Journal of Computational Physics 230 (2011) 6959–6976

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Smoothed aggregation multigrid solvers for high-order discontinuous Galerkin methods for elliptic problems

Luke N. Olson [a], Jacob B. Schroder [b,*]

[a] Department for Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
[b] Department of Applied Mathematics, University of Colorado at Boulder, UCB 526, Boulder, CO 80309, USA

ARTICLE INFO

Article history:
Received 2 September 2010
Received in revised form 19 March 2011
Accepted 12 May 2011
Available online 20 May 2011

Keywords:
High-order
Discontinuous Galerkin
Algebraic multigrid
AMG
Smoothed aggregation

ABSTRACT

We develop a smoothed aggregation-based algebraic multigrid solver for high-order discontinuous Galerkin discretizations of the Poisson problem. Algebraic multigrid is a popular and effective method for solving the sparse linear systems that arise from discretizing partial differential equations. However, high-order discontinuous Galerkin discretizations have proved challenging for algebraic multigrid. The increasing condition number of the matrix and loss of locality in the matrix stencil as p increases, in addition to the effect of weakly enforced Dirichlet boundary conditions all contribute to the challenging algebraic setting.

We propose a smoothed aggregation approach that addresses these difficulties. In particular, the approach effectively coarsens degrees-of-freedom centered at the same spatial location as well as degrees-of-freedom at the domain boundary. Moreover, the character of the near null-space, particularly at the domain boundary, is captured by interpolation. One classic prolongation smoothing step of weighted-Jacobi is also shown to be ineffective at high-order, and a more robust energy-minimization approach is used, along with block relaxation that more directly utilizes the block diagonal structure of the discontinuous Galerkin discretization. Finally, we conclude by examining numerical results in support our proposed method.

1. Introduction

In this paper, we develop a smoothed aggregation (SA) based algebraic multigrid (AMG) solver for high-order discontinuous Galerkin discretizations of the Poisson problem. As p increases, the problem becomes more challenging due to an increase in condition number. Moreover, the algebraic structure changes, resulting in a loss of locality in the degrees-of-freedom, although maintaining element level locality. Furthermore, the effect of weakly enforced Dirichlet boundary conditions affects the performance of a purely algebraic solver.

High-order methods are popular where high accuracy is of great importance because of the spectral convergence in p of the error for sufficiently smooth solutions. Because of this property, per degree-of-freedom, high-order discretizations often yield improved accuracy in comparison to low-order methods. However, the overall efficiency of high-order discretizations is debated. Matrices become more dense and the matrix conditioning deteriorates as p increases. As the density of the matrix increases, the equations become coupled to degrees-of-freedom farther away and a loss-of-locality occurs. Together, this

* Corresponding author.
E-mail addresses: jacob.schroder@colorado.edu, jacob.bb.schroder@gmail.com (J.B. Schroder).

0021-9991/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2011.05.009

impacts the effectiveness of iterative solvers and AMG in particular. Classic SA degrades quickly as p increases for either continuous or discontinuous Galerkin discretizations.

To address this problem, work has been done on applying multilevel methods to continuous Galerkin high-order finite element discretizations [1–3]. This previous work applies standard multigrid techniques to a low-order approximation of the high-order problem in order to form an effective preconditioner. The geometric nature of these methods requires expensive rediscretizations when forming the preconditioner, in addition to a coupling between the solver and discretization code.

High-order elements have been successful in a discontinuous Galerkin framework, which provides flexibility for non-conforming meshes and neighboring elements of differing p. Moreover, discontinuous Galerkin discretization s have grown in popularity for elliptic operators [4–9], which are the target problem of this paper. For instance with convection–diffusion problems, where the convection terms benefit from the discontinuous formulation, Poisson solvers are needed for Schur-complement type preconditioners for the pressure block [10,11]. However, the linear systems produced by discontinuous Galerkin discretizations have more degrees-of-freedom than for the analogous continuous Galerkin discretization, thus compounding the added complexity from high-order. This further motivates the need for scalable linear solvers.

Even for the Poisson problem, classic SA techniques are inadequate, especially at high-order, for discontinuous Galerkin discretizations. The performance seriously degrades for either increasing p or h—see the first column of Table 2. Classic SA cannot effectively aggregate or smooth the prolongator in the high-order discontinuous Galerkin setting, because of the complicated non-M-matrix stencil and the rising condition number with p. There has been work on multilevel solvers for discontinuous Galerkin discretizations of the Poisson operator [12–14,10,15,16]. However, these approaches are typically either two-level, or non-algebraic. The non-algebraic (p-multigrid) methods [12–14,16] have been successful for several applications, but have some inherent drawbacks such as the potential expense of rediscretizing the problem for coarser p. That is, the operator for the kth coarse level is constructed by rediscretizing the problem for $p = p_{orig} - k$, where p_{orig} is the original polynomial order. These constructions are often costly since the hierarchy is either too expensive when p is gradually coarsened or is inaccurate when more aggressive p-coarsening is used. Alternatively, the approach [10,15] rediscretizes the problem at the original p for a hierarchy of nested meshes, which may also result in high computational cost. This further motivates an algebraic approach.

At $p = 1$ or 0 (e.g., when $k = p_{orig}$ or $k = p_{orig} - 1$ during p-multigrid), the problem may be sufficiently small and suitable for a direct solution or may be efficiently solved using standard geometric or algebraic multigrid approaches. A previous SA approach to low-order ($p = 0, 1$) discontinuous Galerkin discretizations has shown the effectiveness of aggregation-based multigrid for diffusion and convection–diffusion problems [17]. However, it is unclear how to extend this method to the high-order setting. The use of classic weighted-Jacobi prolongation smoothing and of a matrix stencil-based strength-of-connection measure to guide relaxation do not extend to the complicated non-M-matrix setting of high-order. Additionally, agglomerating entire elements together is not effective when the elements contain themselves many degrees-of-freedom. Furthermore, issues relating to aggregating degrees-of-freedom centered at the same spatial location as well as aggregation and interpolation at the domain boundary are not explicitly addressed.

The goal is an algebraic solver for the Poisson problem that is both p- and h-independent. We depart from earlier non-algebraic (p-multigrid) approaches that require the discretization code to be coupled to the iterative solver. This is accomplished by designing an algebraic method that assumes access to the matrix A, a collection of *a priori* near null-space modes B, element orders, degree-of-freedom type (e.g., nodal), and possibly the fine level mesh coordinates. Here, B is initially the standard constant for diffusion problems. Yet, the creation of a p-independent method is not straight-forward. As p increases, both the condition number of A increases and locality in each matrix row is lost. Additionally, no standard aggregation method exists for the high-order setting. Further difficulties arise due to aggregation and interpolation at the boundary. Dirichlet boundary conditions are enforced weakly in the discontinuous Galerkin setting, necessitating an automatic approach in the solver.

In Section 2, a brief overview of discontinuous Galerkin methods for the Poisson problem is given. In Section 3, the SA method is briefly presented, including two recent SA developments, the energy-minimization prolongation smoothing method and the strength-of-connection method, the Evolution Measure. In Section 4, motivation for the proposed SA methods is given, with a focus on the proposed aggregation schemes. In Section 5, we propose effective SA methods for this problem. In Section 6, encouraging numerical results in support of our approach are given.

2. Discontinuous Galerkin method

We now derive both the weak and strong-weak forms of the discontinuous Galerkin [6] method for the Poisson problem. We examine the local discontinuous Galerkin (LDG) method [18], the interior penalty (IP) method [19] and the Brezzi et al. method [20]. Consider,

$$\Delta u = f \text{ in } \Omega, \tag{1a}$$

$$u = g_{\mathcal{D}} \text{ on } \Gamma_{\mathcal{D}}, \tag{1b}$$

$$\nabla u \cdot \boldsymbol{n} = \boldsymbol{g}_{\mathcal{N}} \cdot \boldsymbol{n} \text{ on } \Gamma_{\mathcal{N}}, \tag{1c}$$

where Ω is a bounded domain in $\mathbb{R}^d$, $\boldsymbol{n}$ is the unit outward normal to the boundary $\Gamma = \Gamma_D \bigcup \Gamma_N$, and f is a given function in $L^2(\Omega)$. The functions $g_{\mathcal{D}}$ and $\boldsymbol{g}_{\mathcal{N}}$ respectively define the boundary conditions on the Dirichlet and Neumann portions of Γ, which are denoted by $\Gamma_{\mathcal{D}}$ and $\Gamma_{\mathcal{N}}$. Discontinuous Galerkin methods are formulated with respect to a first-order system, so we introduce a new vector variable $\boldsymbol{q}$ to yield

$$\boldsymbol{q} - \nabla u = 0 \quad \text{in } \Omega, \tag{2a}$$
$$\nabla \cdot \boldsymbol{q} = f \quad \text{in } \Omega, \tag{2b}$$
$$u = g_{\mathcal{D}} \quad \text{on } \Gamma_{\mathcal{D}}, \tag{2c}$$
$$\boldsymbol{q} \cdot \boldsymbol{n} = \boldsymbol{g}_{\mathcal{N}} \cdot \boldsymbol{n} \quad \text{on } \Gamma_{\mathcal{N}}. \tag{2d}$$

The first-order system is only used to derive the method and $\boldsymbol{q}$ does not require an additional linear solve.

2.1. Weak formulation

To derive the weak formulation, we initially multiply (2a) and (2b) by smooth test functions $\boldsymbol{\psi}$ and v, respectively. Next, we integrate by parts over a triangulation $\mathcal{T}$ of Ω. This yields, for each element K,

$$\int_K \boldsymbol{q} \cdot \boldsymbol{\psi}\, d\boldsymbol{x} = -\int_K u \nabla \cdot \boldsymbol{\psi}\, d\boldsymbol{x} + \int_{\partial K} u\, \boldsymbol{\psi} \cdot \boldsymbol{n}_{\boldsymbol{K}}\, ds, \tag{3a}$$
$$\int_K -\boldsymbol{q} \cdot \nabla v\, d\boldsymbol{x} + \int_{\partial K} v\, \boldsymbol{q} \cdot \boldsymbol{n}_{\boldsymbol{K}}\, ds = \int_K f v\, d\boldsymbol{x}, \tag{3b}$$

where $\boldsymbol{n}_{\boldsymbol{K}}$ is the unit outward normal on K. Now, we introduce the *broken* finite element spaces associated with $\mathcal{T}$. Let

$$\boldsymbol{V}_{\boldsymbol{h}} := \left\{ v_h \in L^2(\Omega) : v_h|_K \in \mathcal{P}_p(K),\ \forall K \in \mathcal{T} \right\}, \tag{4a}$$
$$\boldsymbol{\Psi}_{\boldsymbol{h}} := \left\{ \boldsymbol{\psi}_h \in (L^2(\Omega))^d : \boldsymbol{\psi}_h|_K \in (\mathcal{P}_p(K))^d,\ \forall K \in \mathcal{T} \right\}, \tag{4b}$$

where $\mathcal{P}_p(K)$ is the space of all polynomial functions on K of degree at most p. Using the finite element spaces, we replace the exact solution $(u, \boldsymbol{q})$ with an approximate solution $(u_h, \boldsymbol{q}_{\boldsymbol{h}})$. Additionally, the numerical fluxes $\hat{u}_h$ and $\hat{\boldsymbol{q}}_{\boldsymbol{h}}$ are discrete approximations to the traces of u_h and $\boldsymbol{q}_{\boldsymbol{h}}$, respectively, and replace u_h and $\boldsymbol{q}_{\boldsymbol{h}}$ in the element boundary integrals. The flux choices are discussed below. The result is the weak statement of the problem to find $u_h \in \boldsymbol{V}_{\boldsymbol{h}}$ and $\boldsymbol{q}_{\boldsymbol{h}} \in \boldsymbol{\Psi}_{\boldsymbol{h}}$ such that $\forall K \in \mathcal{T}$,

$$\int_K \boldsymbol{q}_{\boldsymbol{h}} \cdot \boldsymbol{\psi}_h\, d\boldsymbol{x} = -\int_K u_h \nabla \cdot \boldsymbol{\psi}_h\, d\boldsymbol{x} + \int_{\partial K} \hat{u}_h \boldsymbol{\psi}_h \cdot \boldsymbol{n}_{\boldsymbol{K}}\, ds, \tag{5a}$$
$$\int_K -\boldsymbol{q}_{\boldsymbol{h}} \cdot \nabla v_h\, d\boldsymbol{x} + \int_{\partial K} v_h\, \hat{\boldsymbol{q}}_{\boldsymbol{h}} \cdot \boldsymbol{n}_{\boldsymbol{K}}\, ds = \int_K f v_h\, d\boldsymbol{x}, \tag{5b}$$

holds $\forall v_h \in \boldsymbol{V}_{\boldsymbol{h}}$ and $\forall\ \boldsymbol{\psi}_h \in \boldsymbol{\Psi}_{\boldsymbol{h}}$.

2.2. Strong-weak formulation

The strong-weak formulation of the problem is obtained through another step of integration by parts on Eqs. (5a) and (5b), This yields the terms $(\hat{\boldsymbol{q}}_{\boldsymbol{h}} - \boldsymbol{q}_{\boldsymbol{h}})$ and $(\hat{u}_h - u_h)$, which resemble a penalty method on each element boundary. The result is the strong-weak statement of the problem to find $u_h \in \boldsymbol{V}_{\boldsymbol{h}}$ and $\boldsymbol{q}_{\boldsymbol{h}} \in \boldsymbol{\Psi}_{\boldsymbol{h}}$ such that $\forall K \in \mathcal{T}$,

$$\int_K \boldsymbol{q}_{\boldsymbol{h}} \cdot \boldsymbol{\psi}_h\, d\boldsymbol{x} = \int_K \nabla u_h \cdot \boldsymbol{\psi}_h\, d\boldsymbol{x} + \int_{\partial K} (\hat{u}_h - u_h) \boldsymbol{n}_{\boldsymbol{K}} \cdot \boldsymbol{\psi}_h\, ds, \tag{6a}$$
$$\int_K \nabla \cdot \boldsymbol{q}_{\boldsymbol{h}} v_h\, d\boldsymbol{x} + \int_{\partial K} v_h (\hat{\boldsymbol{q}}_{\boldsymbol{h}} - \boldsymbol{q}_{\boldsymbol{h}}) \cdot \boldsymbol{n}_{\boldsymbol{K}}\, ds = \int_K f v_h\, d\boldsymbol{x}, \tag{6b}$$

holds $\forall v_h \in \boldsymbol{V}_{\boldsymbol{h}}$ and $\forall \boldsymbol{\psi}_h \in \boldsymbol{\Psi}_{\boldsymbol{h}}$. This is called the strong-weak form because the differentiability requirements on the solution are higher than necessary.

2.3. Fluxes

A specific discontinuous Galerkin method is defined through the flux choice. In order to define this choice for the LDG method, let K^+ and K^- be two adjacent elements in $\mathcal{T}$, $\boldsymbol{n}^+$ and $\boldsymbol{n}^-$ be the corresponding unit outward normals, and $(\boldsymbol{q}^+, u^+)$

and $(\boldsymbol{q}^-, u^-)$ be the traces of $(\boldsymbol{q}, u)$ on the boundaries of K^+ and K^-, respectively. The fluxes are defined using the jump $[\![\cdot]\!]$ and average $\{\cdot\}$ operators, which are

$$\{u\} := (u^+ + u^-)/2, \quad \{\boldsymbol{q}\} := (\boldsymbol{q}^+ + \boldsymbol{q}^-)/2,$$
$$[\![u]\!] := u^+\boldsymbol{n}^+ + u^-\boldsymbol{n}^-, \quad [\![\boldsymbol{q}]\!] := \boldsymbol{q}^+ \cdot \boldsymbol{n}^+ + \boldsymbol{q}^- \cdot \boldsymbol{n}^-.$$

The LDG method defines fluxes as

$$\hat{\boldsymbol{q}} := \{\boldsymbol{q}\} + \beta[\![\boldsymbol{q}]\!] - \tau_K[\![u]\!], \tag{7a}$$
$$\hat{u} := \{u\} - \beta \cdot [\![u]\!]. \tag{7b}$$

Importantly, this flux choice yields a consistent and stable method [6]. For stability, the scalar function τ_K yields an element-wise penalty term that is typically chosen to be $\mathcal{O}(1/h_K)$ [21], where h_K is the local element diameter of element K. For consistency, care must be taken that $\hat{\boldsymbol{q}}$ and $\hat{u}$ are equivalent over all shared edges (or faces) between adjacent elements. This additionally ensures symmetry of the discrete operator [22,6]. To guarantee consistent $\hat{u}$ and $\hat{\boldsymbol{q}}$, β must be equivalent over all shared edges (or faces) between adjacent elements. Common choices for β are $\beta = 0$ or $|\beta(e) \cdot \boldsymbol{n}^{\pm}| = 1/2$, over any shared edge e. The latter choice guarantees superconvergence on regular grids [22] and a more compact matrix stencil. We focus on $\beta = 0$, although the performance for the proposed solver is similar for either choice of β.

This flux choice is also the source of the name, *local* discontinuous Galerkin method. The flux $\hat{u}$ does not depend on $\boldsymbol{q}$, thus allowing elimination of $\boldsymbol{q}$ from (5b), with the help of (6a). The result is a problem with only u as the unknown—i.e., an additional linear solve for $\boldsymbol{q}$ is not required.

The fluxes for the IP method and the Brezzi et al. method also share this favorable property of not requiring an additional linear solve. For the IP method, the fluxes are

$$\hat{\boldsymbol{q}} := \{\boldsymbol{q}\} - \tau_K[\![u]\!], \tag{8a}$$
$$\hat{u} := \{u\}. \tag{8b}$$

For a stable and consistent method [23] the element-wise penalty term must satisfy $\tau_K > N_K/h_K$, where N_K is the number of nodal basis functions in element K. For the Brezzi et al. method, the fluxes are

$$\hat{\boldsymbol{q}} := \{\boldsymbol{q}\} + \tau_K r_e([\![u]\!]), \tag{9a}$$
$$\hat{u} := \{u\}. \tag{9b}$$

The function r_e is defined as a lifting operator on each edge e that vanishes outside the union of triangles containing e. Specifically,

$$\int_\Omega r_e(\boldsymbol{\varphi}) \cdot \boldsymbol{\psi} d\boldsymbol{x} = -\int_e \boldsymbol{\varphi} \cdot \{\boldsymbol{\psi}\} ds, \tag{10}$$

$\forall \boldsymbol{\psi} \in \boldsymbol{\Psi}$ and $\boldsymbol{\varphi} \in [L^1(e)]^2$. For a stable and consistent method [6], the element-wise penalty term must satisfy $\tau_K > 0$. By considering stability [6], we must satisfy $\tau_K \geqslant 3$, leading to our choice of $\tau_K = 4$.

Fluxes on Γ are given implicitly by defining the basic flux operations on Γ as

$$[\![u]\!] = u\boldsymbol{n} \quad \text{and} \quad \{q\} = q. \tag{11}$$

The quantities $\{u\}$ and $[\![q]\!]$ are not needed on Γ (cf. [6]). An important implication of the boundary fluxes is that Dirichlet boundary conditions are enforced weakly, in contrast to standard finite elements. This affects our linear solvers.

We choose to study these three representative discontinuous Galerkin methods with the expectation that the results here extend to other methods. The issues addressed by our solver are present, regardless of the flux choice. The LDG, IP and Brezzi et al. methods are logical choices given their popularity and favorable properties such as consistency and stability.

3. Smoothed aggregation overview

In this section, we give a high-level overview of our SA algorithm, which conforms to standard SA with the exception of pre-smoothing the near null-space vectors. SA automatically constructs a multilevel hierarchy of interpolation operators and coarse sets of degrees-of-freedom through Algorithm 1, the setup phase. The goal of Algorithm 1 is to create a coarse space and associated interpolation (i.e., prolongation) operators which capture algebraically smooth error—i.e., low energy error. In other words, interpolation is designed to complement relaxation (also called smoothing). We define algebraically smooth error modes to be grid functions with a small Rayleigh quotient [24], and therefore equivalent to the near null-space or low energy modes. The multilevel hierarchy is then used in the solve phase outlined in Algorithm 2 to reduce the residual for a given right-hand side and initial guess. The solve phase interpolates the residual equation between levels and at each level, uses an inexpensive relaxation method to reduce the error. If interpolation and relaxation are complementary, the process reduces the residual quickly. The error that is slow to relax is interpolated to coarser levels for further reduction.

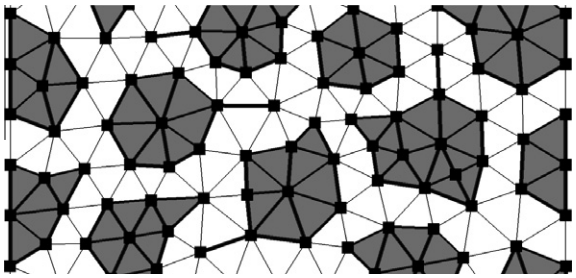

Fig. 1. Example aggregation of the matrix graph of S_0.

Algorithm 1 requires the matrix A, a set of m user-provided near null-space modes B, and a list of near null-space mode improvement steps to be taken at each level, $\{\eta_0,\ldots,\eta_N\}$, for $\eta_i \in \{0,1,2,\ldots\}$. B is used to form the tentative prolongation operator and is typically defined through the null-space of the governing PDE with no boundary conditions. For example in diffusion and linearized elasticity problems, B is the constant and rigid-body-modes, respectively. Algorithm 1 constructs coarse operators A_k, with $k = 0$ being the index for the finest level, and related interpolation operators $P_k : \mathbb{R}^{n_{k+1}} \rightarrow \mathbb{R}^{n_k}$, where n_k and n_{k+1} are the sizes of two successively coarser grids.

When constructing P_k, Algorithm 1 first pre-smooths the near null-space. For our experiments, we take η_k steps of relaxation on $A_k B_k = \mathbf{0}$, for each vector in B_k, the near null-space modes on level k. Improving B with relaxation combines the two previous approaches for generating B. The classic SA approach uses the null-space of the unrestricted PDE and yields a B that is often inaccurate, particularly near the boundaries. This classic approach was later extended to create an adaptive SA [25,26] approach, which in part targets near null-space mode improvement near the domain boundaries, although at high computational expense. The ability to adapt B is critical here because, unlike with classic SA, the prolongation smoothing we use directly incorporates B_k exactly into span(P_k).

The next calculation is a strength-of-connection matrix S_k [27–29] which is used in the graph coarsening process and thus impacts the sparsity pattern of P_k. The matrix graph of A_k is one approach, but, this graph does not necessarily reflect the nature of algebraically smooth error—e.g., during anisotropic diffusion, algebraically smooth error varies slowly in only one direction [29]. As an alternative, we construct the strength matrix S_k so that its graph is the subset of A_k that retains only those couplings that represent accurate interpolation of algebraically smooth error between neighboring degrees-of-freedom.

Therefore, we let S_k be nonzero at (i,j) if algebraically smooth error is accurately interpolated between i and j. The classic SA strength measure bases these decisions directly on the matrix stencil such that degree-of-freedom i is strongly connected to j if

$$|A_{ij}| \geqslant \theta\sqrt{A_{ii}A_{jj}}, \tag{12}$$

for some user supplied $\theta \in [0,1]$, typically 0.25. This measure is motivated by an M-matrix assumption on A, which does not hold here and thus leads to problems with (12).

S_k controls the sparsity pattern of P_k through the process of aggregation, which places the n_k vertices of the matrix graph of S_k into a_{k+1} aggregates. The aggregates form a disjoint, i.e., non-overlapping, covering of the matrix graph of S_k and we aggregate through a greedy algorithm on the matrix graph of S_k by collecting distance one neighbors. Fig. 1 depicts the result of aggregation applied to the matrix graph of S_0 for a sample 2D Poisson problem discretized with standard $p = 1$ continuous Galerkin finite elements. Each shaded region depicts an aggregate, with any hanging nodes connected with a thick line.

With `inject_modes()`, a tentative prolongation operator $P_k^{(0)}$ is formed based on aggregation and B_k. An intermediate sparse matrix $C_k \in \mathbb{R}^{n_k \times a_{k+1}}$ is formed such that entry (i,j) is nonzero only if degree-of-freedom i is in aggregate j. $P_k^{(0)}$ is formed by injecting B_k in a block column-wise fashion into C_k. Each block column of $P_k^{(0)}$ corresponds to an aggregate and is nonzero only for the degrees-of-freedom in that aggregate, where B_k is injected. Let indexing for matrices and vectors start at 1 and i be in aggregate j, then the entry $(i,(j-1)m+s)$ of $P_k^{(0)}$ is equal to the entry (i,s) of B_k, where $1 \leqslant s \leqslant m$. If i is not in aggregate j, then the corresponding entry of $P_k^{(0)}$ is 0. Thus, $P_k^{(0)}$ is of size $(n_k \times n_{k+1})$, where $n_{k+1} = ma_{k+1}$.

As a final step we apply a local QR factorization to the nonzero portion of each block column. The Q replaces the nonzero portion of $P_k^{(0)}$ to improve conditioning, while the R forms B_{k+1}. The result is that span($P_k^{(0)}$) exactly represents the algebraically smooth near null-space so that $P_k^{(0)} B_{k+1} = B_k$.

Often, the tentative prolongator is suboptimal and is improved through prolongation smoothing, which widens the interpolation stencil and lowers the energy of each column so that the quality of the coarse space is improved. The result is a more accurate P_k that captures algebraically smooth error and is complementary to relaxation. Classic SA uses one iteration of weighted-Jacobi for $A_k P_k^{(0)} = \mathbf{0}$ to improve P_k. This step is particularly important for classic SA, because it is the only opportunity to ameliorate the effects of the boundary. We further reduce the boundary impact in Algorithm 1 by pre-smoothing B.

Finally, a coarse level operator A_{k+1} is formed through the Galerkin condition through $R_k = P_k^T$ and $A_{k+1} = R_k A_k P_k$. The entire process repeats itself until the dimensions of A_k drop below a threshold.

Algorithm 1: `sa_setup`$(A, B, \{\eta_0, \ldots, \eta_N\})$

$A_0 = A$
$B_0 = B$
$k = 0$

while `size`(A_k) > max_coarse **do**
  smooth η_k times on $A_k B_k = \mathbf{0}$

  S_k = `strength`(A_k)
  C_k = `aggregate`(S_k)
  $P_k^{(0)}, B_{k+1} =$ `inject_modes`(C_k, B_k)
  $P_k =$ `prolongator_smoother`$\left(A_k, P_k^{(0)}\right)$
  $R_k = P_k^*$
  $A_{k+1} = R_k A_k P_k$
  $k = k + 1$
return $\{A_0, \ldots, A_k\}$, $\{P_0, \ldots, P_{k-1}\}$, $\{R_0, \ldots, R_{k-1}\}$

After construction of the hierarchy, Algorithm 2 is used for standard multigrid cycling to reduce the residual for a given right-hand side and initial guess. The recursive algorithm `sa_solve()` first pre-smoothes at level k—e.g., using Gauss-Seidel or weighted-Jacobi. The residual is then formed and restricted to level $k+1$ to give $R_k r_k = b_{k+1}$, where the coarse residual equation is solved exactly or is recursively sent for further coarsening. The solution x_{k+1}, which is an approximation of the error, is interpolated to level k and used to correct x_k. A final post-smoothing sweep is then used. The variable γ controls the number of recursive calls and hence, cycle type. A value of 1 corresponds to a V-cycle and 2 corresponds to a W-cycle, which results in more visits to coarse grids and is valuable for the problems considered here. The number of pre- and post-smoothing steps may also vary. For example, a V(1,2)-cycle corresponds to $\gamma = 1$, 1 pre-smoothing step, and 2 post-smoothing steps.

Algorithm 2: `sa_solve`$(k, \gamma, \{A_0, \ldots, A_{N+1}\}, \{P_0, \ldots, P_N\}, \{R_0, \ldots, R_N\}, x_k, b_k)$

if $k = N + 1$ **then**
  return `solve`(A_k, b_k)
else
  x_k = `pre_smooth`(A_k, x_k, b_k)
  $r_k = b_k - A_k x_k$
  $b_{k+1} = R_k r_k$
  $x_{k+1} = \mathbf{0}$

  for $i = 1$ **to** γ **do**
    x_{k+1} = `sa_solve`$(k+1, \gamma, \{A_0, \ldots, A_{N+1}\}, \{P_0, \ldots, P_N\}, \{R_0, \ldots, R_N\}, x_{k+1}, b_{k+1})$

  $x_k = x_k + P_k x_{k+1}$
  x_k = `post_smooth`(A_k, x_k, b_k)
  return x_k

3.1. Prolongation smoothing

In this section, we give a brief overview of the energy-minimization framework used for prolongation smoothing [30], which improves interpolation accuracy. We utilize a Krylov-based strategy that computes for each column P_j of P

$$\min_{P \in V} \sum_j \|P_j\|_A, \tag{13}$$

where V is a Krylov sub-space over which two constraints are enforced: $P B_{k+1} = B_k$ and a sparsity pattern constraint. The near null-space preservation constraint utilizes our *a priori* knowledge of algebraically smooth error and, as a result, improves the conditioning of the minimization problem. The sparsity pattern constraint ensures moderate complexities and is taken to be the sparsity pattern of $S P^{(0)}$, so that nonzeros only grow in the direction of strong connections.

Since A is SPD for this problem, a constrained conjugate gradient (CG) method is used. With respect to complexity, each Krylov-based prolongation smoothing step is on the order of one classic weighted-Jacobi prolongation smoothing step.

3.2. Evolution Measure

In this section, we give a brief overview of a strength-of-connection measure (Evolution Measure) [29] used in our experiments. Overall, this measure combines local knowledge of both algebraically smooth error and the behavior of interpolation, in order to compute strength-of-connection. To do this, consider computing strength values for a degree-of-freedom i at all of its algebraic neighbors in the graph of matrix A.

In order to account for algebraically smooth error, a δ-function centered at i is evolved according to weighted-Jacobi, such that

$$z = (I - \omega\ D^{-1}A)^k \delta_i. \tag{14}$$

For small k, z is a locally supported algebraically smooth function. Typical values are $k = 2$ and $\omega = 1/\rho(D^{-1}A)$.

Local knowledge of interpolation is then accounted for. Consider a mock aggregate composed of all the algebraic neighbors of i. We assess the ability of the aggregate's block column in $P^{(0)}$—i.e., slice of B—to interpolate z. The mock interpolation accuracy is only measured around the algebraic neighborhood of i, with exact interpolation enforced at point i. In response, strength-of-connection is then defined as the point-wise accuracy of the mock interpolant. For a single near null-space vector $B \in \mathbb{R}^n$, the Evolution Measure reduces to the simple ratio

$$S_{ij} = \left| 1 - \frac{B_j z_i}{B_i z_j} \right|. \tag{15}$$

Generalizations of the Evolution Measure to $B \in \mathbb{R}^{n,m}$ exist [29], but the target problems here only concern the standard case of one near null-space mode.

A drop-tolerance is then applied to S. Since the measure assesses accuracy, smaller values indicate a stronger coupling, leading to the following dropping strategy: drop entry S_{ij} if

$$S_{ij} > \hat{\theta} \min_m \{S_{im} : m \neq i,\ A_{im} \neq 0\}. \tag{16}$$

Typical values of $\hat{\theta}$ are in the range of 2.0–4.0, with $\hat{\theta} = 2.0$ the choice used here.

4. Motivation

4.1. Difficulties

Here, we enumerate the difficulties that cause classic SA to degrade for both h- and p-refinement. This degradation is observed by considering $p = 1$ for our model problem (see Table 2):

h	1/16	1/32	1/64	1/128
PCG iterations:	26	38	65	122

Here, we see that the iterations (capped at 150) increase simply due to the discontinuous jumps between elements; classic SA performs optimally in the continuous case for this example. Likewise, refinement in p at $h = 1/8$ leads to (see Table 2):

p	2	3	4	5	6	7	8	9	10	11
PCG iterations:	29	35	40	35	52	50	80	112	151+	151+

Even use of an advanced strength-of-connection approach such as the Evolution Measure and an improved interpolation strategy such as energy-minimization prolongation smoothing does not lead to h or p independence. For example, with both advanced strength-of-connection and improved interpolation, for p refinement at $h = 1/8$ we have (see Table 2):

p	2	3	4	5	6	7	8	9	10	11
PCG iterations:	8	8	15	20	26	27	26	53	54	69

Similarly, h-independence is lost for $p > 4$. Incidentally, $p = 4$ is often considered a threshold where straightforward multigrid methods remain competitive [2,3].

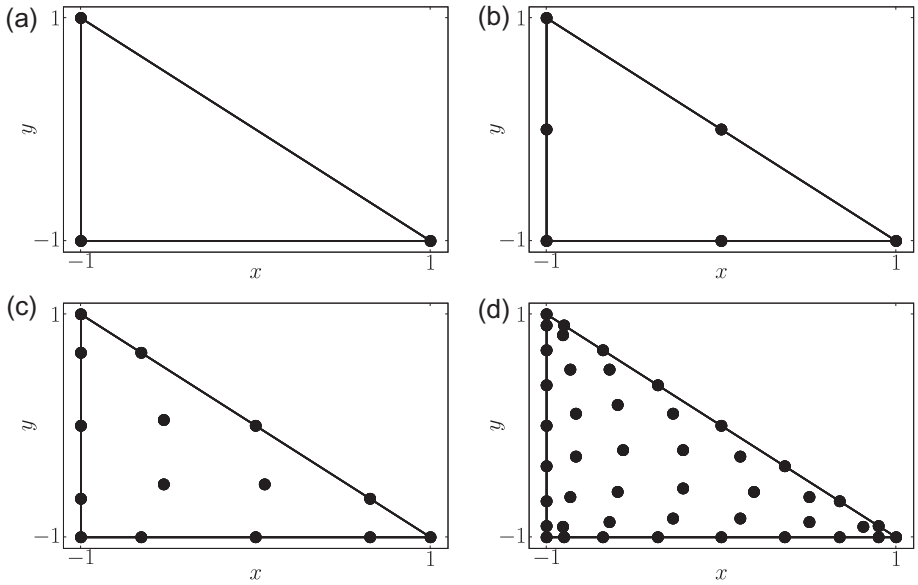

Fig. 2. Nodal locations on reference elements (a) $p = 1$, (b) $p = 2$, (c) $p = 4$ (d) $p = 8$.

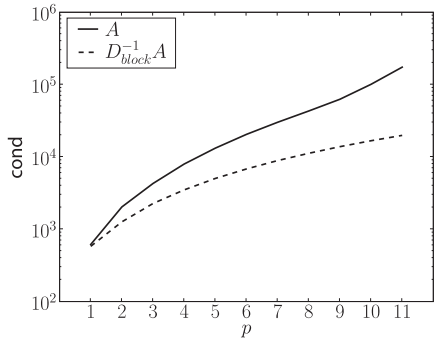

Fig. 3. Growth in condition number for $h = 1/8$.

One of the inherent difficulties for iterative solvers is the rising condition number of A with p, which is the normal scenario for high-order finite elements. The Sledge++ [31] package is used to discretize (1) and chooses near-optimal nodal locations for Lagrangian basis functions, so that conditioning is improved in comparison to equispaced nodal locations. In Fig. 2, we show example nodal locations for various p.

In Fig. 3, we use Arnoldi iteration to approximate the spectral condition number of A and $D_{block}^{-1}A$ for p-refinement. The model problem is isotropic diffusion on an unstructured simplicial mesh of the unit box with $h = 1/8$. D_{block}^{-1} refers to the block inverse diagonal of A, with a blocksize of $(p+1)(p+2)/2$, or the number of degrees-of-freedom in each element. The condition number rises less dramatically for $D_{block}^{-1}A$, but the cost of D_{block}^{-1} grows with p^2. Consequently, developing a p-independent method remains the key challenge.

Remark 1. The condition number also rises for a fixed p and decreasing h, but in this case the basic algebraic nature of the matrix does not change (assuming all meshes are well-conditioned). This makes h-independent solvers easier to construct. The rising condition number with p reflects a different situation, where the underlying algebraic structure of A changes and each matrix row becomes more dense and less local.

To partially address the rising condition number and loss of locality, we leverage the improved conditioning of $D_{block}^{-1}A$ and use block Gauss-Seidel on the finest level for both the pre- and post-smoothers and for pre-smoothing the near null-space. Blocksize again equals the number of degrees-of-freedom in an element. Block relaxation on the finest level has already been shown to be beneficial [11,16,17] for discontinuous discretizations. However, given the algebraic nature of the multilevel hierarchy, there is no natural concept of elements or corresponding blocksizes on coarse levels, which leads us to use standard Gauss-Seidel for all coarse levels.

Additionally, the number of iterations for energy-minimization prolongation smoothing and of time steps k taken by the Evolution Measure needs to rise with p. However, there is a bottleneck with the Evolution Measure and k, since the largest computationally feasible value is $k = 4$ and this still results in a strength measure that is of the same complexity as computing A^2. For some problems, even squaring the matrix is too expensive. Thus, larger values of k results in intolerable complexity and limited value.

4.2. Conforming aggregation

The main issue not addressed so far is aggregation in the high-order discontinuous setting, for which no standard method exists. One clear difference for discontinuous Galerkin discretizations, in comparison to standard discretizations, is that each element has a distinct set of degrees-of-freedom, leading to multiple degrees-of-freedom at each nodal location on shared element boundaries. A similar problem occurs for PDE systems where multiple variables exist at the same spatial location. The classic SA approach to this issue is supernodes [32], which are designed to aggregate vector problems with multiple variables at the same spatial location. The supernode approach yields an aggregation in two stages at the finest level. In the first stage, supernodes are formed by aggregating degrees-of-freedom at the same spatial location together. The second stage then aggregates strongly connected supernodes together.

A supernode-like approach here corresponds to *conforming* aggregation on the finest level, wherein all degrees-of-freedom at the same spatial location are collected. All other degrees-of-freedom are injected as singleton aggregates to the next level. We label this conforming aggregation due to its similarity with a C^0-projection. To accomplish this, only the strength matrix S must be changed on the finest level. Moreover, no prolongation smoothing is required on the finest level, so that the tentative $P = P^{(0)}$.

A conforming step is well-motivated. Degrees-of-freedom at the same spatial location have similar solution values, thus constant-based interpolation remains tractable. Additionally, a conforming step allows SA to separate decisions about two entirely different types of algebraic connections into two distinct phases—i.e., connections between degrees-of-freedom at the same spatial location are considered only on the finest level, and longer-range connections are considered only on coarse levels. Lastly considering the motivating example of isotropic diffusion, we know that degrees-of-freedom close together spatially should be aggregated together. Hence, degrees-of-freedom at the same location that are also algebraic neighbors are in a unique class of exceptionally strong connections.

In order to better visualize a conforming aggregation step, Fig. 4 shows a sample conforming aggregation on level 0. The black triangles are the element boundaries and each element is shrunk towards its barycenter in order to depict the discontinuous nature of the discretization. The colored polygons and lines represent aggregates.

4.3. Distance-based aggregation

Building on the idea of conforming aggregation and restricting ourselves to isotropic problems, we next consider a distance-based strength-of-connection measure. For isotropic diffusion on an isotropic mesh, where algebraically smooth error is slowly varying in all directions, aggregates should be evenly sized and convex in shape. Thus, a distance-based measure is appropriate. Since we (and classic SA) assume access to the underlying mesh points, we define a distance-based measure. The weak neighbors for a degree-of-freedom i are those degrees-of-freedom whose distance from i is larger than twice the distance of the nearest neighbor.

This approach is computationally inexpensive, because the strength matrix S essentially becomes A, with A's entries replaced with distance values. Entries in S are then dropped according to the criteria above. Aggregation then proceeds as normal, making the computation of S on the finest level the only algorithmic modification of this approach. Advantageously, use of a distance-based measure on the finest level results in a lower operator complexity than using only a conforming step on the finest level. This is because the distance-based measure coarsens more aggressively by combining both a conforming aggregation step and aggregation of nearby degrees-of-freedom into the fine level aggregation phase.

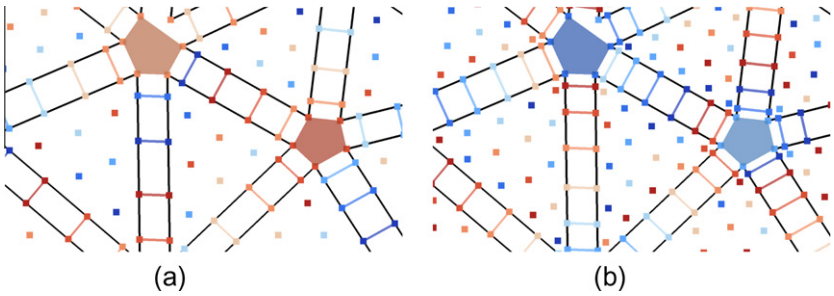

Fig. 4. Example of conforming aggregation in the interior of a domain (a) $p = 5$ Conforming Aggregation, (b) $p = 7$ Conforming Aggregation.

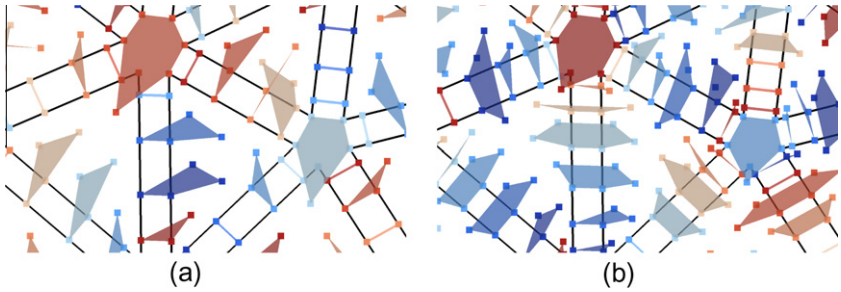

Fig. 5. Example distance-based aggregation in domain interior (a) $p = 5$, (b) $p = 7$.

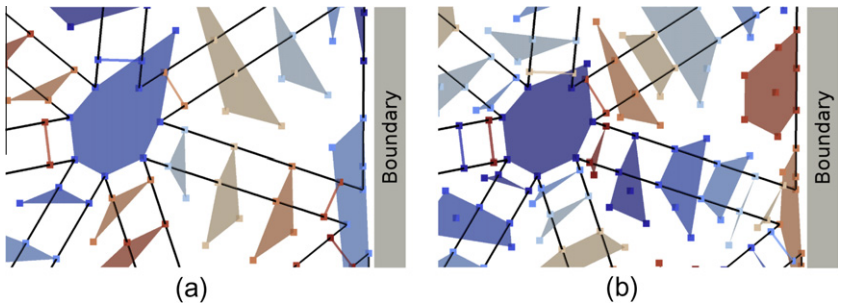

Fig. 6. Example distance-based aggregation on domain boundary (a) $p = 5$, (b) $p = 7$.

The distance-based measure provides effective aggregates on the finest level as shown in Figs. 5 and 6. Example aggregation in the domain interior is shown in Fig. 5, where degrees-of-freedom at the same spatial location are aggregated together and degrees-of-freedom in the interior of an element are aggregated isotropically. Fig. 6 shows similar aggregation at the domain boundary.

In conclusion, the distance-based measure effectively aggregates on level 0, but only for isotropic problems. Additionally, it is straightforward to implement and requires little computational cost. Although we consider only isotropic problems in the remainder of the paper, we remark on several extensions to the anisotropic case. For example, in the case of a rotated anisotropic problem, the algebraically smooth error only varies slowly in one direction, resulting in a poor measure of strength when using a distance-based approach. In the anisotropic setting, conforming aggregation on the finest level is appropriate, because no assumptions are made about the nature or direction of algebraically smooth error. On subsequent levels, the Evolution Measure is effective since the problem specific nature of algebraically smooth error is taken into account.

4.4. Hybrid strength schemes and domain boundary considerations

We now consider using the distance-based and conforming strength strategies in a multilevel method. Vertex information is generally only available on level 0, thus necessitating a hybrid approach, using the Evolution Measure or the classic measure on coarse levels.

The boundary poses an unusual difficulty because the discontinuous Galerkin discretization leaves all Dirichlet degrees-of-freedom in the matrix and enforces Dirichlet conditions weakly. Thus, SA must aggregate and interpolate Dirichlet degrees-of-freedom, a task for which classic SA is not designed. With respect to interpolation, we begin with the standard choice for diffusion problems of $B = \mathbf{1}$. However, it is critical that B be adapted to A, so that interpolation is improved near the boundaries—i.e., B should be close to 0 near the boundaries. For this, we employ relaxation on $AB = \mathbf{0}$ to effectively enforce the boundary conditions on B; this is a critical part of the solver's overall performance.

This introduces, however, a problematic situation for the Evolution Measure, because B is now on the order of 10^{-4} or 10^{-5} at the domain boundary, but on the order of 1 in the interior. This change in B occurs even over the scale of just one element. Consider measure (15). Since B is small near the boundary, but on the order of 1 in the interior, this yields degrees-of-freedom on the boundary that have strong connections only with other boundary degrees-of-freedom. However, for degrees-of-freedom in the domain interior and in elements on the boundary, there exist strong connections to same-element boundary degrees-of-freedom. As a result, the values in S are typically on the order of 10^3 or 10^4 for same-element connections from the boundary to the interior, but on the order of 1 for same-element connections from the interior to the boundary. Smaller entries imply a stronger connections. Consequently, the Evolution Measure gives nonsymmetric strength information at the domain boundary.

This nonsymmetric strength information is correct, despite the fact that the matrix A is symmetric and based on an elliptic PDE. Degrees-of-freedom near the Dirichlet boundary are strongly connected to the degrees-of-freedom on the Dirichlet boundary, but not vice-versa. Since strength information communicates the accuracy of interpolation for algebraically smooth error between two degrees-of-freedom, we consider the desired action of interpolation in order to highlight this subtle issue. Interpolation from the interior to the Dirichlet boundary is appropriate, because interpolation targets zero boundary values with small interpolation weights. However, interpolation from the Dirichlet boundary to the domain interior is inaccurate because the solution value on the boundary, which is approximately 0, contains no useful information about algebraically smooth error in the interior.

This is problematic because each aggregate is represented by a single coarse grid variable, and this coarse grid variable (and corresponding coarse grid matrix row), must accurately represent algebraically smooth error for the entire aggregate on the fine grid. This is achieved by using symmetric strength information. For example, a strong connection between i and j indicates that algebraically smooth error can be accurately interpolated from i to j and from j to i.

Therefore, we propose to symmetrize the Evolution Measure [29], so that the update $S \leftarrow S + S^T$ occurs before the drop-tolerance is applied. This change greatly improves aggregation at the boundary and we use it in our experiments; if the unsymmetrized version is used, performance deteriorates significantly.

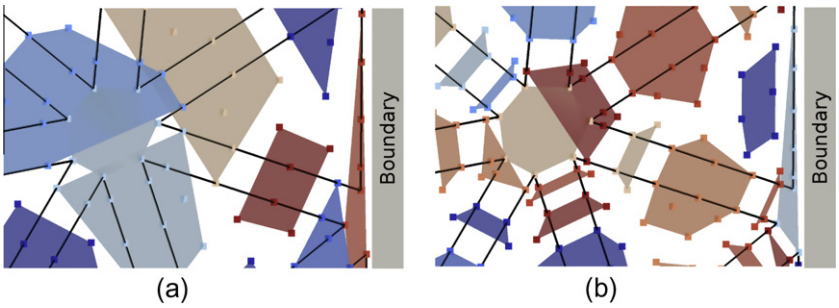

Fig. 7. Example Level 1 Evolution Measure aggregation on domain boundary, conforming aggregation used on level 0 (a) $p = 5$, Symmetrized S, (b) $p = 7$, Symmetrized S.

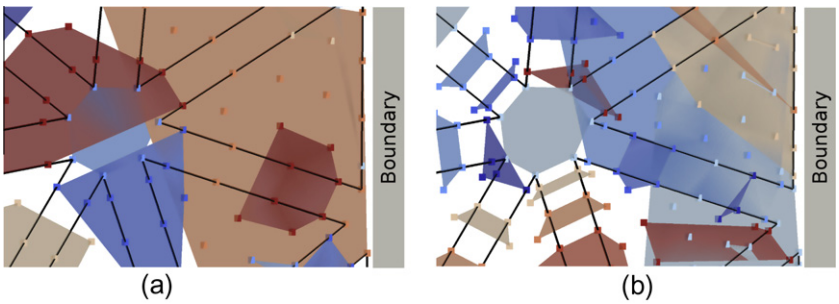

Fig. 8. Example Level 1 Evolution Measure aggregation on domain boundary, conforming aggregation used on level 0 (a) $p = 5$, Unsymmetrized S, (b) $p = 7$, Unsymmetrized S.

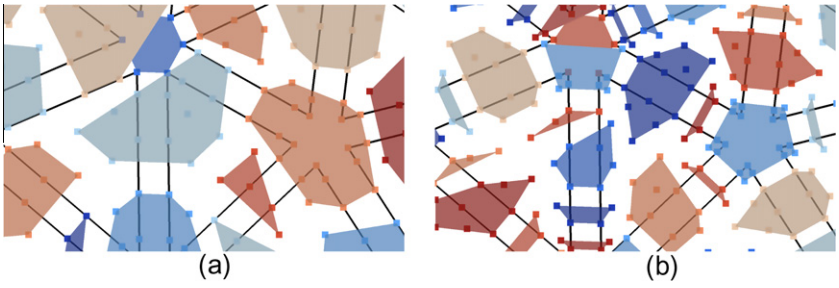

Fig. 9. Example Level 1 Evolution Measure aggregation in domain interior, conforming aggregation used on level 0 (a) $p = 5$, Symmetrized or Unsymmetrized S, (b) $p = 7$, Symmetrized or Unsymmetrized S.

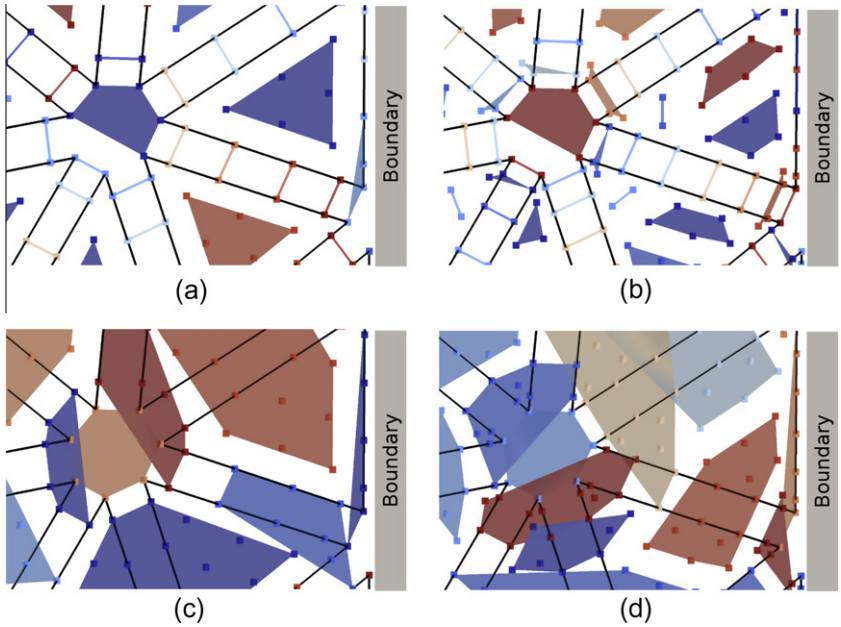

Fig. 10. Example stand-alone Evolution Measure aggregation on domain boundary (a) $p = 5$, Level 0, (b) $p = 7$, Level 0, (c) $p = 5$, Level 1, (d) $p = 7$, Level 1.

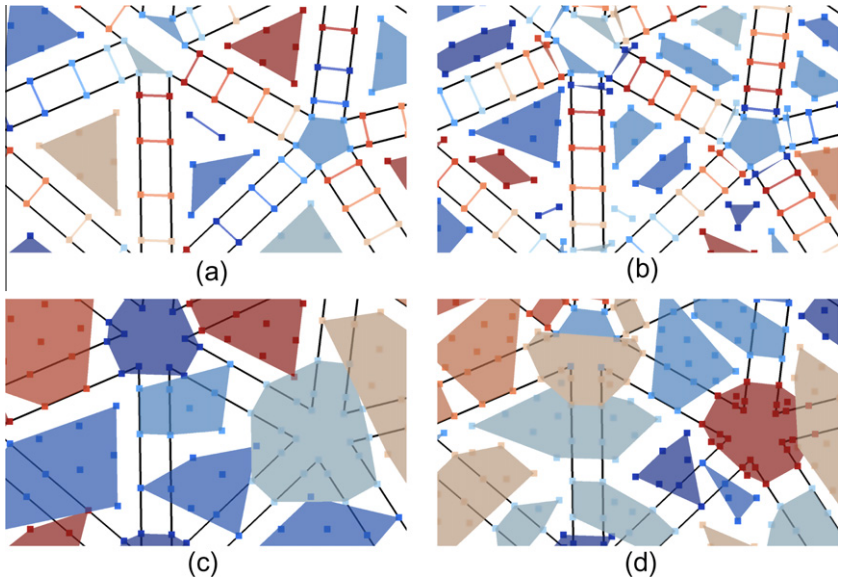

Fig. 11. Example stand-alone Evolution Measure aggregation in domain interior (a) $p = 5$, Level 0, (b) $p = 7$, Level 0, (c) $p = 5$, Level 1, (d) $p = 7$, Level 1.

To illustrate the importance of symmetrizing S, Figs. 7 and 8 depict aggregation at the domain boundary for a symmetrized and unsymmetrized S, respectively. The aggregates are projected from level 1, because a conforming step is used on the finest level. In Fig. 7, aggregates are sensible with boundary degrees-of-freedom aggregated together and degrees-of-freedom in the interior aggregated isotropically. If the Evolution Measure is not modified, then very poor aggregation occurs, yielding irregular and oversized aggregates near the boundary as depicted in Fig. 8. The result is that SA performance suffers.

In the domain interior, symmetric and nonsymmetric S yield similar results. Sample aggregation is depicted in Fig. 9, where degrees-of-freedom at the same spatial location are aggregated together and degrees-of-freedom in the interior of an element are aggregated isotropically. The difficulty posed by using a nonsymmetric S has not been observed for problems

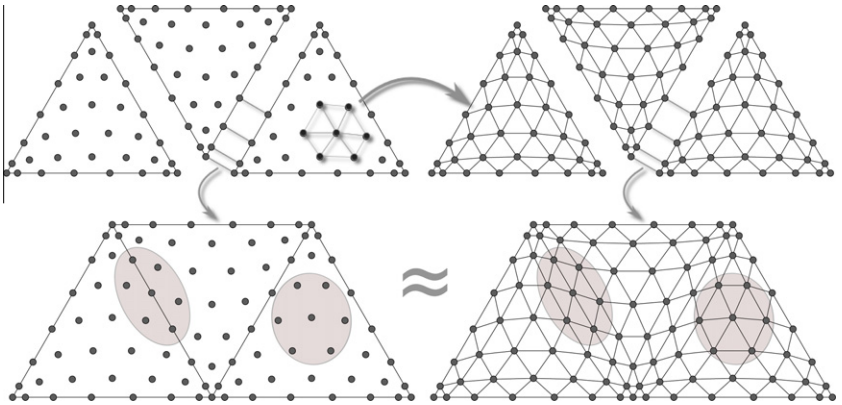

Fig. 12. Low-order finite element preconditioning through aggregation and the relationship to direct high-order aggregation. *Top-left*: non-conforming spectral mesh; *top-right*: non-conforming low-order finite element mesh; *bottom-right*: conforming low-order finite element mesh; *bottom-left*: conforming spectral mesh.

Table 1
High-order SA frameworks.

		Near null-space smoothing	Pre- post-smoothing	Strength measure	Prolongation smoothing
Naive	All levels	5 GS	1 GS	Evol. $k = 2$	EMin. $\eta = 4$
Block	Level 0	p Block GS	1 Block GS	Evol. $k = 4$	EMin. $\eta = p + 2$
	Levels 1+	p GS	1 GS	↓	↓
Distance	Level 0	p Block GS	1 Block GS	Distance	EMin. $\eta = p + 2$
	Levels 1+	p GS	1 GS	Evol. $k = 4$	↓
Conforming	Level 0	p Block GS	1 Block GS	Conforming	None
	Levels 1+	p GS	1 GS	Evol. $k = 4$	EMin. $\eta = p + 2$

where continuous Galerkin discretizations are used. The critical difference is that no boundary degrees-of-freedom are explicitly represented in the matrix.

4.5. Stand-alone Evolution Measure aggregation

Stand-alone use of the Evolution Measure on all levels of the hierarchy also produces an effective SA solver, particularly at low to moderate p. We therefore visualize how the stand-alone Evolution Measure aggregates. We show only aggregation for the symmetrized Evolution Measure, as the unsymmetrized version again has very poor aggregation near the boundary. Figs. 10 and 11 show example aggregation on levels 0 and 1 on the domain boundary and in the domain interior, respectively. The Evolution Measure aggregates as expected in both cases.

4.6. Connection to low-order preconditioners

An additional motivation for the proposed aggregation strategies is from the construction of low-order finite element preconditioners for the spectral problem [3]. In this approach, a low-order finite element mesh based on the high-order mesh is constructed, as depicted in Fig. 12 (top-left to top-right). From the low-order mesh, AMG performs well after a conforming step or by using long-range interpolation [2] (bottom-right). An artifact of our approach is that it produces aggregates that reasonably reflect this low-order process, as illustrated in the example aggregates given by the shaded regions in the bottom row of figures.

5. Proposed solvers

We now combine the above algorithmic components for computational experiments. In the algebraic setting, there is no concept of distance on coarse levels, so use of either the distance-based measure or conforming aggregation on the finest level must be coupled with either the classic measure (12) or the Evolution Measure on coarse levels to yield a hybrid strength-of-connection scheme.

Table 1 outlines the proposed solver components. The conforming and distance-based frameworks are the most advanced because of the special fine level strength-of-connection strategies discussed above. In our numerical experiments, we use Gauss-Seidel smoothing (denoted "j GS" for j sweeps), along with the Evolution Measure (denoted "Evol." with k sweeps) and energy-minimization prolongation smoothing to improve interpolation (denoted "EMin." with η iterations). We label the classic use of SA as the "Naive" method, while "Block" refers to the use of D_{block}^{-1}. The hybrid strength frameworks, "Conforming" and "Distance", correspond to the block framework enhanced on the finest level with conforming aggregation and the distance-based strength measure, respectively.

Table 2
PCG iterations.

p	h	Naive framework				Block methods framework			
		Jacobi		En-Min		Jacobi		En-Min	
		Classic	Evol.	Classic	Evol.	Classic	Evol.	Classic	Evol.
1	1/16	26	10	23	9	25	9	23	8
	1/32	38	10	31	9	37	9	30	9
	1/64	65	10	50	9	64	9	51	9
	1/128	122	12	94	9	121	12	95	8
2	1/8	29	10	27	8	29	9	27	8
	1/16	34	11	29	8	34	11	28	8
	1/32	50	14	35	9	49	14	35	8
	1/64	86	19	57	9	86	20	54	9
3	1/8	35	11	32	8	36	11	32	8
	1/16	40	12	34	8	40	12	34	8
	1/32	59	14	46	9	59	15	44	8
	1/64	99	20	76	9	102	21	71	9
4	1/4	33	20	32	15	33	19	31	15
	1/8	40	19	39	17	39	18	39	16
	1/16	44	19	41	17	44	19	41	16
	1/32	73	22	66	17	71	21	66	17
5	1/4	30	22	28	20	29	23	27	20
	1/8	35	25	33	22	34	26	33	21
	1/16	48	29	42	23	47	29	41	22
	1/32	83	39	73	24	82	49	72	23
6	1/4	34	31	31	26	33	30	29	22
	1/8	52	36	36	27	51	38	35	24
	1/16	73	41	49	29	72	49	46	24
7	1/4	45	37	41	25	42	33	39	20
	1/8	50	36	44	27	48	31	42	22
	1/16	76	42	50	27	74	44	49	25
8	1/4	65	47	55	25	60	42	50	19
	1/8	80	40	60	26	74	39	57	20
	1/16	111	42	66	25	107	43	61	21
9	1/4	92	55	72	47	80	38	63	22
	1/8	112	72	78	53	101	56	68	24
	1/16	151+	78	86	62	151+	61	75	25
10	1/2	75	43	66	36	49	25	46	18
	1/4	129	80	100	52	102	44	76	21
	1/8	151+	74	106	54	136	48	79	23
11	1/2	88	57	82	53	53	28	50	23
	1/4	145	88	123	65	102	54	83	30
	1/8	151+	102	140	69	146	59	91	30

Table 3
PCG iterations.

p	h	Distance-based framework				Conforming framework			
		Jacobi		En-Min		Jacobi		En-Min	
		Classic	Evol.	Classic	Evol.	Classic	Evol.	Classic	Evol.
1	1/16	9	9	9	9	14	12	12	9
	1/32	9	10	9	9	16	17	14	9
	1/64	10	10	9	9	17	17	14	9
	1/128	10	10	10	10	17	24	16	9
2	1/8	10	10	9	9	12	10	12	8
	1/16	11	11	11	9	15	15	13	9
	1/32	13	11	12	9	18	25	15	9
	1/64	14	12	13	9	19	45	16	10
3	1/8	13	11	11	8	16	11	14	8
	1/16	15	12	13	9	17	13	14	9
	1/32	18	16	15	10	20	19	17	9
	1/64	19	27	16	11	21	29	18	9
4	1/4	17	17	10	8	14	10	13	8
	1/8	17	16	13	8	21	14	18	9
	1/16	20	19	16	8	23	18	19	9
	1/32	23	22	18	9	28	27	22	9
5	1/4	18	17	12	8	17	11	15	7
	1/8	19	17	15	10	26	11	23	8
	1/16	23	22	18	10	30	15	24	9
	1/32	27	23	20	11	35	21	28	9
6	1/4	37	36	15	11	22	14	20	8
	1/8	39	38	17	12	34	16	30	8
	1/16	36	35	20	12	40	24	31	9
7	1/4	36	34	15	11	27	14	25	10
	1/8	40	38	19	13	41	17	36	11
	1/16	41	40	22	13	51	25	39	11
8	1/4	48	46	17	15	35	19	32	11
	1/8	51	48	21	15	56	21	47	13
	1/16	56	51	25	15	68	28	51	13
9	1/4	43	39	19	15	40	19	39	12
	1/8	63	61	23	17	64	20	55	13
	1/16	80	73	27	16	76	25	60	13
10	1/2	31	30	12	12	21	18	21	11
	1/4	66	58	21	16	48	26	48	17
	1/8	65	62	27	17	78	29	68	18
11	1/2	27	26	14	14	24	21	23	17
	1/4	69	61	24	16	56	30	55	19
	1/8	78	55	31	18	85	31	78	20

6. Results

A numerical study in h and p is done. The LDG, IP and Brezzi et al. methods are used to discretize (1) with the strong-weak form. Our experiments with uniform Dirichlet boundary conditions and mixtures of Neumann and Dirichlet yielded no qualitative difference in performance, so we reproduce results primarily for the Dirichlet case. Using Sledge++, matrices are generated for non-nested unstructured simplicial meshes of the unit box in 2D for decreasing h and various p. W(1,1)-cycles precondition CG to a relative residual tolerance of 10^{-8}, with the number of iterations capped at 150. W(1,1)-cycles are used because they exhibit grid independent behavior in cases that V-cycles do not. The result is that W-cycles give lower cost per digit-of-accuracy than V-cycles. The coarsest grid is 100 degrees-of-freedom. The test problem is a zero initial guess and a random right-hand side.

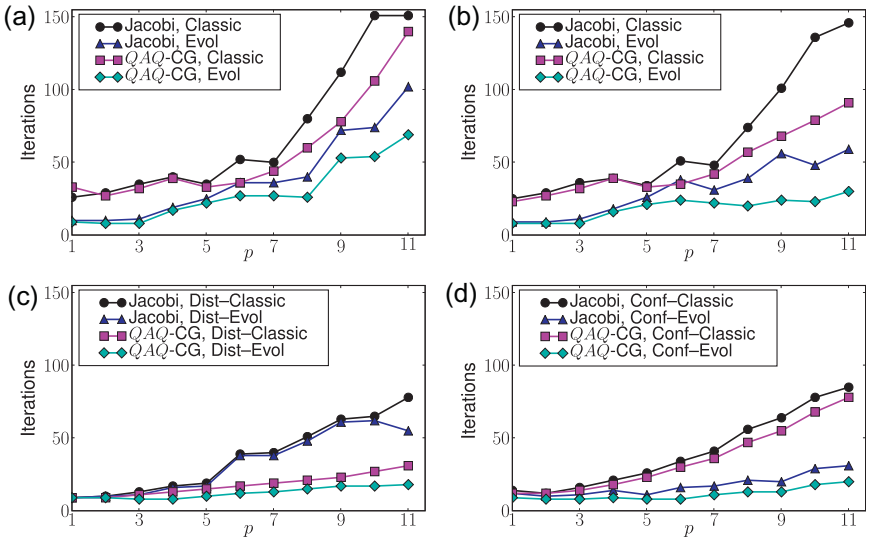

Fig. 13. PCG Iterations for constant $h = 1/8$ (a) Naive Framework, (b) Block Methods Framework, (c) Distance Framework, (d) Conforming Framework.

Table 4
PCG iterations, $h = 1/8$, Dirichlet boundary conditions.

	p=	1	2	3	4	5	6	7	8	9	10	11
LDG	Dist.–Evol.	9	9	8	8	10	12	13	15	17	17	18
	Conf.–Evol.	9	8	8	9	8	8	11	13	13	18	20
IP	Dist.–Evol.	9	8	8	8	12	9	12	15	16	16	16
	Conf.–Evol.	7	8	9	9	9	10	11	13	13	17	21
Brezzi	Dist.–Evol.	10	9	9	8	12	9	13	15	16	16	17
	Conf.–Evol.	10	9	9	9	10	10	11	13	13	16	21

Table 5
PCG iterations, $h = 1/8$, mixed Dirichlet and Neumann boundary conditions.

	p=	1	2	3	4	5	6	7	8	9	10	11
LDG	Dist.–Evol.	9	8	8	9	15	10	16	18	20	19	19
	Conf.–Evol.	8	8	9	9	10	11	12	13	15	19	24
IP	Dist.–Evol.	9	8	8	9	13	10	15	16	16	16	17
	Conf.–Evol.	8	8	9	9	10	10	12	13	13	18	22
Brezzi	Dist.–Evol.	11	9	9	9	14	10	15	17	18	16	18
	Conf.–Evol.	10	9	9	9	10	10	12	13	13	17	21

Tables 2 and 3 depict the performance for the four different SA frameworks in Table 1. We restrict ourselves to showing results for the LDG method, because the results for the IP and Brezzi et al. methods were qualitatively no different. For the purposes of comparison, we also experiment with weighted-Jacobi prolongation smoothing and the classic strength measure.

The classic measure is not well-suited to this complex algebraic setting of non-M-matrices, but $\theta = 0.1$ is empirically a fair choice for general p and is used here. Additionally, the use of filtered-matrix prolongation smoothing for classic weighted-Jacobi was critical for controlling complexity—e.g., an operator-complexity of 9 versus 2.25. Filtered-matrix prolongation smoothing essentially lumps weak connections on the diagonal of A for only the prolongation smoothing step. The sparsity

pattern constraint for energy-minimization prolongation smoothing accomplishes the same goal of controlling complexity. In general, operator complexities are in the range 2.25–2.50 for all frameworks except the conforming framework, which has operator complexities in the range 2.60–3.20.

Table 2 gives results for the naive framework from Table 1. It is observed that the combination of energy-minimization prolongation smoothing and the Evolution Measure provides h-independence until $p = 6$ and p-independence until $p = 4$. No other method in Table 2 is competitive for $p > 1$ with the combination of energy-minimization prolongation smoothing and the Evolution Measure.

Table 2 also gives results for the block methods framework from Table 1, which generally yields better PCG iteration counts than the naive framework for $p > 6$. Combination of energy-minimization prolongation smoothing and the Evolution Measure provides h-independence until $p = 7$, with a slow degradation for increasing h and fixed p thereafter . There is p-independence again until $p = 4$. No other method in Table 2 is competitive for $p > 1$ with the combination of energy-minimization prolongation smoothing and the Evolution Measure.

Table 3 gives results for the "Distance" framework from Table 1. The hybrid strength schemes are referred to as "Dist.–Classic" and "Dist.–Evol." and correspond to the use of the distance-based measure on the finest level and the classic measure and the Evolution Measure on coarse levels, respectively. Importantly, $\theta = 0.1$ for the classic measure is detrimental to performance for the hybrid strength schemes, but $\theta = 0.0$ performed well and is used here for the two hybrid schemes. Despite tuning θ for the classic measure, the "Dist.–Evol." scheme exhibits superior performance beyond $p = 2$. In particular, the combination of energy-minimization prolongation smoothing and "Dist.–Evol." yields h-independence until $p = 9$, with a small degradation for increasing h at $p = 10$ and 11. Additionally, p-independence is extended until $p = 6$. Degradation for increasing p and fixed h is thereafter slow.

Table 3 also gives results for the "Conforming" framework from Table 1. The hybrid strength schemes are referred to as "Conf.–Classic" and "Conf.–Evol." and correspond to the use of conforming aggregation on the finest level and the classic measure and the Evolution Measure on coarse levels, respectively. The results are largely similar to the distance framework, with combination of "Conf.–Evol." and energy-minimization prolongation smoothing providing the most robust method. The one downside to using conforming aggregation on the finest level is that operator complexities are higher because this is the least aggressive coarsening approach.

In order to better visualize how iterations increase with p, Fig. 13 depicts plots of p versus PCG iterations, for a constant $h = 1/8$. The data is taken from Tables 2 and 3. This is the only h value shared by all p values. The two hybrid strength schemes produce the solvers with behavior closest to p-independence.

In order to provide an indication of how the solvers perform consistently for different boundary conditions and for the three discontinuous Galerkin methods considered, we present Tables 4 and 5. Table 4 presents PCG iterations for a constant $h = 1/8$ and $p = 1,2,\ldots,11$ for each of the three discretizations and the case of uniform Dirichlet boundary conditions. Table 5 does the same, but for the case of Neumann boundary conditions on the North and South walls of the unit box and Dirichlet boundary conditions on the East and West walls of the unit box. We show results for the two most successful frameworks, Conf.–Evol and Dist.–Evol., which exhibit consistent performance.

7. Conclusions and future work

The proposed methods are the first scalable AMG methods for high-order discontinuous Galerkin discretizations of the Poisson operator. SA has been extended to address the following difficulties posed by this problem: the rising condition number of A and loss of locality in each matrix row as p increases; the inclusion of Dirichlet degrees-of-freedom in the matrix; and the coarsening needs of the high-order discontinuous Galerkin discretization. These difficulties motivate the proposed specialized frameworks that provide appropriate relaxation, interpolation and aggregation.

Strength-of-connection is a critical component on both the finest level and coarse levels. This is evidenced by the fact that the hybrid strength schemes exhibit better performance than application of the Evolution Measure at all levels. Additionally when using the same strength measure on the finest level (distance-based or conforming), use of the Evolution Measure on coarse levels is superior to use of the classic measure on coarse levels.

Prolongation smoothing is another important component. Use of energy-minimization prolongation smoothing consistently out-performs classic weighted-Jacobi prolongation smoothing.

Given the increasing condition numbers in Fig. 3, it is not surprising that p-independence is difficult. Thus, future work will focus on achieving better performance at high p for purely algebraic methods. Additionally, the scope of problem types should be expanded to convection–diffusion and anisotropic diffusion.

References

[1] J.W. Lottes, P.F. Fischer, Hybrid multigrid/Schwarz algorithms for the spectral element method, J. Sci. Comput. 24 (2005) 45–78.
[2] L. Olson, Algebraic multigrid preconditioning of high-order spectral elements for elliptic problems on a simplicial mesh, SIAM J. Sci. Comput. 29 (5) (2007) 2189–2209.
[3] J.J. Heys, T.A. Manteuffel, S.F. McCormick, L.N. Olson, Algebraic multigrid for higher-order finite elements, J. Comput. Phys. 204 (2) (2005) 520–532.
[4] P. Castillo, Performance of discontinuous Galerkin methods for elliptic PDEs, SIAM J. Sci. Comput. 24 (2) (2002) 524–547.
[5] F. Brezzi, G. Manzini, D. Marini, P. Pietra, A. Russo, Discontinuous Galerkin approximations for elliptic problems, Numer. Meth. Partial Diff. Eq. 16 (2000) 365–378.

[6] D.N. Arnold, F. Brezzi, B. Cockurn, L.D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM J. Numer. Anal. 39 (2002) 1749–1779.
[7] B. Cockburn, J. Gopalakrishnan, R. Lazarov, Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems, SIAM J. Numer. Anal. 47 (2) (2009) 1319–1365.
[8] N.C. Nguyen, J. Peraire, B. Cockburn, A hybridizable discontinuous Galerkin method for stokes flow, Comput. Methods Appl. Mech. Eng. 199 (9-12) (2010) 582–597.
[9] Y. Jeon, E.. Park, A hybrid discontinuous Galerkin method for elliptic problems, SIAM J. Numer. Anal. 48 (5) (2010) 1968–1983.
[10] G. Kanschat, Preconditioning methods for local discontinuous Galerkin discretizations, SIAM J. Sci. Comput. 25 (3) (2003) 815–831.
[11] G. Kanschat, Robust smoothers for high-order discontinuous Galerkin discretizations of advection–diffusion problems, J. Comput. Appl. Math. 218 (1) (2008) 53–60 (special Issue: Finite Element Methods in Engineering and Science (FEMTEC 2006)).
[12] H. Atkins, B. Helenbrook, Application of *p*-multigrid to discontinuous Galerkin formulations of the Poisson operator, AIAA J. 44 (3) (2005) 566–575.
[13] K.J. Fidkowski, T.A. Oliver, J. Lu, D.L. Darmofal, *p*-multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations, J. Comput. Phys. 207 (1) (2005) 92–113.
[14] V.A. Dobrev, R.D. Lazarov, P.S. Vassilevski, L.T. Zikatanov, Two-level preconditioning of discontinuous Galerkin approximations of second-order elliptic equations, Numer. Linear Algebra Appl. 13 (9) (2006) 753–770.
[15] J. Gopalakrishnan, G. Kanschat, A multilevel discontinuous Galerkin method, Numer. Math. 95 (3) (2003) 527–550.
[16] P.W. Hemker, W. Hoffmann, M.H. v. Raalte, Two-level Fourier analysis of a multigrid approach for discontinuous Galerkin discretizations, SIAM J. Sci. Comput. 25 (3) (2003) 1018–1041.
[17] F. Prill, M. Lukáčová-Medviďová, R. Hartmann, Smoothed aggregation multigrid for the discontinuous Galerkin method, SIAM J. Sci. Comput. 31 (5) (2009) 3503–3528.
[18] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection–diffusion systems, SIAM J. Numer. Anal. 35 (6) (1998) 2440–2463.
[19] J.J. Douglas, T. Dupont, Interior penalty procedures for elliptic and parabolic Galerkin methods, in: Computing Methods in Applied Sciences, Lecture Notes in Physics, vol. 58, Springer, Berlin, 1976, pp. 207–216.
[20] F. Brezzi, G. Manzini, D. Marini, P. Pietra, A. Russo, Discontinuous finite elements for diffusion problems, in: in Atti Convegno in onore di F. Brioschi (Milan, 1997), Instituto Lombardo, Accademia di Scienze e Lettere, Italy, 1999, pp. 197–217.
[21] P. Castillo, B. Cockurn, I. Perugia, D. Schötzau, An a priori error analysis of the local discontinuous Galerkin method for elliptic problems, SIAM J. Numer. Anal. 38 (2000) 1676–1706.
[22] B. Cockurn, G. Kanschat, I. Perugia, D. Schötzau, Superconvergence of the local discontinuous Galerkin method for elliptic problems on Cartesian grids, SIAM J. Numer. Anal. 39 (2001) 264–285.
[23] J. Hesthaven, T. Warburton, Nodal discontinuous Galerkin methods: algorithms, analysis, and applications, Texts in applied mathematics, vol. 54, Springer, Berlin, 2008.
[24] S.F. McCormick, J.W. Ruge, Multigrid methods for variational problems, SIAM J. Numer. Anal. 19 (5) (1982) 924–929.
[25] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, J. Ruge, Adaptive smoothed aggregation (αSA) multigrid, SIAM Rev. 47 (2) (2005) 317–346.
[26] M. Brezina, T. Manteuffel, S. Mccormick, J. Ruge, G. Sanders, Towards adaptive smoothed aggregation (αSA) for nonsymmetric problems, SIAM J. Sci. Comput.
[27] P. Vaněk, J. Mandel, M. Brezina, Algebraic multigrid based on smoothed aggregation for second and fourth order problems, Computing 56 (1996) 179–196.
[28] J. Brannick, M. Brezina, S. MacLachlan, T. Manteuffel, S. McCormick, J. Ruge, An energy-based AMG coarsening strategy, Numer. Linear Algebra Appl. 13 (2-3) (2006) 133–148.
[29] L.N. Olson, J.B. Schroder, R.S. Tuminaro, A new perspective on strength measures in algebraic multigrid, Numer. Linear Algebra Appl. 17 (4) (2010) 713–733.
[30] L.N. Olson, J.B. Schroder, R.S. Tuminaro, A general interpolation strategy for algebraic multigrid using energy-minimization, SIAM J. Sci. Comput. Submitted.
[31] T. Warburton, J. Hesthaven, L. Wilcox, Sledge++ users' guide (2006). <http://www.caam.rice.edu/timwar/TimWarburton/Sledge++.html>.
[32] P. Vaněk, M. Brezina, J. Mandel, Convergence of algebraic multigrid based on smoothed aggregation, Numer. Math. 88 (3) (2001) 559–579.