A Scalable Preconditioner for a Primal DPG Method

Andrew T. Barker
*Lawrence Livermore National Laboratory*

Veselin A. Dobrev
*Lawrence Livermore National Laboratory*

Jay Gopalakrishnan
*Portland State University, gjay@pdx.edu*

Tzanio Kolev
*Lawrence Livermore National Laboratory*

Citation Details
A SCALEABLE PRECONDITIONER FOR A PRIMAL DPG METHOD∗

A. T. BARKER†, V. DOBREV†, J. GOPALAKRISHNAN‡, AND T. KOLEV†

Abstract. We show how a scalable preconditioner for the primal discontinuous Petrov-Galerkin (DPG) method can be developed using existing algebraic multigrid (AMG) preconditioning techniques. The stability of the DPG method gives a norm equivalence which allows us to exploit existing AMG algorithms and software. We show how these algebraic preconditioners can be applied directly to a Schur complement system arising from the DPG method. One of our intermediate results shows that a generic stable decomposition implies a stable decomposition for the Schur complement. This justifies the application of algebraic solvers directly to the interface degrees of freedom. Combining such results, we obtain the first massively scalable algebraic preconditioner for the DPG system.

Key words. algebraic multigrid, BoomerAMG, ADS, Schur complement, Discontinuous Petrov-Galerkin.

AMS subject classifications. 65F10, 65M55, 65N30.

1. Introduction. Discontinuous Petrov-Galerkin (DPG) methods, introduced in [12, 13], constructed test spaces that guarantee stability. Today these methods are known to be simultaneously viewable as Galerkin mixed methods, as least-squares methods in nonstandard norms, or as Petrov-Galerkin methods using discontinuous functions [16]. DPG methods have a great deal of flexibility, allowing them to be applied to a wide variety of problems [10, 14, 17], and their convergence theory has now matured [9].

However, there is a lack of fast scalable solvers for the DPG method. In [3], an overlapping Schwarz preconditioner is analyzed: because it has no coarse level, the preconditioner expectedly deteriorates as overlap size become small. A coarse level was added for improved scalability in [25], where the authors analyzed a two-level additive Schwarz preconditioner for an ultraweak DPG method applied to the Poisson equation with Robin boundary condition. Going beyond the Poisson problem to the harmonic wave equation, there are numerical reports of good performance of certain preconditioning strategies [20].

It was clear from the inception of the DPG method that certain norm equivalences stemming from its stability give a natural avenue for the design of preconditioners, as was presented abstractly in [29]. We pursue the same avenue, but investigate all the details necessary to prove optimality. For example, the norm in which the DPG method is stable involves an infimum over an infinite dimensional space. Its equivalence with a finite dimensional infimum arising from a discrete Schur complement is an essential ingredient in our preconditioner analysis, an equivalence not proved in [29]. Another example of an observation immediate from our analysis (but not clear from [29]) is that the interface variable in the DPG method admits a preconditioner which uses only the interface degrees of freedom. In the recent work of [28] geometric multilevel strategies are investigated numerically but without theoretical analysis. In this paper we show how existing algebraic multilevel preconditioners can
be effectively combined to precondition a DPG system, including at very large scale on parallel supercomputers.

The particular DPG method we consider is the so-called primal DPG method [15], reviewed in the next section. After describing a basic norm equivalence associated with the method, we proceed to analyze one of the component norms in Section 3. We show that this interface norm, obtained as an infimum over an infinite-dimensional space, is equivalent to an infimum over a finite-dimensional space. An auxiliary algebraic Schur complement result is presented in Section 4. Section 5 identifies the finite-dimensional infimum as a Schur complement norm and proceeds to analyze an auxiliary-space preconditioner for the Schur complement. The preconditioner and the main result are summarized in Section 6. Section 7 reports results from numerical studies of the proposed preconditioner. We conclude by summarizing the main results from the paper in Section 8.

2. The primal DPG method. For completeness and consistency of notation, we recall some definitions and results from the work introducing the primal DPG system [15]. The model problem we consider is the Poisson problem of finding \( u \in H^1_0(\Omega) \) such that

\[
\int_\Omega \kappa \nabla u \cdot \nabla v \, dx = \int_\Omega f v \, dx \tag{2.1}
\]

for all \( v \in H^1_0(\Omega) \), where \( \Omega \) is a polygonal (in \( \mathbb{R}^2 \)) or polyhedral domain (in \( \mathbb{R}^3 \)) with Lipschitz boundary and \( \kappa > 0 \) is a piecewise constant coefficient. Zero Dirichlet boundary conditions on \( \partial \Omega \) are essentially imposed in (2.1). In practice the method can also handle more general problems with varying coefficients and different boundary conditions, but we choose this setting for simplicity.

Even before discretization, the DPG formulation uses a mesh-dependent weak form. We assume that \( \Omega \) is given together with a mesh \( \Omega_h \) that partitions \( \Omega \) into elements of varying shapes. For example, in the \( \mathbb{R}^2 \) case, the mesh elements \( K \in \Omega_h \) may be triangles or quadrilaterals, while in \( \mathbb{R}^3 \) the elements may be tetrahedra, prisms, hexahedra, etc. Precise assumptions on the mesh and element shapes will be specified later, but for now we only require that the boundary of each element be Lipschitz, so that traces of Sobolev space functions on the element boundaries are well-defined. Specifically, we require a well-defined normal trace operator

\[
\text{trc}_n : H(\text{div}, \Omega) \to \prod_{K \in \Omega_h} H^{-1/2}(\partial K),
\]

which maps to element-wise traces \( (\text{trc}_n q)|_{\partial K} = q \cdot n|_{\partial K} \). One can also define \( \text{trc}_n q \) as a single valued function on the mesh facets: Indeed, if each interface facet \( \gamma = \bar{K}_1 \cap \bar{K}_2 \) between two elements \( K_1, K_2 \in \Omega_h \), as well as boundary facets \( \gamma = \bar{K} \cap \partial \Omega \) are Lipschitz, then we may fix a continuous unit normal vector function \( n_\gamma \) on \( \gamma \) and define \( (\text{trc}_n q)|_{\gamma} = n_\gamma \cdot q|_{\gamma} \). This is a well-defined function in the dual space of \( H^{1/2}(\gamma) \) (denoted also by \( H^{1/2}_{\partial \Omega}(\gamma) \)), whenever \( q \in H(\text{div}, \Omega) \).

The numerical fluxes of the DPG method lie in the range of \( \text{trc}_n \), i.e., in the space \( Q = \text{ran}(\text{trc}_n) \) with norm given by

\[
\|q\|_Q = \inf_{\tau \in \text{trc}_n^{-1}\{q\}} \|\tau\|_{H(\text{div}, \Omega)}. \tag{2.2}
\]

Here, as usual, \( \text{trc}_n^{-1}\{q\} \) denotes the pre-image of the singleton \( \{q\} \). It is standard to prove that the minimal extension operator \( E : Q \to H(\text{div}, \Omega) \), defined by \( \text{trc}_n(Eq) = q \),
and set

\[ u \in F \] with uniquely solvable for \((\cdot,\cdot)_{W}\), to denote its norm and inner product, respectively. When the space is uniquely understood from the argument or other context, we will drop the subscript. Let

\[ H^1(\Omega_h) = \prod_{K \in \Omega_h} H^1(K). \]

This product space is endowed with the standard Cartesian product norm and inner product. For brevity, put \( X = H^1_0(\Omega) \times Q \) and \( Y = H^1(\Omega_h) \). Define the bilinear form \( b : X \times Y \to \mathbb{R} \) by

\[ b((w,r),v) = \sum_{K \in \Omega_h} \left( \int_K \kappa \nabla w \cdot \nabla v \, dx + \langle (r,v)_{H^{-1/2}(\partial K)} \right) \]

where \( (r,v)_{H^{-1/2}(\partial K)} \) denotes the duality pairing between \( H^{-1/2}(\partial K) \) and \( H^{1/2}(\partial K) \). The Dirichlet problem \((2.1)\) can then be reformulated [15] as the problem of finding a pair \((u,q) \in X\) satisfying

\[ b((u,q),v) = F(v), \quad \forall v \in Y, \quad (2.4) \]

with \( F(v) = (f,v)_{L^2(\Omega)} \). It is proved in [15, Lemma 3.4] that the problem \((2.4)\) is uniquely solvable for \((u,q) \in X\) given any \( F \) in the dual space of \( Y \) (see also [9, Example 3.6] for a simplified analysis).

The primal DPG method uses the formulation \((2.4)\) and finite element subspaces \( U_h \subset H^1_0(\Omega) \), \( Q_h \subset Q \), and \( Y_h \subset H^1(\Omega_h) \). Here \( h = \max_{K \in \Omega_h} \text{diam}(K) \) denotes the mesh size of \( \Omega_h \). To describe a computational version of the method, let \( \{u_i\}, \{q_j\}, \{v_k\} \) denote a finite element basis for \( U_h, Q_h \) and \( Y_h \), respectively. Define the matrices

\[
[B_0]_{kj} = b((u_i,0),v_k), \quad [B_1]_{kj} = b((0,q_j),v_k), \\
[M]_{kl} = (v_l,v_k)_Y \equiv (v_l,v_k)_{L^2(\Omega)} + (\kappa \nabla v_l, \nabla v_k)_{L^2(\Omega)}
\]

and set

\[ B = [B_0 \ B_1], \quad A = B^T M^{-1} B = \begin{bmatrix} B_0^T M^{-1} B_0 & B_0^T M^{-1} B_1 \\ B_1^T M^{-1} B_0 & B_1^T M^{-1} B_1 \end{bmatrix}. \]

Let \( u \) denote the vector in \( \mathbb{R}^{\dim(U_h)} \) representing a function \( u \in U_h \) by the basis expansion formula \( u = \sum_i [u]_i u_i \). The vectors \( q \) in \( \mathbb{R}^{\dim(Q_h)} \) and \( v \) in \( \mathbb{R}^{\dim(Y_h)} \) are similarly defined. Restricting \((2.4)\) to the finite dimensional spaces formally gives

\[ B \begin{bmatrix} u \\ q \end{bmatrix} = F \]

where \([F]_k = F(v_k)\). The DPG discretization of \((2.4)\) solves instead the following symmetric and positive definite problem for \( u \) and \( q \):

\[ A \begin{bmatrix} u \\ q \end{bmatrix} = g \quad (2.5) \]
where \( g = B^T M^{-1} F \). Note that \( M \) is the Gram matrix in the (weighted) “broken” \( H^1(\Omega_h) \)-inner product and so is block diagonal (one block per element). Thus, \( M^{-1} \) can be evaluated fast locally.

The DPG method admits three well-known interpretations. The early papers on the DPG method used the concept of optimal test functions [12]. Its interpretation as a least-squares method in a nonstandard inner product was pointed out in [13, p. 6]. Its interpretation as a mixed method is now well known (see e.g., [6, Theorem 2.4]). It is easy to see that all these three interpretations, in practice, yield the same matrix system (2.5) when the same spaces and bases are used.

The starting point of our analysis is the stability of the DPG method (2.5). Let \( X_h = U_h \times Q_h \) and let \( x \) and \( z \) be vectors representing two functions \( x \) and \( z \) in \( X_h \), respectively. Per the above-mentioned notational conventions, \((Ax, z)\) denotes the Euclidean inner product \( z^T Ax \). It defines a bilinear form in the function space \( X_h \), namely \( a(x, z) = (Ax, z) \). Note that both \( X_h \) and \( Y_h \) are used in the definition of \( A \).

Throughout this paper we assume that the mesh \( \Omega \) and the spaces \( X_h \) and \( Y_h \) are such that there exist mesh-independent constants \( c_1 \) and \( c_2 \) satisfying

\[
c_1\|x\|_X^2 \leq a(x, x) \leq c_2\|x\|_X^2, \tag{2.6}
\]

for all \( x \in X_h \). The connection between (2.6) and the stability of the method is described next.

**Proposition 2.1.** Assumption (2.6) holds if and only if

\[
c_1\|x\|_X \leq \sup_{0 \neq v \in Y_h} \frac{|b(x, v)|}{\|v\|_Y} \leq c_2\|x\|_X \tag{2.7}
\]

for all \( x \) in \( X_h \).

**Proof.** Define \( T_h : X_h \to Y_h \) by \((T_h x, y)_Y = b(x, y)\), for all \( x \in X_h \) and \( y \in Y_h \). Then, for any \( x \in X_h \),

\[
\|T_h x\|_Y^2 = \sup_{0 \neq v \in Y_h} \frac{(T_h x, v)_Y}{\|v\|_Y} = \sup_{0 \neq v \in Y_h} \frac{b(x, v)}{\|v\|_Y}.
\]

Letting \( x \) and \( v \) denote the vector representations of \( x \in X_h \) and \( v = T_h x \in Y_h \), respectively, it is easy to see that \( v = M^{-1} B x \). Hence the result follows from \( a(x, x) = (MM^{-1}Bx, MM^{-1}Bx) = \|T_h x\|_Y^2 \).

Clearly, the upper inequality of (2.7) follows from the continuity of the bilinear form \( b(\cdot, \cdot) \), and therefore holds independently of the choice of the discrete spaces. The lower inequality of (2.7) is an inf-sup condition. It follows from the fact that (2.4) is well-posed whenever the discrete spaces are chosen so that a Fortin operator [19] can be constructed. Here are a few known examples of cases where a \( c_1 \) independent of \( h \) can be obtained for the Dirichlet problem under consideration:

1. Suppose the mesh \( \Omega_h \) is a quasiuniform tetrahedral geometrically conforming mesh, \( U_h \) is the Lagrange finite element space of degree \( p \), \( Q_h = \{ q : q|_\gamma \) is a polynomial of degree at most \( p - 1 \) on each mesh facet \( \gamma \} \), and \( Y_h = \{ v : v|_K \) is a polynomial of degree at most \( p + 2 \} \). Then a Fortin operator provided in [19] yields a \( c_1 \) independent of \( h \), as proved in [15].

2. When \( \Omega_h \) is a uniform mesh of rectangular elements, \( U_h = \{ w \in H^1_0(\Omega) : w|_K \) is in the tensor product space of polynomials of degree at most \( p \) in each coordinate direction, for all elements \( K \in \Omega_h \}, Q_h = \{ q : q|_\gamma \in P_p(\gamma) \) on
each mesh facet $\gamma$, and $Y_h = \{ v : v|_K \text{ is a polynomial of degree at most } p + 3 \text{ in each coordinate direction} \}$, a Fortin operator in [8] gives a mesh-independent $c_1$.

In the remainder of this paper, we examine an important implication of (2.6). Namely, in order to precondition the large Hermitian positive definite DPG system (2.5), it suffices to obtain a preconditioner for the $\| \cdot \|_X$ norm. In our model problem, this norm is

$$\| x \|^2_X = \| u \|^2_{H^1(\Omega)} + \| q \|^2_{Q},$$

for any $x = (u, q) \in U_h \times Q_h$, so it suffices to combine preconditioners for the $H^1(\Omega)$ and $Q$ norms. Since the former is standard, we focus on the latter in the next section.

3. Characterizing the $Q$-norm. The $Q$-norm (2.2) is defined through a minimization over an infinite dimensional space (the minimal extension $E$ in (2.3) is not computable). In this section, we relate this norm to a minimum over a finite dimensional subspace.

3.1. Tetrahedral case. To present the idea transparently, we first detail the case when $\Omega_h$ is a geometrically conforming mesh of tetrahedral elements. For any tetrahedron $K$, let $R_p(K) = P_p(K)^3 + xP_p(K)$, where $x$ is the coordinate vector and $P_p(K)$ denotes the set of all polynomials of total degree at most $p \geq 1$. The Raviart-Thomas finite element space is $R_h = \{ r \in H(\text{div}, \Omega) : r|_K \in R_p(K) \text{ for all } K \in \Omega_h \}$. Let $Q_h = \text{trc}_n(R_h)$. Clearly $Q_h$ is a finite dimensional subspace of $Q$.

Define $E_h : Q_h \to R_h$ by $\text{trc}_n(E_h q) = q$, and

$$(E_h q, v)_{H(\text{div}, K)} = 0, \quad \forall v \in R_p(K) \cap \hat{H}(\text{div}, K),$$

for all $K \in \Omega_h$. This computable approximation of the minimal extension operator defines a new norm on $Q_h$,

$$\| q \|_{Q_h} = \| E_h q \|_{H(\text{div}, \Omega)}.$$

We now proceed to prove the equivalence of this norm with the $Q$-norm (in Theorem 3.3 below). Throughout this section, let $c$ denote a generic positive constant whose value might change from one occurrence to another, but will remain independent of $h$ and $p$. Let $\hat{K}$ denote the unit tetrahedron, $\hat{n}$ denotes its outward unit normal on $\partial \hat{K}$, and $\bar{\sigma}_n = \hat{\sigma} \cdot \hat{n}|_{\partial \hat{K}}$. Let $Q_p(\partial \hat{K}) = \{ \bar{\sigma}_n : \bar{\sigma} \in R_p(\hat{K}) \}$. For any $\bar{\sigma} \in R_p(\hat{K})$ define the constant function

$$\bar{\sigma}_n = \frac{1}{|\partial \hat{K}|} \int_{\partial \hat{K}} \bar{\sigma}_n \ ds,$$

where $|\partial \hat{K}|$ denotes the surface area of $\partial \hat{K}$.

**Lemma 3.1.** There is a $c > 0$ and a $\hat{\mathcal{G}} : \mathbb{R} \to R_p(\hat{K})$ such that for any $\bar{\sigma}$ in $H(\text{div}, \hat{K})$ with $\bar{\sigma}_n \in Q_p(\partial \hat{K})$, we have $\hat{n} \cdot (\hat{\mathcal{G}} \bar{\sigma}_n)|_{\partial \hat{K}} = \bar{\sigma}_n$,

$$\| \hat{\mathcal{G}} \bar{\sigma}_n \|_{L^2(\hat{K})} \leq c \| \bar{\sigma} \|_{H(\text{div}, \hat{K})}, \quad \text{and} \quad \| \text{div} \hat{\mathcal{G}} \bar{\sigma}_n \|_{L^2(\hat{K})} \leq c \| \text{div} \bar{\sigma} \|_{\hat{K}}.$$

Proof. We show that the following stronger inequality holds on the unit tetrahedron:

$$\|\hat{G}\hat{\sigma}_n\|_{H(\text{div}, \bar{K})} \leq c \|\text{div} \hat{\sigma}\|_{\bar{K}}.$$  

Let $\hat{x}_I$ denote the incenter of the unit tetrahedron $\bar{K}$. Then $(\hat{x} - \hat{x}_I) \cdot \hat{n} = d$ is constant for any $\hat{x} \in \partial\bar{K}$ ($d = 3|\bar{K}|/|\partial\bar{K}|$ is the radius of the insphere). Define

$$\hat{G}\hat{\sigma}_n = \frac{\hat{n}}{d}(\hat{x} - \hat{x}_I).$$

Then, setting $\hat{c} = \|\hat{G}1\|_{H(\text{div}, \bar{K})}/|\partial\bar{K}|$ and $c = \hat{c}|\bar{K}|^{1/2}$, we have

$$\|\hat{G}\hat{\sigma}_n\|_{H(\text{div}, \bar{K})}^2 = \hat{c}^2 \left| \int_{\partial\bar{K}} \hat{\sigma}_n \, ds \right|^2 = \hat{c}^2 \left| \int_{\bar{K}} \text{div} \hat{\sigma} \, dx \right|^2 \leq \hat{c}^2 \|\text{div} \hat{\sigma}\|_{\bar{K}}^2$$

and $\hat{G}\hat{\sigma}_n = \hat{\sigma}_n(\hat{x} - \hat{x}_I) \cdot \hat{n}/d = \hat{\sigma}_n$, for all $\hat{x} \in \partial\bar{K}$. \[\square\]

**Lemma 3.2.** There is a $c > 0$ and a $\hat{E} : Q_p(\partial\bar{K}) \to R_p(\bar{K})$ such that for any $\hat{\sigma}$ in $H(\text{div}, \bar{K})$ with $\hat{\sigma}_n \in Q_p(\partial\bar{K})$, we have $\hat{n} \cdot (\hat{E}\hat{\sigma}_n)|_{\partial\bar{K}} = \hat{\sigma}_n$,

$$\|\hat{E}(\hat{\sigma}_n - \hat{\sigma}_n)\|_{L^2(\bar{K})} \leq c \|\hat{\sigma}\|_{H(\text{div}, \bar{K})}, \quad \text{and} \quad \text{div}(\hat{E}(\hat{\sigma}_n - \hat{\sigma}_n)) = 0.$$

**Proof.** We use the polynomial extension operator $\mathcal{E}^{\text{div}}$ from [18, Theorem 7.1]: Accordingly (a) if $\hat{q} \in Q_p(\partial\bar{K})$, then $\mathcal{E}^{\text{div}}\hat{q}$ is in $R_p(\bar{K})$, (b) if $\hat{q}$ has zero mean, then $\text{div}(\mathcal{E}^{\text{div}}\hat{q}) = 0$, and (c) if $\hat{\tau}$ is any extension of $\hat{q}$ (i.e., $\hat{\tau}$ is a function in $H(\text{div}, \bar{K})$ satisfying $\hat{n} \cdot \hat{\tau}|_{\partial\bar{K}} = \hat{q}$), then

$$\|\mathcal{E}^{\text{div}}\hat{q}\|_{H(\text{div}, \bar{K})} \leq c \|\hat{\tau}\|_{H(\text{div}, \bar{K})}.$$  

Since $\hat{\tau}$ is an extension of $\hat{\sigma}_n$ and $\hat{G}\hat{\sigma}_n$ is an extension of $\hat{\sigma}_n$,

$$\|\mathcal{E}^{\text{div}}(\hat{\sigma}_n - \hat{\sigma}_n)\|_{H(\text{div}, \bar{K})} \leq \|\mathcal{E}^{\text{div}}\hat{\sigma}_n\|_{H(\text{div}, \bar{K})} + \|\mathcal{E}^{\text{div}}\hat{\sigma}_n\|_{H(\text{div}, \bar{K})} \leq c \left( \|\hat{\sigma}\|_{H(\text{div}, \bar{K})} + \|\hat{G}\hat{\sigma}_n\|_{H(\text{div}, \bar{K})} \right) \leq c \|\hat{\sigma}\|_{H(\text{div}, \bar{K})}.$$  

Finally, since $\hat{q} = \hat{\sigma}_n - \hat{\sigma}_n$ has zero mean, we have $\text{div}(\mathcal{E}^{\text{div}}(\hat{\sigma}_n - \hat{\sigma}_n)) = 0$. \[\square\]

Before the next result, recall that for any tetrahedral element $K$, there is an affine homeomorphism $\Phi : \bar{K} \to K$. Let $[D\Phi]$ denote the Jacobian matrix of its derivatives and let $J = \text{det}[D\Phi]$. Define the Piola maps

$$\Phi_*\hat{\sigma} = J^{-1}[D\Phi_K]\hat{\sigma} \circ \Phi^{-1}, \quad \Phi^*\sigma = J[D\Phi_K]^{-1}\sigma \circ \Phi.$$  

Clearly $\Phi_*$ maps functions on $\bar{K}$ to $K$, while $\Phi^*$ maps in the opposite direction, from $K$ to $\bar{K}$. Letting $|K|$ denote the volume of the $K$ and $h_K = \text{diam}(K)$, we recall the following standard estimates [5] for affine $\Phi$: There is a $c > 0$, depending only on shape regularity of $K$, such that

$$\|\Phi_*\hat{\sigma}\|_{L^2(K)}^2 \leq c \frac{|K|^2}{h_K^2} \|\hat{\sigma}\|_{L^2(\bar{K})}^2, \quad \|\text{div}(\Phi_*\hat{\sigma})\|_{L^2(K)}^2 \leq c \frac{|K|}{|\bar{K}|} \|\text{div}\hat{\sigma}\|_{L^2(\bar{K})}^2,$$

$$\|\Phi^*\sigma\|_{L^2(K)}^2 \leq c \frac{|K|^2}{|\bar{K}|^2} \|\sigma\|_{L^2(\bar{K})}^2, \quad \|\text{div}(\Phi^*\sigma)\|_{L^2(K)}^2 \leq c \frac{|\bar{K}|}{|K|} \|\text{div}\sigma\|_{L^2(\bar{K})}^2.$$  

(3.2a)
for all \( \hat{\sigma} \in H(\text{div}, K) \) and \( \sigma \in H(\text{div}, K) \).

We are now in a position to put everything together and prove the main norm equivalence result of this section.

**Theorem 3.3.** If \( \Omega_h \) is shape regular, then there is a \( c_3 > 0 \) independent of \( h \) and \( p \) (and depending only on the shape regularity) such that

\[
\|q\|_Q \leq \|q\|_{Q_h} \leq c_3 \|q\|_Q
\]

(3.3)

for all \( q \in Q_h \).

**Proof.** The lower inequality follows from

\[
\|q\|_Q = \inf_{\tau \in \text{trc}_n^{-1}(\{q\})} \|\tau\|_{H(\text{div}, \Omega)} \leq \inf_{\tau_h \in \mathcal{R}_h \cap \text{trc}_n^{-1}(\{q\})} \|\tau_h\|_{H(\text{div}, \Omega)} = \|q\|_{Q_h}.
\]

To prove the upper inequality, pick any \( K \in \Omega_h \), set

\[
\sigma = (Eq)_K, \quad \hat{\sigma} = \Phi^* \sigma, \quad F_K q = \Phi_* \hat{\sigma}_n,
\]

where \( E \) is the minimal extension in (2.3),

\[
\hat{\sigma}_n = \hat{G} \sigma_n + \hat{E} (\hat{\sigma}_n - \bar{\sigma}_n),
\]

and \( \hat{G} \) and \( \hat{E} \) are as given by Lemmas 3.1 and 3.2. Clearly, \( n \cdot (F_K q)|_{\partial K} = q \) and the function \( F q \), defined by \( (F q)|_K = F_K q \) for all \( K \in \Omega_h \), is in \( H(\text{div}, \Omega) \). Moreover the estimates of Lemmas 3.1 and 3.2, together with (3.2), imply

\[
\|F q\|^2_{L^2(K)} \leq \frac{c}{|K|} \left( \|\hat{G} \sigma_n\|^2_{L^2(\hat{K})} + \|\hat{E} (\hat{\sigma}_n - \bar{\sigma}_n)\|^2_{L^2(\hat{K})} \right) \leq \frac{c}{|K|} \|\hat{\sigma}\|^2_{H(\text{div}, \hat{K})}
\]

\[
= \frac{c}{|K|} \|\Phi^* \sigma\|^2_{H(\text{div}, K)} \leq \frac{c}{|K|} \left( \|\sigma\|^2_{L^2(\hat{K})} + h_K^2 \|\text{div} \sigma\|^2_{L^2(\hat{K})} \right),
\]

\[
\|\text{div} F q\|^2_{L^2(K)} \leq \frac{c}{|K|} \|\text{div} (\hat{G} \sigma_n)\|^2_{L^2(\hat{K})} \leq \frac{c}{|K|} \|\text{div} \hat{\sigma}\|^2_{L^2(\hat{K})} = \frac{c}{|K|} \|\text{div} (\Phi^* \sigma)\|^2_{L^2(\hat{K})}
\]

We have thus obtained, for any \( q \in Q_h \), an extension \( F q \in \mathcal{R}_h \) satisfying

\[
\|F q\|_{H(\text{div}, \Omega)} \leq c \|\sigma\|_{H(\text{div}, \Omega)} = c \|Eq\|_{H(\text{div}, \Omega)} = c \|q\|_Q.
\]

Since \( \|q\|_{Q_h} \) is the infimum of \( \|\tau_h\|_{H(\text{div}, \Omega)} \) over all \( \tau_h \in \mathcal{R}_h \) satisfying \( \text{trc}_n \tau_h = q \), the inequality \( \|q\|_{Q_h} \leq \|F q\|_{H(\text{div}, \Omega)} \) holds and completes the proof. \( \square \)

**3.2. General meshes.** We now briefly remark on how the norm equivalence of Theorem 3.3 may be extended to more general elements and meshes. While a general theorem for all element shapes is beyond the scope of this paper, we wish to provide pointers on what arguments need extension. The proof of Theorem 3.3 depends on three ingredients: (a) Lemma 3.1, (b) Lemma 3.2, and (c) the scaling estimates (3.2). Moving from tetrahedral to other element shapes, we must first obtain generalizations of the extension operators of Lemmas 3.1 and 3.2 on the reference element \( \hat{K} \) for the new shapes. We show how this can be done for two other element shapes, one in two dimensions and another in three dimensions.

**Triangles:** The extension \( \hat{G} \) constructed in the proof of Lemma 3.1 continues to work for the unit triangle if we set \( \hat{x}_I \) to be the center of the inscribed circle of the
triangle. As for Lemma 3.2, if $\hat{K}$ is a triangle, then the extension of [1, Corollary 2.2] has all the properties stated in the lemma.

Cubes: To obtain the result of Lemma 3.1 when $\hat{K}$ is the unit cube, we set $\hat{x}_I=(1/2,1/2,1/2)$ and $\hat{G}\hat{\sigma}_n=2(\hat{x}-\hat{x}_I)\hat{\sigma}_n$. Then proceeding as in (3.1), we obtain the result. The extension operators constructed in [11] for each $p$ provide the required $\hat{E}$ in Lemma 3.2 when $\hat{K}$ is a cube.

The scaling estimates (3.2) are valid for affine mappings $\Phi$. We next comment on meshes with curved elements, which are images of reference elements under a possibly nonlinear $\Phi$. If $\Phi$ is such that the estimates of (3.2) with a properly (re)defined $h_K$ and $|K|$ for curvilinear elements $K$ hold, then the proof of Theorem 3.3 can be generalized. Examples of nonlinear $\Phi$ where such geometrical quantities can be identified can be found in [4, 5].

4. An algebraic Schur complement result. The purpose of this section, which can be read independently of the rest of the paper, is to present a simple matrix result, whose relevance to our problem will be clear in the next section. The result is a generalization of [7, Lemma 4.2]. Suppose $i\cup f=\{1,2,\ldots,m\}$ and $j\cup e=\{1,2,\ldots,l\}$ are disjoint partitions of two index sets. Let $D$ be an $m \times m$ symmetric positive definite matrix and $H$ be an $m \times l$ matrix (both with real entries). We use standard block notations, e.g., $x_f$ denotes the restriction of a vector $x$ to $f$-indices, and the matrices have block forms

$$D = \begin{bmatrix} D_{ii} & D_{if} \\ D_{fi} & D_{ff} \end{bmatrix}, \quad H = \begin{bmatrix} H_{ij} & H_{ie} \\ H_{fj} & H_{fe} \end{bmatrix}. \quad (4.1)$$

Define $S$ to be the Schur complement $S = D_{ff} - D_{fi}D_{ii}^{-1}D_{if}$. Let $\text{diag}(D)$ denote the diagonal matrix formed from the diagonal of $D$.

**Lemma 4.1.** Suppose there is a $c_4 > 0$ such that every $u \in \mathbb{R}^m$ can be decomposed as $u = v + Hr$, for some $v \in \mathbb{R}^m$ and $r \in \mathbb{R}^l$, such that

$$(\text{diag}(D)v, v) + (DHr, Hr) \leq c_4(Du, u). \quad (4.2)$$

Then for any $u \in \mathbb{R}^m$ there exist $v \in \mathbb{R}^m$ and $r \in \mathbb{R}^l$ (not necessarily the same as in the assumption), depending only on $u_f$, such that the decomposition $u_f = v_f + [Hr]_f$ holds and satisfies

$$(\text{diag}(S)v_f, v_f) + (S[Hr]_f, [Hr]_f) \leq c_4(Su_f, u_f).$$

**Proof.** Let $E$ be the matrix representation of the extension operator $E_h$

$$E = \begin{bmatrix} -D_{ii}^{-1}D_{if} \\ I_{ff} \end{bmatrix}. \quad (4.3)$$

Since $S = E^TDE$, from the well-known properties of Schur complements

$$(Sx_f, x_f) = (DEx_f, Ex_f) = \inf_{\{y \in \mathbb{R}^m: y_f = x_f\}} (Dy, y) \leq (Dx, x), \quad \forall x \in \mathbb{R}^m. \quad (4.4)$$

Now, given any $u \in \mathbb{R}^m$, let us set $w = Eu_f$ and let $v \in \mathbb{R}^m$, $r \in \mathbb{R}^l$ be such that $w = v + Hr$ (and in particular $u_f = v_f + [Hr]_f$) and

$$(\text{diag}(D)v, v) + (DHr, Hr) \leq c_4(Dw, w). \quad (4.5)$$
By (4.3), with \( x = Hr \)
\[
(S[hr]f, [hr]f) \leq (Dhr, Hr). \tag{4.5}
\]
Next, consider the \( k \)-th diagonal entry of \( S \) which can be expressed as \( S_{kk} = (Se_k, e_k) \)
where \( e_k \) is the vector with entries \( [e_k]_s = \delta_{ks} \). Setting \( x^T = [0^T e_k^T] \) in (4.3), we get
\[
S_{kk} = (Se_k, e_k) \leq (D_{ff} e_k, e_k) = [D_{ff}]_{kk}.
\]
Since all diagonal entries of \( D \) are positive, we conclude that
\[
(diag(S)v_f, v_f) \leq (diag(D)v_f, v_f) \leq (diag(D)v, v). \tag{4.6}
\]
Adding the estimates (4.5) and (4.6) and then using (4.4) we arrive at
\[
(diag(S)v_f, v_f) + (S[hr]f, [hr]f) \leq (diag(D)v, v) + (Dhr, Hr) \leq c_4(Du, w).
\]
Noting that \((Du, w) = (DEu_f, Eu_f) = (Su_f, u_f)\) completes the proof. \( \square \)

The statement of Lemma 4.1 can be easily extended to the case of more than one matrix \( H \): assume that we have a sequence of real matrices \( H_k \) with dimensions \( m \times l_k, \)
\( k = 1, \ldots, n \).

**Corollary 4.2.** Suppose there is \( c_4 > 0 \) such that for all \( u \in \mathbb{R}^m \) there exist \( v \in \mathbb{R}^m \) and \( r \in \mathbb{R}^{l_k}, \ k = 1, \ldots, n \), such that

\[
u = v + \sum_{k=1}^n H_k r_k, \quad \text{and} \quad (diag(D)v, v) + \sum_{k=1}^n (DH_k r_k, H_k r_k) \leq c_4(Du, u).
\]

Then for any \( u \in \mathbb{R}^m \) there exist \( v \in \mathbb{R}^m \) and \( r \in \mathbb{R}^{l_k}, \ k = 1, \ldots, n \) (not necessarily the same as in the assumption), depending only on \( u_f \), such that

\[
u_f = v_f + \sum_{k=1}^n [H_k r_k]_f, \quad \text{and} \quad (diag(S)v_f, v_f) + \sum_{k=1}^n (S[H_k r_k]_f, [H_k r_k]_f) \leq c_4(Su_f, u_f).
\]

5. **Preconditioning the \( Q_h \)-norm using an interface decomposition.** In Section 3, we reduced the problem of preconditioning \( \| \cdot \|^2_{Q_h} \) to that of preconditioning \( \| \cdot \|^2_{Q_h} \). In this section we propose a scalable method for preconditioning the \( Q_h \)-norm, by further reducing the problem to that of preconditioning the Gram matrix of the \( H(\text{div}, \Omega) \) inner product. Such matrices can be efficiently handled by recent algebraic multigrid techniques [24], resulting ultimately in a good preconditioner for the DPG matrix \( A \), as shown in the next section. We will also show that only computations involving the interface degrees of freedom are needed to precondition the \( Q_h \)-norm.

Let \( \{r_m\} \) denote a finite element basis of \( R_h \). Define \( D \) to be the Gram matrix of the \( H(\text{div}, \Omega) \) inner product in the \( \{r_m\} \) basis. We partition the degrees of freedom of \( \{r_m\} \) into those associated with the interior of elements – denoted by \( i \) – and those on the element interfaces – denoted by \( f \) – and block partition \( D \) as in (4.1). Recall the notational conventions from Section 2 that allow us to move from functions \( q \) to their vector representations \( q \) using appropriate basis expansions. As already noted in (4.3), the Schur complement \( S = D_{ff} - D_{fi} D_{ii}^{-1} D_{if} \) satisfies

\[
(Sq, q) = \inf_{\{r \in R_h : r_f = q\}} (Dr, r) = \|E_h q\|_{H(\text{div}, \Omega)}^2 = \|q\|_{Q_h}^2,
\tag{5.1}
\]
i.e., to precondition the $Q_h$-norm we need to construct a good preconditioner for $S$.

The characterization of the $Q_h$-norm in terms of an $H(\text{div})$-norm suggests the use of an $H(\text{div})$ preconditioner. Indeed, if $T = [0_{fi}, I_{ff}]$ denotes the restriction operator such that $T_r = r_f$ for all $r \in R_h$, then it follows from

$$D^{-1} = \begin{bmatrix} I & -D_{ii}^{-1}D_{ij} \\ 0 & I \end{bmatrix} \begin{bmatrix} D^{-1}_{ii} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -D_{ij}D_{ii}^{-1} & I \end{bmatrix}$$

that $S^{-1} = [D^{-1}]_{ff} = TD^{-1}T^T$. Thus, replacing $D^{-1}$ by any spectrally equivalent $H(\text{div}, \Omega)$-preconditioner will give us a spectrally equivalent preconditioner for $S$. In particular, we may use the Auxiliary-space Divergence Solver (ADS) of [24].

It is well known that ADS is a good preconditioner for many problems set in the $H(\text{div}, \Omega)$-conforming space $R_h$. However, we want to precondition the interface operator $S$ using only the interface degrees of freedom. The ADS preconditioner when applied to $R_h$ uses all degrees of freedom of $R_h$, and not merely the interface degrees of freedom in $Q_h$. This can become a significant addition to the cost as the order $p$ increases.

What can we expect when the algebraic ADS is directly applied to the interface space $Q_h$? To answer this, we examine below the stable decomposition underpinning the theory of ADS and employ Corollary 4.2 to get an analogous stable decomposition restricted to the interface. For simplicity, we now focus on the three-dimensional case. (The two-dimensional case is similar once curl is properly defined.) Let $N_h$ denote the $H(\text{curl})$-conforming Nedelec space of the first kind on the same mesh, which is in correspondence with $R_h$ in the standard finite element exact sequence.

ADS is based on a decomposition of an arbitrary $H(\text{div}, \Omega)$ function acting on an $H^1$ component, the curl of an $H(\text{curl})$ component, and a component that is “small” in the sense that it can be handled by simple smoothing. This decomposition leads to an additive preconditioner for $D$ in the form

$$B^{\text{ADS}} = R + \Pi B^\Pi \Pi^T + \Pi C B^C C^T$$  \hspace{1cm} (5.2)

where the ingredients are as follows:

1. $R$ is a simple smoother for the global matrix $D$, for example, one symmetrized Gauss-Seidel iteration.
2. $\Pi$ is the matrix representation of the Raviart-Thomas interpolation operator from $U_h \times U_h \times U_h$ (or simply $U^3_h$) to $R_h$ obtained using a standard basis \{u_l\} of $U_h$ and the basis \{r_m\} of $R_h$.
3. $C$ is the matrix representation of curl : $N_h \rightarrow R_h$ using a standard basis \{n_k\} of $N_h$ and the basis \{r_m\} of $R_h$.
4. $B^\Pi$ is a standard algebraic $H^1$ solver, for example BoomerAMG from [21, 2], applied to the matrix $\Pi^T D \Pi$.
5. $B^C$ is an algebraic Maxwell solver, such as the Auxiliary-space Maxwell Solver (AMS) of [23] applied to $C^T D C$.

Just as we partitioned the degrees of freedom of $R_h$ into interior ($i$) and interface ($f$) ones, we can partition the degrees of freedom of $U^3_h$ into its interior $\bar{i}$ and its interface $\bar{f}$ degrees of freedom. Similarly the degrees of freedom of $N_h$ are partitioned into sets $\bar{i}$ (interior) and $\bar{f}$ (interface). An important property of the matrices $\Pi$ and $C$ is that when we decompose them into the interior and interface degrees of freedom, their block form is

$$\Pi = \begin{bmatrix} \Pi_{ii} & \Pi_{ij} \\ 0 & \Pi_{jj} \end{bmatrix}, \quad C = \begin{bmatrix} C_{ii} & C_{ij} \\ 0 & C_{jj} \end{bmatrix}.$$  \hspace{1cm} (5.3)
The fact that \( \Pi_{ij} \) and \( \mathbf{C}_{ij} \) are zero blocks follows from the definition of the finite element spaces \( \bar{U}_h, R_h, \hat{N}_h \) and their degrees of freedom, e.g., the \( R_h \) degrees of freedom on a face for the curl of a function in \( N_h \) depend only on the \( N_h \) degrees of freedom associated with that face.

The rationale behind the preconditioner construction in (5.2) comes from the theory of auxiliary space preconditioners [22]. For example, it is possible to prove [24, Section 5.2] under further simplifying assumptions that any \( u \in R_h \) can be decomposed into

\[
   u = v + \Pi z + Cy
\]

with \( z \in U_h \times U_h \times U_h, y \in N_h \), and \( v \in R_h \) such that

\[
   (\text{diag}(D)v, v) + (D\Pi z, \Pi z) + (DCy, Cy) \leq c_5(Dv, u)
\]

where \( c_5 > 0 \) is a constant independent of the size of the problem. This is enough to conclude [27] that \( B^{AB5} \) is a good preconditioner for \( D^{-1} \) (and the “goodness” is measured by \( c_5 \) and the condition numbers of \( B^T \Pi D \Pi \) and \( B^T C F D C \)). In practice, \( B^{AB5} \) often serves as a good preconditioner for \( D^{-1} \) even when a rigorous proof of (5.4) is difficult (such as for non-conforming irregular meshes and discontinuous material coefficients). Loosely speaking, (5.4) means that \( u \) can be decomposed into well-behaved components in the ranges of \( \Pi \) and \( C \) with a small remainder \( v \).

When a purely algebraic implementation of ADS is applied to \( S \), it results in the preconditioner

\[
   B^f = R^f + \Pi_{ij} \tilde{B}^T \Pi_{ij} + C_{ij} \tilde{B}^T \tilde{C}^T
\]

which uses only the interface degrees of freedom of all the spaces involved. Here \( R^f \) is a simple point smoother, like the symmetrized Gauss-Seidel iteration, applied to \( S \), \( \tilde{B}^S \) is BoomerAMG solver applied to \( \Pi^T_{ij} S \Pi_{ij} \), and \( \tilde{B}^C \) is AMS applied to \( C^T_{ij} S C_{ij} \). Arguments for the effectiveness of the interface preconditioners \( \tilde{B}^C \) and \( \tilde{B}^A \) are presented in [7].

Our goal is to discuss the effectiveness of \( B^f \). Just as (5.4) implies that \( B \) is a good preconditioner for \( D \), a stable interface decomposition is required for \( B^f \) to be a good preconditioner for \( S \). We will now show that the decomposition (5.4) implies a stable interface decomposition.

**Lemma 5.1.** If (5.4) holds, then any \( q \in Q_h \) can be decomposed as

\[
   q = v_f + \Pi_{ij} z_f + C_{ij} y_f
\]

where \( v \in R_h \), \( z \in U_h \) and \( y \in N_h \) and their interface degrees of freedom satisfy

\[
   (\text{diag}(S)v_f, v_f) + (S \Pi_{ij} z_f, \Pi_{ij} z_f) + (SC_{ij} y_f, C_{ij} y_f) \leq c_5(Sq, q).
\]

In addition, if

\[
   \kappa \left( \tilde{B}^S \Pi_{ij}^T \Pi_{ij} \right) \leq c_\kappa
\]

then the condition number \( \kappa(B/S) \) depends only on \( c_5 \) and \( c_\kappa \).

**Proof.** Apply Corollary 4.2 with \( H_1 = \Pi \) and \( H_2 = C \), and observe that \( [\Pi z]_f = \Pi_{ij} z_f \) and \( [Cy]_f = C_{ij} y_f \) due to (5.3). \( \square \)

Informally, the result of the lemma can be stated as follows: if ADS works for the matrix \( D \) (a volumetric discretization of \( \| \cdot \|_{H(\mathrm{div}, \Omega)} \)), it will also work for its Schur complement \( S \) (an interfacial discretization of \( \| \cdot \|_{Q_h} \)). Since we assume the former, we can conclude that ADS will be an effective preconditioner for \( S \).
6. Scalable preconditioner. We are now ready to put all the pieces together and define a scalable preconditioner for the original DPG matrix $A$. Our basic premise is that (i) the algebraic ADS is a good solver for the Gram matrix of the $H(\text{div}, \Omega)$-inner product in $R_h$, in the sense that (5.4) holds, and (ii) the algebraic solver BoomerAMG [21], denoted by $B^\sigma$, is a good preconditioner for the Gram matrix $G$ of the $H^1(\Omega)$-inner product on $U_h$, in the sense that the spectral condition number is bounded independent of discretization size $h$ and polynomial order $p$, that is,

$$\kappa(B^\sigma G) \leq c_\kappa. \tag{6.1}$$

Combining this with the $B^f$ defined in (5.5), we have the following result.

**Theorem 6.1.** Assume that (2.6), (3.3), (5.4) and (6.1) hold. Then the block-diagonal matrix

$$\begin{bmatrix}
B^\sigma \\
B^f
\end{bmatrix} \tag{6.2}
$$

is a preconditioner for $A$ and the condition number of the preconditioned system depends only on $c_1, c_2, c_3, c_5$ and $c_\kappa$.

**Proof.** From (2.6), for any $x = (u, q) \in X$, we have

$$c_1(\|u\|^2_{H^1(\Omega)} + \|q\|_Q^2) \leq (Ax, x) \leq c_2(\|u\|^2_{H^1(\Omega)} + \|q\|_Q^2).$$

Using (3.3),

$$c_1\|u\|^2_{H^1(\Omega)} + c_1 c_3^{-2}\|q\|_Q^2 \leq (Ax, x) \leq c_2\|u\|^2_{H^1(\Omega)} + c_2\|q\|_Q^2 \tag{6.3}$$

Hence the result follows from Lemma 5.1 and (6.1).

In practice, the application of $B^\sigma$ and $B^f$ requires the availability of the Gram matrices $G$ and $D$, which may be inconvenient. What we have in hand is $A$. Hence instead of the preconditioner in (6.2), we may use the block preconditioner

$$\begin{bmatrix}
P^\sigma \\
P^f
\end{bmatrix}$$

where $P^\sigma$ and $P^f$ are the algebraic solvers BoomerAMG and ADS applied directly to the principal minors of $A$ corresponding to $U_h$ and $Q_h$, namely to $A_0 = B^\sigma T_0 M^{-1} B_0$ and $A_1 = B^f T_1 M^{-1} B_1$ respectively. The justification for this comes from the observation that by taking $q = 0$ in (6.3), we can conclude that

$$c_1\|u\|^2_{H^1(\Omega)} \leq (A_0 u, u) \leq c_2\|u\|^2_{H^1(\Omega)},$$

i.e., $A_0$ is spectrally equivalent to $G$. Similarly, by taking $u = 0$ in (6.3), we have

$$c_1 c_3^{-2}\|q\|_Q^2 \leq (A_1 q, q) \leq c_2\|q\|_Q^2.$$

Thus instead of preconditioning the matrices $G$ and $S$, whose quadratic forms give the norms $\|\cdot\|_{H^1(\Omega)}$ and $\|\cdot\|_{Q_h}$ respectively, we can directly precondition their spectrally equivalent principal minors $A_0$ and $A_1$. In our implementation it is in fact straightforward to construct the Gram matrix $G$, and we do so in order to build the AMG preconditioner $B^\sigma$, but we use the principal minor $A_1$ to construct the ADS preconditioner $P^f$, so that the preconditioner we use in the numerical results below takes the form

$$\begin{bmatrix}
B^\sigma \\
P^f
\end{bmatrix}. \tag{6.4}$$
Table 7.1
Number of CG iterations and average reduction factors per iteration (in parenthesis) for various h and p refinement levels.

<table>
<thead>
<tr>
<th>elements</th>
<th>order (p)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>64</td>
<td>5 (0.06)</td>
</tr>
<tr>
<td>512</td>
<td>7 (0.12)</td>
</tr>
<tr>
<td>4096</td>
<td>8 (0.18)</td>
</tr>
<tr>
<td>32768</td>
<td>10 (0.22)</td>
</tr>
<tr>
<td>262144</td>
<td>10 (0.22)</td>
</tr>
</tbody>
</table>

7. Numerical results. In this section we report some numerical results with the proposed DPG preconditioner that test its performance with respect to the mesh size $h$, the polynomial order of the trial space $p$, as well as the orders of the test and interfacial spaces. We also examine the parallel scalability of the new algorithm and examine its behavior on more challenging problems with unstructured meshes and large coefficient jumps.

We apply a Conjugate Gradients (CG) solver to the problem (2.5) preconditioned with the preconditioner (6.4) where $B^o$ and $P^f$ use a single V-cycle of BoomerAMG and ADS respectively. The CG relative tolerance we used was $10^{-6}$.

Our implementation is freely available in the MFEM finite element library [26] and we used a slightly modified version of MFEM’s parallel Example 8 (version 3.2) to perform the numerical experiments in this section. Specific ADS and BoomerAMG parameters and additional details can be found in the source code of that example.

7.1. Scalability with respect to $h,p$ for structured mesh. Here we solve the test problem (2.1) on the domain $\Omega = (0,1)^3$ with constant coefficient $\kappa = 1$ meshed with a uniform hexahedral grid. The right hand side $f$ is set to the constant one and zero Dirichlet boundary conditions are imposed on all of $\partial \Omega$.

Table 7.1 reports results for experiments with varying mesh size $h$ (reported as number of finite elements) and polynomial orders $p$. The order $p$ sets the polynomial degree of $U_h$ to $p$, the order of $Q_h$ to $p-1$, and the order of $Y_h$ to $p + d - 1$ where $d = 3$ is the spatial dimension of $\Omega$. As mentioned in Section 2, Assumption (2.6) holds in this setting. The table reports iteration counts as well as the average reduction factors in the PCG iteration. We observe that both of these convergence metrics are quite stable with respect to $h$ and $p$.

In Table 7.2, we explore the parallel scalability of this algorithm, doing a weak scaling study where the number of elements per processor is kept almost constant as we increase the number of processors. This particular experiment uses a trial space order of $p = 1$ but a test space order of 2 rather than the theoretically necessary 3 (see the remarks in Section 7.2). The test was run on an IBM BlueGene/Q machine, where we use four MPI tasks per node.

While the number of iterations in Table 7.2 exhibits some growth, the overall performance is reasonably scalable, and we are continuing to work on improving the per-iteration run time in our implementation.

7.2. Influence of the order of the test space. Currently known theoretical results on the DPG method requires one to set the test space a few degrees higher than
Table 7.2
Weak scaling for the solver with polynomial order fixed at $p = 1$.

<table>
<thead>
<tr>
<th>processors</th>
<th>elements</th>
<th>iterations</th>
<th>conv. factor</th>
<th>solve time</th>
<th>time/iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2.62e+5</td>
<td>9</td>
<td>0.21</td>
<td>249.42s</td>
<td>27.7s</td>
</tr>
<tr>
<td>32</td>
<td>2.10e+6</td>
<td>11</td>
<td>0.26</td>
<td>473.84s</td>
<td>43.1s</td>
</tr>
<tr>
<td>256</td>
<td>1.68e+7</td>
<td>12</td>
<td>0.29</td>
<td>547.95s</td>
<td>45.7s</td>
</tr>
<tr>
<td>2048</td>
<td>1.34e+8</td>
<td>13</td>
<td>0.32</td>
<td>665.81s</td>
<td>51.2s</td>
</tr>
<tr>
<td>16384</td>
<td>1.07e+9</td>
<td>14</td>
<td>0.37</td>
<td>745.69s</td>
<td>53.3s</td>
</tr>
</tbody>
</table>

Table 7.3
Effect of test space order $r$ on scalability with respect to $h$-refinement for a triangular mesh.

<table>
<thead>
<tr>
<th>refine</th>
<th>$r = 4$</th>
<th>$r = 5$</th>
<th>$r = 5$</th>
<th>$r = 6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14</td>
<td>12</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>14</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>28</td>
<td>15</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>46</td>
<td>15</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td>67</td>
<td>17</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>

the trial space. Higher order test spaces can significantly add to the size of the discrete system (2.5) and the overall computational cost. Our numerical results indicate that test spaces of one degree lower than the theoretical requirement often continue to yield a scalable method. We have observed this for triangles, quadrilaterals, tetrahedra, and hexahedra. In Table 7.3, we present some representative results for the interesting case of triangles in two dimensions, where the scalability depends on the parity of $p$. For even $p = 4$, scalability requires a test space order one degree higher than for the odd order $p = 5$. The dependence of the error convergence rate on the parity of $p$ was discussed in [6]. It is interesting to observe that our preconditioner also exhibits such dependence.

7.3. Scalability with respect to $h$ on unstructured meshes. Next we consider problems with different meshes, including unstructured triangular, quadrilateral, and tetrahedral meshes, using in particular the meshes shown in Figure 7.1 at various levels of refinement. The problem is the same as in Section 7.1 except for the mesh. We fix $p = 1$ and focus on the scalability with respect to $h$. The convergence results in Table 7.4 demonstrate that the preconditioner continues to be scalable in these more general settings.

7.4. Behavior of solver with respect to contrast in coefficient. In the following numerical results the coefficient $\kappa$ in (2.1) is piecewise constant, chosen randomly on each element, so that it is 1 with probability 1/2 and $\kappa_0$ with probability 1/2, where $\kappa_0$ is a specified constant across the mesh. Here the $B^0$ component in (6.4) is constructed from an $H^1$ matrix assembled from the bilinear form in (2.1) using the varying coefficient $\kappa$, and $P^{f}$ is constructed as usual using the principal minor of $A$, which also includes the coefficient $\kappa$. In Table 7.5 we report the number of iterations and average reduction factor for several refinement levels and choice of contrast $\kappa_0$. As
the underlying discretization we use here is not robust to contrast in coefficients, we do not expect the solver to perform perfectly with respect to variation in coefficients. Nonetheless, we see that the preconditioner performs acceptably for a wide range of contrasts and we are able to solve the problem effectively.

8. Conclusions. In this paper we presented a scalable preconditioner for the primal DPG formulation of the Poisson problem based on parallel algebraic multigrid techniques. We proved that the preconditioner is optimal under certain assumptions on the mesh and problem coefficients. We also demonstrated that the new algorithm performs well on a wide variety of problems, including some where the theory is not applicable. Due to its algebraic nature, the preconditioner is easy to apply in practice, and has a freely available implementation in the MFEM library.

REFERENCES

Table 7.5
Number of CG iterations and average reduction factors per iteration (in parenthesis) for various values of the contrasts coefficient $\kappa_0$.

<table>
<thead>
<tr>
<th>contrast elements</th>
<th>1e-06</th>
<th>1e-04</th>
<th>1e-02</th>
<th>1e+00</th>
<th>1e+02</th>
<th>1e+04</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>13 (0.29)</td>
<td>12 (0.31)</td>
<td>10 (0.24)</td>
<td>5 (0.06)</td>
<td>8 (0.15)</td>
<td>12 (0.24)</td>
</tr>
<tr>
<td>512</td>
<td>31 (0.64)</td>
<td>29 (0.61)</td>
<td>14 (0.36)</td>
<td>7 (0.10)</td>
<td>11 (0.27)</td>
<td>17 (0.44)</td>
</tr>
<tr>
<td>4096</td>
<td>64 (0.80)</td>
<td>49 (0.75)</td>
<td>15 (0.39)</td>
<td>8 (0.16)</td>
<td>13 (0.33)</td>
<td>33 (0.64)</td>
</tr>
<tr>
<td>32768</td>
<td>119 (0.89)</td>
<td>73 (0.83)</td>
<td>16 (0.41)</td>
<td>9 (0.20)</td>
<td>14 (0.36)</td>
<td>43 (0.72)</td>
</tr>
</tbody>
</table>