

ROBUST MULTIGRID FOR HIGH-ORDER DISCONTINUOUS GALERKIN METHODS: A FAST POISSON SOLVER SUITABLE FOR HIGH-ASPECT RATIO CARTESIAN GRIDS

JÖRG STILLER

Technische Universität Dresden, Institute of Fluid Mechanics and Center for Advancing Electronics Dresden, 01062 Dresden, Germany

ABSTRACT. We present a polynomial multigrid method for nodal interior penalty and local discontinuous Galerkin formulations of the Poisson equation on Cartesian grids. For smoothing we propose two classes of overlapping Schwarz methods. The first class comprises element-centered and the second face-centered methods. Within both classes we identify methods that achieve superior convergence rates, prove robust with respect to the mesh spacing and the polynomial order, at least up to $P = 32$. Consequent structure exploitation yields a computational complexity of $O(PN)$, where N is the number of unknowns. Further we demonstrate the suitability of the face-centered method for element aspect ratios up to 32.

1. INTRODUCTION

High-order discretization methods are exciting because of their promise to deliver higher accuracy at lower cost than first and second order methods. Much confidence has been put in the discontinuous Galerkin (DG) method because it combines multiple desirable properties of finite element and finite volume methods, including geometric flexibility, variable approximation order, straightforward adaptivity and suitability for conservation laws [6, 19]. Traditionally, DG methods have been used in the numerical solution of hyperbolic and convection-dominated problems. Nevertheless, the need for implicit diffusion schemes and application to other problem classes, such as incompressible flow and elasticity, led to a growing interest in DG methods and related solution techniques for elliptic equations [2, 30].

The most efficient elliptic solvers are based on multigrid (MG) techniques and can be classified into polynomial or p -MG [11, 17, 18], geometric or h -MG [10, 15, 22, 23, 25, 31] or, combining both concepts, hp -MG [1, 35], and algebraic MG [3, 28, 29, 32]. Apart from their different coarsening strategy, polynomial and geometric multigrid are closely related to each other and can be applied with the same smoothing methods. Early work

E-mail address: joerg.stiller@tu-dresden.de.

Key words and phrases. Discontinuous Galerkin, elliptic problems, multigrid method, Schwarz method, Krylov acceleration, Cartesian grids.

The final article is available at <http://dx.doi.org/10.1016/j.jcp.2016.09.041>.

on p -MG goes back to Helenbrook and coworkers [17, 18] who explored various smoothers for DG formulations of the Poisson equation. For isotropic grids they identified block Gauss-Seidel as the best choice, whereas more expensive line smoothing proved necessary on high-aspect-ratio grids. At about the same time, Gopalakrishnan and Kanschat [15] developed h -MG preconditioners for Poisson and convection-diffusion problems, which also use element-based block-Gauss-Seidel methods for smoothing. Kanschat [22, 23] extended this approach to locally refined Cartesian grids in two and three space dimensions. In both cases, polynomial and geometric multigrid, block-Gauss-Seidel smoothing yields acceptable convergence rates for low to moderate polynomial degrees, e.g. $\rho \approx 0.5$ with one pre-smoothing for $P = 4$. However, the convergence degrades with increasing P , which renders the approach unfeasible for higher polynomial degrees.

Several researchers proposed algebraic multigrid methods for various DG formulations of elliptic equations. Olson and Schroder [28] presented a preconditioned conjugate gradient (PCG) method based on smoothed aggregation. Using block relaxation combined with energy-minimizing prolongation it attains mesh independent convergence rates corresponding to residual reductions of about 0.3 per smoothing step for $P \leq 6$, but degrades with increasing approximation order. Bastian et al. [3] proposed a non-smoothed aggregation approach. For smoothing they use block relaxations which operate on extended aggregates and, hence, can be regarded as overlapping Schwarz methods. This approach yields by far the most efficient DG-MG method reported until now. The method proved robust with respect to the polynomial order, up to at least $P = 6$, though the iteration count exhibits a logarithmic dependence on the mesh spacing. For the DG method of Oden, Babuška and Baumann it achieved convergence rates of $\rho \approx 0.04$ with one pre- and post-smoothing, which corresponds to a residual reduction by a factor of 25 in one step. The approach was also shown to work with symmetric and non-symmetric interior penalty methods, although it required nearly twice as many iterations in the latter case. A possible drawback is the rise of cost with increasing polynomial order. The authors did not specify the complexity of their algorithm, however the solver runtimes indicate that the cost per unknown grows as P^3 in 2D and P^5 in 3D.

To the best of our knowledge, none of the proposed MG methods is robust with respect to both, the polynomial order and the mesh spacing. Computational complexity and robustness against high aspect ratios are further issues that need to be considered to strengthen the competitiveness of DG methods for elliptic equations. As a step into that direction we present a new p -multigrid method for interior penalty and local discontinuous Galerkin discretizations of the Poisson equation on Cartesian grids. Our approach is motivated and strongly influenced by previous work dedicated to the continuous spectral element method [16, 20, 26, 33]. We propose two classes of multiplicative and weighted additive Schwarz methods, which use an adjustable overlap depending on the polynomial level. The first class comprises element-centered and the second face-centered methods. Within both classes we identify methods that achieve superior convergence rates, prove robust with respect to the mesh spacing and the polynomial order and reach a computational complexity of $O(PN)$, where N is the number of unknowns. Further we demonstrate the suitability of the face-centered method for high element aspect ratios.

The paper is organized as follows: In the next section we derive a unified nodal DG formulation of the Poisson problem comprising the symmetric interior penalty method and the local discontinuous Galerkin method. Then we describe the solution methods, i.e. Schwarz, multigrid, and inexact PCG, in Section 3. Section 4 presents the numerical experiments and Section 5 concludes the paper.

2. DISCONTINUOUS GALERKIN METHOD

2.1. Problem definition. As a model problem we consider the Poisson equation

$$-\nabla^2 u = f \quad (1)$$

in the rectangular periodic domain $\Omega = [0, l_1] \times [0, l_2]$. By introducing the flux vector $\boldsymbol{\sigma} = \nabla u$, the problem can be rewritten into the first-order system

$$\boldsymbol{\sigma} = \nabla u, \quad (2a)$$

$$-\nabla \cdot \boldsymbol{\sigma} = f. \quad (2b)$$

This form serves as the starting point for the discontinuous Galerkin method.

2.2. Spatial discretization. The domain Ω is decomposed into $N_E = N_{E,1} \times N_{E,2}$ rectangular elements

$$\Omega^{m_1, m_2} = (x_1^{m_1-1/2}, x_1^{m_1+1/2}) \times (x_2^{m_2-1/2}, x_2^{m_2+1/2})$$

with dimensions $\Delta x_d^{m_d} = x_d^{m_d+1/2} - x_d^{m_d-1/2}$ for $d = 1, 2$. Each element is mapped to the standard region $(-1, 1)^2$ by

$$\xi_d^{m_d}(x_d) = \frac{2}{\Delta x_d^{m_d}}(x_d - x_d^{m_d-1/2}) - 1.$$

For simplicity we use the array notations $\mathbf{m} = (m_1, m_2)$, $\mathbf{x} = (x_1, x_2)$ and $\boldsymbol{\xi} = (\xi_1, \xi_2)$ where possible.

Let $\{\varphi_i(\xi)\}_{i=0}^P$ be a polynomial basis of degree P in the interval $[-1, 1]$. Using a tensor-product ansatz the solution to (2) can be approximated in $\Omega^{\mathbf{m}}$ as

$$u_h(\mathbf{x})|_{\Omega^{\mathbf{m}}} = u^{\mathbf{m}}(\boldsymbol{\xi}^{\mathbf{m}}(\mathbf{x})) = \sum_{i,j=0}^P u_{ij}^{\mathbf{m}} \varphi_i(\xi_1) \varphi_j(\xi_2) \quad (3)$$

and

$$\boldsymbol{\sigma}_h(\mathbf{x})|_{\Omega^{\mathbf{m}}} = \boldsymbol{\sigma}^{\mathbf{m}}(\boldsymbol{\xi}^{\mathbf{m}}(\mathbf{x})) = \sum_{i,j=0}^P \boldsymbol{\sigma}_{ij}^{\mathbf{m}} \varphi_i(\xi_1) \varphi_j(\xi_2). \quad (4)$$

The global solution u_h and the fluxes $\boldsymbol{\sigma}_h$ belong to the function spaces

$$V_h = \left\{ v \in L^2(\Omega) : v|_{\Omega^{\mathbf{m}}} \in Q_p(\Omega^{\mathbf{m}}) \quad \forall \Omega^{\mathbf{m}} \subset \Omega \right\} \quad \text{and}$$

$$\boldsymbol{\Sigma}_h = \left\{ \boldsymbol{\tau} \in [L^2(\Omega)]^2 : \boldsymbol{\tau}|_{\Omega^{\mathbf{m}}} \in [Q_p(\Omega^{\mathbf{m}})]^2 \quad \forall \Omega^{\mathbf{m}} \subset \Omega \right\},$$

respectively, where $Q_p(\Omega^{\mathbf{m}})$ is the space spanned by the tensor-product polynomials $x_1^k x_2^l$ with $0 \leq k, l \leq P$ in $\Omega^{\mathbf{m}}$.

2.3. Elemental DG formulation. Following Cockburn and Shu [5] we consider discontinuous Galerkin formulations of the form: Find $u_h \in V_h$ and $\boldsymbol{\sigma}_h \in \boldsymbol{\Sigma}_h$ such that for all $\Omega^m \subset \Omega$

$$\int_{\Omega^m} \boldsymbol{\tau} \cdot \boldsymbol{\sigma}_h \, d\Omega = - \int_{\Omega^m} (\nabla \cdot \boldsymbol{\tau}) u_h \, d\Omega + \int_{\partial\Omega^m} \boldsymbol{\tau} \cdot \hat{\mathbf{n}} \, d\Gamma \quad \forall \boldsymbol{\tau} \in [Q_p(\Omega^m)]^2, \quad (5a)$$

$$\int_{\Omega^m} \nabla v \cdot \boldsymbol{\sigma}_h \, d\Omega = \int_{\Omega^m} v f \, d\Omega + \int_{\partial\Omega^m} v \hat{\boldsymbol{\sigma}} \cdot \mathbf{n} \, d\Gamma \quad \forall v \in Q_p(\Omega^m), \quad (5b)$$

where the numerical fluxes \hat{u} and $\hat{\boldsymbol{\sigma}}$ are approximations to u and to $\boldsymbol{\sigma} = \nabla u$, respectively, on the element boundary $\partial\Omega^m$. To concretize the numerical fluxes we need some additional notation. For given functions $v \in V_h$, $\boldsymbol{\tau} \in \boldsymbol{\Sigma}_h$ let v^- , $\boldsymbol{\tau}^-$ and v^+ , $\boldsymbol{\tau}^+$ denote the interior and exterior traces on $\partial\Omega^m$, respectively. Now we define the average and jump operators by

$$\{v\} = \frac{1}{2} (v^- + v^+), \quad \{\boldsymbol{\tau}\} = \frac{1}{2} (\boldsymbol{\tau}^- + \boldsymbol{\tau}^+), \quad (6a)$$

$$[[v]] = (v^- - v^+) \mathbf{n}, \quad [[\boldsymbol{\tau}]] = (\boldsymbol{\tau}^- - \boldsymbol{\tau}^+) \cdot \mathbf{n}. \quad (6b)$$

We remark that, despite element oriented notation, these definitions are in fact element independent and equivalent to those given in [2].

For constructing the numerical fluxes we consider the interior penalty method (IP) and the local discontinuous Galerkin method (LDG). We closely follow the notation used in [2]. With IP the numerical fluxes take the form

$$\hat{u}_{\text{IP}} = \{u_h\}, \quad (7a)$$

$$\hat{\boldsymbol{\sigma}}_{\text{IP}} = \{\nabla u_h\} - \mu_{\text{IP}} [[u_h]] \quad (7b)$$

and with LDG

$$\hat{u}_{\text{LDG}} = \{u_h\} - \boldsymbol{\beta} \cdot [[u_h]], \quad (8a)$$

$$\hat{\boldsymbol{\sigma}}_{\text{LDG}} = \{\boldsymbol{\sigma}_h\} + \boldsymbol{\beta} [[\boldsymbol{\sigma}_h]] - \mu_{\text{LDG}} [[u_h]]. \quad (8b)$$

Here μ_{IP} and μ_{LDG} are positive penalty functions that are defined on the edges and typically piecewise constant. According to [2], $\mu_{\text{LDG}} > 0$ is sufficient for stability with LDG, whereas no general stability threshold is known for IP. The auxiliary parameter $\boldsymbol{\beta}$ can be used to improve the sparsity of the stiffness matrix [5]. Moreover, Cockburn et al. [7] devised a special form of $\boldsymbol{\beta}$ which yields superconvergence in the L^2 norm when used with Cartesian grids. This form includes constant vectors with components satisfying $|\beta_d| = 1/2$. As a possible drawback, however, any non-trivial choice of $\boldsymbol{\beta}$ also implies a directional bias and thus breaks the symmetry of diffusive transport.

2.4. Discrete equations. In the following we constrain ourselves to nodal bases formed by the Lagrange polynomials to the Gauss-Lobatto-Legendre (GLL) quadrature points $\{\eta_i\}$ in the standard interval $[-1, 1]$, see e.g. [9, 19, 24]. The GLL quadrature is used for evaluating integrals over element domains, which yields a diagonal mass matrix without degrading the overall accuracy of the method.

Before stating the discrete equations, let us introduce some notation: $\underline{u}^m = [u_{ij}^m]$ represents the vector of solution coefficients in Ω^m and $\underline{u} = [\underline{u}^m]$ the global solution vector.

Underscores indicate nodal vectors and operator matrices, and double indices refer to the directions of the tensor-product decomposition introduced in (3) and (4). Due to the latter, we have to distinguish between one-dimensional and two-dimensional operators. In particular, the Kronecker delta $\delta_{i,k}$ denotes the components of the 1D unit matrix, $M_{i,k}^s = \int \varphi_i \varphi_j d\xi = \rho_i \delta_{i,k}$ the (approximate) standard 1D mass matrix resulting from GLL quadrature with weights ρ_i , and $D_{i,k}^s = \varphi'_k(\eta_i)$ the standard 1D differentiation matrix with respect to the GLL points [9, 24]. The 2D element mass matrix \underline{M}^m constitutes the tensor product of the one-dimensional mass matrices $\underline{M}_1^{m_1}$ and $\underline{M}_2^{m_2}$, i.e.,

$$M_{ij,kl}^m = M_{1,i,k}^{m_1} M_{2,j,l}^{m_2} \quad (9)$$

with

$$M_{d,i,k}^m = \int_{x_d^{m-1/2}}^{x_d^{m+1/2}} \varphi_i(\xi_d) \varphi_k(\xi_d) dx_d = \frac{\Delta x_d^m}{2} M_{i,k}^s, \quad d = 1, 2. \quad (10)$$

Note that the mass matrices are diagonal, which is a welcome side effect of the GLL quadrature.

Next we derive explicit expressions for the discrete fluxes. Application of the divergence theorem to (5a) yields the equivalent form

$$\int_{\Omega^m} \boldsymbol{\tau} \cdot \boldsymbol{\sigma}_h d\Omega = \int_{\Omega^m} \boldsymbol{\tau} \cdot \nabla u_h d\Omega + \int_{\partial\Omega^m} \boldsymbol{\tau} \cdot (\hat{u} - u_h^-) \mathbf{n} d\Gamma.$$

Substituting the numerical flux (7a) or (8a) for \hat{u} , respectively, and using (6) we obtain

$$\int_{\Omega^m} \boldsymbol{\tau} \cdot \boldsymbol{\sigma}_h d\Omega = \int_{\Omega^m} \boldsymbol{\tau} \cdot \nabla u_h d\Omega - \int_{\partial\Omega^m} \left(\frac{1}{2} + \boldsymbol{\beta} \cdot \mathbf{n} \right) \boldsymbol{\tau} \cdot \llbracket u_h \rrbracket d\Gamma,$$

where $\boldsymbol{\beta}$ vanishes for IP and is constant for LDG. Choosing $\boldsymbol{\tau}|_{\Omega^m} = \varphi_i(\xi_1) \varphi_j(\xi_2) \mathbf{e}_d$, substituting the approximate solution (3, 4) and evaluating the integrals by means of GLL quadrature leads to

$$\underline{M}^m \underline{\boldsymbol{\sigma}}^m = \underline{M}^m (\nabla u)^m + \sum_{\gamma=1}^4 \left(\frac{1}{2} + \boldsymbol{\beta} \cdot \mathbf{n}_\gamma \right) \underline{M}_\gamma^\Gamma (u_\gamma^+ - u_\gamma^-) \mathbf{n}_\gamma, \quad (11)$$

where $\underline{\boldsymbol{\sigma}}^m = [\boldsymbol{\sigma}_{ij}^m]$ and

$$(\nabla u)_{ij}^m = \begin{bmatrix} \frac{2}{\Delta x_1^{m_1}} \sum_{p=0}^P D_{i,p}^s u_{pj}^m \\ \frac{2}{\Delta x_2^{m_2}} \sum_{q=0}^P D_{j,q}^s u_{iq}^m \end{bmatrix} \quad (12)$$

are the nodal coefficients of ∇u_h in Ω^m . The last term in (11) comprises the contributions of the boundary edges. See Fig. 1 for illustration and Tab. 1 for the involved operators and variables. Multiplying Eq. (11) with the inverse of the element mass matrix yields

$$\underline{\boldsymbol{\sigma}}^m = (\nabla u)^m + \sum_{\gamma=1}^4 \left(\frac{1}{2} + \boldsymbol{\beta} \cdot \mathbf{n}_\gamma \right) (\underline{M}^m)^{-1} \underline{M}_\gamma^\Gamma (u_\gamma^+ - u_\gamma^-) \mathbf{n}_\gamma. \quad (13)$$

For further simplification it is essential that the boundary operators $\underline{M}_\gamma^\Gamma$ are products

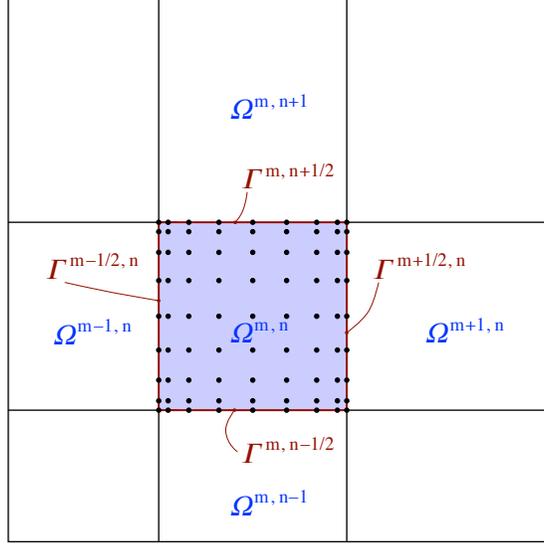
FIGURE 1. Element domain with collocation points of order $P = 8$.

TABLE 1. Element boundary operators and variables.

γ	edge	\mathbf{n}_γ	$\underline{M}_\gamma^\Gamma$	\underline{u}_γ^-	\underline{u}_γ^+
1	$\Gamma^{m_1-1/2,m_2}$	$-\mathbf{e}_1$	$\delta_{i,0} M_{2,j,q}^{m_2}$	$u_{0q}^{m_1,m_2}$	$u_{Pq}^{m_1-1,m_2}$
2	$\Gamma^{m_1,m_2-1/2}$	$-\mathbf{e}_2$	$\delta_{j,0} M_{1,i,p}^{m_1}$	$u_{p0}^{m_1,m_2}$	$u_{pP}^{m_1,m_2-1}$
3	$\Gamma^{m_1+1/2,m_2}$	\mathbf{e}_1	$\delta_{i,P} M_{2,j,q}^{m_2}$	$u_{Pq}^{m_1,m_2}$	$u_{0q}^{m_1+1,m_2}$
4	$\Gamma^{m_1,m_2+1/2}$	\mathbf{e}_2	$\delta_{j,P} M_{1,i,p}^{m_1}$	$u_{pP}^{m_1,m_2}$	$u_{0p}^{m_1,m_2+1}$

of the injection operator for the normal direction and the one-dimensional mass matrix (10) for the tangential direction. For example, for $\gamma = 1$, i.e. $\Gamma^{m_1-1/2,m_2}$, we find

$$\begin{aligned}
[(\underline{M}^m)^{-1} \underline{M}_\gamma^\Gamma]_{ij,q} &= \sum_{k,l=0}^P (M_{1,i,k}^{m_1})^{-1} (M_{2,j,l}^{m_2})^{-1} \delta_{k,0} M_{2,l,q}^{m_2} \\
&= (M_{1,i,0}^{m_1})^{-1} \delta_{j,q} \\
&= \frac{2}{\Delta x_1^{m_1} \rho_0} \delta_{i,0} \delta_{j,q}.
\end{aligned}$$

Using this result and observing $\mathbf{n}_1 = -\mathbf{e}_1 = -[10]^T$, the contribution of the edge in Eq. (13) becomes

$$\left(\left(\frac{1}{2} + \boldsymbol{\beta} \cdot \mathbf{n}_1 \right) (\underline{M}^m)^{-1} \underline{M}_1^\Gamma (\underline{u}_1^+ - \underline{u}_1^-) \mathbf{n}_1 \right)_{ij} = \frac{1 - 2\beta_1}{\Delta x_1^{m_1} \rho_0} \delta_{i,0} (u_{Pj}^{m_1-1,m_2} - u_{0j}^{m_1,m_2}) \begin{bmatrix} -1 \\ 0 \end{bmatrix}.$$

Substitution of this and the corresponding expressions for $\gamma = 2, 3, 4$ in (11) yields the flux vector coefficients

$$\begin{aligned} \boldsymbol{\sigma}_{ij}^m &= (\nabla u)_{ij}^m \\ &+ \left[\begin{aligned} &\frac{1-2\beta_1}{\Delta x_1^{m_1} \rho_0} \delta_{i,0} \left(u_{0j}^{m_1, m_2} - u_{Pj}^{m_1-1, m_2} \right) + \frac{1+2\beta_1}{\Delta x_1^{m_1} \rho_P} \delta_{i,P} \left(u_{0j}^{m_1+1, m_2} - u_{Pj}^{m_1, m_2} \right) \\ &\frac{1-2\beta_2}{\Delta x_2^{m_2} \rho_0} \delta_{j,0} \left(u_{i0}^{m_1, m_2} - u_{iP}^{m_1, m_2-1} \right) + \frac{1+2\beta_2}{\Delta x_2^{m_2} \rho_P} \delta_{j,P} \left(u_{i0}^{m_1, m_2+1} - u_{iP}^{m_1, m_2} \right) \end{aligned} \right]. \end{aligned} \quad (14)$$

Equations (12, 14) immediately lead to explicit expressions for the numerical fluxes $\hat{\boldsymbol{\sigma}}$ defined by (7b) and (8b), respectively. As the discrete equations require only the flux normal to the boundary, it is sufficient to consider the x_1 -component at edge $\Gamma^{m_1+1/2, m_2}$ and, correspondingly, the x_2 -component at $\Gamma^{m_1, m_2+1/2}$. For example, the IP flux (7b) through $\Gamma^{m_1+1/2, m_2}$ becomes

$$(\hat{\boldsymbol{\sigma}}_{\text{IP}} \cdot \mathbf{n})|_{\Gamma^{m_1+1/2, m_2}} = \mathbf{e}_1 \cdot \left(\{\nabla u_h\} - \mu_{\text{IP}} \llbracket u_h \rrbracket \right) \Big|_{\Gamma^{m_1+1/2, m_2}} =: \hat{\sigma}_{\text{IP}, 1}^{m_1+1/2, m_2}.$$

To obtain the nodal coefficients, this expression has to be evaluated at the collocation points. Expanding the average and jump operators (6) and substituting (12) yields

$$\begin{aligned} \hat{\sigma}_{\text{IP}, 1, j}^{m_1+1/2, m_2} &= \frac{1}{\Delta x_1^{m_1}} \sum_{k=0}^P D_{P,k}^s u_{kj}^{m_1, m_2} + \frac{1}{\Delta x_1^{m_1+1}} \sum_{l=0}^P D_{0,l}^s u_{lj}^{m_1+1, m_2} \\ &+ \mu_{\text{IP}}^{m_1+1/2, m_2} \left(u_{0j}^{m_1+1, m_2} - u_{Pj}^{m_1, m_2} \right) \end{aligned} \quad (15a)$$

and, similarly, for the x_2 -direction

$$\begin{aligned} \hat{\sigma}_{\text{IP}, 2, i}^{m_1, m_2+1/2} &= \frac{1}{\Delta x_2^{m_2}} \sum_{k=0}^P D_{P,k}^s u_{ik}^{m_1, m_2} + \frac{1}{\Delta x_2^{m_2+1}} \sum_{l=0}^P D_{0,l}^s u_{il}^{m_1, m_2+1} \\ &+ \mu_{\text{IP}}^{m_1, m_2+1/2} \left(u_{i0}^{m_1, m_2+1} - u_{iP}^{m_1, m_2} \right). \end{aligned} \quad (15b)$$

The corresponding LDG flux is

$$(\hat{\boldsymbol{\sigma}}_{\text{LDG}} \cdot \mathbf{n})|_{\Gamma^{m_1+1/2, m_2}} = \mathbf{e}_1 \cdot \left(\{\boldsymbol{\sigma}_h\} + \beta \llbracket \boldsymbol{\sigma}_h \rrbracket - \mu_{\text{LDG}} \llbracket u_h \rrbracket \right) \Big|_{\Gamma^{m_1+1/2, m_2}} =: \hat{\sigma}_{\text{LDG}, 1}^{m_1+1/2, m_2}.$$

With (6) and (14) this leads to the nodal coefficients

$$\begin{aligned} \hat{\sigma}_{\text{LDG}, 1, j}^{m_1+1/2, m_2} &= \frac{1+2\beta_1}{\Delta x_1^{m_1}} \sum_{k=0}^P D_{P,k}^s u_{kj}^{m_1, m_2} + \frac{1-2\beta_1}{\Delta x_1^{m_1+1}} \sum_{l=0}^P D_{0,l}^s u_{lj}^{m_1+1, m_2} \\ &+ \left[\frac{(1+2\beta_1)^2}{2\Delta x_1^{m_1} \rho_P} + \frac{(1-2\beta_1)^2}{2\Delta x_1^{m_1+1} \rho_0} + \mu_{\text{LDG}}^{m_1+1/2, m_2} \right] \left(u_{0j}^{m_1+1, m_2} - u_{Pj}^{m_1, m_2} \right). \end{aligned} \quad (16a)$$

For direction 2 we obtain

$$\begin{aligned} \hat{\sigma}_{\text{LDG}, 2, i}^{m_1, m_2+1/2} &= \frac{1+2\beta_2}{\Delta x_2^{m_2}} \sum_{k=0}^P D_{P,k}^s u_{ik}^{m_1, m_2} + \frac{1-2\beta_2}{\Delta x_2^{m_2+1}} \sum_{l=0}^P D_{0,l}^s u_{il}^{m_1, m_2+1} \\ &+ \left[\frac{(1+2\beta_2)^2}{2\Delta x_2^{m_2} \rho_P} + \frac{(1-2\beta_2)^2}{2\Delta x_2^{m_2+1} \rho_0} + \mu_{\text{LDG}}^{m_1, m_2+1/2} \right] \left(u_{i0}^{m_1, m_2+1} - u_{iP}^{m_1, m_2} \right). \end{aligned} \quad (16b)$$

A closer inspection of Equations (14–16) shows that IP and LDG coincide if $\beta = 0$ and

$$\begin{aligned}\mu_{\text{IP}}^{m_1+1/2, m_2} &= \left(\frac{1}{2\Delta x_1^{m_1} \rho_P} + \frac{1}{2\Delta x_1^{m_1+1} \rho_0} \right) + \mu_{\text{LDG}}^{m_1+1/2, m_2} = \mu_0^{m_1+1/2, m_2} + \mu_{\text{LDG}}^{m_1+1/2, m_2}, \\ \mu_{\text{IP}}^{m_1, m_2+1/2} &= \left(\frac{1}{2\Delta x_2^{m_2} \rho_P} + \frac{1}{2\Delta x_2^{m_2+1} \rho_0} \right) + \mu_{\text{LDG}}^{m_1, m_2+1/2} = \mu_0^{m_1, m_2+1/2} + \mu_{\text{LDG}}^{m_1, m_2+1/2}.\end{aligned}$$

This observation allows to relate the interior penalty method to the LDG stability condition $\mu_{\text{LDG}} > 0$ and motivates the generic penalty coefficient

$$\mu = \mu_0 (1 + \mu_\star),$$

where $\mu_\star > 0$ is a dimensionless parameter and μ_0 the LDG stability threshold, e.g., at $\Gamma^{m_1+1/2, m_2}$

$$\mu_0^{m_1+1/2, m_2} = \frac{1}{2\Delta x_1^{m_1} \rho_P} + \frac{1}{2\Delta x_1^{m_1+1} \rho_0} = \left\{ \frac{1}{\Delta x_1} \right\} \frac{P(P+1)}{2}.$$

The original coefficients are related to the generic one by $\mu_{\text{IP}} = \mu$ and $\mu_{\text{LDG}} = \mu - \mu_0$. As the penalty coefficients depend only on the mesh spacing normal to the edge, we simplify the notation by dropping the index referring to the tangential direction, i.e. $\mu^{m_1+1/2, m_2}$ becomes $\mu^{m_1+1/2}$ etc. Introducing these definitions in (15) and (16) yields the unified numerical fluxes

$$\begin{aligned}\hat{\sigma}_{1,j}^{m_1+1/2, m_2} &= \frac{1 + 2\beta_1}{\Delta x_1^{m_1}} \sum_{k=0}^P D_{P,k}^s u_{kj}^{m_1, m_2} + \frac{1 - 2\beta_1}{\Delta x_1^{m_1+1}} \sum_{l=0}^P D_{0,l}^s u_{lj}^{m_1+1, m_2} \\ &+ \left[2 \frac{\beta_1^2 + \beta_1}{\Delta x_1^{m_1} \rho_P} + 2 \frac{\beta_1^2 - \beta_1}{\Delta x_1^{m_1+1} \rho_0} + \mu^{m_1+1/2} \right] \left(u_{0j}^{m_1+1, m_2} - u_{Pj}^{m_1, m_2} \right),\end{aligned}\tag{17a}$$

$$\begin{aligned}\hat{\sigma}_{2,i}^{m_1, m_2+1/2} &= \frac{1 + 2\beta_2}{\Delta x_2^{m_2}} \sum_{k=0}^P D_{P,k}^s u_{ik}^{m_1, m_2} + \frac{1 - 2\beta_2}{\Delta x_2^{m_2+1}} \sum_{l=0}^P D_{0,l}^s u_{il}^{m_1, m_2+1} \\ &+ \left[2 \frac{\beta_2^2 + \beta_2}{\Delta x_2^{m_2} \rho_P} + 2 \frac{\beta_2^2 - \beta_2}{\Delta x_2^{m_2+1} \rho_0} + \mu^{m_2+1/2} \right] \left(u_{i0}^{m_1, m_2+1} - u_{iP}^{m_1, m_2} \right).\end{aligned}\tag{17b}$$

With these prerequisites we are ready to tackle (5b). To obtain the discrete equations we set $v|_{\Omega^m} = \varphi_i(\xi_1)\varphi_j(\xi_2)$ and evaluate each integral by means of GLL quadrature. The first term of (5b) becomes

$$\int_{\Omega^m} \nabla v \cdot \boldsymbol{\sigma}_h \, d\Omega = \frac{\Delta x_1^{m_1} \Delta x_2^{m_2}}{4} \sum_{p,q=0}^P \rho_p \rho_q \boldsymbol{\sigma}_{pq}^m \cdot \left[\frac{2}{\Delta x_1^{m_1}} \varphi'_i(\eta_p) \varphi_j(\eta_q) \right],$$

where η_p and η_q denote the GLL points for direction 1 and 2, respectively. Using the already introduced 1D mass and differentiation matrices, and exploiting the orthogonality of the former, this can be rewritten to

$$\int_{\Omega^m} \nabla v \cdot \boldsymbol{\sigma}_h \, d\Omega = M_{2,jj}^{m_2} \sum_{p=0}^P \rho_p D_{p,i}^s \sigma_{1,pj}^m + M_{1,ii}^{m_1} \sum_{q=0}^P \rho_q D_{q,j}^s \sigma_{2,iq}^m.$$

Substitution of $\sigma_{1,pj}^m$ and $\sigma_{2,iq}^m$ with (14) and, therein, $(\nabla u)_{ij}^m$ with (12) gives

$$\begin{aligned} \int_{\Omega^m} \nabla v \cdot \boldsymbol{\sigma}_h \, d\Omega &= M_{2,j,j}^{m_2} \sum_{p=0}^P \rho_p D_{p,i}^s \left[\frac{2}{\Delta x_1^{m_1}} \sum_{k=0}^P D_{p,k}^s u_{kj}^m + \frac{1-2\beta_1}{\Delta x_1^{m_1} \rho_0} \delta_{p,0} (u_{0j}^{m_1,m_2} - u_{Pj}^{m_1-1,m_2}) \right. \\ &\quad \left. + \frac{1+2\beta_1}{\Delta x_1^{m_1} \rho_P} \delta_{p,P} (u_{0j}^{m_1+1,m_2} - u_{Pj}^{m_1,m_2}) \right] \\ &\quad + M_{1,i,i}^{m_1} \sum_{q=0}^P \rho_q D_{q,j}^s \left[\frac{2}{\Delta x_2^{m_2}} \sum_{l=0}^P D_{p,l}^s u_{il}^m + \frac{1-2\beta_2}{\Delta x_2^{m_2} \rho_0} \delta_{q,0} (u_{i0}^{m_1,m_2} - u_{iP}^{m_1,m_2-1}) \right. \\ &\quad \left. + \frac{1+2\beta_2}{\Delta x_2^{m_2} \rho_P} \delta_{q,P} (u_{i0}^{m_1,m_2+1} - u_{iP}^{m_1,m_2}) \right]. \end{aligned}$$

Introduction of the one-dimensional standard stiffness matrix

$$L_{i,k}^s = \int_{-1}^1 \varphi'_i(\xi) \varphi'_k(\xi) \, d\xi = \sum_{j=0}^P \rho_j D_{j,i}^s D_{j,k}^s$$

and exploitation of the Kronecker deltas yields the final form

$$\begin{aligned} \int_{\Omega^m} \nabla v \cdot \boldsymbol{\sigma}_h \, d\Omega &= M_{2,j,j}^{m_2} \left[\frac{1}{\Delta x_1^{m_1}} \sum_{k=0}^P L_{i,k}^s u_{pj}^m + \frac{1-2\beta_1}{\Delta x_1^{m_1}} D_{0,i}^s (u_{0j}^{m_1,m_2} - u_{Pj}^{m_1-1,m_2}) \right. \\ &\quad \left. + \frac{1+2\beta_1}{\Delta x_1^{m_1}} D_{P,i}^s (u_{0j}^{m_1+1,m_2} - u_{Pj}^{m_1,m_2}) \right] \\ &\quad + M_{1,i,i}^{m_1} \left[\frac{1}{\Delta x_2^{m_2}} \sum_{l=0}^P L_{j,l}^s u_{il}^m + \frac{1-2\beta_2}{\Delta x_2^{m_2}} D_{0,j}^s (u_{i0}^{m_1,m_2} - u_{iP}^{m_1,m_2-1}) \right. \\ &\quad \left. + \frac{1+2\beta_2}{\Delta x_2^{m_2}} D_{P,j}^s (u_{i0}^{m_1,m_2+1} - u_{iP}^{m_1,m_2}) \right]. \end{aligned}$$

The second term in Eq. (5b) becomes

$$\int_{\Omega^m} v f \, d\Omega = \frac{\Delta x_1^{m_1} \Delta x_2^{m_2}}{4} \rho_i \rho_j f(x_1^{m_1}(\eta_i), x_2^{m_2}(\eta_j)) =: g_{ij}^{m_1,m_2},$$

where $x_d^{m_d}(\cdot)$ is the inverse element mapping. Finally, the last term represents an integral over the element boundary, which can be decomposed into contributions of the four edges:

$$\begin{aligned} \int_{\partial\Omega^m} v \hat{\boldsymbol{\sigma}} \cdot \mathbf{n} \, d\Gamma &= - \int_{\Gamma^{m_1-1/2,m_2}} v \hat{\sigma}_1 \, d\Gamma - \int_{\Gamma^{m_1,m_2-1/2}} v \hat{\sigma}_2 \, d\Gamma \\ &\quad + \int_{\Gamma^{m_1+1/2,m_2}} v \hat{\sigma}_1 \, d\Gamma + \int_{\Gamma^{m_1,m_2+1/2}} v \hat{\sigma}_2 \, d\Gamma. \end{aligned}$$

Applying GLL quadrature on the edges yields

$$\begin{aligned} \int_{\partial\Omega^m} v \hat{\boldsymbol{\sigma}} \cdot \mathbf{n} \, d\Gamma &= - \delta_{i,0} M_{2,j,j}^{m_2} \hat{\sigma}_{1,j}^{m_1-1/2,m_2} - \delta_{j,0} M_{1,i,i}^{m_1} \hat{\sigma}_{2,i}^{m_1,m_2-1/2} \\ &\quad + \delta_{i,P} M_{2,j,j}^{m_2} \hat{\sigma}_{1,j}^{m_1+1/2,m_2} + \delta_{j,P} M_{1,i,i}^{m_1} \hat{\sigma}_{2,i}^{m_1,m_2+1/2}. \end{aligned}$$

The terms on the right side can be expanded using the expressions for the numerical fluxes given in (17). Doing this and substituting the above results in (5b) leads to the

discrete equations

$$\begin{aligned}
& M_{jj}^{m_2} \left[\frac{2}{\Delta x_1^{m_1}} \sum_{k=0}^P L_{i,k}^s u_{kj}^{m_1, m_2} \right. \\
& + \frac{1-2\beta_1}{\Delta x_1^{m_1}} D_{0,i}^s \left(u_{0j}^{m_1, m_2} - u_{Pj}^{m_1-1, m_2} \right) \\
& + \frac{1+2\beta_1}{\Delta x_1^{m_1}} D_{P,i}^s \left(u_{0j}^{m_1+1, m_2} - u_{Pj}^{m_1, m_2} \right) \\
& + \delta_{i,0} \left(\left(\frac{1+2\beta_1}{\Delta x_1^{m_1-1}} \sum_{k=0}^P D_{P,k}^s u_{kj}^{m_1-1, m_2} + \frac{1-2\beta_1}{\Delta x_1^{m_1}} \sum_{l=0}^P D_{0,l}^s u_{lj}^{m_1, m_2} \right) \right. \\
& \quad \left. + \left(2 \frac{\beta_1^2 + \beta_1}{\Delta x_1^{m_1-1} \rho_P} + 2 \frac{\beta_1^2 - \beta_1}{\Delta x_1^{m_1} \rho_0} + \mu^{m_1-1/2} \right) \left(u_{0j}^{m_1, m_2} - u_{Pj}^{m_1-1, m_2} \right) \right) \\
& - \delta_{i,P} \left(\left(\frac{1+2\beta_1}{\Delta x_1^{m_1}} \sum_{k=0}^P D_{P,k}^s u_{kj}^{m_1, m_2} + \frac{1-2\beta_1}{\Delta x_1^{m_1+1}} \sum_{l=0}^P D_{0,l}^s u_{lj}^{m_1+1, m_2} \right) \right. \\
& \quad \left. + \left(2 \frac{\beta_1^2 + \beta_1}{\Delta x_1^{m_1} \rho_P} + 2 \frac{\beta_1^2 - \beta_1}{\Delta x_1^{m_1+1} \rho_0} + \mu^{m_1+1/2} \right) \left(u_{0j}^{m_1+1, m_2} - u_{Pj}^{m_1, m_2} \right) \right) \left. \right] \\
& + M_{ii}^{m_1} \left[\frac{2}{\Delta x_2^{m_2}} \sum_{l=0}^P L_{j,l}^s u_{il}^{m_1, m_2} \right. \\
& + \frac{1-2\beta_2}{\Delta x_2^{m_2}} D_{0,j}^s \left(u_{i0}^{m_1, m_2} - u_{iP}^{m_1, m_2-1} \right) \\
& + \frac{1+2\beta_2}{\Delta x_2^{m_2}} D_{P,j}^s \left(u_{i0}^{m_1, m_2+1} - u_{iP}^{m_1, m_2} \right) \\
& + \delta_{j,0} \left(\left(\frac{1+2\beta_2}{\Delta x_2^{m_2-1}} \sum_{k=0}^P D_{P,k}^s u_{ik}^{m_1, m_2-1} + \frac{1-2\beta_2}{\Delta x_2^{m_2}} \sum_{l=0}^P D_{0,l}^s u_{il}^{m_1, m_2} \right) \right. \\
& \quad \left. + \left(2 \frac{\beta_2^2 + \beta_2}{\Delta x_2^{m_2-1} \rho_P} + 2 \frac{\beta_2^2 - \beta_2}{\Delta x_2^{m_2} \rho_0} + \mu^{m_2-1/2} \right) \left(u_{i0}^{m_1, m_2} - u_{iP}^{m_1, m_2-1} \right) \right) \\
& - \delta_{j,P} \left(\left(\frac{1+2\beta_2}{\Delta x_2^{m_2}} \sum_{k=0}^P D_{P,k}^s u_{ik}^{m_1, m_2} + \frac{1-2\beta_2}{\Delta x_2^{m_2+1}} \sum_{l=0}^P D_{0,l}^s u_{il}^{m_1, m_2+1} \right) \right. \\
& \quad \left. + \left(2 \frac{\beta_2^2 + \beta_2}{\Delta x_2^{m_2} \rho_P} + 2 \frac{\beta_2^2 - \beta_2}{\Delta x_2^{m_2+1} \rho_0} + \mu^{m_2+1/2} \right) \left(u_{i0}^{m_1, m_2+1} - u_{iP}^{m_1, m_2} \right) \right) \left. \right] \\
& = g_{ij}^{m_1, m_2} \tag{18}
\end{aligned}$$

for $0 \leq i, j \leq P$ and $1 \leq m_d \leq N_{E,d}$.

To emphasize the tensor-product structure of (18) we introduce the global solution coefficients $u_{IJ} = u_{ij}^{m_1, m_2}$ with periodic indices

$$I = \ell_1(i, m_1) := i + (P + 1)\tilde{(m_1 - 1)} \quad (19a)$$

$$J = \ell_2(j, m_2) := j + (P + 1)\tilde{(m_2 - 1)}, \quad (19b)$$

where $\tilde{m}_d = m_d + kN_{E,d}$ is the 1D element index mapped into the range $1 \leq \tilde{m}_d \leq N_{E,d}$ by properly choosing $k \in \mathbb{I}$. Further, we define the global 1D mass matrices

$$M_{d,I,K} = M_{d,\ell_d(i,m),\ell_d(k,m+r)} = M_{d,i,k}^m \delta_{0,r} \quad (20)$$

with $d = 1, 2$ for directions 1 and 2. The bracketed expressions in (18) can be represented by means of global 1D stiffness matrices that are applied to the solution coefficients. Careful examination of the expressions shows that these matrices possess the form

$$L_{d,I,K} = L_{\ell_d(i,m),\ell_d(k,m+r)} = \begin{cases} L_{d,i,k}^- & r = -1 \\ L_{d,i,k}^0 & r = 0 \\ L_{d,i,k}^+ & r = 1 \\ 0 & |r| > 1 \end{cases} \quad (21)$$

with

$$\begin{aligned} L_{d,i,k}^0 &= \frac{2}{\Delta x_d^m} L_{i,k}^s \\ &+ \frac{1 - 2\beta_d}{\Delta x_d^m} D_{0,i}^s \delta_{k,0} + \frac{1 - 2\beta_d}{\Delta x_d^m} \delta_{i,0} D_{0,k}^s + \left[2 \frac{\beta_d^2 + \beta_d}{\Delta x_d^{m-1} \rho_P} + 2 \frac{\beta_d^2 - \beta_d}{\Delta x_d^m \rho_0} + \mu^{m-1/2} \right] \delta_{i,0} \delta_{k,0} \\ &- \frac{1 + 2\beta_d}{\Delta x_d^m} D_{P,i}^s \delta_{k,P} - \frac{1 + 2\beta_d}{\Delta x_d^m} \delta_{i,P} D_{P,k}^s + \left[2 \frac{\beta_d^2 + \beta_d}{\Delta x_d^m \rho_P} + 2 \frac{\beta_d^2 - \beta_d}{\Delta x_d^{m+1} \rho_0} + \mu^{m+1/2} \right] \delta_{i,P} \delta_{k,P} \\ L_{d,i,k}^- &= - \frac{1 - 2\beta_d}{\Delta x_d^{m-1}} D_{0,i}^s \delta_{k,P} + \frac{1 + 2\beta_d}{\Delta x_d^{m-1}} \delta_{i,0} D_{P,k}^s - \left[2 \frac{\beta_d^2 + \beta_d}{\Delta x_d^{m-1} \rho_P} + 2 \frac{\beta_d^2 - \beta_d}{\Delta x_d^m \rho_0} + \mu^{m-1/2} \right] \delta_{i,0} \delta_{k,P} \\ L_{d,i,k}^+ &= \frac{1 + 2\beta_d}{\Delta x_d^{m+1}} D_{P,i}^s \delta_{k,0} - \frac{1 - 2\beta_d}{\Delta x_d^{m+1}} \delta_{i,P} D_{0,k}^s - \left[2 \frac{\beta_d^2 + \beta_d}{\Delta x_d^m \rho_P} + 2 \frac{\beta_d^2 - \beta_d}{\Delta x_d^{m+1} \rho_0} + \mu^{m+1/2} \right] \delta_{i,P} \delta_{k,0}, \end{aligned}$$

where it should be noted that $\mu^{m\pm 1/2}$ depends on Δx_d^m and $\Delta x_d^{m\pm 1}$. Adopting the global 1D operators allows to rewrite (18) as

$$\sum_{L=0}^{N_2} \sum_{K=0}^{N_1} \underbrace{(M_{2,J,L} L_{1,I,K} + L_{2,J,L} M_{1,I,K})}_{=: A_{IJ,KL}} u_{KL} = g_{IJ} \quad (22)$$

for $0 \leq I \leq N_1 = N_P N_{E,1}$ and $0 \leq J \leq N_2 = N_P N_{E,2}$, where $N_P = P + 1$. Alternatively, the system can be written in matrix form as follows:

$$\underline{A} \underline{u} = (\underline{M}_2 \otimes \underline{L}_1 + \underline{L}_2 \otimes \underline{M}_1) \underline{u} = \underline{g}. \quad (23)$$

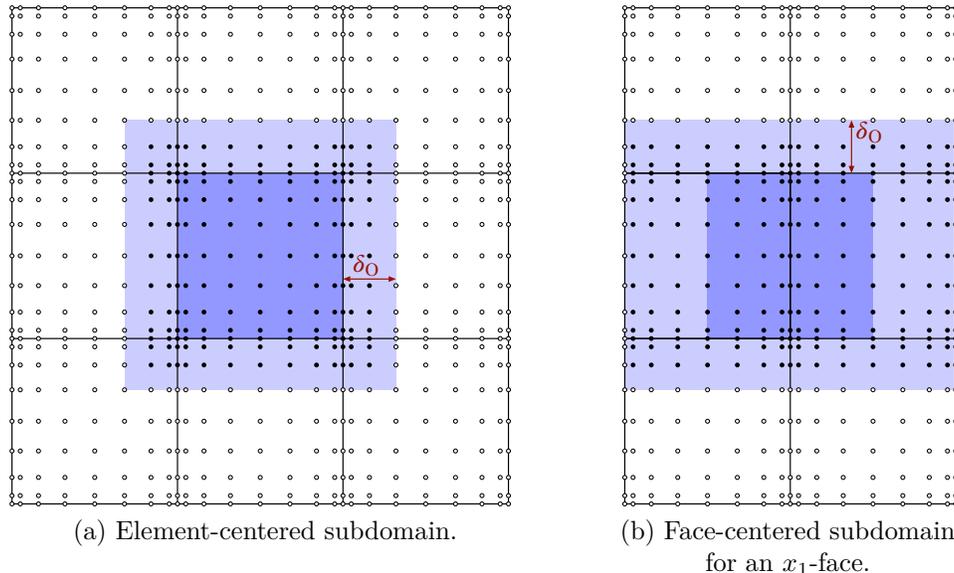


FIGURE 2. Subdomains used with the Schwarz method. Each subdomain consists of a core region (dark shaded) and an overlap zone of width δ_o (light shaded). The circles are the GLL nodes for polynomial order $p = 8$. Filled circles indicate the nodes that are solved for and updated.

3. SOLUTION METHODS

The linear system (23) is symmetric positive semi-definite. Moreover, its structure closely resembles the discrete equations generated with the continuous spectral element method [26, 33]. This coincidence inspired us to adopt the multigrid techniques developed in [33] for the present discontinuous formulation. In particular, we examine polynomial multigrid (MG) and multigrid-preconditioned conjugate gradients (MGCG). Both approaches employ overlapping Schwarz methods for smoothing. We first present the Schwarz methods and then sketch MG and MGCG.

3.1. Schwarz methods. Schwarz methods are iterative domain decomposition techniques which improve the approximate solution by parallel or sequential subdomain solves, leading to additive or multiplicative methods, respectively. Here we consider element-centered and face-centered subdomains as illustrated in Fig. 2. The element-centered subdomain was already used in [33]. It covers the element region extended by a strip including N_o layers of additional nodes from the neighbor elements, excluding the nodes located on the subdomain boundary (Fig. 2a). The overlap is defined as

$$\delta_o = \frac{1}{2} \Delta \xi_o \Delta x_n,$$

where $\Delta \xi_o$ is the overlap in standard coordinates and Δx_n the extension of the abutting element normal to the shared edge. For the element-centered subdomain, we set

$\Delta\xi_o = \eta_{N_o} + 1$. With this definition, the overlap width corresponds to the distance between the edge and the nearest node layer that is not included. In the case $N_o = 0$ no nodes are adopted and the overlap width equals zero, since $\eta_{N_o} = -1$. Choosing $N_o = 1$ includes the boundary nodes of the adjoining elements that are located on the common edge or vertex. Note, however, that these nodes are counted separately, though they coincide geometrically with nodes of the element representing the core of the subdomain.

Except for the non-overlapping case, the use of element-centered subdomains implies a diagonal coupling, which is natural to continuous elements, but seems artificial to DG. This motivated the use of face-centered subdomains, including only nodes from the two elements sharing one face (i.e. one edge in 2D). However, in course of our studies it proved necessary to allow for a lateral overlap analogous to the element-centered approach, which finally led to the face-centered domains sketched in Fig. 2b. The core region of the face-centered subdomain is defined as the union of the two adjoining half elements, and the opposite halves form a part of the overlap zone. This “normal” overlap is fixed and has always an extension of half the element width. In contrast, the tangential or “lateral” overlap is adjustable and defined as in the element-centered case. According to the face orientation, the subdomains can be divided into two groups, where the normal is aligned either with the x_1 - or the x_2 -direction. In terms of core regions, each group represents a complete partition of the computational domain and, hence, will be treated separately.

In the following $\{\Omega_s\}_{s=1}^{N_D}$ denotes the set of subdomains constituting a single partition, i.e. the set of element-centered subdomains, or the set of face-centered subdomains with normals oriented in direction 1, or the set of face-centered subdomains with normals oriented in direction 2. Within each set the subdomains are linearly numbered according to a lexicographical ordering based on their row and column indices. Note that for periodic Cartesian grids $N_D = N_E$, i.e., the number of subdomains equals the number of elements in all three cases.

The main idea of the Schwarz method is to solve a subproblem for every subdomain Ω_s and to construct a correction to a given approximate solution \tilde{u} by combining the resulting local corrections. For establishing the subproblems we introduce the (exact) correction $\Delta u = u - \tilde{u}$ and convert Eq. (23) into the equivalent residual form

$$\underline{A}\Delta u = g - \underline{A}\tilde{u} = \tilde{r}.$$

For each subdomain Ω_s we define the restriction operator \underline{R}_s such that $\underline{u}_s = \underline{R}_s u$ gives the associated coefficients. Conversely, the transposed restriction operator, \underline{R}_s^T is used to globalize the local coefficients by adding zeros for exterior nodes. With these prerequisites the correction contributed by Ω_s is defined as the solution of the subproblem

$$\underline{A}_{ss}\Delta u_s = \underline{r}_s, \quad (24)$$

where $\underline{A}_{ss} = \underline{R}_s \underline{A} \underline{R}_s^T$ represents the restricted system matrix and $\underline{r}_s = \underline{R}_s \tilde{r}$ the restricted residual. Due to the rectangular shape of the subdomain, the restriction operator possesses the tensor-product factorization $\underline{R}_s = \underline{R}_{s,1} \otimes \underline{R}_{s,2}$ and \underline{A}_{ss} inherits the structure of the full system matrix \underline{A} , i.e.

$$\underline{A}_{ss} = \underline{M}_{s,2} \otimes \underline{L}_{s,1} + \underline{L}_{s,2} \otimes \underline{M}_{s,1}.$$

Note that the one-dimensional mass matrices $\underline{M}_{s,d} = \underline{R}_{s,d}\underline{M}_d\underline{R}_{s,d}^T$ are positive diagonal, while the stiffness matrices $\underline{L}_{s,d} = \underline{R}_{s,d}\underline{L}_d\underline{R}_{s,d}^T$ are symmetric and — because of the implied Dirichlet conditions — positive definite. These properties allow for application of the fast diagonalization technique (FDM) developed by Lynch et al. [27] and adopted for SEM in [8], which yields the inverse of \underline{A}_{ss} in the form

$$\underline{A}_{ss}^{-1} = (\underline{S}_2 \otimes \underline{S}_1)(\underline{I} \otimes \underline{A}_1 + \underline{A}_2 \otimes \underline{I})^{-1}(\underline{S}_2^T \otimes \underline{S}_1^T),$$

where \underline{I} is the unity matrix, \underline{S}_d a matrix composed of the eigenvectors to the generalized eigenproblem for $\underline{L}_{s,d}$ and $\underline{M}_{s,d}$, and \underline{A}_d the diagonal matrix of eigenvalues. The central term on the right side is the diagonal matrix composed of the reciprocal 2D eigenvalues. Apart from this factor, the evaluation of $\underline{A}_{ss}^{-1}\underline{r}_s$ requires the subsequent application of four 1D operators, namely \underline{S}_1 , \underline{S}_2 and their transposes, to a 2D operand. For an element-centered domain, \underline{S}_1 and \underline{S}_2 are square matrices of dimension $N_P + 2N_O$, where $N_P = P + 1$ is the number of collocation points in one direction, and \underline{r}_s is of the size $(N_P + 2N_O)^2$. Hence, $\Delta\underline{u}_s = \underline{A}_{ss}^{-1}\underline{r}_s$ can be computed with $\Theta(4(N_P + N_O)^3)$ operations. In the face-centered case, \underline{S}_1 and \underline{S}_2 are square matrices of dimensions $2N_P - 2$ and $N_P + 2N_O$, or vice versa, depending on the face orientation, and \underline{r}_s is of the size $(2N_P - 2)(N_P + 2N_O)$. Assuming $2N_P - 2 \approx 2N_P$, the cost for evaluating \underline{u}_s amounts to $4(3N_P + 2N_O)(N_P + 2N_O)N_P$. Finally, defining $C_D = 4(1 + 2C_O)^3$ for element-centered and $C_D = 12(1 + 2C_O)(1 + 2C_O/3)$ for face-centered subdomains, where $C_O = N_O/N_P$, both estimates can be cast in the form $\Theta(C_D N_P^3)$.

Several options exist for combining the local solutions. Following [33] we consider the multiplicative Schwarz method and a weighted version of the additive Schwarz method. The multiplicative Schwarz method solves the subproblems (24) consecutively, while continually updating the residual. In the element-centered case, one iteration corresponds to one sweep over all subdomains, whereas two sweeps are performed in the face-centered case: one for over the domains oriented in direction 1 and another one for direction 2. Though \underline{A} is symmetric, one multiplicative Schwarz iteration corresponds to the application of a non-symmetric linear operator. However, for an even number of iterations, the method is symmetrized by reversing the order of subdomains and, in the face-centered case, also the order of sweeps in each iteration.

The weighted additive Schwarz method determines all local corrections independently and computes the global correction as a linear combination of these results, i.e.

$$\Delta\underline{u} \simeq \sum_s \underline{R}_s^T(\underline{w}\Delta\underline{u}_s), \quad (25)$$

where $\underline{w} = \underline{w}_2 \otimes \underline{w}_1$ is a diagonal local weighting matrix that is generated from generic 1D weight distributions \underline{w}_d . For the element-centered approach we reuse the weight distributions introduced in [33] for the continuous case, i.e., \underline{w}_d is computed from the hat-shaped weighting function

$$w_H(\xi_H) = \frac{1}{2} \left[\phi_i \left(\frac{1 + \xi_H}{\Delta\xi_O} \right) + \phi_i \left(\frac{1 - \xi_H}{\Delta\xi_O} \right) \right], \quad (26)$$

where ξ_H is the 1D standard coordinate extended beyond $[-1, 1]$ and ϕ_i a transitional function. Let Ω^m denote the element associated with the core region of subdomain Ω_s ,

Ω^{m-1} its predecessor in the weighting direction and, respectively, Ω^{m+1} its successor. Then ξ_{H} is computed as

$$\xi_{\text{H}} = \begin{cases} \xi & \text{in } \Omega^m \\ \xi \pm 2 & \text{in } \Omega^{m\pm 1} \end{cases} .$$

For ϕ_i we consider the smoothed sign functions defined as

$$\phi_i(x) = \begin{cases} \tilde{\phi}_i(x) & x \in [-1, 1] \\ \text{sgn}(x) & \text{else} \end{cases} ,$$

where $\tilde{\phi}_i$ is a polynomial of degree $i \in \{1, 3, 5, \dots\}$ satisfying the conditions

$$\begin{aligned} \tilde{\phi}_i(\pm 1) &= \pm 1, \\ \frac{d^k \tilde{\phi}_i}{dx^k}(\pm 1) &= 0, \quad 0 < k \leq (i-1)/2. \end{aligned}$$

The $\tilde{\phi}_i$ are strictly monotonic in $(-1, 1)$ and yield a smooth transition of the weight function in the overlap zone. In the following we use the cubic and quintic transitions

$$\begin{aligned} \tilde{\phi}_3 &= (3x - x^3)/2, \\ \tilde{\phi}_5 &= (15x - 10x^3 + 3x^5)/8. \end{aligned}$$

Figure 3 exemplifies the resulting weight distribution for the quintic case. Other possible choices include conventional additive Schwarz, $\phi_i = 1$, and arithmetic averaging, $\tilde{\phi}_i = 0$, as proposed by Lottes and Fischer [26] for SEM.

With the face-centered approach, one iteration consists of one sweep over all subdomains of one orientation, followed by second sweep for the other orientation. Note that this method is not strictly additive, since the second sweep builds on the result of the first one. The weight matrix \underline{w} is constructed similarly as in the element-centered case. For the direction normal to the face consider the two adjacent elements Ω^m and Ω^{m+1} such that the face is located at $\xi = 1$ in the former and at $\xi = -1$ in the latter. Note that we use a 1D notation for simplicity. The normal weighting function is then defined as

$$w_{\text{F}}(\xi_{\text{F}}) = \frac{1}{2} [1 + \phi_i(|\xi_{\text{F}}|)] , \quad (27)$$

where

$$\xi_{\text{F}} = \begin{cases} \xi - 1 & \text{in } \Omega^m \\ \xi + 1 & \text{in } \Omega^{m+1} \end{cases} .$$

Figure 4 illustrates w_{F} for for the quintic case ($i = 5$). The tangential weights are identical to those used in the element-centered case.

3.2. Multigrid. For MG we define a series of polynomial levels $\{P_l\}$ with $P_l = 2^l$ increasing from 1 at $l = 0$ to P at top level L . More general series can be used supposing that $P_{l+1} > P_l$, but are not considered here. Correspondingly, let \underline{u}_l denote the global coefficients and \underline{A}_l the system matrix on level l . On the top level we have $\underline{u}_L = \underline{u}$ and $\underline{A}_L = \underline{A}$, whereas on lower levels \underline{u}_l is the defect correction and \underline{A}_l the counterpart of \underline{A}

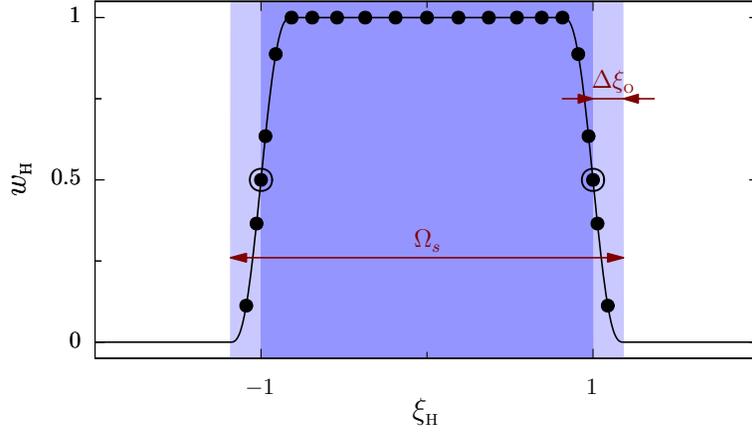


FIGURE 3. Hat-shaped weight distribution w_H for element-centered subdomains and tangential weighting in face-centered subdomains using a quintic shape function with $P = 16$ and $N_o = 3$. The core region and the overlap zone of the subdomain are identified by dark and light shading, respectively. Filled circles indicate the node positions. On each side, the overlap zone includes three nodes from the adjoining element. The enclosing circles mark the positions where one of these coincides with a node of the core element.

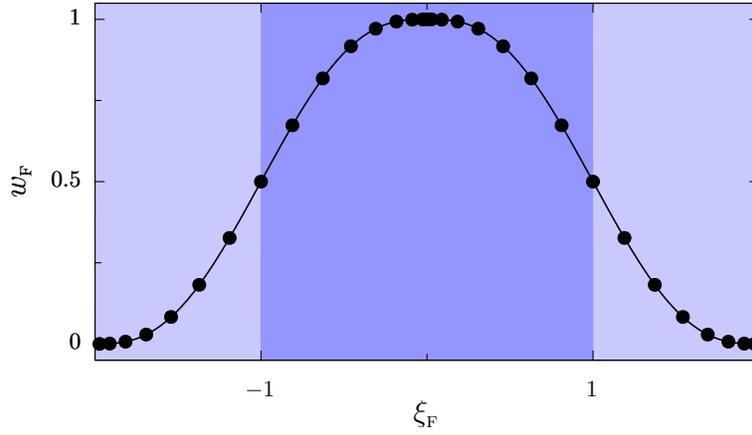


FIGURE 4. Weight distribution w_F for normal weighting in face-centered subdomains using a quintic shape function with $P = 16$. The core region and the overlap zone of the subdomain are identified by dark and light shading, respectively. Filled circles indicate the node positions.

obtained with elements of order P_l . For transferring the correction from level $l - 1$ to level l we use the embedded interpolation operator \mathcal{I}_l , and for restricting the residual its transpose. These ingredients allow to build a multigrid V-cycle, which is identical to the

continuous case [33], but repeated for convenience in Algorithm 1. The SMOOTHER is designed as a generic procedure which employs either the multiplicative or the weighted additive Schwarz method with element-centered or face-centered subdomains. To allow for variable V-cycles [4], the number of pre- and post-smoothing steps, $N_{s1,l}$ and $N_{s2,l}$, may change from level to level. Line 11 of Algorithm 1 defines the coarse grid solution by means of the pseudoinverse \underline{A}_0^+ . In our implementation the coarse problem is solved using the conjugate gradient method. To ensure convergence in spite of singularity, the right side is projected to the null space of \underline{A}_0 , as advocated by Kaasschieter [21].

Algorithm 1 Multigrid V-cycle.

```

1: function MULTIGRIDCYCLE( $\underline{u}$ ,  $\underline{f}$ ,  $N_s$ )
2:    $\underline{u}_L \leftarrow \underline{u}$ 
3:    $\underline{f}_L \leftarrow \underline{f}$ 
4:   for  $l = L, 1$  step  $-1$  do
5:     if  $l < L$  then
6:        $\underline{u}_l \leftarrow 0$ 
7:     end if
8:      $\underline{u}_l \leftarrow \text{SMOOTHER}(\underline{u}_l, \underline{f}_l, N_{s1,l})$  ▷ Pre-smoothing
9:      $\underline{f}_{l-1} \leftarrow \underline{\mathcal{I}}_l^T(\underline{f}_l - \underline{A}_l \underline{u}_l)$  ▷ Residual restriction
10:  end for
11:   $\underline{u}_0 \leftarrow \underline{A}_0^+ \underline{f}_0$  ▷ Coarse grid solution
12:  for  $l = 1, L$  do
13:     $\underline{u}_l \leftarrow \underline{u}_l + \underline{\mathcal{I}}_l \underline{u}_{l-1}$  ▷ Correction prolongation
14:     $\underline{u}_l \leftarrow \text{SMOOTHER}(\underline{u}_l, \underline{f}_l, N_{s2,l})$  ▷ Post-smoothing
15:  end for
16:  return  $\underline{u} \leftarrow \underline{u}_L$ 
17: end function

```

3.3. Preconditioned conjugate gradients. Robustness and efficiency of multigrid can be enhanced by Krylov acceleration [34]. Here we follow the strategy devised [33], where the inexact preconditioned conjugate gradients of Golub and Ye [14] were adopted to cope with the (slight) asymmetry introduced by the Schwarz method. The resulting MGCG method is summarized in Algorithm 2. Note that, as before with the coarse problem, the right side \underline{f} must be in the null space of \underline{A} if the system is singular.

4. RESULTS

For assessing robustness and efficiency, the described methods were implemented in Fortran and applied to the test case of Lottes and Fischer [12, 26], i.e.,

$$-\nabla^2 u = 2\pi^2 \sin(\pi x_1) \sin(\pi x_2)$$

in the domain $\Omega = (0, 2AR) \times (0, 2)$ with the aspect ratio $AR \in \mathbb{N}$. Assuming periodic boundary conditions, the exact solution is $u = \sin(\pi x_1) \sin(\pi x_2)$ for arbitrary AR . To keep the test series manageable, we constrained ourselves to equidistant grids with an

Algorithm 2 Inexact multigrid-preconditioned conjugate gradients.

```

1: function MGCG( $\underline{u}$ ,  $\underline{f}$ ,  $\underline{N}_s$ ,  $i_{\max}$ ,  $r_{\max}$ )
2:    $\underline{r}_{\text{old}} \leftarrow \underline{0}$ 
3:    $\underline{r} \leftarrow \underline{f} - \underline{A}\underline{u}$ 
4:    $\underline{p} \leftarrow \text{MULTIGRIDCYCLE}(\underline{0}, \underline{r}, \underline{N}_s)$ 
5:    $\underline{\delta} \leftarrow \underline{p}^T \underline{r}$ 
6:   for  $i = 1, i_{\max}$  do
7:      $\underline{q} \leftarrow \underline{A}\underline{p}$ 
8:      $\alpha \leftarrow \underline{\delta} / (\underline{p}^T \underline{q})$ 
9:      $\underline{u} \leftarrow \underline{u} + \alpha \underline{p}$ 
10:     $\underline{r} \leftarrow \underline{r} - \alpha \underline{q}$ 
11:    if  $\|\underline{r}\| \leq r_{\max}$  exit
12:     $\underline{z} \leftarrow \text{MULTIGRIDCYCLE}(\underline{0}, \underline{r}, \underline{N}_s)$ 
13:     $\underline{\beta} \leftarrow \underline{q}^T (\underline{r} - \underline{r}_{\text{old}}) / \underline{\delta}$ 
14:     $\underline{p} \leftarrow \underline{z} + \underline{\beta} \underline{p}$ 
15:     $\underline{\delta} \leftarrow \underline{z}^T \underline{r}$ 
16:     $\underline{r}_{\text{old}} \leftarrow \underline{r}$ 
17:  end for
18:  return  $\underline{u}$ 
19: end function

```

identical number of elements in each direction, i.e., $N_{E,1} = N_{E,2}$. As a consequence, the element aspect ratio $\Delta x_1 / \Delta x_2$ is equivalent to the domain aspect ratio AR . The code was compiled using the GNU compiler collection 6.0 with optimization `-O3` and executed on a 3.1 GHz Intel Core i7-5557U CPU. In all test runs, the initial guess was chosen at random with values confined to the interval $[0, 1]$.

4.1. Performance metrics. The primary convergence measure is the average multigrid convergence rate

$$\rho = \sqrt[n]{\frac{r_n}{r_0}}, \quad (28)$$

where r_n is the Euclidean norm of the residual vector after the n th cycle. Since ρ varied by several orders of magnitude in some tests, we use alternatively the logarithmic convergence rate

$$\bar{r} = -\log_{10} \rho \quad (29)$$

and the number of cycles n_{10} needed to reduce the residual by a factor of 10^{10} .

Remark. The multigrid convergence rate and, consequently, the logarithmic convergence rate depend on the number of cycles n . However, this dependence is weak, except for very low n or the case that machine accuracy is reached. Both effects are negligible in the reported tests and, hence, not further considered here.

As an efficiency measure we define the average equivalent number of operator applications required for reducing the residual by a factor of 10,

$$\bar{\omega} = \bar{r} \frac{W_{\text{cyc}}}{W_{\text{op}}}, \quad (30)$$

where W_{op} is the cost for one application of the system matrix \underline{A} and W_{cyc} for one cycle. Assuming that sum factorization is exploited, the former can be estimated as $W_{\text{op}} = 2N_{\text{P}}^3 N_{\text{E}}$. The cycle cost comprises the contributions of Schwarz iterations, residual evaluations, the coarse grid solver and transfer operators. Using the estimates given in Sec. 3.1 for the first and neglecting the latter two we arrive at

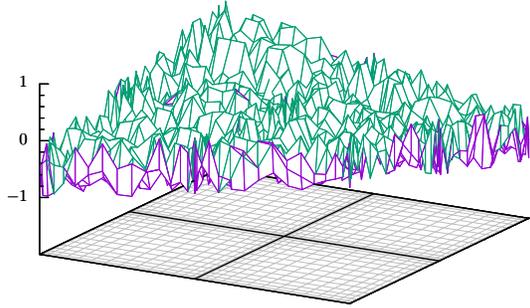
$$W_{\text{cyc}} \simeq \left[C_{\text{s}} N_{\text{s}} \left(\frac{1}{2} C_{\text{D}} M_{\text{D}} + 1 \right) + C_{\text{CG}} \right] W_{\text{op}}, \quad (31)$$

where N_{s} is the number of pre- and post-smoothing steps on the finest level, $C_{\text{s}} = 4/3$ for the classical V-cycle and 2 for a variable V-cycle doubling the number of smoothing steps with lower levels [4], M_{D} the number of sweeps per Schwarz iteration, i.e. $M_{\text{D}} = 1$ for element-centered and $M_{\text{D}} = 2$ for face-centered subdomains, and $C_{\text{CG}} = 1$ the extra cost for conjugate gradients when using MGCG. The “1” in the inner braces stems from the residual evaluation, which is also the dominant cost with CG.

Occasionally, the number of operations to achieve a certain residual reduction may be of interest. As a representative measure we consider the average number of multiplications per unknown for a reduction by ten orders of magnitude, which can be estimated as

$$w_{10} \simeq \frac{10}{\bar{r}} \frac{W_{\text{cyc}}}{N_{\text{P}}^2 N_{\text{E}}} = \frac{20}{\bar{r}} \left[C_{\text{s}} N_{\text{s}} \left(\frac{1}{2} C_{\text{D}} M_{\text{D}} + 1 \right) + C_{\text{CG}} \right] N_{\text{P}}. \quad (32)$$

4.2. Qualitative behavior of Schwarz methods. Figure 5 illustrates the smoothing properties of selected additive Schwarz methods for a discretization using 8×8 elements of order $P = 16$ with stabilization $\mu_{\star} = 1$ and auxiliary parameter $\beta = 0$. The displayed error is defined as the difference between the approximate and exact solutions adjusted to a zero median. For clarity, the plots were restricted to a subregion comprising four elements. Figure 5a depicts the error of the random initial guess and Figs. 5b-f the error after one iteration with different Schwarz methods. In particular, Fig. 5b and Fig. 5c reveal that the non-overlapping element-centered method and the unweighted element-centered method with overlap $N_{\text{o}} = 3$ fail to smooth the error across the element boundaries. The jumps produced with both methods tend to dominate the residual and lead to a severe degradation of MG efficiency. Arithmetically weighting the overlapping Schwarz updates greatly improves this behavior, although the error still exhibits ridges near the element boundaries (Fig. 5d). Using a smooth hat-shaped weight distribution removes these ridges and yields the best smoothing properties for element-centered subdomains (Fig. 5e). Finally, Figure 5f illustrates the excellent performance of the overlapping face-centered Schwarz method with cubic weighting. It should be noted, however, that one face-centered iteration has two sweeps instead of one with element-centered domains and, in addition to this, employs a larger overlap into the face normal direction.



(a) Initial error.

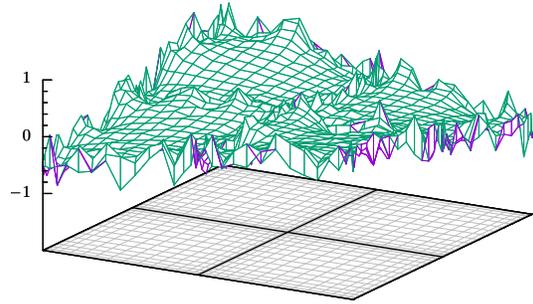
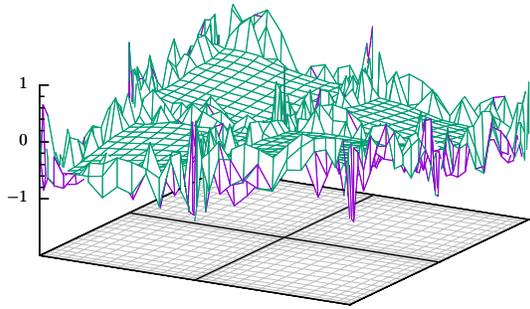
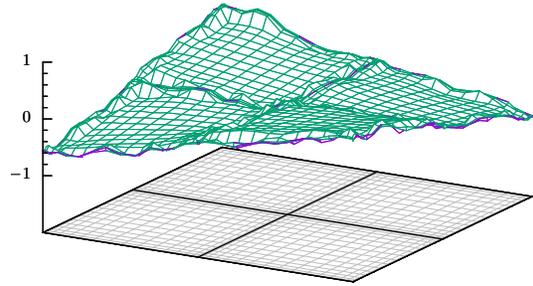
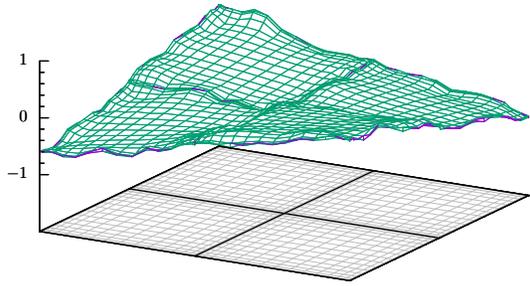
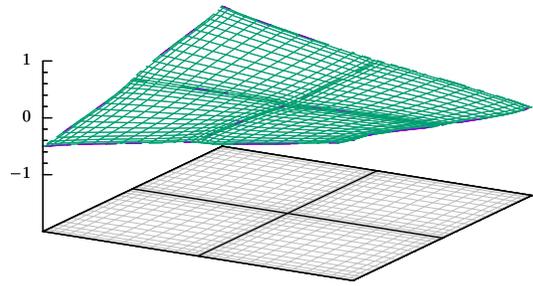
(b) Element-centered additive, $N_0 = 0$.(c) Element-centered additive,
 $N_0 = 3$, unweighted(d) Element-centered additive,
 $N_0 = 3$, arithmetic averaging.(e) Element-centered additive,
 $N_0 = 3$, cubic weighting.(f) Face-centered additive, $N_0 = 3$,
cubic weighting.

FIGURE 5. Smoothing properties of selected Schwarz methods. DG with 8×8 elements of order $P = 16$, $\mu_\star = 1$ and $\beta = 0$. Graph (a) shows the initial error in a subregion consisting of four elements, and (b–f) the error after one Schwarz iteration.

4.3. Convergence and robustness. To investigate the influence of overlap and weighting, and to assess the robustness of the multigrid approach we performed several numerical experiments on grids consisting of up to 256^2 elements of order $P = 4$ to 32 and aspect ratios between 1 and 32. For concise notation we use two-letter acronyms, where the first letter identifies the subdomain type (“E” – element-centered, “F” – face-centered) and the second the iteration method (“M” – multiplicative, “A” additive). Additionally, subscript “0” indicates zero overlap, $N_o = 0$, and “ ℓ ” a level-dependent overlap of $N_{o,\ell} = 1 + \lfloor P_\ell/8 \rfloor$. For example, FM_0 denotes the face-centered multiplicative Schwarz method with $N_o = 0$, i.e. no lateral overlap. If not indicated otherwise, one pre-smoothing and one post-smoothing step were applied on each level $l > 0$. The weighting method (cubic or quintic) is stated where necessary.

Table 2 shows the convergence rates of selected MG and MGCG methods on a uniform 16×16 tessellation of the domain $[0, 2]^2$. The results for the element-centered smoothers resemble those obtained with continuous spectral elements [33]. Using the multiplicative smoother with no overlap, EM_0 , MG reaches convergence rates up to $\bar{r} = 0.63$ with $P = 4$ and MGCG up to $\bar{r} = 0.9$, but both degrade with growing polynomial order P . In the additive case, EA_0 , MG fails to converge (not shown), whereas MGCG just succeeds. Lines 7–14 show the results obtained with the level-dependent overlap, $N_{o,\ell} = 1 + \lfloor P_\ell/8 \rfloor$. It imposes a lower bound on the overlap width δ_o that corresponds approximately to one eighth of the neighbor element width in the direction normal to the boundary. With this choice MG becomes robust against increasing P for multiplicative as well as additive Schwarz smoothing. Comparing the smoothers reveals that EA_ℓ consistently achieved a higher convergence rates than EM_ℓ . As a possible reason we found that the instantaneous updates in the multiplicative method tend to produce excessive gradients in overlap regions, whereas the additive method avoids this peculiarity by applying a weighted average. Both, cubic as well as quintic weighting are suited, though the latter proved slightly more efficient.

Lines 15–28 of Tab. 2 show the results for the face-centered smoothers. Compared to the element-centered smoothers, they achieve notably higher convergence rates, which can be attributed to the inbuilt overlap normal to the face and the double sweep over both coordinate directions. Even with no lateral overlap the approach proved robust against P . Application of a level-dependent lateral overlap, $N_{o,\ell} = 1 + \lfloor P_\ell/8 \rfloor$, increased the logarithmic convergence rate to magnitudes in the range between 2 and 3.5. In contrast to the element-centered method, the difference between cubic and quintic weighting is marginal and hence not considered here. Moreover, the face-centered method exhibits a lower sensitivity with respect to the auxiliary LDG parameter β . In most cases, similar convergence rates were obtained for central and non-central numerical fluxes, i.e. $\beta = 0$ and $\beta = 1/2$, whereas the latter tends to converge slower than the former when using element-centered smoothers.

In the following we focus our attention to MGCG for IP/LDG with central numerical fluxes and abandon cubic in favor of quintic weighting. Table 3 shows the logarithmic convergence rates \bar{r} , cycle counts n_{10} and equivalent operator applications $\bar{\omega}$ for smoothers

EM₀, EA_ℓ, FA₀ and FA_ℓ on equidistant grids ranging from 8² to 256² elements and polynomial orders from $P = 4$ to 32. The results clearly confirm that the methods are robust with respect to the grid size. As expected, EM₀ degrades for growing P , whereas with EA_ℓ, FA₀ and FA_ℓ the increased order leads to even higher convergence rates and, hence, lower cycle counts. Comparing the average number of operator applications required for reducing the residual by a factor of 10, we find that EA_ℓ is the most efficient method with $\bar{\omega} \in [6, 11]$. For example, consider order $P = 16$ with $\bar{r} \approx 2.2$: taking into account $C_s = 4/3$, $N_s = 2$, $C_D = 4(1 + 2C_o)^3$, $C_o \lesssim 3/17$, $C_{CG} = 1$ and $M_D = 1$, the estimate (32) results in just 2608 multiplications per unknown to achieve a residual reduction by ten orders of magnitude. EM₀ uses no overlap and, hence attains lower cycle costs. As a result it remains competitive with EA_ℓ for $P = 4$ and still the second-best method for $P = 8$, despite the higher cycle count. FA_ℓ converges about 1.4 times faster than EA_ℓ, but fails to compensate the sixfold higher operation count and thus remains about twice as expensive in terms of $\bar{\omega}$.

Figure 6a shows the multigrid convergence rates in comparison with the spectral element (SE) version of EA_ℓ. In line with our expectations the comparison asserts that the discontinuous and continuous methods converge nearly with identical rates, improving from $\rho \approx 0.02$ at $P = 4$ to 0.003 at $P = 32$. The face-centered additive smoother yields even faster convergence with rates between 0.003 and 0.0003. To assess the actual computational cost, Figure 6b depicts the runtimes required to reduce the residual by ten orders of magnitude for different polynomial degrees. Note that the number of elements was adjusted according to $N_{E,d} = 256/P$ to assure a nearly constant problem size. Sum factorization was exploited on levels $P_l \geq 8$. Except for the CG method, which included for comparison, all graphs exhibit a gentle downward slope which becomes increasingly horizontal with growing P . This indicates that the increased operation count scaling as $\bar{\omega}P$ is more than compensated by the gain in computational efficiency due to the larger operator size. For the same reason, the face-centered methods, FA₀ and FA_ℓ come considerably closer to EA_ℓ than predicted by $\bar{\omega}$. In contrast, EM₀ performs much worse than expected. It consumes more than 10 times the runtime of EA_ℓ and barely outperforms CG. This behavior is a consequence of the recursive solution strategy of multiplicative Schwarz: It prevents the “stacking” of operands as in the additive case, where it allows to convert matrix-vector multiplications into vastly more efficient matrix-matrix multiplications. The fastest discontinuous method, MGCG with EA_ℓ, solves the problem with $P = 16$ in 0.112s or, respectively, 1.5 μ s per unknown. Counting only multiplications, this corresponds to a performance of 1.9 GFLOPS. The spectral element counterpart attains nearly identical convergence rates, but succeeds with just two thirds of the runtime, which corresponds almost exactly to the lower operation count.

For assessing the sensitivity to the element aspect ratio, AR was varied from 1 to 32. The domains were decomposed into 16×16 rectangular elements of the order P , which ranged from 4 to 32. In addition to the methods considered above, the tests included variants using a variable V-cycle. With the latter the number of smoothing steps is doubled when switching to the next coarser level, i.e. $N_{s,l} = (2^{L-l}, 2^{L-l})$. The variable V-cycle improves convergence speed and robustness, but also raises the cost of one cycle

TABLE 2. Convergence rates of MG and MGCG with different overlaps and weighting methods. DG with $\mu_\star = 1$ and constant β using a uniform 16×16 grid.

#	method	smoother	weighting	β	\bar{r}			
					$P=4$	$P=8$	$P=16$	$P=32$
1	MG	EM ₀	–	0	0.63	0.36	0.22	0.15
2			–	½	0.43	0.26	0.17	0.13
3	MGCG	EM ₀	–	0	0.90	0.72	0.52	0.36
4			–	½	0.73	0.58	0.40	0.28
5		EA ₀	–	0	0.20	0.09	0.03	0.01
6			–	½	0.16	0.06	0.02	0.01
7	MG	EM _ℓ	–	0	1.02	1.01	1.13	1.45
8			–	½	0.61	0.84	0.92	1.16
9		EA _ℓ	cubic	0	1.39	1.64	1.82	1.99
10			cubic	½	1.52	1.69	1.70	1.98
11			quintic	0	1.66	1.65	2.11	2.51
12			quintic	½	1.56	1.68	2.04	2.49
13	MGCG	EA _ℓ	quintic	0	1.76	1.84	2.20	2.49
14			quintic	½	1.60	1.74	2.07	2.40
15	MG	FM ₀	–	0	1.64	1.71	1.87	1.96
16			–	½	1.45	1.34	1.32	1.34
17		FA ₀	quintic	0	1.15	1.22	1.32	1.37
18			quintic	½	1.20	1.14	1.13	1.16
19	MGCG	FM ₀	–	0	1.93	2.03	2.28	2.41
20			–	½	1.65	1.66	1.72	1.84
21		FA ₀	quintic	0	1.45	1.57	1.70	1.82
22			quintic	½	1.43	1.54	1.61	1.67
23	MG	FM _ℓ	–	0	2.41	2.53	2.66	2.83
24			–	½	2.10	2.54	3.01	3.18
25		FA _ℓ	quintic	0	2.02	2.35	2.56	3.11
26			quintic	½	2.47	2.61	3.26	3.53
27	MGCG	FA _ℓ	quintic	0	2.54	2.71	3.10	3.50
28			quintic	½	2.51	2.62	3.19	3.30

by approximately 50%. In our tests we observed runtime savings in face-centered case, whereas the extra cost prevailed in the element-centered case. According to Tab. 4, both element-centered methods show a strong sensitivity to the aspect ratio: Similar to CG, EM₀ degrades severely as soon as AR exceeds 4, whereas EA_ℓ retains still 30 to 40 percent of the original convergence rate with $AR = 8$. In comparison, the face-centered methods proved rather robust, in particular with higher ansatz order. For example, with

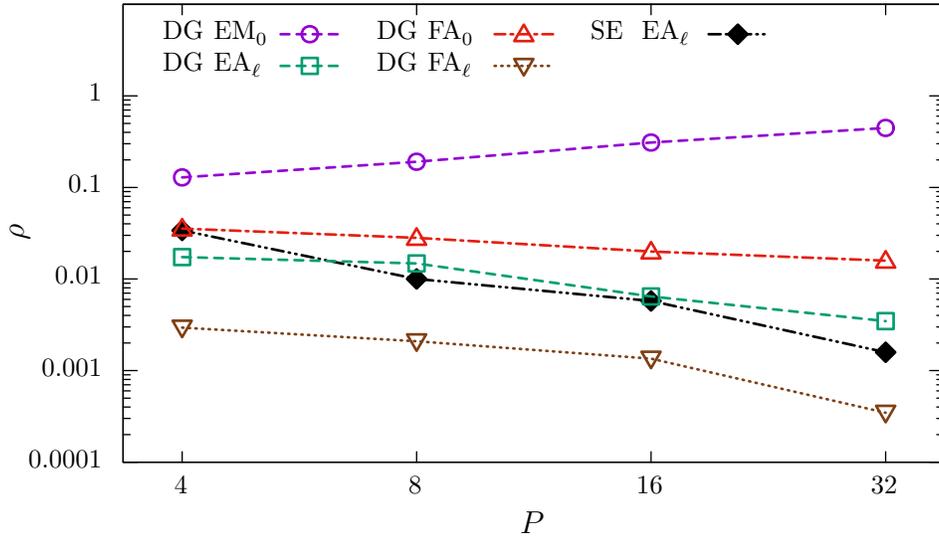
TABLE 3. Robustness of MGCG against the problem size; $\mu_* = 1$, $\beta = 0$, tessellation with $N_{E,d} \times N_{E,d}$ square elements.

P	$N_{E,d}$	EM ₀			EA _ℓ			FA ₀			FA _ℓ		
		\bar{r}	n_{10}	$\bar{\omega}$									
4	8	0.92	11	9.8	1.78	6	10.3	1.45	7	24.6	2.53	4	21.5
	16	0.90	12	10.0	1.76	6	10.4	1.45	7	24.6	2.54	4	21.4
	32	0.89	12	10.1	1.76	6	10.4	1.45	7	24.6	2.53	4	21.5
	64	0.89	12	10.1	1.76	6	10.4	1.45	7	24.6	2.53	4	21.5
	128	0.89	12	10.1	1.76	6	10.4	1.45	7	24.6	2.54	4	21.4
	256	0.89	12	10.1	1.76	6	10.4	1.45	7	24.6	2.53	4	21.5
	8	8	0.73	14	12.3	1.85	6	10.7	1.55	7	23.0	2.61	4
16		0.72	14	12.5	1.84	6	10.7	1.57	7	22.7	2.71	4	20.9
32		0.72	14	12.5	1.84	6	10.7	1.57	7	22.7	2.63	4	21.6
64		0.72	14	12.5	1.84	6	10.7	1.57	7	22.7	2.68	4	21.2
128		0.72	14	12.5	1.84	6	10.7	1.57	7	22.7	2.68	4	21.2
256		0.72	14	12.5	1.84	6	10.7	1.57	7	22.7	2.68	4	21.2
16		8	0.52	20	17.3	2.26	5	7.5	1.67	6	21.4	3.15	4
	16	0.52	20	17.3	2.20	5	7.7	1.70	6	21.0	3.10	4	16.8
	32	0.52	20	17.3	2.19	5	7.7	1.70	6	21.0	3.17	4	16.4
	64	0.52	20	17.3	2.19	5	7.7	1.70	6	21.0	3.11	4	16.7
	128	0.52	20	17.3	2.19	5	7.7	1.70	6	21.0	3.11	4	16.7
	256	0.52	20	17.3	2.19	5	7.7	1.70	6	21.0	3.12	4	16.7
	32	8	0.36	28	25.0	2.46	5	6.3	1.77	6	20.2	3.47	3
16		0.36	29	25.0	2.49	5	6.2	1.82	6	19.6	3.50	3	14.2
32		0.36	28	25.0	2.47	5	6.3	1.82	6	19.6	3.46	3	14.3
64		0.36	28	25.0	2.46	5	6.3	1.82	6	19.6	3.38	3	14.7
128		0.36	28	25.0	2.46	5	6.3	1.82	6	19.6	3.52	3	14.1
256		0.36	28	25.0	2.46	5	6.3	1.82	6	19.6	3.53	3	14.0

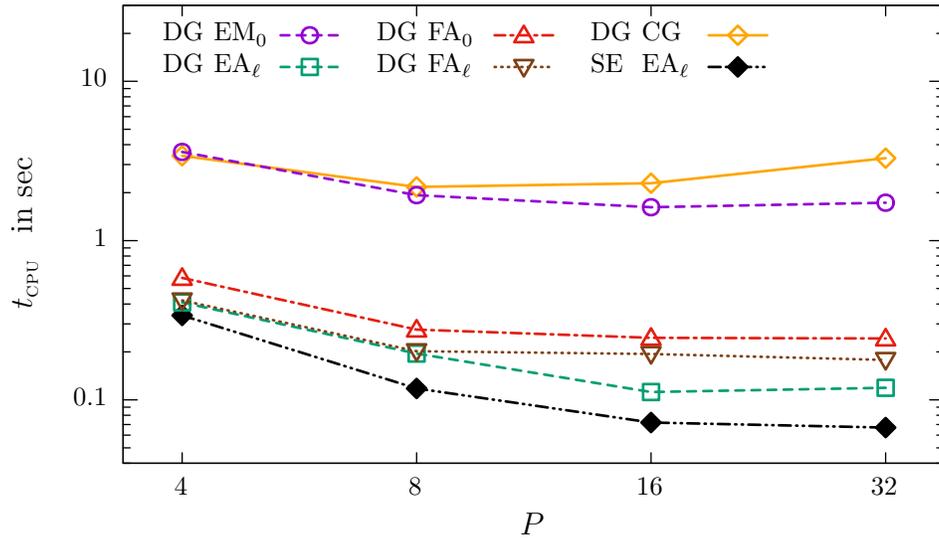
FA₀, n_{10} multiplies by 5 when increasing AR from 1 to 16 for $P = 4$, but only by 1.4 for $P = 16$. The overlapping face-centered method, FA_ℓ exhibits a similar behavior. Finally, Figure 7 shows the corresponding runtimes for $P = 16$. EA_ℓ remains the fastest method for $AR \leq 2$, whereas EA₀ is by far the slowest. Due to their better robustness, FA_ℓ and FA₀ close up with growing aspect ratio. They break even with EA_ℓ at $AR = 4$ and $AR = 8$, respectively, and gain a clear advantage for higher aspect ratios.

5. CONCLUSIONS

We presented a multigrid method for nodal discontinuous Galerkin formulations of the Poisson equation on two-dimensional Cartesian grids. The method adopts and extends techniques developed recently for the continuous spectral element method [26, 33]. Using



(a) Convergence rates.



(b) Runtimes.

FIGURE 6. MGCG convergence rates and runtimes for a 10^{10} residual reduction using $(256/P)^2$ square elements of order P . DG denotes the discontinuous Galerkin method with parameters and smoothers according to Tab. 3, SE the corresponding spectral element method, and CG the DG conjugate gradient solver.

TABLE 4. Robustness of MGCG against the aspect ratio $AR = \Delta x_1/\Delta x_2$; $\mu_\star = 1$, $\beta = 0$; EM_0 , EA_ℓ using one pre- and one post-smoothing and FA_0 , FA_ℓ a variable V-cycle with $N_{s,l} = (2^{L-l}, 2^{L-l})$.

P	AR	EM_0		EA_ℓ		FA_0		FA_ℓ	
		\bar{r}	n_{10}	\bar{r}	n_{10}	\bar{r}	n_{10}	\bar{r}	n_{10}
4	1	0.90	12	1.76	6	1.52	7	2.78	4
	2	0.74	14	1.26	8	1.33	8	2.49	5
	4	0.32	32	0.88	12	1.18	9	1.86	6
	8	0.13	80	0.47	22	0.85	12	1.05	10
	16	0.08	120	0.04	236	0.30	34	0.41	25
	32	0.07	140	0.03	321	0.13	79	0.16	62
8	1	0.72	14	1.84	6	1.63	7	3.10	4
	2	0.56	18	1.76	6	1.49	7	3.38	3
	4	0.29	35	1.20	9	1.43	7	2.63	4
	8	0.12	87	0.70	15	1.18	9	1.57	7
	16	0.07	141	0.25	40	0.77	14	0.91	12
	32	0.06	178	0.10	98	0.30	34	0.36	28
16	1	0.52	20	2.20	5	1.78	6	3.63	3
	2	0.37	28	2.07	5	1.62	7	3.64	3
	4	0.21	48	1.43	7	1.58	7	3.33	3
	8	0.10	97	0.85	12	1.57	7	2.58	4
	16	0.07	137	0.34	30	1.19	9	1.53	7
	32	0.06	161	0.13	76	0.60	17	0.80	13
32	1	0.35	29	2.49	5	1.89	6	3.96	3
	2	0.23	44	2.39	5	1.78	6	4.05	3
	4	0.15	65	1.71	6	1.80	6	4.22	3
	8	0.09	116	1.07	10	1.80	6	4.55	3
	16	0.07	150	0.41	25	1.64	7	2.55	4
	32	0.06	157	0.17	61	1.07	10	1.40	8

the nodal basis corresponding to the Gauss-Lobatto-Legendre points in conjunction with the related quadrature we derived a unified form of the discrete equations, which embodies the interior penalty method as well as the local discontinuous Galerkin method. These equations are solved by means of polynomial multigrid with multiplicative or weighted additive Schwarz methods for smoothing and, optionally, the inexact preconditioned conjugate gradient method [14] for acceleration. The Schwarz methods operate on a set of overlapping rectangular subdomains, which are either element- or face-centered. The resulting multigrid methods achieved excellent convergence rates independent from the problem size. Using a level-dependent overlap of $1 + \lfloor P_l/8 \rfloor$ nodes proved sufficient for robustness against the ansatz order up to $P = 32$. Taking advantage of tensor-product

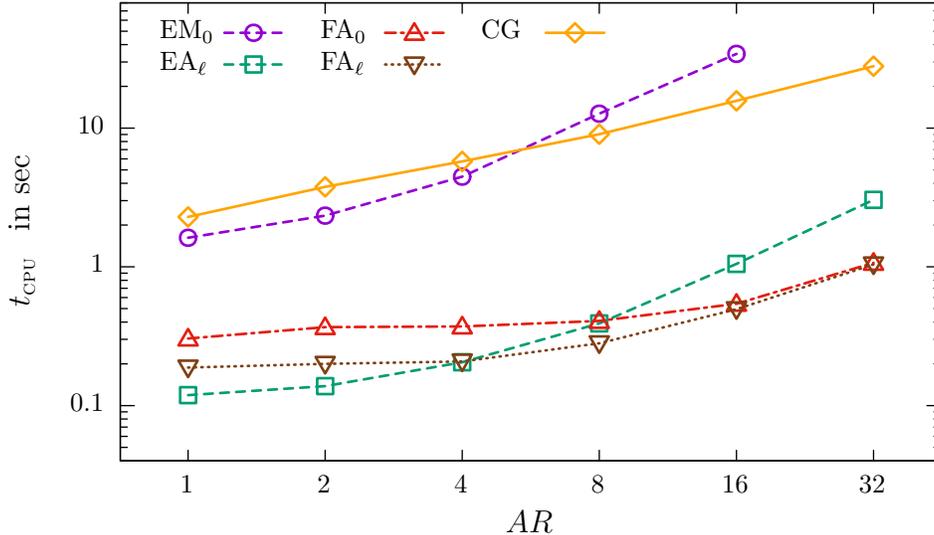


FIGURE 7. MGCG and CG runtimes for a 10^{10} residual reduction at different aspect ratios. Discretization using 16×16 elements of order $P=16$ with 73,984 unknowns; parameters and smoothers as in Tab. 4.

factorization and fast diagonalization techniques, the methods attain a computational complexity of $O(PN)$ per cycle. In terms of runtime, the solvers actually achieve linear complexity, since the convergence rate and the computational efficiency improve with growing order. Multigrid with conjugate gradient acceleration and the element-centered additive smoother with level-dependent overlap is the best choice for equidistant grids with nearly square elements, for which it achieves convergence rates between $\rho = 0.017$ and 0.003. With $P = 16$ it needs about 2600 multiplications per unknown to reduce the residual by ten orders of magnitudes. Multigrid with conjugate gradient acceleration and the face-centered additive smoother is twice as costly under these conditions, but proves more robust against the aspect ratio and becomes the preferred choice for aspect ratios greater than 4.

The proposed multigrid methods present an opportunity to enhance the competitiveness of high-order discontinuous Galerkin methods in more complex applications such as computational fluid dynamics. Due to its tensor-product structure, the approach offers a straightforward extension to three-dimensional problems, which is the subject to ongoing work. Further challenges include the development of multigrid preconditioners for variable diffusion and deformed meshes, as demonstrated by Fischer and Lottes [13] for the spectral element case.

ACKNOWLEDGEMENTS

Funding by German Research Foundation (DFG) in frame of the project STI 157/4-1 is gratefully acknowledged.

REFERENCES

- [1] P. F. Antonietti, M. Sarti, and M. Verani. Multigrid algorithms for hp -discontinuous Galerkin discretizations of elliptic problems. *SIAM Journal on Numerical Analysis*, 53(1):598–618, 2015.
- [2] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 39(5):1749–1779, 2001.
- [3] P. Bastian, M. Blatt, and R. Scheichl. Algebraic multigrid for discontinuous Galerkin discretizations of heterogeneous elliptic problems. *Numer. Linear Algebra Appl.*, 19(2):367–388, 2012.
- [4] J. Bramble. *Multigrid methods*. Pitman Res. Notes Math. Ser. 294. Longman Scientific & Technical, Harlow, UK, 1995.
- [5] B. Cockburn and C.-W. Shu. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, 1998.
- [6] B. Cockburn, G. E. Karniadakis, and C.-W. Shu. *Discontinuous Galerkin Methods: Theory, Computation and Applications*. Springer Berlin Heidelberg, 2000.
- [7] B. Cockburn, G. Kanschat, I. Perugia, and D. Schötzau. Superconvergence of the local discontinuous Galerkin method for elliptic problems on Cartesian grids. *SIAM Journal on Numerical Analysis*, 39(1):264–285, 2002.
- [8] W. Couzy and M. O. Deville. A fast Schur complement method for the spectral element discretization of the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 116:135–142, January 1995.
- [9] M. O. Deville, P. F. Fischer, and E. H. Mund. *High-Order Methods for Incompressible Fluid Flow*, volume 1. Cambridge University Press, 2002.
- [10] V. A. Dobrev, R. D. Lazarov, P. S. Vassilevski, and L. T. Zikatanov. Two-level preconditioning of discontinuous Galerkin approximations of second-order elliptic equations. *Numerical Linear Algebra with Applications*, 13(9):753–770, 2006.
- [11] K. J. Fidkowski, T. A. Oliver, J. Lu, and D. L. Darmofal. p -multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 207(1):92–113, 2005.
- [12] P. F. Fischer. Personal communication, September 2015.
- [13] P. F. Fischer and J. W. Lottes. Hybrid Schwarz-multigrid methods for the spectral element method: Extensions to Navier-Stokes. In *Domain Decomposition Methods in Science and Engineering Series*, pages 35–49. Springer, 2004.
- [14] G. H. Golub and Q. Ye. Inexact preconditioned conjugate gradient method with inner-outer iteration. *SIAM Journal on Scientific Computing*, 21(4):1305–1320, Dec. 1999. ISSN 10648275.
- [15] J. Gopalakrishnan and G. Kanschat. A multilevel discontinuous Galerkin method. *Numerische Mathematik*, 95(3):527–550, 2003.
- [16] L. Haupt, J. Stiller, and W. Nagel. A fast spectral element solver combining static condensation and multigrid techniques. *J. Comput. Phys.*, 255:384–395, 2013.

- [17] B. T. Helenbrook and H. L. Atkins. Application of p -multigrid to discontinuous Galerkin formulations of the poisson equation. *AIAA Journal*, 44(3):566–575, 2006.
- [18] B. T. Helenbrook, H. L. Atkins, and D. J. Mavriplis. Analysis of p -multigrid for continuous and discontinuous finite element discretizations. AIAA Paper 2003-3989, AIAA, June 2003.
- [19] J. S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods*. Springer, 2008.
- [20] I. Huismann, J. Stiller, and J. Fröhlich. Factorizing the factorization - a spectral-element solver for elliptic equations with linear operation count. *J. Computational Physics*, 2016. (submitted).
- [21] E. F. Kaasschieter. Preconditioned conjugate gradients for solving singular systems. *J. Comput. Appl. Math.*, 24:265–275, 1988.
- [22] G. Kanschat. Multilevel methods for discontinuous Galerkin FEM on locally refined meshes. *Computers and Structures*, 82:2437–2445, 2004.
- [23] G. Kanschat. Robust smoothers for high-order discontinuous Galerkin discretizations of advection-diffusion problems. *Journal of Computational and Applied Mathematics*, 218(1):53–60, 2008.
- [24] G. E. Karniadakis and S. J. Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, 2nd edition, 2005.
- [25] J. K. Kraus and S. K. Tomar. A multilevel method for discontinuous Galerkin approximation of three-dimensional anisotropic elliptic problems. *Numer. Linear Algebra Appl.*, 15(5):417–438, 2008.
- [26] J. W. Lottes and P. F. Fischer. Hybrid multigrid/Schwarz algorithms for the spectral element method. *J. Sci. Comput.*, 24:45–78, 2005.
- [27] R. E. Lynch, J. R. Rice, and D. H. Thomas. Direct solution of partial difference equations by tensor product methods. *Numer. Math.*, 6:185–199, 1964.
- [28] L. N. Olson and J. B. Schroder. Smoothed aggregation multigrid solvers for high-order discontinuous Galerkin methods for elliptic problems. *Journal of Computational Physics*, 230(18):6959–6976, 2011.
- [29] F. Prill, M. Lukáčová-Medvidová, and R. Hartmann. Smoothed aggregation multigrid for the discontinuous Galerkin method. *SIAM J. Sci. Comput.*, 31(5):3503–3528, 2009.
- [30] B. Rivière. *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation*, volume 35 of *Frontiers in Mathematics*. SIAM, Philadelphia, PA, USA, 2008.
- [31] K. Shahbazi, D. J. Mavriplis, and N. K. Burgess. Multigrid algorithms for high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations. *Journal of Computational Physics*, 228(21):7917–7940, 2009.
- [32] C. Siefert, R. Tuminaro, A. Gerstenberger, G. Scovazzi, and S. S. Collis. Algebraic multigrid techniques for discontinuous Galerkin methods with varying polynomial order. *Computational Geosciences*, 18(5):597–612, 2014.
- [33] J. Stiller. Nonuniformly weighted Schwarz smoothers for spectral element multigrid. *J. Sci. Comput.*, 2017. (DOI 10.1007/s10915-016-0345-z).

- [34] U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, 2000.
- [35] J. J. W. van der Vegt and S. Rhebergen. Hp-multigrid as smoother algorithm for higher order discontinuous Galerkin discretizations of advection dominated flows. Part II: Optimization of the Runge-Kutta smoother. *J. Comput. Phys.*, 231(22): 7564–7583, 2012.