

PROPOSITION 2 (cf. section 6.3.4 of [23]). *There exist constants c and C , independent of h and p , such that*

$$c \mathbf{v}^T K_h \mathbf{v} \leq \mathbf{v}^T K_p \mathbf{v} \leq C \mathbf{v}^T K_h \mathbf{v}$$

and

$$c \mathbf{v}^T M_h \mathbf{v} \leq \mathbf{v}^T M_p \mathbf{v} \leq C \mathbf{v}^T M_h \mathbf{v},$$

where M_p and K_p are the high-order finite element operators from (2.4) defined on the mesh \mathcal{T}_p , and M_h and K_h are the low-order operators defined on the Gauss–Lobatto low-order refined mesh \mathcal{T}_h .

As a consequence of this result, a good preconditioner for the low-order refined system K_h will also be a good preconditioner for the original high-order system K_p . Thus, we devote our attention to the construction of robust and scalable preconditioners for the systems K_h , which are obtained through a low-order refinement procedure.

Remark 2. The main challenge in designing a preconditioner for the low-order refined operator K_h is that the resulting mesh is anisotropic. The spacing of Gauss–Lobatto points near the interval endpoints scales like $1/p^2$, and the spacing of Gauss–Lobatto points near the midpoint of the interval scales like $1/p$ [14]. Thus, the aspect ratio of the parallelepipeds making up the Cartesian grid generated from the one-dimensional Gauss–Lobatto points scales like p . As a consequence, the low-order refined meshes \mathcal{T}_h are not shape regular in p , and we can expect degraded convergence of multigrid-type algorithms [35].

3.2. Additive Schwarz and parallel subspace correction methods. In order to reduce the global problem K_h into a set of smaller, local problems that can be solved in parallel, we make use of the additive Schwarz domain decomposition framework [59, 29, 51]. The main idea of the additive Schwarz method is to decompose the finite element space V_h into a sum of subspaces

$$(3.1) \quad V_h = V_0 + V_1 + \cdots + V_J.$$

At this point, it will be useful to introduce some notation. We define the operator $A_h : V_h \rightarrow V_h$ by $(A_h u_h, v_h) = a(u_h, v_h)$, where (\cdot, \cdot) is the L^2 inner product, and $a(u_h, v_h) = (\nabla u_h, \nabla v_h)$. Furthermore, for each subspace V_j , we define the elliptic projection $P_j : V_h \rightarrow V_j$ satisfying

$$(3.2) \quad a(P_j u_h, v_j) = a(u_h, v_j) \quad \text{for all } v_j \in V_j,$$

and the L^2 projection $Q_j : V_h \rightarrow V_j$ satisfying

$$(3.3) \quad (Q_j u_h, v_j) = (u_h, v_j) \quad \text{for all } v_j \in V_j.$$

On each subspace V_j , let A_j be the restriction of A_h to V_j . Then, we have the useful identity

$$(3.4) \quad A_j P_j = Q_j A_h,$$

and so $P_j = A_j^{-1} Q_j A_h$. However, it is usually not feasible to invert the subspace operators A_j exactly, and thus we introduce approximate local solvers $R_j \approx A_j^{-1}$. The additive Schwarz preconditioner is then given by

$$(3.5) \quad B = \sum_{j=0}^J R_j Q_j.$$

Each of the local approximate solvers R_j may be applied in parallel. Thus, this method is also often referred to as the method of parallel subspace corrections [63].

In what follows, we will be interested in studying the convergence properties of a particular choice of additive Schwarz method. The main quantity we are interested in studying is the *iterative condition number* of the preconditioned system. For a generic operator A , which we assume is similar to a symmetric positive-definite matrix, the

iterative condition number is defined as the ratio of largest eigenvalue to the smallest eigenvalue,

$$(3.6) \quad \kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}.$$

This quantity will control the speed of convergence of the preconditioned system when used with a Krylov subspace method such as conjugate gradients (CG).

We now define the space decomposition used for the present solver. The space V_0 is taken to be a coarse subspace, defined as the multilinear ($p = 1$) finite element space on the original (coarse) mesh \mathcal{T}_p . V_0 consists of piecewise polynomials of degree $p = 1$, whose degrees of freedom are defined at the (low-order) vertices of the coarse mesh \mathcal{T}_p . Note that the size of V_0 is independent of the polynomial degree p of the high-order space. Since we are interested in the case of high polynomial degree p , the mesh \mathcal{T}_p is typically fairly coarse, and thus the space V_0 is only a small fraction of the size of the space V_h (which is refined according to the degree p). However, this space will usually still be too large for use with direct solvers. In this work, we choose the coarse solver R_0 to be given by one V-cycle of the algebraic multigrid method BoomerAMG implemented in *hypre* [30, 40]. In the massively parallel setting, the choice of coarse solver can be quite important. We choose to use *hypre* for this component of the solver because of its demonstrated scalability. In principle, any scalable solver suitable for standard low-order finite element discretizations on unstructured meshes can be used for R_0 . Our experience suggests that for relatively small problems with coarse meshes at high order the convergence of the preconditioner B is often not overly sensitive to the choice of coarse solver R_0 , and in this case the cost of applying R_0 represents a relatively small portion of the overall algorithm. For the remainder of this section, we will assume that R_0 is a uniform preconditioner for A_0 , i.e., that $\kappa(R_0 A_0)$ is bounded independent of mesh size h .

The spaces V_1, \dots, V_J are defined in terms of overlapping, unstructured *vertex patches* (cf. [51, 29]). These patches are defined in terms of sets of vertices of the original coarse mesh \mathcal{T}_p . We begin by partitioning the vertices of \mathcal{T}_p into J disjoint sets, E_1, \dots, E_J . To each set of vertices E_j , we associate a subdomain $\Omega_j \subseteq \Omega$, which is obtained by taking the union of all coarse elements $D \in \mathcal{T}_p$ containing any vertex $\mathbf{x}_k \in E_j$. In other words, Ω_j is defined by

$$(3.7) \quad \Omega_j = \bigcup_{\mathbf{x}_k \in E_j} \bigcup_{\substack{D \in \mathcal{T}_p \\ \mathbf{x}_k \in D}} D.$$

Note that, in general, the subdomains Ω_j overlap, since each element $D \in \mathcal{T}_p$ has 2^d vertices, each of which may belong to a different vertex set E_j . The number of overlapping subdomains containing a single element is bounded by the number of vertices per element. The subspace V_j , for $j \geq 1$, is defined as the multilinear finite element space on the mesh \mathcal{T}_h restricted to the subdomain Ω_j . Homogeneous Dirichlet conditions are enforced at the subdomain boundary $\partial\Omega_j$. The linear systems A_j associated with each patch subspace V_j may be quite large, and so the approximate solvers R_j associated with each of these subspaces are given by a structured geometric multigrid V-cycle with ordered ILU smoothing. These approximate solvers will be described and studied in greater detail in section 3.3. For now, we will assume that R_j is a uniform preconditioner for A_j , i.e., that $\kappa(R_j A_j)$ is bounded independent of mesh size h and polynomial degree p .

We are now interested in studying the speed of convergence of the preconditioned system BA_h , where the additive Schwarz preconditioner B is defined by (3.5). To estimate the condition number of the preconditioned system, we largely follow the domain decomposition theory developed in [29, 59, 63, 64]. Note that the condition number $\kappa(BA_h)$ is bounded by c_1/c_0 , where c_1 and c_0 are constants satisfying

$$(3.8) \quad c_0 a(u_h, u_h) \leq a(BA_h u_h, u_h) \leq c_1 a(u_h, u_h).$$

To begin, we first make use the assumption that each of the subspace solvers gives a uniformly well-conditioned system. In other words, there exist constants C_0 and C_1 , independent of h and p , such that (for all $0 \leq j \leq J$)

$$(3.9) \quad C_0 a(u_j, u_j) \leq a(R_j A_j u_j, u_j) \leq C_1 a(u_j, u_j) \quad \text{for all } u_j \in V_j.$$

Using this assumption and making use of the identity (3.4), we have

$$\begin{aligned} a(BA_h u_h, u_h) &= \sum_{j=0}^J a(R_j Q_j A_h u_h, u_h) = \sum_{j=0}^J a(R_j A_j P_j u_h, u_h) \\ &= \sum_{j=0}^J a(R_j A_j P_j u_h, P_j u_h), \end{aligned}$$

and by (3.9) we see

$$(3.10) \quad C_0 \sum_{j=0}^J a(P_j u_h, u_h) \leq a(BA_h u_h, u_h) \leq C_1 \sum_{j=0}^J a(P_j u_h, u_h).$$

From this, we conclude that it suffices to find bounds \tilde{c}_0 and \tilde{c}_1 such that

$$(3.11) \quad \tilde{c}_0 a(u_h, u_h) \leq \sum_{j=0}^J a(P_j u_h, u_h) \leq \tilde{c}_1 a(u_h, u_h),$$

from which we immediately have $c_0 \geq \tilde{c}_0 C_0$ and $c_1 \leq \tilde{c}_1 C_1$.

For the upper bound, we make a standard argument using the finite overlap of the mesh patches. Note that, since the projection operators P_j have norm one,

$$(3.12) \quad \sum_{j=0}^J a(P_j u_h, u_h) = \sum_{j=0}^J a(P_j u_h, P_j u_h) = \sum_{j=0}^J a_j(P_j u_h, P_j u_h) \leq \sum_{j=0}^J a_j(u_h, u_h).$$

Each element is contained in a bounded number of overlapping patches (bounded by the number of vertices per element, 2^d). Accounting for the coarse space V_0 , we then have $\tilde{c}_1 \leq 2^d + 1$.

It remains to estimate the lower bound \tilde{c}_0 , for which we recall Lions' lemma [43].

LEMMA 1 (Lions' lemma). *Let $u_h \in V_h$ be written as an element of $V_0 + V_1 + \dots + V_J$, $u_h = \sum_{j=0}^J u_j$, where $u_j \in V_j$. If*

$$(3.13) \quad \sum_{j=0}^J a(u_j, u_j) \leq \tilde{c}_0^{-1} a(u_h, u_h),$$

then

$$\tilde{c}_0 a(u_h, u_h) \leq \sum_{j=0}^J a(P_j u_h, u_h).$$

By means of this lemma, it suffices to demonstrate a stable decomposition $u_h = \sum_{j=0}^J u_j$ for $u_h \in V_h$ satisfying inequality (3.13). On the coarse space, we use the L^2 projection to define $u_0 = Q_0 u_h$. Making use of the results from Bramble and Xu [12], we have the following properties:

$$(3.14) \quad a(Q_0 u_h, Q_0 u_h) \lesssim a(u_h, u_h) \quad \text{and} \quad \|u_h - Q_0 u_h\|_{L^2}^2 \lesssim h^2 a(u_h, u_h).$$

Now it remains to write $w = \sum_{j=1}^J u_j$, where $w = u_h - u_0$.

We begin by defining a partition of unity $\{\theta_j\}$ subordinate to the patches Ω_j . Each function of the partition of unity will be a member of the low-order coarse finite element space V_0 . Recall that the space V_0 is given by the $p = 1$ finite element space defined on the coarse mesh \mathcal{T}_p , and therefore, $\theta_j : \Omega \rightarrow \mathbb{R}$ is completely determined by its values at the coarse mesh vertices \mathbf{x}_k . Additionally, recall that the subdomains Ω_j are defined in terms of sets of vertices E_j . Then, let $\theta_j \in V_0$ be defined by

$$(3.15) \quad \theta_j(\mathbf{x}_k) = \begin{cases} 1 & \text{if } \mathbf{x}_k \in E_j, \\ 0 & \text{if } \mathbf{x}_k \notin E_j. \end{cases}$$

It is clear from this definition that $\text{supp}(\theta_j) \subseteq \Omega_j$. Since each coarse mesh vertex \mathbf{x}_k is included in exactly one vertex set E_j , we have that $\sum_{j=1}^J \theta_j(\mathbf{x}) = 1$ for all $\mathbf{x} \in \Omega$. Letting $w_j = \theta_j w$, we have $\sum_{j=1}^J w_j = w$. However, note that, in general, $w_j \notin V_j$, since w_j may include quadratic terms. Therefore, define $u_j = I_h(w_j) \in V_j$, where I_h denotes multilinear interpolation on the refined mesh \mathcal{T}_h . By linearity of I_h , the functions u_j also satisfy $\sum_{j=1}^J u_j = w$ and thus $u_0 + \sum_{j=1}^J u_j = u_h$.

We make the following claim concerning the H^1 seminorm of u_j . This lemma is the key technical result allowing us to prove uniformity of the preconditioner with respect to p as well as h . We thank D. Kalchev for his contributions to the proof of this lemma.

LEMMA 2. *Let $w \in V_h$ be given, and let $w_j = \theta_j w$, where θ_j is defined by (3.15). Let $u_j = I_h(w_j)$. Then,*

$$|u_j|_{H^1(\Omega_j)}^2 \lesssim h^{-2} \|w\|_{L^2(\Omega_j)}^2 + |w|_{H^1(\Omega_j)}^2.$$

Remark 3. The main issue that must be addressed in this lemma is that the mesh \mathcal{T}_h is not shape-regular with respect to p , even though we assume that the original coarse mesh \mathcal{T}_p is shape regular. The aspect ratio of elements in \mathcal{T}_h increases linearly with the polynomial degree p of the high-order space V_p . For this reason, standard inverse inequalities (such as those used in, e.g., [29]) are insufficient to prove this claim.

Proof. We first claim that

$$(3.16) \quad |w_j|_{H^1(\Omega_j)}^2 \lesssim h^{-2} \|w\|_{L^2(\Omega_j)}^2 + |w|_{H^1(\Omega_j)}^2.$$

We work on a single element in the refined mesh $D \in \mathcal{T}_h$. On D , both w and θ_j are bilinear functions (and hence smooth). So, a simple application of the product rule gives

$$|\nabla(\theta_j w)|^2 = |(\nabla\theta_j)w + \theta_j \nabla w|^2 \leq 2|(\nabla\theta_j)w|^2 + 2|\theta_j \nabla w|^2.$$

Given the definition of θ_j , we have the pointwise

$$0 \leq \theta_j \leq 1, \quad |\nabla\theta_j| \lesssim 1/h.$$

Making use of these bounds, we obtain

$$|w_j|_{H^1(D)}^2 \lesssim h^{-2} \|w\|_{L^2(D)}^2 + |w|_{H^1(D)}^2.$$

Thus, summing over all elements $D \subseteq \Omega_j$, we arrive at (3.16).

Now, it remains to show that $|I_h(w_j)|_{H^1(D)}^2 \lesssim |w_j|_{H^1(D)}^2$, where the constant is independent of the aspect ratio of the element D . It suffices to consider the case that D is the rectangle $[0, h_1] \times \cdots \times [0, h_d]$. This simplification can be performed because the coarse mesh element containing D can be mapped to the reference element $[0, 1]^d$, and the resulting H^1 seminorms will be equivalent up to a constant depending on the regularity of the coarse mesh \mathcal{T}_p .

We proceed dimension-by-dimension. Define the i th-directional seminorm $|\cdot|_{H_i^1(D)}$ by

$$|u|_{H_i^1(D)}^2 = \int_D \left| \frac{\partial u}{\partial x_i} \right|^2 d\mathbf{x}.$$

From this definition, it follows that

$$|u|_{H^1(D)}^2 = \sum_{i=1}^d |u|_{H_i^1(D)}^2.$$

Define \hat{u}_j on the reference element $[0, 1]^d$ by $\hat{u}_j(\mathbf{x}) = u_j(\mathbf{hx})$, where $\mathbf{hx} = (h_1 x_1, \dots, h_d x_d)$. Then, a simple change of variables shows that

$$|u_j|_{H_i^1(K)}^2 = \frac{\prod_{j=1}^d h_j}{h_i^2} |\hat{u}_j|_{H_i^1([0,1]^d)}^2.$$

Note that since $u_j = I_h(w_j)$, we have $\hat{u}_j = \widehat{I_h(w_j)} = I_h(\hat{w}_j)$. Furthermore, let $\mathcal{Q}_2([0, 1]^d)$ be the space of polynomials on the unit cube that are at most quadratic in each variable, and let $\mathcal{Q}_1([0, 1]^d)$ be the space of multilinear polynomials on the unit cube. Then, since this space is finite-dimensional, it is clear that $I_h : \mathcal{Q}_2([0, 1]^d) \rightarrow \mathcal{Q}_1([0, 1]^d)$ is a bounded linear operator with respect to the $|\cdot|_{H_i^1(D)}$ seminorms. In other words, there exists some constant c_I such that, for $1 \leq i \leq d$,

$$(3.17) \quad |I_h(\hat{w}_j)|_{H_i^1([0,1]^d)}^2 \leq c_I |\hat{w}_j|_{H_i^1([0,1]^d)}^2.$$

Therefore, we have

$$|u_j|_{H^1(D)}^2 = \sum_{i=1}^d |u_j|_{H_i^1(D)}^2 = \sum_{i=1}^d \frac{\prod_{j=1}^d h_j}{h_i^2} |\hat{u}_j|_{H_i^1([0,1]^d)}^2 = \sum_{i=1}^d \frac{\prod_{j=1}^d h_j}{h_i^2} |I_h(\hat{w}_j)|_{H_i^1([0,1]^d)}^2.$$

Making use of the inequality (3.17) gives

$$(3.18) \quad |u_j|_{H^1(D)}^2 \leq c_I \sum_{i=1}^d \frac{\prod_{j=1}^d h_j}{h_i^2} |\hat{w}_j|_{H_i^1([0,1]^d)}^2 = c_I \sum_{i=1}^d |w_j|_{H_i^1(D)}^2 = c_I |w_j|_{H^1(D)}^2,$$

where it is important to note that the constant c_I is independent of the aspect ratio of the rectangle $[0, h_1] \times \cdots \times [0, h_d]$, and hence independent of the high-order degree p . Summing over all elements $D \subseteq \Omega_j, D \in \mathcal{T}_h$, we combine (3.18) with (3.16) to conclude that

$$|u_j|_{H^1(\Omega_j)}^2 \lesssim h^{-2} \|w\|_{L^2(\Omega_j)}^2 + |w|_{H^1(\Omega_j)}^2. \quad \square$$

Remark 4. Note that the above lemma additionally applies to mesh patches Ω_j featuring anisotropy from sources other than Gauss–Lobatto refinement, suggesting that this additive Schwarz technique could prove to be useful when applied to problems with boundary layers or other more general anisotropic features. An example of such a problem is considered in section 4.5.

We are now ready to state the main result concerning the additive Schwarz preconditioner applied to the low-order refined system.

THEOREM 1. *Define the preconditioner B by (3.5), where each local solver R_j is an approximation to A_j^{-1} that is uniform in h and p . Then, there exists a constant \tilde{C} , independent of h and p , such that*

$$\kappa(BA_h) \leq \tilde{C}.$$

Proof. The constant \tilde{C} is bounded above by \tilde{c}_1/\tilde{c}_0 from (3.11). Using the finite overlap of the mesh patches, the estimate given by (3.12) implies that $\tilde{c}_1 \leq 2^d + 1$.

To estimate the lower-bound \tilde{c}_0 we use Lions' lemma and the decomposition $u_h = \sum_{j=0}^J u_j$ described above. Property (3.14) implies that $a(u_0, u_0) \lesssim a(u_h, u_h)$ and $\|w\|_{L^2}^2 \lesssim h^2 a(u_h, u_h)$. Combining these estimates with the result of Lemma 2, we have

$$\begin{aligned} \sum_{j=0}^J a(u_j, u_j) &\lesssim a(u_h, u_h) + \sum_{j=1}^J a(u_j, u_j) \\ &\lesssim a(u_h, u_h) + \sum_{j=1}^J \left(h^{-2} \|w\|_{L^2(\Omega_j)}^2 + |w|_{H^1(\Omega_j)}^2 \right) \\ &\lesssim a(u_h, u_h) + \sum_{j=1}^J |u_h|_{H^1(\Omega_j)}^2 \\ &\lesssim (2^d + 1) a(u_h, u_h). \end{aligned}$$

Thus, $\tilde{c}_0 \gtrsim 1/(2^d + 1)$, completing the proof. \square

Combining the estimates from Theorem 1 and Proposition 2, this result extends to the high-order system (2.4).

COROLLARY 1. *Let B be the preconditioner defined by (3.5). Then, there exists a constant C , independent of h and p , such that*

$$\kappa(BA_p) \leq C.$$

Remark 5. The components described above can also be used to construct a simple two-grid scheme, using $V_1 + \dots + V_J$ as a smoother and V_0 as a coarse solver. Furthermore, the multiplicative version of the Schwarz solver is easily constructed. In the remainder of this paper, we consider only the additive version of the algorithm for reasons of simplicity and ease of parallelization.

3.3. Element-structured multigrid V-cycle. In order to precondition the local patch operators A_j , $1 \leq j \leq J$, we introduce an element-structured multigrid V-cycle. This V-cycle is used as a local approximate solver in an additive Schwarz framework, which in turn is used as a preconditioner in a Krylov subspace method. Recall that each element $D \subseteq \Omega_j$, $D \in \mathcal{T}_p$, from the original high-order mesh is refined

by meshing the Gauss–Lobatto nodes, resulting in a (nonuniform) Cartesian grid of subelements on each coarse element. This gives rise to a natural geometric hierarchy of meshes, which can be obtained from the refined mesh by coarsening the Cartesian subgrid within each coarse element.

Each level in the hierarchy is constructed from the previous level by removing half of the interior Gauss–Lobatto points within each element, resulting in $\mathcal{O}(\log(p))$ levels. For instance, this can be achieved by deleting every other interior one-dimensional Gauss–Lobatto point (proceeding, for example, left-to-right) within the coarse element D . This process terminates when all interior Gauss–Lobatto points have been removed. At this point, the coarsened elements coincide with the elements from the original mesh \mathcal{T}_p . This multigrid procedure bears some resemblance to p -multigrid methods, with $p = 1$ as the coarsest level [38, 39, 31]. The key difference is the current method operates only on low-order refined operators, avoiding the prohibitive cost of assembling the stiffness matrices associated with high-order operators. At each level of the multigrid hierarchy, we perform one presmoothing and one postsmoothing step. The construction of appropriate smoothers is discussed in section 3.4. Depending on the size of the mesh patch, at the coarsest level we can use either a direct solver, or a less expensive approximate solver such as ILU.

3.4. Ordered ILU smoothers. As mentioned previously, the refined mesh \mathcal{T}_h is highly anisotropic. It is well known that the performance of standard smoothers such as Jacobi and Gauss–Seidel depends on the anisotropy of the mesh, and thus degrades as the polynomial degree p of the high-order system is increased. To illustrate this, we apply a variety of algebraic and geometric multigrid methods to the Poisson problem on a 2×2 coarse mesh. Each coarse element is subdivided using a regular grid defined by a given number of refinement points in each direction. These points are either distributed uniformly or according to the Gauss–Lobatto rule. In Figure 1, we present the number of CG iterations required to reduce the residual by a factor of 10^{12} while increasing the number of refinement points per direction in each element. We apply the commonly used BoomerAMG [40] and PyAMG [49] algebraic multigrid preconditioners, as well as a standard geometric multigrid V-cycle based on block-structured mesh coarsening. For each of these multigrid methods, we consider both Jacobian and Gauss–Seidel smoothing. When the grid is refined uniformly, geometric and algebraic multigrid methods converge uniformly using the simple point Jacobi smoother. However, when the grid is refined anisotropically using Gauss–Lobatto points, the iteration counts quickly grow. Using a more powerful smoother such as Gauss–Seidel can improve performance. However, the iteration counts are no longer robust in the refinement level.

A common method for addressing this difficulty is line smoothing [62, 57]. On general, unstructured meshes, properly identifying nonintersecting lines of degrees of freedom along which to perform the smoothing is no longer trivial [36]. To avoid this difficulty, we make use of incomplete LU (ILU) smoothing. Different orderings of the degrees of freedom result in different ILU factorizations. The choice of ordering is known to have a large effect on the convergence of ILU-preconditioned conjugate gradient methods [26]. To make the connection between ILU smoothing and alternating line smoothing, we see that if the ordering is chosen such that the degrees of freedom on each line appear consecutively, then alternating applications of the ILU factorization can be used to perform the alternating line smoothing [37]. We refer to this smoother as “alternating line-ordered ILU.” This procedure still requires the identification of lines corresponding to strong coupling. We are interested in order-

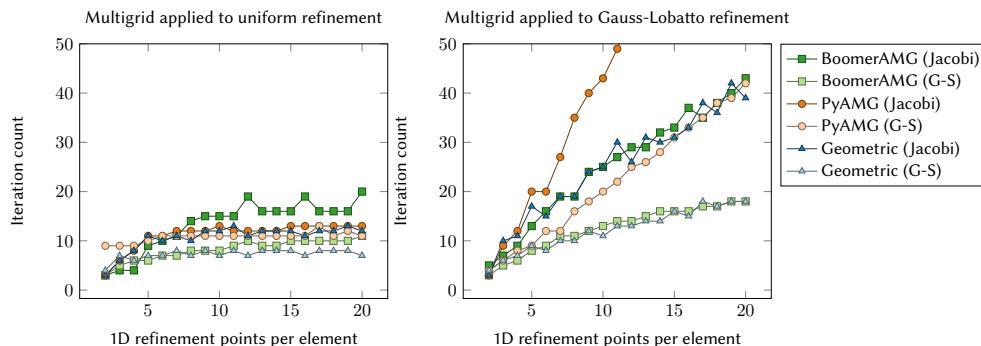


FIG. 1. *Multigrid-preconditioned CG iteration counts (residual reduction of 10^{12}) for a Poisson problem on a Cartesian grid. The grids are obtained by refining a 2×2 Cartesian grid using either uniform or Gauss-Lobatto refinement.*

ing the degrees of freedom in the context of unstructured meshes, and thus turn to algebraic ordering methods. In particular, we consider approximate minimum degree (AMD) [2], reverse Cuthill–McKee (RCM) [25], and minimum discarded fill (MDF) [26, 54] orderings.

We apply alternating line Jacobi, alternating line Gauss–Seidel, and ordered ILU smoothing to the model problem with Gauss–Lobatto refinement. In Figure 2 we show the number of CG iterations required to reduce the residual by a factor of 10^{12} . We first note that the alternating line relaxation methods all result in iteration counts that remain constant, independent of the refinement level. Line Gauss–Seidel smoothing results in iteration counts that are about one-half of those obtained using line Jacobi smoothing. Alternating line-ordered ILU results in iteration counts slightly less than alternating line Gauss–Seidel. MDF-ordered and RCM-ordered ILU give comparable iteration counts, and both remain robust in the number of refinements. AMD-ordered ILU results in iteration counts that exhibit an odd-even effect and grow with the number of refinements, indicating that the AMD ordering is not suitable for our ILU smoothers. The MDF and RCM orderings perform similarly for this simple test problem. In the remainder of this work, we will make use of the MDF ordering because it has been shown to perform well on problems with high anisotropy and complex coefficients [27]. In Figure 3, we illustrate the smoothing properties of the above methods by displaying the error after two multigrid iterations, with a random initial guess and zero right-hand side. It is clear that the pointwise smoothers such as Jacobi and Gauss–Seidel do not smooth the error in regions of anisotropy (i.e., at the medians and towards the edges of the domain). Line Jacobi and AMD-ordered ILU smoothing perform better, but still suffer in anisotropic regions. Line Gauss–Seidel and the remaining ILU methods all result in smooth errors, even in regions of high anisotropy.

3.5. Extension to DG methods. In this section, we consider the extension of the above preconditioners to DG discretizations of (2.1). Of particular interest are the interior penalty and BR2 discretizations, given by (2.5) and (2.8), respectively. The goal is to obtain preconditioners whose convergence is independent of the DG penalty parameter η . This is achieved very naturally in the additive Schwarz framework by making use of the space decomposition introduced in [6]. Recall that V_{DG} is the space of degree- p piecewise polynomial space defined on the mesh \mathcal{T}_p , where no continuity

Multigrid with line smoothing applied and Gauss-Lobatto refinement

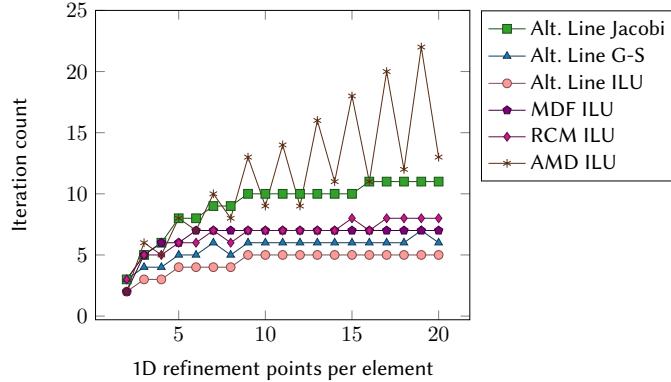


FIG. 2. *Line and ILU smoothing: multigrid-preconditioned CG iteration counts (residual reduction of 10^{12}) for a Poisson problem on a Cartesian grid. The grids are obtained by refining a 2×2 Cartesian grid using Gauss–Lobatto refinement.*

is enforced between elements. We write

$$(3.19) \quad V_{DG} = V_B + V_p,$$

where V_p is the standard H^1 conforming degree- p finite element space. The space V_B consists of all functions that vanish at all interior Gauss–Lobatto nodes. Using the space decomposition (3.19), the additive Schwarz preconditioner for the DG discretization becomes

$$(3.20) \quad B_{DG} = R_B Q_B + R_p Q_p,$$

where R_B and R_p are approximate solvers, $R_B \approx A_B^{-1}$ and $R_p \approx A_p^{-1}$. We take R_p to be the additive Schwarz preconditioner $R_p = B$ for the conforming problem, defined by (3.5), which, according to Theorem 1, is a uniform preconditioner for A_p . As in [6], we take R_B to be a simple point Jacobi preconditioner. Then, we have the following result.

THEOREM 2 (see [6, Theorem 2]). *Let A_{IP} denote the operator corresponding to the interior penalty bilinear form a_{IP} . Let R_B denote the point Jacobi preconditioner on V_B , and let R_p be a uniform (in h and p) approximation to A_p^{-1} . Then, the additive Schwarz preconditioner B_{DG} defined by (3.20) satisfies*

$$\kappa(B_{DG} A_{IP}) \leq C,$$

where the constant C is independent of h , p , and penalty parameter η .

We now extend this result to the BR2 discretization given by (2.8). We begin by defining the interior penalty and BR2 norms,

$$(3.21) \quad \|v\|_{IP}^2 = \|\nabla v\|_{L^2(\Omega)}^2 + \sum_e \frac{p^2}{h} \|\eta^{1/2} [\![v]\!]\|_{L^2(e)}^2,$$

$$(3.22) \quad \|v\|_{BR2}^2 = \|\nabla v\|_{L^2(\Omega)}^2 + \sum_e \|\eta^{1/2} r_e([\![v]\!])\|_{L^2(\Omega)}^2,$$

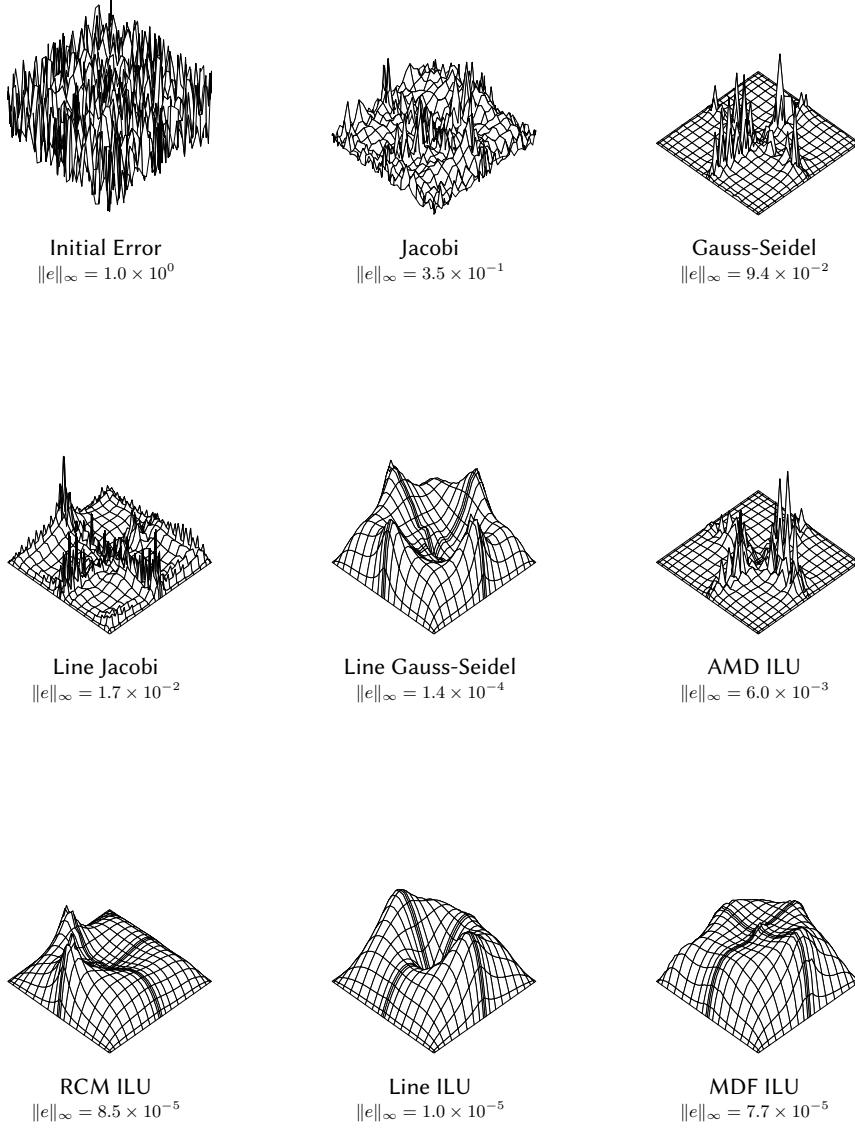


FIG. 3. Two multigrid iterations with different smoothers applied to 2×2 Cartesian grid with Gauss-Lobatto refinement and random initial error.

where we recall that the lifting operator r_e is defined by

$$(3.23) \quad \int_{\Omega} r_e(\varphi) \cdot \tau \, dx = - \int_e \varphi \cdot \{\tau\} \, ds.$$

Then, we have the following boundedness and coercivity estimates.

LEMMA 3. *For the IP bilinear form, we have (e.g., from [4])*

$$\begin{aligned} a_{IP}(u, v) &\lesssim \|u\|_{IP} \|v\|_{IP}, \\ a_{IP}(u, u) &\gtrsim \|u\|_{IP}^2. \end{aligned}$$

For the BR2 bilinear form, we have (cf. [13])

$$\begin{aligned} a_{BR2}(u, v) &\lesssim \|u\|_{BR2}\|v\|_{BR2}, \\ a_{BR2}(u, u) &\gtrsim \|u\|_{BR2}^2. \end{aligned}$$

Thus, to prove an estimate for the BR2 bilinear form, it suffices to prove equivalence of the norms $\|\cdot\|_{IP}$ and $\|\cdot\|_{BR2}$. We will make use of the following results from [17].

LEMMA 4 (see [17, Lemma 3.1]). *The following trace and inverse trace inequalities hold:*

$$(3.24) \quad \|v\|_{L^2(\partial K)}^2 \lesssim \frac{p^2}{h} \|v\|_{L^2(K)}^2 \quad \text{for all } v \in V_{DG},$$

$$(3.25) \quad \|v_B\|_{L^2(K)}^2 \lesssim \frac{h}{p^2} \|v_B\|_{L^2(\partial K)}^2 \quad \text{for all } v_B \in V_B,$$

where we recall that the space V_B denotes the space of functions in V_{DG} that vanish on all interior Gauss–Lobatto nodes.

PROPOSITION 3. *The norms $\|\cdot\|_{IP}$ and $\|\cdot\|_{BR2}$ are equivalent.*

Proof. It is clear that it suffices to prove for every face e that

$$\begin{aligned} \|r_e([\![v]\!])\|_{L^2(\Omega)}^2 &\lesssim \frac{p^2}{h} \|[\![v]\!]\|_{L^2(e)}^2, \\ \frac{p^2}{h} \|[\![v]\!]\|_{L^2(e)}^2 &\lesssim \|r_e([\![v]\!])\|_{L^2(\Omega)}^2. \end{aligned}$$

Let $[\![v]\!]$ be given on e . Let \mathbf{v} denote the extension of $[\![v]\!]$ to the element K by setting $\mathbf{v} = [\![v]\!]$ componentwise on all Gauss–Lobatto nodes lying on the face e , and $\mathbf{v} = 0$ at all other Gauss–Lobatto nodes. Then, choosing $\boldsymbol{\tau} = \mathbf{v}$ and $\boldsymbol{\varphi} = [\![v]\!]$ in (3.23), we have

$$\begin{aligned} \|[\![v]\!]\|_{L^2(e)}^2 &= - \int_{\Omega} r_e([\![v]\!]) \cdot \mathbf{v} \, dx \leq \|r_e([\![v]\!])\|_{L^2(\Omega)} \|\mathbf{v}\|_{L^2(\Omega)} \\ &\lesssim \frac{h^{1/2}}{p} \|r_e([\![v]\!])\|_{L^2(\Omega)} \|[\![v]\!]\|_{L^2(e)}, \end{aligned}$$

where for the last inequality we made use of (3.25).

For the other direction, we let $\boldsymbol{\tau} = r_e([\![v]\!])$ and $\boldsymbol{\varphi} = [\![v]\!]$, and obtain from (3.23)

$$\|r_e([\![v]\!])\|_{L^2(\Omega)}^2 \leq \|[\![v]\!]\|_{L^2(e)} \|r_e([\![v]\!])\|_{L^2(e)} \lesssim \frac{p}{h^{1/2}} \|[\![v]\!]\|_{L^2(e)} \|r_e([\![v]\!])\|_{L^2(\Omega)},$$

making use of inequality (3.24). \square

COROLLARY 2. *Let A_{BR2} denote the operator corresponding to the BR2 bilinear form a_{BR2} . Then, the additive Schwarz preconditioner B_{DG} defined by (3.20) satisfies*

$$\kappa(B_{DG}A_{BR2}) \leq C,$$

where the constant C is independent of h , p , and penalty parameter η .

In section 4.3, we present numerical results using both the IP and BR2 discretizations, verifying the robustness of the resulting preconditioner.

3.6. Computational cost and memory requirements. We briefly discuss the computational cost associated with the matrix-free preconditioners described above. Recall that the number of degrees of freedom for the high-order problem (2.4) scales like $n_{dof} = \mathcal{O}(p^d n_{el})$, where n_{el} is the number of elements in the mesh \mathcal{T}_p . Using sum-factorized operator evaluation, the action of the operator K_p can be computed in $\mathcal{O}(p^{d+1} n_{el}) = \mathcal{O}(pn_{dof})$ operations [50, 52]. The matrices associated with the low-order refined discretization are assembled, requiring a constant number of operations per degree of freedom. Similarly, because of the inherent sparsity of the low-order operator, the number of nonzeros per row of these matrices is constant, independent of p . Let n_{coarse} denote the number of degrees of freedom in the coarse (multilinear) finite element space defined on the mesh \mathcal{T}_p . It is clear that $n_{coarse} < n_{dof}$, and in the very high-order case, we have $n_{coarse} \ll n_{dof}$. Additionally, let n_{loc} denote an upper bound on the number of degrees of freedom in a given patch and recall that J denotes the total number of mesh patches, so that $n_{dof} \lesssim Jn_{loc}$. We will assume that the number of elements per patch is bounded by some constant, and so $n_{loc} = \mathcal{O}(p^d)$. The MDF ordering on each patch is performed in $\mathcal{O}(n_{loc} \log(n_{loc}))$ time using a min-heap data structure [54]. This cost is asymptotically dominated by the cost of the high-order operator evaluation, and is only performed once as a preprocessing step, rather than at every iteration. The computational cost and memory requirements for the preconditioning operations are summarized in Table 1. Given that the preconditioner is robust in the discretization parameters, we conclude that the total runtime for the algorithm scales like the cost of a matrix-free application of the high-order operator. Furthermore, the total memory cost associated with building and applying the preconditioner is optimal, scaling linearly in the number of degrees of freedom.

TABLE 1
Computational complexity and memory requirements for preconditioning operations.

| | Operations | Memory |
|----------------------------------|--------------------------------------|---------------------------|
| Matrix-free application of K_p | $\mathcal{O}(pn_{dof})$ | $\mathcal{O}(n_{dof})$ |
| Matrix-based assembly of K_h | $\mathcal{O}(n_{dof})$ | $\mathcal{O}(n_{dof})$ |
| Coarse solver R_0 | $\mathcal{O}(n_{coarse})$ | $\mathcal{O}(n_{coarse})$ |
| MDF ordering | $\mathcal{O}(n_{dof} \log(n_{loc}))$ | $\mathcal{O}(n_{dof})$ |
| ILU smoothing | $\mathcal{O}(n_{dof})$ | $\mathcal{O}(n_{dof})$ |
| Total preconditioner application | $\mathcal{O}(pn_{dof})$ | $\mathcal{O}(n_{dof})$ |

4. Numerical results. In this section, we present a variety of numerical results intended to study the performance of the preconditioner and verify the properties shown in the preceding sections. The preconditioner was implemented in the MFEM finite element discretization library [3].

4.1. Cartesian grid. For a first numerical example, we consider the h - and p -refinement of a Cartesian grid with constant coefficient $b = 1$. We begin with a 2×2 grid, and refine uniformly by factors of two, so that the finest mesh is a 32×32 grid. Polynomial degrees between 2 and 20 are used, such that the largest system considered has a total of 410,881 degrees of freedom. We first study the performance of the local multigrid solver by using a single additive Schwarz patch for the entire domain. This corresponds to preconditioning the high-order system K_p with an element-structured multigrid V-cycle applied to the low-order refined system K_h , coarsening each element using the strategy described in section 3.3. The coarsening procedure terminates when all interior Gauss–Lobatto points have been eliminated. One application of

TABLE 2
Number of CG iterations required to reduce the residual by a factor of 10^8 . Constant coefficient Cartesian grid with one patch.

| p | $n_x = 2$ | $n_x = 4$ | $n_x = 8$ | $n_x = 16$ | $n_x = 32$ |
|-----|-----------|-----------|-----------|------------|------------|
| 2 | 4 | 10 | 12 | 12 | 13 |
| 4 | 13 | 14 | 14 | 14 | 14 |
| 6 | 15 | 16 | 15 | 16 | 16 |
| 8 | 16 | 16 | 16 | 15 | 16 |
| 10 | 17 | 17 | 17 | 17 | 18 |
| 12 | 17 | 18 | 17 | 18 | 19 |
| 14 | 18 | 18 | 17 | 19 | 19 |
| 16 | 18 | 17 | 17 | 16 | 18 |
| 18 | 18 | 18 | 17 | 19 | 19 |
| 20 | 18 | 18 | 17 | 19 | 19 |

TABLE 3
Number of CG iterations required to reduce the residual by a factor of 10^8 . Constant coefficient Cartesian grid with vertex patches.

| p | $n_x = 2$ | $n_x = 4$ | $n_x = 8$ | $n_x = 16$ | $n_x = 32$ |
|-----|-----------|-----------|-----------|------------|------------|
| 2 | 4 | 10 | 14 | 20 | 26 |
| 4 | 12 | 17 | 22 | 26 | 29 |
| 6 | 17 | 22 | 26 | 31 | 32 |
| 8 | 19 | 24 | 28 | 33 | 34 |
| 10 | 22 | 26 | 31 | 35 | 36 |
| 12 | 24 | 27 | 30 | 35 | 36 |
| 14 | 24 | 30 | 32 | 36 | 37 |
| 16 | 25 | 31 | 33 | 36 | 37 |
| 18 | 26 | 31 | 34 | 37 | 38 |
| 20 | 27 | 32 | 34 | 37 | 38 |

MDF-ordered ILU(0) is used for presmoothing and postsmoothing at each level. The coarse solve is performed on the $p = 1$ discretization on the coarse mesh \mathcal{T}_p , using one V-cycle of BoomerAMG. The number of CG iterations required to reduce the residual by a factor of 10^8 is shown in Table 2. We observe that for this problem, fewer than 20 iterations were required for all cases considered. We also observe a mild preasymptotic dependence of the number of iterations on p for low polynomial degrees. However, for p larger than about 10, the iterations remain almost constant.

Additionally, we study the performance of the additive Schwarz method with vertex patches. In this case, the domain is decomposed into overlapping subdomains corresponding to each vertex of the coarse mesh \mathcal{T}_p . Each subdomain consists of all elements containing the given vertex. The same geometric multigrid procedure with ILU smoothing as described above is applied to each of the low-order refined subdomain local problems independently. As before, the coarse solver is one V-cycle of BoomerAMG on the global coarse problem. The number of CG iterations required to reduce the residual by a factor of 10^8 is shown in Table 3. Similar to the previous case, we observe a preasymptotic dependence on the mesh size and polynomial degree. The number of iterations is larger by about a factor of two when compared with one patch per domain. However, in the case of vertex patches, the computations can be performed in parallel on each patch.

Note that the iteration counts shown in Tables 2 and 3 are largely consistent with past results from the literature on low-order preconditioning of high-order operators.

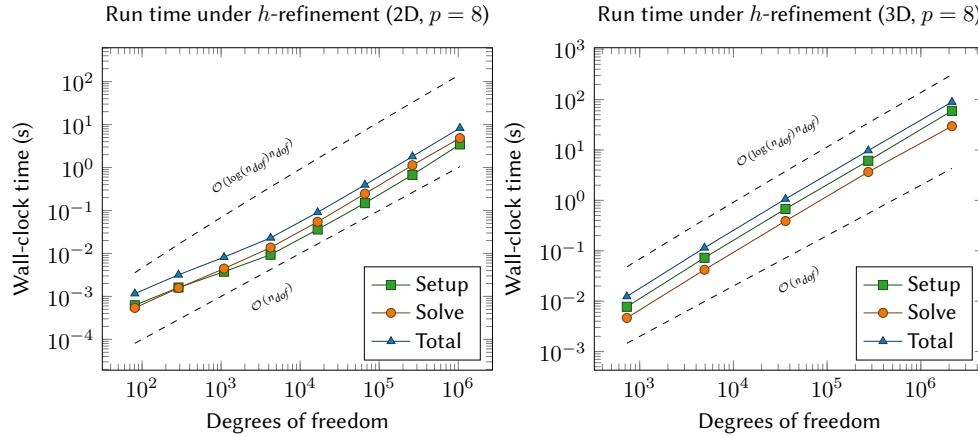


FIG. 4. Wall-clock run times in 2D and 3D under uniform h -refinement using fixed polynomial degree $p = 8$.

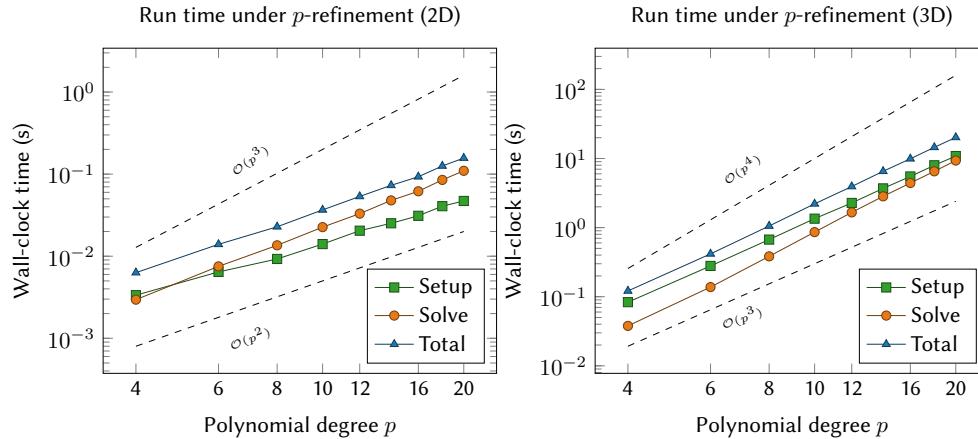


FIG. 5. Wall-clock run times in 2D and 3D under p -refinement using a fixed Cartesian mesh.

In particular, we compare these results to the low-order finite element preconditioning of spectral element methods presented in [44], in which several additive Schwarz and multigrid preconditioners were studied. The results from [44] suggest that increased performance could be obtained by using a hybrid Schwarz strategy (incorporating the coarse solve in a multiplicative rather than additive way), and by using a larger coarse space. Additionally, in [44], the importance of weighting the smoother by a diagonal counting matrix was emphasized. Investigating these considerations in the context of the present solver is saved for future work.

We also study the performance of the preconditioner in terms of wall-clock run time under h - and p -refinement in two dimensions (2D) and three dimensions (3D). We first fix the polynomial degree to be $p = 8$, and perform a sequence of uniform refinements. The wall-clock time required for the solution (i.e., total time over all CG iterations) scales optimally (i.e., $\mathcal{O}(n_{\text{dof}})$). The time required for setup (i.e., construction of the preconditioner) scales like $\mathcal{O}(n_{\text{dof}} \log(n_{\text{dof}}))$, because of the $n \log(n)$ complexity of the MDF ordering used in the ILU factorization. In 2D, the cost of

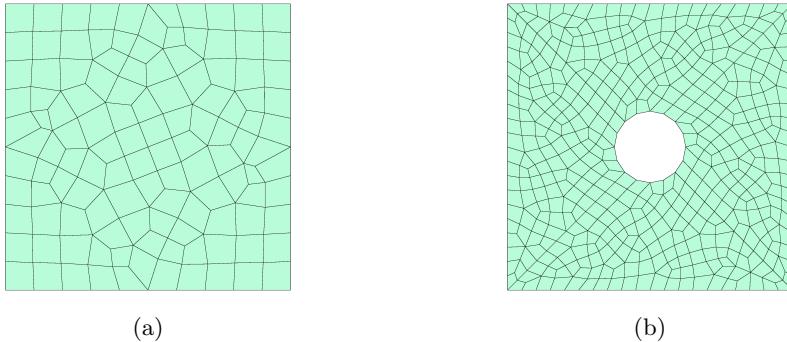


FIG. 6. *Unstructured quadrilateral meshes used for the numerical experiments.*

the MDF factorization appears to be negligible, and we observe the setup run time to scale close to linearly in the number of degrees of freedom. However, in 3D, the cost of computing the MDF ordering is no longer negligible. We also perform p -refinement on a fixed mesh in 2D and 3D. In both cases, we observe that the setup run time scales linearly in the number of degrees of freedom ($\mathcal{O}(p^d)$) since the construction of the preconditioner requires only the low-order operator K_h . The run time for the CG iterations scales like $\mathcal{O}(p^{d+1})$ because of the cost of applying the high-order operator K_p using the sum factorization technique. Figures 4 and 5 show timings performed in serial running on an Intel Xeon Gold 6130 CPU (2.10 GHz), confirming these scalings.

4.2. Variable coefficients and unstructured meshes. The coefficients (the first three of which are from [57]) are shown in Figure 7 and are defined as follows:

- $b_1(x, y) = 10^4(1-x^2)(1-y^2)$, which features sharp gradients near the boundaries of the domain $[-1, 1]^2$.
- $b_2(x, y) = 100x^2 + y^2 + 1$, which is anisotropic, and results in an anisotropic solution with a steep gradient.
- $b_3(x, y) = (1+x^2+y^2)^4$.
- $b_4(x, y)$ is chosen to be a discontinuous, piecewise constant coefficient with values 10 and 1 on elements, distributed randomly on the mesh.

The iteration counts for these coefficients are presented in Table 4. We also consider unstructured meshes and more complicated geometries. The meshes used for the numerical experiments are shown in Figure 6. The first mesh is an unstructured quadrilateral mesh of the domain $[-1, 1]^2$. The domain of the second mesh consists of the unit square $[-1, 1]^2$ with a disk of radius $1/4$ removed. The iteration counts remain bounded for all coefficients, although convergence is slower for b_1 than for the remaining coefficients. The additive Schwarz method with one patch per vertex converges in about twice as many iterations as using a single patch for the domain. Iteration counts are similar between the two meshes, and are about 50% larger than in the case of the Cartesian grid. Iteration counts (shown in Table 5) are similar between the two meshes, and are about 50% larger than in the case of the Cartesian grid.

4.3. Discontinuous Galerkin methods. To test the performance of the additive Schwarz preconditioner B_{DG} applied to interior penalty and BR2 discontinuous Galerkin discretizations, we again make use of the unstructured meshes shown in Figure 6. We measure the number of CG iterations required to reduce the residual by a factor of 10^8 under h - and p -refinement. We present these results in Tables 6 and

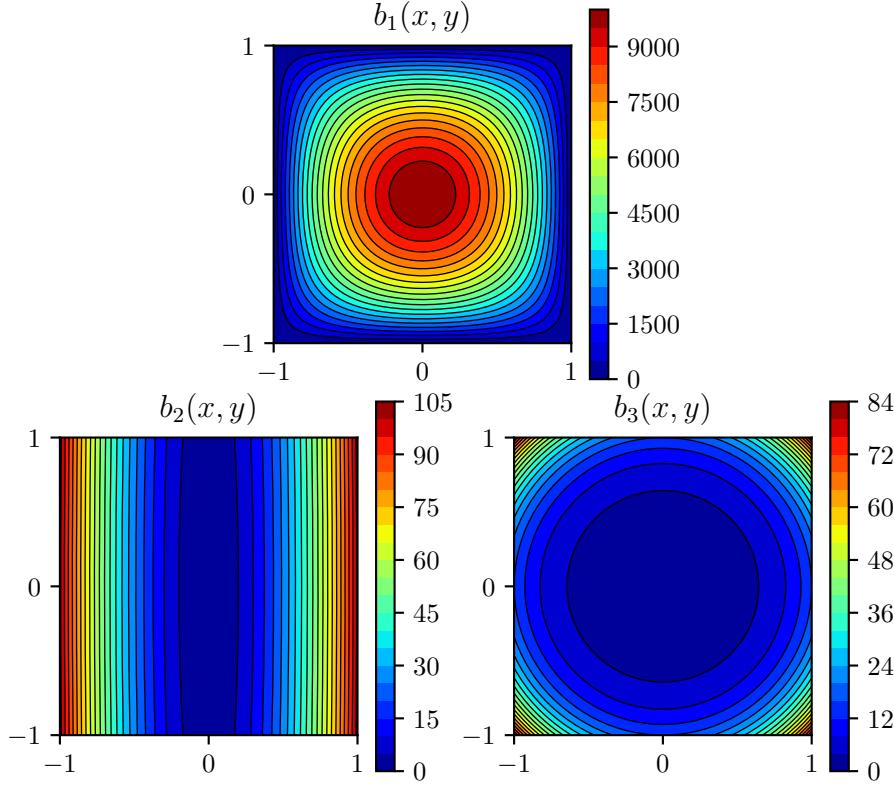
FIG. 7. Variable coefficients $b_i(x, y)$ used for the examples in section 4.2.

TABLE 4

Variable coefficient case on unstructured mesh (a). Number of iterations required to reduce the residual by 10^8 .

| p | Single patch | | | | Vertex patches | | | |
|-----|--------------|-------------|-------------|-------------|----------------|-------------|-------------|-------------|
| | $b_1(x, y)$ | $b_2(x, y)$ | $b_3(x, y)$ | $b_4(x, y)$ | $b_1(x, y)$ | $b_2(x, y)$ | $b_3(x, y)$ | $b_4(x, y)$ |
| 2 | 15 | 15 | 14 | 16 | 24 | 23 | 22 | 26 |
| 4 | 18 | 17 | 17 | 17 | 35 | 30 | 29 | 31 |
| 6 | 23 | 19 | 19 | 19 | 42 | 34 | 34 | 35 |
| 8 | 23 | 21 | 21 | 21 | 49 | 38 | 37 | 38 |
| 10 | 29 | 22 | 22 | 23 | 55 | 40 | 40 | 41 |
| 12 | 30 | 22 | 22 | 23 | 58 | 41 | 41 | 43 |
| 14 | 35 | 23 | 23 | 26 | 63 | 43 | 43 | 44 |
| 16 | 30 | 24 | 24 | 25 | 66 | 44 | 44 | 46 |
| 18 | 41 | 25 | 25 | 28 | 71 | 46 | 45 | 47 |
| 20 | 37 | 24 | 24 | 28 | 72 | 46 | 46 | 48 |

7. We note that the number of iterations does not increase as the mesh is further refined, verifying robustness in h . Similar to the continuous Galerkin examples presented previously, we observe a slight preasymptotic increase in the iteration count with p , though the number of iterations remains bounded, verifying robustness in p . Since one of the main features of the preconditioner is its robustness in the IP and BR2 penalty parameters η , we consider a fixed mesh and set $p = 6$. We increase the value of η from 1 to 10,000 by factors of 10, and report the iterations required

TABLE 5
Variable coefficient case on unstructured mesh (b). Number of iterations required to reduce the residual by 10^8 .

| p | Single patch | | | | Vertex patches | | | |
|-----|--------------|-------------|-------------|-------------|----------------|-------------|-------------|-------------|
| | $b_1(x, y)$ | $b_2(x, y)$ | $b_3(x, y)$ | $b_4(x, y)$ | $b_1(x, y)$ | $b_2(x, y)$ | $b_3(x, y)$ | $b_4(x, y)$ |
| 2 | 15 | 15 | 15 | 16 | 31 | 28 | 27 | 31 |
| 4 | 18 | 17 | 17 | 18 | 39 | 34 | 34 | 33 |
| 6 | 23 | 20 | 20 | 22 | 46 | 39 | 38 | 37 |
| 8 | 24 | 22 | 22 | 22 | 52 | 42 | 41 | 41 |
| 10 | 31 | 23 | 23 | 26 | 56 | 44 | 43 | 44 |
| 12 | 30 | 24 | 24 | 27 | 60 | 45 | 44 | 46 |
| 14 | 36 | 25 | 25 | 30 | 66 | 46 | 46 | 50 |
| 16 | 29 | 25 | 25 | 27 | 66 | 47 | 47 | 50 |
| 18 | 41 | 27 | 26 | 34 | 72 | 48 | 47 | 55 |
| 20 | 37 | 26 | 26 | 31 | 71 | 48 | 48 | 53 |

TABLE 6
BR2 discontinuous Galerkin method. Number of CG iterations required to reduce the residual by a factor of 10^8 under p -refinement.

| p | Mesh (a) | Mesh(b) |
|-----|----------|---------|
| 2 | 21 | 22 |
| 4 | 22 | 23 |
| 6 | 25 | 26 |
| 8 | 23 | 26 |
| 10 | 26 | 29 |
| 12 | 26 | 30 |
| 14 | 27 | 31 |
| 16 | 26 | 29 |
| 18 | 29 | 32 |
| 20 | 28 | 31 |

TABLE 7
BR2 discontinuous Galerkin method. Number of CG iterations required to reduce the residual by a factor of 10^8 under h -refinement.

| Refinements | Mesh (a) | | Mesh (b) | |
|-------------|------------|---------|------------|-----------|
| | Iterations | DOFs | Iterations | DOFs |
| 0 | 25 | 5,880 | 26 | 23,814 |
| 1 | 24 | 23,520 | 26 | 95,256 |
| 2 | 23 | 94,080 | 25 | 381,024 |
| 3 | 22 | 376,320 | 24 | 1,524,096 |

to converge in Table 8. Confirming the results from section 3.5, the iteration count remains robust in η for both the IP and BR2 methods.

4.4. Comparison with AMG methods. In order to compare the present method with a possible alternative matrix-free method for high-order operators, we consider a FEM-SEM preconditioner using an algebraic multigrid V-cycle as an approximate solver for the low-order preconditioner. We make use of BoomerAMG with both ℓ_1 -scaled hybrid symmetric Gauss–Seidel relaxation and ℓ_1 -scaled Jacobi relaxation [40]. We compare the number of CG iterations for increasing polynomial degree p . For this test, we use an unstructured mesh with 80 curved isoparametric quadrilateral elements. We also consider the high-order DG BR2 system with $\eta = 10$ to

TABLE 8

Effect of DG penalty parameter η on convergence. Number of iterations required to reduce the residual by a factor of 10^8 using interior penalty and BR2 DG methods on the unstructured meshes from Figure 6.

| η | Mesh (a) | | Mesh (b) | |
|--------|----------|-----|----------|-----|
| | IP | BR2 | IP | BR2 |
| 10^0 | 29 | 23 | 33 | 25 |
| 10^1 | 27 | 22 | 28 | 23 |
| 10^2 | 25 | 18 | 27 | 21 |
| 10^3 | 23 | 18 | 26 | 21 |
| 10^4 | 22 | 18 | 24 | 19 |

TABLE 9

Comparison with algebraic multigrid methods. Number of CG iterations required to reduce the residual by a factor of 10^8 . AMG(J) and AMG(GS) indicate BoomerAMG using ℓ_1 -scaled Jacobi and hybrid symmetric Gauss–Seidel, respectively. LOR-AMG indicates that the AMG V-cycle is applied to the low-order refined system. B(1) and B(V) indicate the present preconditioner with a single patch, and one patch per vertex, respectively.

| p | Continuous Galerkin | | | | Discontinuous Galerkin | | | |
|-----|---------------------|------|------------|--------------|------------------------|------|--------|----------|
| | B(1) | B(V) | LOR-AMG(J) | LOR-AMG(G-S) | B(1) | B(V) | AMG(J) | AMG(G-S) |
| 2 | 15 | 24 | 26 | 20 | 18 | 37 | 186 | 81 |
| 4 | 18 | 31 | 39 | 31 | 22 | 46 | 329 | 124 |
| 6 | 20 | 35 | 53 | 40 | 23 | 48 | 370 | 145 |
| 8 | 21 | 38 | 65 | 46 | 25 | 49 | 361 | 130 |
| 10 | 23 | 40 | 76 | 52 | 26 | 51 | 381 | 130 |
| 12 | 26 | 42 | 96 | 61 | 27 | 53 | 318 | 122 |
| 14 | 28 | 43 | 106 | 71 | 29 | 53 | 344 | 131 |
| 16 | 30 | 44 | 114 | 75 | 31 | 54 | 296 | 123 |
| 18 | 31 | 45 | 123 | 77 | 34 | 55 | 319 | 124 |
| 20 | 33 | 46 | 132 | 87 | 36 | 55 | 287 | 110 |

ensure coercivity of the system. Because the FEM-SEM equivalence is less straightforward when applied to DG systems, we compare with AMG applied to the fully assembled high-order DG system (i.e., standard matrix-based AMG). The results are shown in Table 9. We notice that the additive Schwarz preconditioner is robust in p and in the number of mesh patches. Consistent with the previous results, using vertex patches requires about 1.5 times more iterations than a single patch for the domain. However, these patches allow for parallelization, because the local solutions on all the patches may be computed simultaneously. The algebraic multigrid methods are not robust in p , likely due to the anisotropy introduced in the low-order refined mesh by the Gauss–Lobatto points. When applied to DG discretizations, the number of CG iterations required when using AMG methods was quite large, consistent with results previously reported in the literature, indicating that multigrid methods may struggle with DG discretizations with lifting operators such as local DG or BR2 [5, 48].

4.5. Anisotropic meshes. In Remark 4, it is mentioned that the additive Schwarz approach described presently could be used with mesh patches featuring anisotropy originating not only from the low-order refined Gauss–Lobatto mesh. Solvers often struggle with highly anisotropic meshes, which are commonly used to resolve boundary layers when solving convection-dominated problems (e.g., high Reynolds number Navier–Stokes) [45]. In this example, we consider a sequence of two-dimensional unstructured meshes, with a strip of high-aspect-ratio elements. We begin with a mesh

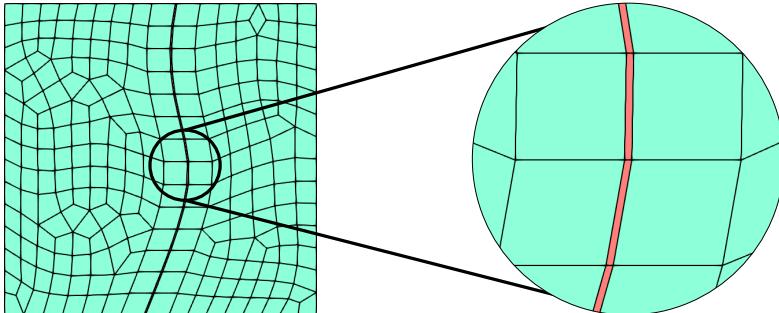


FIG. 8. Example of mesh used for the anisotropic test case, showing elements (highlighted in red) with an aspect ratio of 15.

that is roughly isotropic (maximum aspect ratio 1.5), and increase the aspect ratio of the elements in the strip by a factor of 10, until reaching a maximum aspect ratio of 1500. An example of one such mesh is shown in Figure 8.

We solve the high-order constant-coefficient problem (with $p = 3, 7, 11, 15$) on this sequence of meshes, using the additive Schwarz preconditioner described here, and compare with BoomerAMG with Gauss–Seidel smoothing applied to the low-order refined problem (denoted LOR-AMR(G-S)). For this problem, in order for the additive Schwarz method to be robust in the aspect ratio of the elements, the overlap of the subdomains must be independent of the aspect ratio. To achieve this, subdomain patches containing anisotropic elements are extended to ensure that the overlap remains isotropic. For this test case, we use $n = 1, 2, 3$ additive Schwarz patches (denoted $B(n)$), as well as one patch per vertex (about 250 overlapping patches for this test case), where patches containing anisotropic elements are enlarged to obtain isotropic overlap (denoted $B(V)$). In Figure 9, we compare the number of CG iterations required to reduce the residual by a factor of 10^8 . We note that algebraic multigrid applied to the low-order refined system is strongly dependent on both the polynomial degree and on the anisotropy of the mesh. On the other hand, since the additive Schwarz patches have isotropic overlap, the hypotheses of Lemma 2 are satisfied, and so the resulting space decomposition is stable. Consistent with this, the iteration counts using the additive Schwarz solver remain constant with respect to the aspect ratio. Slight increases in iteration counts are observed for higher polynomial degrees and increased number of subdomains, as is also observed in the preceding numerical examples.

5. Conclusions. In this work, we introduced a matrix-free preconditioner for high-order finite element discretizations of elliptic problems based on a low-order refined methodology. The low-order refined system is sparse, and the associated matrix can be formed in linear time and memory. The low-order refined system is preconditioned using a patch-based overlapping additive Schwarz method. The local approximate solvers in the Schwarz method use a structured geometric multigrid technique with ordered ILU smoothing to treat the anisotropy of the low-order discretization. The extension to discontinuous Galerkin discretizations is performed naturally in the additive Schwarz context. The resulting preconditioners are robust in the discretization parameters h , p , and the DG penalty parameter η . Numerical experiments are performed on a variety of geometries and meshes. The robustness of the preconditioner is verified on test cases involving variable coefficients with sharp gradients

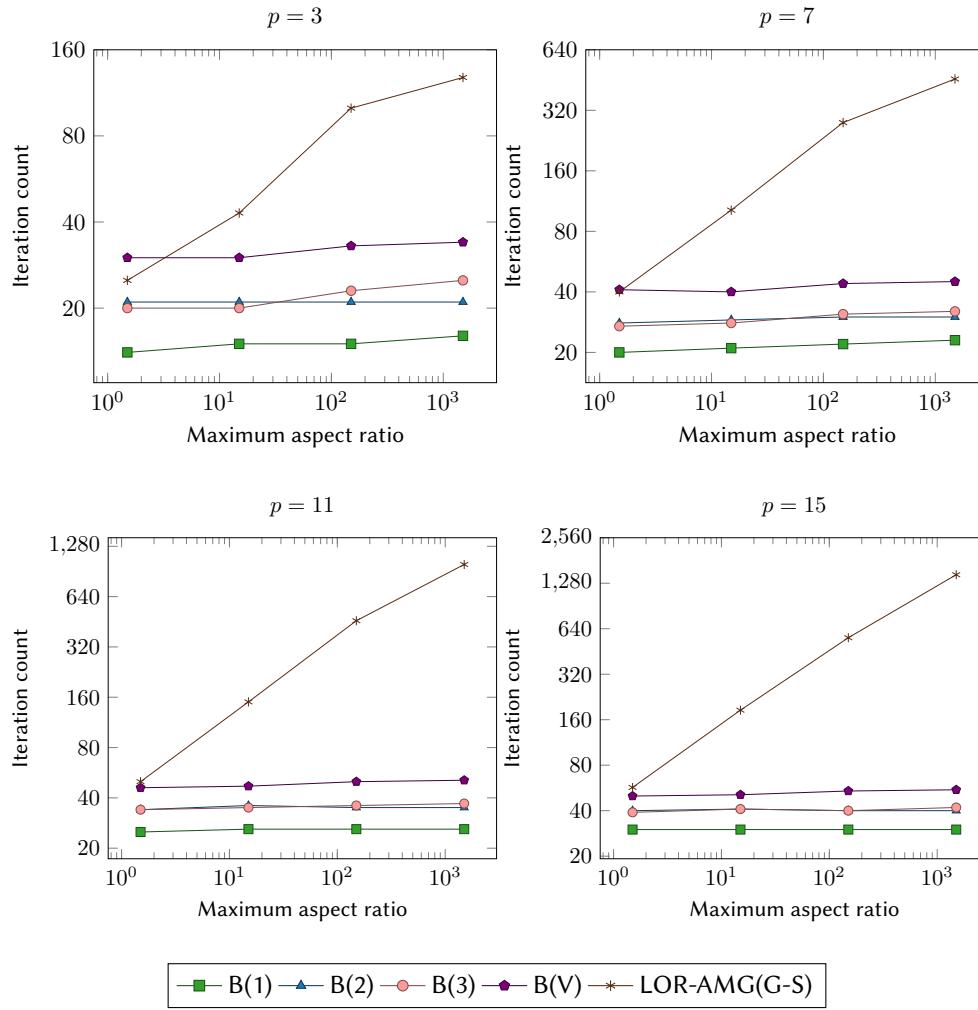


FIG. 9. Convergence on a sequence of meshes with anisotropic elements, showing number of CG iterations required to reduce the residual by a factor of 10^8 . $B(n)$ indicates additive Schwarz with n subdomains, $B(V)$ indicates additive Schwarz with vertex patches with isotropic overlap, and $LOR-AMG(G-S)$ indicates one V -cycle of BoomerAMG applied to the low-order refined system.

and directional biases. The efficiency of the method is compared with a FEM-SEM preconditioner using algebraic multigrid for the low-order refined system.

Acknowledgment. The author thanks Tz. Kolev and D. Kalchev for helpful and insightful conversations.

REFERENCES

- [1] M. ADAMS, M. BREZINA, J. HU, AND R. TUMINARO, *Parallel multigrid smoothing: Polynomial versus Gauss-Seidel*, J. Comput. Phys., 188 (2003), pp. 593–610, [https://doi.org/10.1016/s0021-9991\(03\)00194-3](https://doi.org/10.1016/s0021-9991(03)00194-3).
- [2] P. R. AMESTOY, T. A. DAVIS, AND I. S. DUFF, *An approximate minimum degree ordering algorithm*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 886–905, <https://doi.org/10.1137/s0895479894278952>.

- [3] R. ANDERSON, J. ANDREJ, A. BARKER, J. BRAMWELL, J.-S. CAMIER, J. CERVENY, V. DOBREV, Y. DUDOUIT, A. FISHER, T. KOLEV, W. PAZNER, M. STOWELL, V. TOMOV, I. AKKERMANN, J. DAHM, D. MEDINA, AND S. ZAMPINI, *MFEM: A modular finite element library*, Comput. Math. Appl., in press.
- [4] P. F. ANTONIETTI AND P. HOUSTON, *A class of domain decomposition preconditioners for hp-discontinuous Galerkin finite element methods*, J. Sci. Comput., 46 (2010), pp. 124–149, <https://doi.org/10.1007/s10915-010-9390-1>.
- [5] P. F. ANTONIETTI, M. SARTI, AND M. VERANI, *Multigrid algorithms for hp-discontinuous Galerkin discretizations of elliptic problems*, SIAM J. Numer. Anal., 53 (2015), pp. 598–618, <https://doi.org/10.1137/130947015>.
- [6] P. F. ANTONIETTI, M. SARTI, M. VERANI, AND L. T. ZIKATANOV, *A uniform additive Schwarz preconditioner for high-order discontinuous Galerkin approximations of elliptic problems*, J. Sci. Comput., 70 (2016), pp. 608–630, <https://doi.org/10.1007/s10915-016-0259-9>.
- [7] D. N. ARNOLD, *An interior penalty finite element method with discontinuous elements*, SIAM J. Numer. Anal., 19 (1982), pp. 742–760, <https://doi.org/10.1137/0719052>.
- [8] D. N. ARNOLD, F. BREZZI, B. COCKBURN, AND L. D. MARINI, *Unified analysis of discontinuous Galerkin methods for elliptic problems*, SIAM J. Numer. Anal., 39 (2002), pp. 1749–1779, <https://doi.org/10.1137/s0036142901384162>.
- [9] T. M. AUSTIN, M. BREZINA, B. JAMROZ, C. JHURANI, T. A. MANTEUFFEL, AND J. RUGE, *Semi-automatic sparse preconditioners for high-order finite element methods on non-uniform meshes*, J. Comput. Phys., 231 (2012), pp. 4694–4708, <https://doi.org/10.1016/j.jcp.2012.03.013>.
- [10] F. BASSI, S. REBAY, M. SAVINI, G. MARIOTTI, AND S. PEDINOTTI, *A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flow*, in Proceedings of the Second European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, R. Decuyper and G. Dibelius, eds., Technologisch Instituut Section on Aeronautics, Antwerp, Belgium, 1997, pp. 99–108.
- [11] P. D. BELLO-MALDONADO AND P. F. FISCHER, *Scalable low-order finite element preconditioners for high-order spectral element poisson solvers*, SIAM J. Sci. Comput., 41 (2019), pp. S2–S18, <https://doi.org/10.1137/18m1194997>.
- [12] J. H. BRAMBLE AND J. XU, *Some estimates for a weighted L^2 projection*, Math. Comp., 56 (1991), pp. 463–476, <https://doi.org/10.1090/s0025-5718-1991-1066830-3>.
- [13] F. BREZZI, G. MANZINI, D. MARINI, P. PIETRA, AND A. RUSSO, *Discontinuous Galerkin approximations for elliptic problems*, Numer. Methods Partial Differential Equations, 16 (2000), pp. 365–378, [https://doi.org/10.1002/1098-2426\(200007\)16:4<365::aid-num2>3.0.co;2-y](https://doi.org/10.1002/1098-2426(200007)16:4<365::aid-num2>3.0.co;2-y).
- [14] K. BRIX, C. CANUTO, AND W. DAHMEN, *Nested dyadic grids associated with Legendre-Gauss-Lobatto grids*, Numer. Math., 131 (2014), pp. 205–239, <https://doi.org/10.1007/s00211-014-0691-4>.
- [15] J. BROWN, *Efficient nonlinear solvers for nodal high-order finite elements in 3D*, J. Sci. Comput., 45 (2010), pp. 48–63, <https://doi.org/10.1007/s10915-010-9396-8>.
- [16] J. BROWN, A. ABDELFATA, J.-S. CAMIER, V. DOBREV, J. DONGARRA, P. FISCHER, A. FISHER, Y. DUDOUIT, A. HAIDAR, K. KAMRAN, T. KALEV, M. MIN, T. RATNAYAKA, M. SHEPHARD, C. SMITH, S. TOMOV, V. TOMOV, AND T. WARBURTON, *CEED ECP Milestone Report: Public release of CEED 1.0*, Tech. report, U.S. Department of Energy, Washington, DC, 2018, <https://doi.org/10.5281/zenodo.2542343>.
- [17] E. BURMAN AND A. ERN, *Continuous interior penalty hp-finite element methods for advection and advection-diffusion equations*, Math. Comp., 76 (2007), pp. 1119–1140, <https://doi.org/10.1090/s0025-5718-07-01951-5>.
- [18] C. CANUTO, *Stabilization of spectral methods by finite element bubble functions*, Comput. Methods Appl. Mech. Engrg., 116 (1994), pp. 13–26, [https://doi.org/10.1016/s0045-7825\(94\)80004-9](https://doi.org/10.1016/s0045-7825(94)80004-9).
- [19] C. CANUTO, P. GERVASIO, AND A. QUARTERONI, *Finite-element preconditioning of G-NI spectral methods*, SIAM J. Sci. Comput., 31 (2010), pp. 4422–4451, <https://doi.org/10.1137/090746367>.
- [20] C. CANUTO, M. Y. HUSSAINI, A. QUARTERONI, AND T. A. ZANG, *Spectral Methods*, Springer, Berlin, Heidelberg, 2006, <https://doi.org/10.1007/978-3-540-30726-6>.
- [21] C. CANUTO AND A. QUARTERONI, *Approximation results for orthogonal polynomials in Sobolev spaces*, Math. Comp., 38 (1982), pp. 67–86.
- [22] C. CANUTO AND A. QUARTERONI, *Preconditioned minimal residual methods for Chebyshev spectral calculations*, J. Comput. Phys., 60 (1985), pp. 315–337, [https://doi.org/10.1016/0021-9991\(85\)90010-5](https://doi.org/10.1016/0021-9991(85)90010-5).
- [23] C. CANUTO, A. QUARTERONI, M. Y. HUSSAINI, AND T. A. ZANG, *Spectral Methods*, Springer,

- Berlin, Heidelberg, 2007, <https://doi.org/10.1007/978-3-540-30728-0>.
- [24] N. CHALMERS AND T. WARBURTON, *Low-order preconditioning of high-order triangular finite elements*, SIAM J. Sci. Comput., 40 (2018), pp. A4040–A4059, <https://doi.org/10.1137/17m1149444>.
 - [25] E. CUTHILL AND J. MCKEE, *Reducing the bandwidth of sparse symmetric matrices*, in Proceedings of the 1969 24th National Conference, ACM Press, 1969, pp. 157–172, <https://doi.org/10.1145/800195.805928>.
 - [26] E. F. D’AZEVEDO, P. A. FORSYTH, AND W.-P. TANG, *Ordering methods for preconditioned conjugate gradient methods applied to unstructured grid problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 944–961, <https://doi.org/10.1137/0613057>.
 - [27] E. F. D’AZEVEDO, P. A. FORSYTH, AND W.-P. TANG, *Towards a cost-effective ILU preconditioner with high level fill*, BIT, 32 (1992), pp. 442–463, <https://doi.org/10.1007/bf02074880>.
 - [28] M. O. DEVILLE AND E. H. MUND, *Finite-element preconditioning for pseudospectral solutions of elliptic problems*, SIAM J. Sci. and Stat. Comput., 11 (1990), pp. 311–342, <https://doi.org/10.1137/0911019>.
 - [29] M. DRYJA AND O. B. WIDLUND, *Some domain decomposition algorithms for elliptic problems*, in Iterative Methods for Large Linear Systems, Elsevier, 1990, pp. 273–291, <https://doi.org/10.1016/b978-0-12-407475-0.50022-x>.
 - [30] R. D. FALGOUT AND U. M. YANG, *hypre: A library of high performance preconditioners*, in Computational Science—ICCS 2002, Lecture Notes in Comput. Sci., Springer, Berlin, Heidelberg, 2002, pp. 632–641, https://doi.org/10.1007/3-540-47789-6_66.
 - [31] K. J. FIDKOWSKI, T. A. OLIVER, J. LU, AND D. L. DARMOFAL, *p-multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations*, J. Comput. Phys., 207 (2005), pp. 92–113, <https://doi.org/10.1016/j.jcp.2005.01.005>.
 - [32] P. F. FISCHER, *An overlapping Schwarz method for spectral element solution of the incompressible Navier-Stokes equations*, J. Comput. Phys., 133 (1997), pp. 84–101, <https://doi.org/10.1006/jcph.1997.5651>.
 - [33] P. F. FISCHER AND J. W. LOTTES, *Hybrid Schwarz-multigrid methods for the spectral element method: Extensions to Navier-Stokes*, in Domain Decomposition Methods in Science and Engineering, Springer, 2005, pp. 35–49, https://doi.org/10.1007/3-540-26825-1_3.
 - [34] D. FORTUNATO, C. H. RYCROFT, AND R. SAYE, *Efficient operator-coarsening multigrid schemes for local discontinuous Galerkin methods*, SIAM J. Sci. Comput., 41 (2019), pp. A3913–A3937, <https://doi.org/10.1137/18m1206357>.
 - [35] W. HACKBUSCH, *Multi-grid Methods and Applications*, Springer, Berlin, Heidelberg, 1985, <https://doi.org/10.1007/978-3-662-02427-0>.
 - [36] O. HASSAN, E. PROBERT, K. MORGAN, AND J. PERAIRE, *Line relaxation methods for the solution of 2D and 3D compressible flows*, in 11th Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics, 1993, pp. 743–750, <https://doi.org/10.2514/6.1993-3366>.
 - [37] W. HEINRICH, *Line relaxation for spectral multigrid methods*, J. Comput. Phys., 77 (1988), pp. 166–182, [https://doi.org/10.1016/0021-9991\(88\)90161-1](https://doi.org/10.1016/0021-9991(88)90161-1).
 - [38] B. HELENBROOK, D. MAVRIPLIS, AND H. ATKINS, *Analysis of “p”-multigrid for continuous and discontinuous finite element discretizations*, in Proceedings of the 16th AIAA Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics, 2003, <https://doi.org/10.2514/6.2003-3989>.
 - [39] B. HELENBROOK AND H. L. ATKINS, *Application of p-multigrid to discontinuous Galerkin formulations of the Poisson equation*, AIAA J., 44 (2006), pp. 566–575, <https://doi.org/10.2514/1.15497>.
 - [40] V. E. HENSON AND U. M. YANG, *BoomerAMG: A parallel algebraic multigrid solver and preconditioner*, Appl. Numer. Math., 41 (2002), pp. 155–177, [https://doi.org/10.1016/s0168-9274\(01\)00115-5](https://doi.org/10.1016/s0168-9274(01)00115-5).
 - [41] A. KLÖCKNER, T. WARBURTON, J. BRIDGE, AND J. HESTHAVEN, *Nodal discontinuous Galerkin methods on graphics processors*, J. Comput. Phys., 228 (2009), pp. 7863–7882, <https://doi.org/10.1016/j.jcp.2009.06.041>.
 - [42] M. KRONBICHLER AND K. LJUNGKVIST, *Multigrid for matrix-free high-order finite element computations on graphics processors*, ACM Trans. Parallel Comput., 6 (2019), 2, <https://doi.org/10.1145/3322813>.
 - [43] P. L. LIONS, *On the Schwarz alternating method. I*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. Golub, G. Meurant, and J. Périaux, eds., SIAM, Philadelphia, 1988, pp. 1–42.
 - [44] J. W. LOTTES AND P. F. FISCHER, *Hybrid multigrid/Schwarz algorithms for the spectral element method*, J. Sci. Comput., 24 (2005), pp. 45–78, <https://doi.org/10.1007/>

s10915-004-4787-3.

- [45] D. J. MAVRIPLIS, *Multigrid strategies for viscous flow solvers on anisotropic unstructured meshes*, J. Comput. Phys., 145 (1998), pp. 141–165.
- [46] J. MELENK, K. GERDES, AND C. SCHWAB, *Fully discrete hp-finite elements: Fast quadrature*, Comput. Methods Appl. Mech. Engrg., 190 (2001), pp. 4339–4364, [https://doi.org/10.1016/s0045-7825\(00\)00322-4](https://doi.org/10.1016/s0045-7825(00)00322-4).
- [47] S. MÜTHING, M. PIATKOWSKI, AND P. BASTIAN, *High-performance Implementation of Matrix-free High-order Discontinuous Galerkin methods*, preprint, <https://arxiv.org/abs/1711.10885>, 2017.
- [48] L. N. OLSON AND J. B. SCHRODER, *Smoothed aggregation multigrid solvers for high-order discontinuous Galerkin methods for elliptic problems*, J. Comput. Phys., 230 (2011), pp. 6959–6976, <https://doi.org/10.1016/j.jcp.2011.05.009>.
- [49] L. N. OLSON AND J. B. SCHRODER, *PyAMG: Algebraic multigrid solvers in Python v4.0*, Release 4.0, 2018, <https://github.com/pyamg/pyamg>.
- [50] S. A. ORSZAG, *Spectral methods for problems in complex geometries*, J. Comput. Phys., 37 (1980), pp. 70–92, [https://doi.org/10.1016/0021-9991\(80\)90005-4](https://doi.org/10.1016/0021-9991(80)90005-4).
- [51] L. F. PAVARINO, *Additive Schwarz methods for the p-version finite element method*, Numer. Math., 66 (1993), pp. 493–515, <https://doi.org/10.1007/bf01385709>.
- [52] W. PAZNER AND P.-O. PERSSON, *Approximate tensor-product preconditioners for very high order discontinuous Galerkin methods*, J. Comput. Phys., 354 (2018), pp. 344–369, <https://doi.org/10.1016/j.jcp.2017.10.030>.
- [53] W. PAZNER AND P.-O. PERSSON, *Interior penalty tensor-product preconditioners for high-order discontinuous Galerkin discretizations*, in 2018 AIAA Aerospace Sciences Meeting, American Institute of Aeronautics and Astronautics, 2018, <https://doi.org/10.2514/6.2018-1093>.
- [54] P.-O. PERSSON AND J. PERAIRE, *Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-Stokes equations*, SIAM J. Sci. Comput., 30 (2008), pp. 2709–2733, <https://doi.org/10.1137/070692108>.
- [55] J. W. RUGE AND K. STÜBEN, *Algebraic multigrid*, in Multigrid Methods, S. F. McCormick, ed., SIAM, Philadelphia, 1987, pp. 73–130, <https://doi.org/10.1137/1.9781611971057.ch4>.
- [56] K. SHAHBAZI, *An explicit expression for the penalty parameter of the interior penalty method*, J. Comput. Phys., 205 (2005), pp. 401–407, <https://doi.org/10.1016/j.jcp.2004.11.017>.
- [57] J. SHEN, F. WANG, AND J. XU, *A finite element multigrid preconditioner for Chebyshev-collocation methods*, Appl. Numer. Math., 33 (2000), pp. 471–477, [https://doi.org/10.1016/s0168-9274\(99\)00114-2](https://doi.org/10.1016/s0168-9274(99)00114-2).
- [58] J. SLOTNICK, A. KHODADOUR, J. ALONSO, D. L. DARMOFAL, W. GROPP, E. LURIE, AND D. J. MAVRIPLIS, *CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences*, Tech. Report NASA-CR-2014-218178, NASA, Langley, VA, 2014.
- [59] A. TOSELLI AND O. B. WIDLUND, *Domain Decomposition Methods — Algorithms and Theory*, Springer, Berlin, Heidelberg, 2005, <https://doi.org/10.1007/b137868>.
- [60] B. VERMEIRE, F. WITHERDEN, AND P. VINCENT, *On the utility of GPU accelerated high-order methods for unsteady flow simulations: A comparison with industry-standard tools*, J. Comput. Phys., 334 (2017), pp. 497–521, <https://doi.org/10.1016/j.jcp.2016.12.049>.
- [61] Z. WANG, K. FIDKOWSKI, R. ABGRALL, F. BASSI, D. CARAENI, A. CARY, H. DECONINCK, R. HARTMANN, K. HILLEWAERT, H. HUYNH, N. KROLL, G. MAY, P.-O. PERSSON, B. VAN LEER, AND M. VISBAL, *High-order CFD methods: Current status and perspective*, Internat. J. Numer. Methods Fluids, 72 (2013), pp. 811–845, <https://doi.org/10.1002/fld.3767>.
- [62] P. WESSELING, *Introduction to Multigrid Methods*, Tech. Report NASA-CR-195045, Institute for Applied Mathematics and Computer Science, NASA, 1995.
- [63] J. XU, *Iterative methods by space decomposition and subspace correction*, SIAM Rev., 34 (1992), pp. 581–613, <https://doi.org/10.1137/1034116>.
- [64] J. XU AND L. ZIKATANOV, *The method of alternating projections and the method of subspace corrections in Hilbert space*, J. Amer. Math. Soc., 15 (2002), pp. 573–598.