DOMAIN DECOMPOSITION PRECONDITIONERS FOR DISCONTINUOUS GALERKIN METHODS FOR ELLIPTIC PROBLEMS ON COMPLICATED DOMAINS

PAOLA F. ANTONIETTI *, STEFANO GIANI †, AND PAUL HOUSTON ‡

Abstract. In this article we consider the application of Schwarz-type domain decomposition preconditioners for discontinuous Galerkin finite element approximations of elliptic partial differential equations posed on complicated domains, which are characterized by small details in the computational domain or microstructures. In this setting, it is necessary to define a suitable coarse-level solver, in order to guarantee the scalability of the preconditioner under mesh refinement. To this end, we exploit recent ideas developed in the so-called composite finite element framework, which allows for the definition of finite element methods on general meshes consisting of agglomerated elements. Numerical experiments highlighting the practical performance of the proposed preconditioner are presented.

Key words. Composite finite element methods, discontinuous Galerkin methods, domain decomposition, Schwarz preconditioners

1. Introduction. In recent years, considerable attention has been devoted to the development of efficient iterative solvers for the solution of the linear system of equations arising from the discontinuous Galerkin finite element (DGFEM) discretization of a range of model problems. In the framework of two level preconditioners, scalable non-overlapping Schwarz methods have been proposed and analyzed for the h–version of the DGFEM in the articles [21, 18, 4, 5, 8, 15, 9]. More recently, in [6, 11], it has been proved that the non-overlapping Schwarz preconditioners can also be successfully employed to reduce the condition number of the stiffness matrices arising from a wide class of high–order DGFEM discretizations of elliptic problems; see, also, [7]. We stress that Schwarz-type preconditioners are particularly suited to DGFEMs, in the sense that uniform scalability of the underlying iterative method may be established without the need to overlap the subdomain partition of the computational mesh. This is a particularly attractive property, since the absence of overlapping subdomains reduces communication between processors on parallel machines. By (uniform) scalability, we mean that the number of iterations needed to compute the solution of the underlying system of equations is uniform, as the mesh is refined, provided that an appropriate coarse mesh solution is computed as part of the preconditioning strategy. Of course, for uniformity, the ratio of granularity of the fine and coarse meshes must remain fixed under mesh refinement. However, as for most other multilevel strategies, the main difficulty regarding the implementation of this class of preconditioners is the construction of a coarser mesh starting from a given fine one. Indeed, naive strategies which are typically employed in practice often lead to the generation of "holes" in the coarse mesh, and the poor approximation of fine scale geometric features. Moreover, the application of these techniques is particularly problematic on unstructured meshes, or hybrid meshes consisting of mixed element types, as well as non-conforming meshes containing hanging nodes. The construction of a poor quality coarse mesh can lead to a significant degradation in the performance

* MOX–*Modeling and Scientific Computing*, Dipartimento di Matematica, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy, email: paola.antonietti@polimi.it.

† School of Engineering and Computing Sciences, Durham University , South Road, Durham, DH1 3LE, UK, email: stefano.giani@durham.ac.uk.

‡ School of Mathematical Sciences, University of Nottingham, University Park, Nottingham NG7 2RD, UK, email: Paul.Houston@nottingham.ac.uk.

of iterative solvers, and in particular to a loss of uniform scalability. By this we mean that the number of iterations needed to compute the numerical solution on the fine mesh increases dramatically under mesh refinement.

A new class of finite elements, referred to as Composite Finite Elements (CFEs), have been developed for the numerical solution of partial differential equations, which are particularly suited to problems characterized by small details in the computational domain or micro-structures; see, for example, [24, 23], for details. The key idea of CFEs is to exploit general shaped element domains upon which elemental basis functions may only be locally piecewise smooth. In particular, an element domain within a CFE may consist of a collection of neighboring elements present within a standard finite element method, with the basis function of the CFE being constructed as a linear combination of those defined on the standard finite element subdomains. In this way, CFEs offer an ideal mathematical and practical framework within which finite element solutions on (coarse) agglomerated meshes may be defined. CFEs have been developed in the context of h–version conforming finite element methods by Sauter and co-workers in the series of articles [24, 23, 26]; the generalization to hp–version discontinuous Galerkin composite finite element methods (DGCFEMs) has been considered in our recent article [10]. For related work on the application of DGFEMs on meshes consisting of agglomerated elements, we refer to the recent article [16], cf. also [20]. We point out that the general philosophy of CFE methods is to construct the underlying finite element spaces based on first generating a hierarchy of meshes, such that the finest mesh does indeed provide an accurate representation of the underlying computational domain, followed by the introduction of appropriate prolongation operators which determine how the finite element basis functions on the coarse mesh are defined in terms of those on the fine grid. We stress that CFE methods provide a flexible mathematical and practical framework within which coarse level approximations may be computed as part of a multilevel iterative solver strategy on domain conforming partitions consisting of general polygonal/polyhedral elements.

In this article we consider the application of Schwarz-type domain decomposition preconditioners for DGFEM approximations of elliptic partial differential equations posed on complicated domains, which are characterized by small details in the computational domain or microstructures. In particular, we exploit the DGCFEM proposed and analyzed in [10] to provide the necessary coarse mesh solver. The performance of the proposed preconditioner will be investigated through a series of numerical experiments. This article is organized as follows. In Section 2 we introduce the model problem and its (standard) DGFEM approximation. Section 3 is devoted to defining the DGCFEM, which represents a natural extension of the standard DGFEM on (coarse) agglomerated meshes. In Section 4 we construct the Schwarz preconditioners, and recall the main theoretical results shown in [6]. Section 5 outlines the implementation aspects of the proposed Schwarz preconditioners. In Section 6 we present some numerical results obtained with the additive Schwarz preconditioner. Finally, in Section 7 we summarize the work presented in this paper and draw some conclusions.

2. Model problem and discretization. In this article we consider the following model problem: given $f \in L_2(\Omega)$, find u such that

$$-\Delta u = f \quad \text{in } \Omega, \tag{2.1}$$

$$u = 0 \quad \text{on } \partial\Omega. \tag{2.2}$$

Here, Ω is a bounded, connected Lipschitz domain in $\mathbb{R}^d$, $d > 1$, with boundary $\partial\Omega$; in particular, it is assumed that Ω is a 'complicated' domain, in the sense that it contains small details or micro-structures.

To discretize (2.1)–(2.2), we employ the *hp*–version of the symmetric interior penalty DGFEM (IP DGFEM). We first introduce the necessary notation: we consider shape-regular meshes $\mathcal{T}_h$ that partition $\Omega \subset \mathbb{R}^d$ into open disjoint elements κ, such that $\overline{\Omega} = \bigcup_{\kappa\in\mathcal{T}_h} \overline{\kappa}$. By h_κ we denote the element diameter of $\kappa \in \mathcal{T}_h$, $h = \max_{\kappa\in\mathcal{T}_h} h_\kappa$, and $\boldsymbol{n}_\kappa$ signifies the unit outward normal vector to κ. We allow the meshes $\mathcal{T}_h$ to be *1-irregular*, i.e., each face of any one element $\kappa \in \mathcal{T}_h$ contains at most one hanging node (which, for simplicity, we assume to be the midpoint of the corresponding face). We assume that the family $\{\mathcal{T}_h\}_{h>0}$ is of *bounded local variation*, i.e., there exists a constant $\rho_1 \geq 1$, independent of the element sizes, such that

$$\rho_1^{-1} \leq h_\kappa/h_{\kappa'} \leq \rho_1, \tag{2.3}$$

for any pair of elements $\kappa, \kappa' \in \mathcal{T}_h$ which share a common face $F = \partial\kappa \cap \partial\kappa'$.

To each $\kappa \in \mathcal{T}_h$ we assign a polynomial degree $p_\kappa \geq 1$ (local approximation order) and define the degree vector $\mathbf{p} = \{p_\kappa : \kappa \in \mathcal{T}_h\}$. We suppose that $\mathbf{p}$ is also of bounded local variation, i.e., there exists a constant $\rho_2 \geq 1$, independent of the element sizes and $\mathbf{p}$, such that, for any pair of neighbouring elements $\kappa, \kappa' \in \mathcal{T}_h$,

$$\rho_2^{-1} \leq p_\kappa/p_{\kappa'} \leq \rho_2. \tag{2.4}$$

With this notation, we introduce the finite element space

$$V(\mathcal{T}_h, p) := \{v \in L_2(\Omega) : v|_\kappa \in \mathcal{S}_{p_\kappa}(\kappa) \quad \forall \kappa \in \mathcal{T}_h\},$$

where $\mathcal{S}_p(\kappa)$, $p \geq 1$, is either the space $\mathcal{P}_p(\kappa)$ of polynomials of degree at most p if κ is a simplex, or the space $\mathcal{Q}_p(\kappa)$ of all tensor product polynomials of degree at most p in each variable if κ is a hypercube. To avoid the proliferation of constants, throughout this article the notation $x \lesssim y$ means that there exists a constant $C > 0$, independent of the mesh size and the polynomial approximation order, such that $x \leq Cy$.

We shall now define some suitable face operators that are required for the definition of the IP DGFEM. Associated with the mesh $\mathcal{T}_h$, we denote by $\mathcal{F}^{\mathcal{I}}(\mathcal{T}_h)$ the set of all interior faces of the partition $\mathcal{T}_h$ of Ω, and by $\mathcal{F}^{\mathcal{B}}(\mathcal{T}_h)$ the set of all boundary faces of $\mathcal{T}_h$. In addition, $\mathcal{F}(\mathcal{T}_h) = \mathcal{F}^{\mathcal{B}}(\mathcal{T}_h) \cup \mathcal{F}^{\mathcal{I}}(\mathcal{T}_h)$ denotes the set of all faces in the mesh $\mathcal{T}_h$.

Let v and $\boldsymbol{q}$ be scalar- and vector-valued functions, respectively, which are smooth inside each element $\kappa \in \mathcal{T}_h$. Given two adjacent elements, $\kappa^+, \kappa^- \in \mathcal{T}_h$ which share a common face $F \in \mathcal{F}^{\mathcal{I}}(\mathcal{T}_h)$, i.e., $F = \partial\kappa^+ \cap \partial\kappa^-$, we write $v^\pm$ and $\boldsymbol{q}^\pm$ to denote the traces of the functions v and $\boldsymbol{q}$, respectively, on the face F, taken from the interior of $\kappa^\pm$, respectively. With this notation, the averages of v and $\boldsymbol{q}$ at $\boldsymbol{x} \in F$ are given by

$$\{\!\{v\}\!\} := \frac{1}{2}(v^+ + v^-), \qquad \{\!\{\boldsymbol{q}\}\!\} := \frac{1}{2}(\boldsymbol{q}^+ + \boldsymbol{q}^-),$$

respectively. Similarly, the jumps of v and $\boldsymbol{q}$ at $\boldsymbol{x} \in F$ are given by

$$[\![v]\!] := v^+\boldsymbol{n}_{\kappa^+} + v^-\boldsymbol{n}_{\kappa^-}, \qquad [\![\boldsymbol{q}]\!] := \boldsymbol{q}^+ \cdot \boldsymbol{n}_{\kappa^+} + \boldsymbol{q}^- \cdot \boldsymbol{n}_{\kappa^-},$$

respectively, where $\boldsymbol{n}_{\kappa^\pm}$ denotes the unit outward normal vector on $\partial\kappa^\pm$, respectively. On a boundary face $F \in \mathcal{F}^{\mathcal{B}}(\mathcal{T}_h)$, we set $\{\!\{v\}\!\} = v$, $\{\!\{\boldsymbol{q}\}\!\} = \boldsymbol{q}$, $[\![v]\!] = v\boldsymbol{n}$ and $[\![\boldsymbol{q}]\!] = \boldsymbol{q}\cdot\boldsymbol{n}$, with $\boldsymbol{n}$ denoting the unit outward normal vector on the boundary $\partial\Omega$.

For a face $F \in \mathcal{F}(\mathcal{T}_h)$, we define h_F to be diameter of the face; moreover, the face polynomial degree p_F is defined by

$$p_F := \begin{cases} \max(p_\kappa, p_{\kappa'}), & \text{if } F = \partial\kappa \cap \partial\kappa' \in \mathcal{F}^{\mathcal{I}}(\mathcal{T}_h), \\ p_\kappa, & \text{if } F = \partial\kappa \cap \partial\Omega \in \mathcal{F}^{\mathcal{B}}(\mathcal{T}_h). \end{cases} \tag{2.5}$$

Note that, under the assumption that our decomposition is shape-regular and *1-irregular* (with the hanging node at the midpoint of the corresponding face), we always have that, for any element κ in $\mathcal{T}_h$, and for any face $F \in \mathcal{F}(\mathcal{T}_h)$, $F \subset \partial\kappa$, the following inequality holds $h_F \lesssim h_\kappa$.

With this notation, the IP DGFEM numerical approximation of the problem (2.1)–(2.2) is defined as follows: find $u_h \in V(\mathcal{T}_h, p)$ such that

$$B_{\mathtt{DG}}(u_h, v_h) = F_h(v_h) \tag{2.6}$$

for all $v_h \in V(\mathcal{T}_h, p)$, where $F_h(v) := \int_\Omega f v \, d\mathbf{x}$, and

$$B_{\mathtt{DG}}(u, v) := \sum_{\kappa \in \mathcal{T}_h} \int_\kappa \nabla u \cdot \nabla v \, d\mathbf{x} - \sum_{F \in \mathcal{F}(\mathcal{T}_h)} \int_F \left(\{\!\{ \nabla_h v \}\!\} \cdot [\![u]\!] + \{\!\{ \nabla_h u \}\!\} \cdot [\![v]\!] \right) ds + \sum_{F \in \mathcal{F}(\mathcal{T}_h)} \int_F \sigma_h \, [\![u]\!] \cdot [\![v]\!] \, ds. \tag{2.7}$$

Here, ∇_h denotes the elementwise gradient operator. Furthermore, the function $\sigma_h \in L_\infty(\mathcal{F}(\mathcal{T}_h))$ is the discontinuity stabilization function that is chosen as

$$\sigma_h := \gamma \frac{p_F^2}{h_F}, \tag{2.8}$$

where $\gamma > 0$ is a (sufficiently large) constant. The bilinear form (2.7) is continuous and coercive in $V(\mathcal{T}_h, p)$ with respect to the energy norm

$$\|v\|_{\mathtt{DG}} := \left(\sum_{\kappa \in \mathcal{T}_h} \|\nabla v\|^2_{L_2(\kappa)} + \sum_{F \in \mathcal{F}(\mathcal{T}_h)} \|\sigma_h^{1/2} [\![v]\!]\|^2_{L_2(F)} \right)^{1/2}, \tag{2.9}$$

cf. [13, 14], for example.

We remark that the minimal dimension of the underlying finite element space $V(\mathcal{T}_h, p)$ constructed in this section is dependent on the number and size of any localized geometric features present in Ω. Indeed, *complicated* domains which contain, for example, a large number of small holes will necessarily require a fine mesh $\mathcal{T}_h$ to accurately describe Ω, thereby, giving rise to a (potentially) very large number of degrees of freedom in $V(\mathcal{T}_h, p)$. It is therefore essential to develop suitable iterative solution methods which can efficiently compute the finite element solution u_h defined by (2.6). In Section 4, we consider the construction of two-level Schwarz–type preconditioners. First, however, in the following section, we introduce the so–called composite DGFEM (DGCFEM), which is based on employing arbitrarily shaped (agglomerated) elements; this scheme will then form the basis of the coarse grid solver defined in Section 4.

3. Construction of composite DGFEMs. In this section we briefly introduce the composite version of the IP DGFEM; for further details, we refer to our recent article [10]. In order to develop the *a priori* error analysis outlined in [10], so-called logical meshes were constructed, based on employing 'standard' element shapes, i.e., simplices and hexahedra, for example. For the purposes of defining a coarse level solver for application within a Schwarz–type preconditioner, we consider a simpler and more general construction following the ideas developed in the article [16].

Given the (fine) mesh $\mathcal{T}_h$, we consider a coarsened mesh $\mathcal{T}_H$ which is constructed based on agglomerating elements from $\mathcal{T}_h$. Thereby, $\mathcal{T}_H$ represents a partition of Ω into disjoint elements $\mathcal{K}$, such that (i) $\overline{\Omega} = \cup_{\mathcal{K}\in\mathcal{T}_H}\overline{\mathcal{K}}$; (ii) we may write

$$\overline{\mathcal{K}} = \cup_{\kappa\in\mathcal{R}_\mathcal{K}}\overline{\kappa}, \tag{3.1}$$

where $\mathcal{R}_\mathcal{K}$ denotes the set of elements from the fine mesh $\mathcal{T}_h$, which are employed to construct $\mathcal{K}$. For the purposes of this article, we assume for simplicity, that the elements $\mathcal{K}$ are connected and satify a star-like property; we remark that this restriction may be relaxed, cf. [10]. As in the previous section, we denote by $\mathcal{F}^\mathcal{I}(\mathcal{T}_H)$ the set of all interior faces of the partition $\mathcal{T}_H$ of Ω, and by $\mathcal{F}^\mathcal{B}(\mathcal{T}_H)$ the set of all boundary faces of $\mathcal{T}_H$; furthermore, we set $\mathcal{F}(\mathcal{T}_H) = \mathcal{F}^\mathcal{I}(\mathcal{T}_H) \cup \mathcal{F}^\mathcal{B}(\mathcal{T}_H)$.

With this notation, we make the following key assumption:

(A1) For all elements $\mathcal{K} \in \mathcal{T}_H$, we define

$$C_\mathcal{K} = \text{card}\left\{F \in \mathcal{F}(\mathcal{T}_H) : F \subset \partial\mathcal{K}\right\}.$$

In the following we assume that there exists a positive constant C_F such that

$$\max_{\mathcal{K}\in\mathcal{T}_H} C_\mathcal{K} \leq C_F,$$

uniformly with respect to the mesh size.

To each composite/agglomerated element $\mathcal{K} \in \mathcal{T}_H$, we assign the polynomial degree $q_\mathcal{K} \geq 1$ as follows:

$$q_\mathcal{K} \leq \min_{\kappa\in\mathcal{R}_\mathcal{K}} p_\kappa.$$

Furthermore, we define the polynomial degree vector $\mathbf{q} = \{q_\mathcal{K} : \mathcal{K} \in \mathcal{T}_H\}$.

The construction of the composite finite element space $V(\mathcal{T}_H, q)$, cf. below, may be undertaken based on employing a suitable prolongation operator $R_0^\top$, cf. [24, 10]. We point out that the choice of $R_0^\top$ employed in [24] leads to finite element basis functions, defined on each composite element domain $\mathcal{K}$, which are piecewise polynomials. In contrast, we follow the approach developed in [10], whereby the restriction of a function from the underlying finite element space to an element $\mathcal{K} \in \mathcal{T}_H$ is a polynomial of degree $q_\mathcal{K}$. Thereby, we write

$$V(\mathcal{T}_H, q) = \left\{v \in L_2(\Omega) : v|_\mathcal{K} \in \mathcal{P}_{q_\mathcal{K}}(\mathcal{K}) \quad \forall\mathcal{K} \in \mathcal{T}_H\right\},$$

cf., also, [16]. Following [10], the classical prolongation (injection) operator from $V(\mathcal{T}_H, q)$ to $V(\mathcal{T}_h, p)$ is denoted by $R_0^\top : V(\mathcal{T}_H, q) \to V(\mathcal{T}_h, p)$. With this notation, we may write $V(\mathcal{T}_H, q)$ in the following alternative form

$$V(\mathcal{T}_H, q) = \left\{v \in L_2(\Omega) : v = R_0\phi,\ \phi \in V(\mathcal{T}_h, p)\right\},$$

where the restriction operator R_0 is defined as the transpose of $R_0^\top$ with respect to the $L_2(\Omega)$ inner product.

The DGCFEM discretization of the problem (2.1)–(2.2) is defined as follows: find $u_H \in V(\mathcal{T}_H, q)$ such that

$$B_{\mathtt{CDG}}(u_H, v_H) = F_H(v_H) \tag{3.2}$$

for all $v_H \in V(\mathcal{T}_H, q)$, where $B_{\mathtt{CDG}}(\cdot,\cdot)$ and $F_H(\cdot)$ are defined in an analogous manner to $B_{\mathtt{DG}}(\cdot,\cdot)$ and $F_h(\cdot)$, respectively, relative to the discontinuity stabilization function σ_H. We remark that σ_H is defined in a similar fashion to σ_h, subject to a change in the definition of the *representative* face volume employed and of the, possibly different, polynomial approximation degree employed on the coarse grid, cf. [10].

REMARK 3.1. *Given the choice of the prolongation operator $R_0^\top$, the implementation of the DGCFEM defined by (3.2) may be undertaken in a relatively straightforward manner following the ideas developed in the articles [16, 20]; we also refer to [10] for the case when more general prolongation operators may be employed. In the current article, the element polynomial bases on the (coarse) finite element space $V(\mathcal{T}_H, q)$ are constructed using the bounding box approach outlined in [20]; see [16] for related work.*

REMARK 3.2. *For the purposes of this article the fine and coarse meshes $\mathcal{T}_h$ and $\mathcal{T}_H$, respectively, are constructed based on employing the refinement algorithm outlined in our recent article [10], cf. Algorithm 3.1. Here, the essential idea is to first construct an overlapping coarse mesh, in the sense that it does not resolve the boundary of the computational domain Ω. Subsequently, this mesh is adaptively refined in such a manner that the resulting final (fine) mesh provides an accurate description of Ω; we point out that as this iteration strategy proceeds, sub-elements generated during the refinement process which lie outside of the computational domain are simply deleted. In this manner, after possible movement of the nodes close to the boundary $\partial\Omega$, we may construct $\mathcal{T}_h$. The coarse mesh is then defined in a natural manner, as the agglomeration of elements at a given refinement level, which all share the same parent element from the original overlapping mesh. Full details of this algorithm are given in [10].*

4. Non-overlapping Schwarz preconditioners. In this section we introduce two level non-overlapping Schwarz preconditioners in order to compute the numerical solution u_h defined on the fine space $V(\mathcal{T}_h, p)$ given by (2.6), cf. [4, 5, 6]. We denote by $\mathcal{T}_\mathcal{S} = \{\Omega_i\}_{i=1}^N$ a family of partitions of Ω into N non-overlapping domains, such that $\overline{\Omega} = \cup_{i=1}^N \overline{\Omega}_i$. With this (user–defined) partition, we consider two families of fine and coarse meshes $\mathcal{T}_h$ and $\mathcal{T}_H$, respectively, constructed as in the previous sections, respectively. In particular, we assume that $\mathcal{T}_h$, $\mathcal{T}_H$ and $\mathcal{T}_\mathcal{S}$ are nested, $\mathcal{T}_\mathcal{S} \subseteq \mathcal{T}_H \subseteq \mathcal{T}_h$, i.e., the subdomain partition does not cut any element of $\mathcal{T}_H$ and thereby of $\mathcal{T}_h$.

With this notation, we now introduce the local and coarse level solvers.

Local solvers. For $i = 1, \ldots, N$, the local DGFEM finite element spaces are defined on Ω_i, respectively, in the following manner:

$$V(\mathcal{T}_{h_i}, p) = \{v \in L_2(\Omega_i) : v|_\kappa \in \mathcal{S}_{p_\kappa}(\kappa) \quad \forall \kappa \in \mathcal{T}_{h_i}\},$$

where $\mathcal{T}_{h_i} = \{\kappa \in \mathcal{T}_h : \kappa \subset \Omega_i\}$. The classical prolongation (injection) operator from $V(\mathcal{T}_{h_i}, p)$ to $V(\mathcal{T}_h, p)$ is denoted by $R_i^\top : V(\mathcal{T}_{h_i}, p) \to V(\mathcal{T}_h, p)$. The restriction operator R_i is defined as the transpose of $R_i^\top$. The local solvers $B_{\mathtt{DG}_i}(\cdot,\cdot) : V(\mathcal{T}_{h_i}, p) \times V(\mathcal{T}_{h_i}, p) \to \mathbb{R}$ are defined as follows:

$$B_{\mathtt{DG}_i}(u_i, v_i) := B_{\mathtt{DG}}(R_i^\top u_i, R_i^\top v_i) \qquad \forall u_i, v_i \in V(\mathcal{T}_{h_i}, p), \quad i = 1, \ldots, N. \tag{4.1}$$

Coarse solver. Employing the composite discontinuous Galerkin finite element space $V(\mathcal{T}_H, q) \subset V(\mathcal{T}_h, p)$, the coarse solver $B_{\mathtt{DG}_0}(\cdot,\cdot) : V(\mathcal{T}_H, q) \times V(\mathcal{T}_H, q) \to \mathbb{R}$ is defined by

$$B_{\mathtt{DG}_0}(u_0, v_0) := B_{\mathtt{CDG}}(u_0, v_0) \qquad \forall u_0, v_0 \in V(\mathcal{T}_H, q). \tag{4.2}$$

Local projection operators. For $i = 1, \ldots, N$, the local projection operators $\widetilde{P}_i : V(\mathcal{T}_h, p) \to V(\mathcal{T}_{h_i}, p)$ are defined by:

$$B_{\mathtt{DG}_i}(\widetilde{P}_i u, v_i) := B_{\mathtt{DG}}(u, R_i^\top v_i) \quad \forall v_i \in V(\mathcal{T}_{h_i}, p).$$

Analogously, on the coarse space $V(\mathcal{T}_H, q)$, we let $\widetilde{P}_0 : V(\mathcal{T}_h, p) \to V(\mathcal{T}_H, q)$ be given by

$$B_{\mathtt{DG}_0}(\widetilde{P}_0 u, v_0) := B_{\mathtt{DG}}(u, R_0^\top v_0) \quad \forall v_0 \in V(\mathcal{T}_H, q).$$

With this notation, we define the projection operators

$$P_i := R_i^\top \widetilde{P}_i : V(\mathcal{T}_h, p) \to V(\mathcal{T}_h, p),$$

for $i = 0, 1, \ldots, N$.

Thereby, the additive and multiplicative Schwarz operators are defined, respectively, by

$$P_{\mathtt{ad}} := \sum_{i=0}^{N} P_i, \quad P_{\mathtt{mu}} := I - (I - P_N)(I - P_{N-1}) \cdots (I - P_0). \tag{4.3}$$

As in [6], a symmetrized variant of the multiplicative Schwarz operator may also defined.

To analyse the Schwarz operators (4.3) we follow the abstract framework of Schwarz methods [27]; see also [19] for a new additive convergence analysis of the multiplicative Schwarz operator $P_{\mathtt{mu}}$. Let $\kappa(P_{\mathtt{ad}})$ denote the condition number of the additive Schwarz operator $P_{\mathtt{ad}}$, i.e.,

$$\kappa(P_{\mathtt{ad}}) := \frac{\lambda_{\max}(P_{\mathtt{ad}})}{\lambda_{\min}(P_{\mathtt{ad}})},$$

where $\lambda_{\max}(P_{\mathtt{ad}})$ and $\lambda_{\min}(P_{\mathtt{ad}})$ are the extremal eigenvalues of the operator $P_{\mathtt{ad}}$, and let $E_{\mathtt{mu}}$ be the error propagation operator of the multiplicative Schwarz operator, i.e.,

$$E_{\mathtt{mu}} := (I - P_N)(I - P_{N-1}) \cdots (I - P_0). \tag{4.4}$$

We first recall the following preliminary result that will be needed in the forthcoming analysis; we refer to [6] for the proof.

LEMMA 4.1. *For any $v \in V(\mathcal{T}_h, p)$, let $v_i \in V(\mathcal{T}_{h_i}, p)$ be the restriction of v to Ω_i, i.e.,*

$$v_i := R_i v, \quad i = 1, \ldots, N.$$

Then, we have

$$\left| \sum_{\substack{i,j=1 \\ i \neq j}}^{N} B_{\mathtt{DG}}(R_i^\top v_i, R_j^\top v_j) \right| \lesssim B_{\mathtt{DG}}(v, v) + \sum_{\substack{i,j=1 \\ i \neq j}}^{N} \sum_{F \in \Gamma_{ij}} \|\sigma_h^{1/2} v_i\|_{L_2(F)}^2 + \|\sigma_h^{1/2} v_j\|_{L_2(F)}^2,$$

where Γ_{ij} *is the set of all faces* $F \in \mathcal{F}(\mathcal{T}_h)$ *such that* $F \subset \partial\Omega_i \cap \partial\Omega_j$, $i,j = 1,\ldots,N$. A key point for the analysis of the Schwarz preconditioners is the existence of the following stable splitting.

LEMMA 4.2. *Let* $C_\natural$ *be defined as*

$$C_\natural := \gamma \max_{\mathcal{K}\in\mathcal{T}_H} H_{\mathcal{K}} \frac{\max_{\kappa\in\mathcal{R}_{\mathcal{K}}} p_\kappa^2}{\min_{\kappa\in\mathcal{R}_{\mathcal{K}}} h_\kappa},$$

where $\mathcal{R}_{\mathcal{K}}$ *denotes the set of elements from the fine mesh* $\mathcal{T}_h$, *which are employed to construct* $\mathcal{K} \in \mathcal{T}_H$ *and* $H_{\mathcal{K}}$ *denotes the diameter of the element* $\mathcal{K} \in \mathcal{T}_H$. *Then, for any* $v \in V(\mathcal{T}_h, p)$ *there exist* $v_0 \in V(\mathcal{T}_H, q)$ *and* $v_i \in V(\mathcal{T}_{h_i}, p)$, $i = 1,\ldots,N$, *such that*

$$v = R_0^\top v_0 + \sum_{i=1}^N R_i^\top v_i,$$

and

$$\sum_{i=0}^N B_{\mathtt{DG}_i}(v_i, v_i) \lesssim C_\natural B_{\mathtt{DG}}(v, v).$$

Proof. Given $v \in V(\mathcal{T}_h, p)$, let $\Pi_0(v)$ be defined as the projection of v onto the space of piecewise constant functions defined on $\mathcal{T}_H$, i.e., $\Pi_0(v) \in V(\mathcal{T}_H, 0)$, and

$$\int_{\mathcal{K}} \Pi_0(v)\, w\, d\mathbf{x} := \int_{\mathcal{K}} v\, w\, d\mathbf{x} \quad \forall w \in V(\mathcal{T}_H, 0) \quad \forall \mathcal{K} \in \mathcal{T}_H.$$

Next, we define

$$v_0 := \Pi_0(v), \quad v_i := R_i(v - \Pi_0(v)), \quad i = 1,\ldots,N,$$

and observe that, since $V(\mathcal{T}_H, 0) \subset V(\mathcal{T}_h, 0) \subset V(\mathcal{T}_h, p)$, $R_0^\top \Pi_0(v) \equiv R_0^\top v_0 \in V(\mathcal{T}_h, p)$. Next, we decompose (uniquely) $v - R_0^\top \Pi_0(v) \equiv v - R_0^\top v_0$ as $v - R_0^\top v_0 = \sum_{i=1}^N R_i^\top v_i$. Using the above decomposition and the definition of the local solvers (4.1), we obtain

$$\begin{aligned} B_{\mathtt{DG}}(v - R_0^\top v_0, v - R_0^\top v_0) &= B_{\mathtt{DG}}(\sum_{i=1}^N R_i^\top v_i, \sum_{j=1}^N R_j^\top v_j) \\ &= \sum_{i=1}^N B_{\mathtt{DG}}(R_i^\top v_i, R_i^\top v_i) + \sum_{\substack{i,j=1 \\ i\neq j}}^N B_{\mathtt{DG}}(R_i^\top v_i, R_j^\top v_j) \\ &= \sum_{i=1}^N B_{\mathtt{DG}_i}(v_i, v_i) + \sum_{\substack{i,j=1 \\ i\neq j}}^N B_{\mathtt{DG}}(R_i^\top v_i, R_j^\top v_j). \end{aligned}$$

Rearranging the terms, we get

$$\sum_{i=1}^N B_{\mathtt{DG}_i}(v_i, v_i) = B_{\mathtt{DG}}(v - R_0^\top v_0, v - R_0^\top v_0) - \sum_{\substack{i,j=1 \\ i\neq j}}^N B_{\mathtt{DG}}(R_i^\top v_i, R_j^\top v_j),$$

and then

$$\sum_{i=1}^{N} B_{\mathtt{DG}_i}(v_i, v_i) \leq \left|B_{\mathtt{DG}}(v - R_0^\top v_0, v - R_0^\top v_0)\right| + \left|\sum_{\substack{i,j=1 \\ i\neq j}}^{N} B_{\mathtt{DG}}(R_i^\top v_i, R_j^\top v_j)\right|. \tag{4.5}$$

Next we estimate each of the two terms on the right hand side of (4.5). Before we proceed, we point out that, recalling that the coarse and fine partitions are nested, since $R_0^\top v_0$ is piecewise constant on each coarse element $\mathcal{K} \in \mathcal{T}_H$, it is also piecewise constant on each fine element $\kappa \in \mathcal{T}_h$. For the first term, we use the continuity of the bilinear form in the energy norm (2.9), exploit that $R_0^\top v_0$ is piecewise constant on each coarse element $\mathcal{K} \in \mathcal{T}_H$, and employ coercivity. Thereby, we get

$$\begin{aligned}
\left|B_{\mathtt{DG}}(v - R_0^\top v_0, v - R_0^\top v_0)\right| &\lesssim \sum_{\kappa\in\mathcal{T}_h} \|\nabla(v - R_0^\top v_0)\|^2_{L_2(\kappa)} + \sum_{F\in\mathcal{F}(\mathcal{T}_h)} \|\sigma_h^{1/2}[\![v - R_0^\top v_0]\!]\|^2_{L_2(F)} \\
&= \sum_{\kappa\in\mathcal{T}_h} \|\nabla v\|^2_{L_2(\kappa)} + \sum_{F\in\mathcal{F}(\mathcal{T}_h)} \|\sigma_h^{1/2}[\![v - R_0^\top v_0]\!]\|^2_{L_2(F)} \\
&\lesssim B_{\mathtt{DG}}(v, v) + \sum_{F\in\mathcal{F}(\mathcal{T}_h)} \|\sigma_h^{1/2}[\![v - R_0^\top v_0]\!]\|^2_{L_2(F)}.
\end{aligned}$$

Observing again that $R_0^\top v_0$ is continuous within each coarse element $\mathcal{K} \in \mathcal{T}_H$, the last term on the right hand side can be bounded as

$$\begin{aligned}
\sum_{F\in\mathcal{F}(\mathcal{T}_h)} \|\sigma_h^{1/2}[\![v - R_0^\top v_0]\!]\|^2_{L_2(F)} \leq \sum_{\mathcal{K}\in\mathcal{T}_H} \Bigg(&\sum_{F\in\mathcal{F}(\mathcal{T}_h):F\subset\mathcal{K}} \|\sigma_h^{1/2}[\![v]\!]\|^2_{L_2(F)} \\
&+ \sum_{F\in\mathcal{F}(\mathcal{T}_h):F\subset\partial\mathcal{K}} \|\sigma_h^{1/2}[\![v - R_0^\top v_0]\!]\|^2_{L_2(F)} \Bigg).
\end{aligned}$$

Thereby,

$$\sum_{F\in\mathcal{F}(\mathcal{T}_h)} \|\sigma_h^{1/2}[\![v - R_0^\top v_0]\!]\|^2_{L_2(F)} \leq B_{\mathtt{DG}}(v, v) + \sum_{\mathcal{K}\in\mathcal{T}_H} \gamma \frac{\max_{\kappa\in\mathcal{R}_\mathcal{K}} p_\kappa^2}{\min_{\kappa\in\mathcal{R}_\mathcal{K}} h_\kappa} \|v - R_0^\top v_0\|^2_{L_2(\partial\mathcal{K})},$$

where we have also used the definition (2.8) of the discontinuity stabilization function σ_h. Hence, we obtain

$$\left|B_{\mathtt{DG}}(v - R_0^\top v_0, v - R_0^\top v_0)\right| \lesssim B_{\mathtt{DG}}(v, v) + \sum_{\mathcal{K}\in\mathcal{T}_H} \gamma \frac{\max_{\kappa\in\mathcal{R}_\mathcal{K}} p_\kappa^2}{\min_{\kappa\in\mathcal{R}_\mathcal{K}} h_\kappa} \|v - R_0^\top v_0\|^2_{L_2(\partial\mathcal{K})}. \tag{4.6}$$

For the second term on the right hand side of (4.5), Lemma 4.1 immediately gives

$$\begin{aligned}
\left|\sum_{\substack{i,j=1 \\ i\neq j}}^{N} B_{\mathtt{DG}}(R_i^\top v_i, R_j^\top v_j)\right| &\lesssim B_{\mathtt{DG}}(v, v) + \sum_{\substack{i,j=1 \\ i\neq j}}^{N} \sum_{F\in\Gamma_{ij}} \|\sigma_h^{1/2} v_i\|^2_{L_2(F)} + \|\sigma_h^{1/2} v_j\|^2_{L_2(F)} \\
&\lesssim B_{\mathtt{DG}}(v, v) + \sum_{\mathcal{K}\in\mathcal{T}_H} \gamma \frac{\max_{\kappa\in\mathcal{R}_\mathcal{K}} p_\kappa^2}{\min_{\kappa\in\mathcal{R}_\mathcal{K}} h_\kappa} \|v - R_0^\top v_0\|^2_{L_2(\partial\mathcal{K})},
\end{aligned}$$

where the last step follows by observing that the partitions are nested. Collecting the previous estimates we obtain

$$\sum_{i=1}^{N} B_{\mathtt{DG}_i}(v_i,v_i) \lesssim B_{\mathtt{DG}}(v,v) + \sum_{\mathcal{K}\in\mathcal{T}_H} \gamma \frac{\max_{\kappa\in\mathcal{R}_\mathcal{K}} p_\kappa^2}{\min_{\kappa\in\mathcal{R}_\mathcal{K}} h_\kappa} \|v - R_0^\top v_0\|^2_{L_2(\partial\mathcal{K})}.$$

Next, we add $B_{\mathtt{DG}_0}(v_0,v_0)$ to both sides, i.e., we have

$$\sum_{i=0}^{N} B_{\mathtt{DG}_i}(v_i,v_i) \lesssim B_{\mathtt{DG}}(v,v) + B_{\mathtt{DG}_0}(v_0,v_0) + \sum_{\mathcal{K}\in\mathcal{T}_H} \gamma \frac{\max_{\kappa\in\mathcal{R}_\mathcal{K}} p_\kappa^2}{\min_{\kappa\in\mathcal{R}_\mathcal{K}} h_\kappa} \|v - R_0^\top v_0\|^2_{L_2(\partial\mathcal{K})}.$$

Using the definition of the coarse solver (4.2), recalling that v_0 is piecewise constant on the coarse mesh $\mathcal{T}_H$ (and therefore $R_0^\top v_0$ is piecewise constant on the fine mesh $\mathcal{T}_h$), and adding and subtracting v, we obtain

$$\begin{aligned} B_{\mathtt{DG}_0}(v_0,v_0) &= \sum_{F\in\mathcal{F}(\mathcal{T}_H)} \|\sigma_H^{1/2}[\![v_0]\!]\|^2_{L_2(F)} \\ &\leq \sum_{F\in\mathcal{F}(\mathcal{T}_h)} \|\sigma_H^{1/2}[\![v]\!]\|^2_{L_2(F)} + \sum_{F\in\mathcal{F}(\mathcal{T}_h)} \|\sigma_H^{1/2}[\![R_0^\top v_0 - v]\!]\|^2_{L_2(F)} \\ &\leq \sum_{F\in\mathcal{F}(\mathcal{T}_h)} \frac{\sigma_H}{\sigma_h} \|\sigma_h^{1/2}[\![v]\!]\|^2_{L_2(F)} + \sum_{F\in\mathcal{F}(\mathcal{T}_h)} \frac{\sigma_H}{\sigma_h} \|\sigma_h^{1/2}[\![R_0^\top v_0 - v]\!]\|^2_{L_2(F)}, \end{aligned}$$

where we have also used that any face $F\in\mathcal{F}(\mathcal{T}_H)$ is the union of a number of faces of the fine mesh $\mathcal{T}_h$. Using the definition of the discontinuity stabilization function (2.8) and observing that $\sigma_H/\sigma_h < 1$, we finally obtain

$$\begin{aligned} B_{\mathtt{DG}_0}(v_0,v_0) &\leq \|v\|^2_{\mathtt{DG}} + \sum_{\mathcal{K}\in\mathcal{T}_H} \gamma \frac{\max_{\kappa\in\mathcal{R}_\mathcal{K}} p_\kappa^2}{\min_{\kappa\in\mathcal{R}_\mathcal{K}} h_\kappa} \|v - R_0^\top v_0\|^2_{L_2(\partial\mathcal{K})} \\ &\lesssim B_{\mathtt{DG}}(v,v) + \sum_{\mathcal{K}\in\mathcal{T}_H} \gamma \frac{\max_{\kappa\in\mathcal{R}_\mathcal{K}} p_\kappa^2}{\min_{\kappa\in\mathcal{R}_\mathcal{K}} h_\kappa} \|v - R_0^\top v_0\|^2_{L_2(\partial\mathcal{K})}. \end{aligned}$$

Collecting all the previous estimates we obtain

$$\sum_{i=0}^{N} B_{\mathtt{DG}_i}(v_i,v_i) \lesssim B_{\mathtt{DG}}(v,v) + \sum_{\mathcal{K}\in\mathcal{T}_H} \gamma \frac{\max_{\kappa\in\mathcal{R}_\mathcal{K}} p_\kappa^2}{\min_{\kappa\in\mathcal{R}_\mathcal{K}} h_\kappa} \|v - R_0^\top v_0\|^2_{L_2(\partial\mathcal{K})}. \tag{4.7}$$

To estimate the last term on the right hand side of (4.7) we make use of the following trace inequality given in [21]:

$$\begin{aligned} &\|v - R_0^\top v_0\|^2_{L_2(\partial\mathcal{K})} \lesssim H_\mathcal{K}^{-1} \|v - R_0^\top v_0\|^2_{L_2(\mathcal{K})} \\ &+ H_\mathcal{K} \left(\sum_{\kappa\in\mathcal{T}_h:\kappa\subset\mathcal{K}} \|\nabla_h (v - R_0^\top v_0)\|^2_{L_2(\kappa)} + \sum_{F\in\mathcal{F}(\mathcal{T}_h):F\subset\mathcal{K}} \|\sigma_h^{1/2}[\![v - R_0^\top v_0]\!]\|^2_{L_2(F)} \right). \end{aligned}$$

Recalling that $R_0^\top v_0$ is continuous and constant on each element $\mathcal{K}\in\mathcal{T}_H$ and summing over the coarse elements we obtain

$$\sum_{\mathcal{K}\in\mathcal{T}_H} \|v - R_0^\top v_0\|^2_{L_2(\partial\mathcal{K})} \lesssim \sum_{\mathcal{K}\in\mathcal{T}_H} H_\mathcal{K}^{-1} \|v - R_0^\top v_0\|^2_{L_2(\mathcal{K})} + H_\mathcal{K} \|v\|^2_{\mathtt{DG}},$$

which inserted in (4.7) leads to

$$\sum_{i=0}^{N} B_{\mathtt{DG}_i}(v_i,v_i) \lesssim (1+H_{\mathcal{K}})B_{\mathtt{DG}}(v,v) + \sum_{\mathcal{K}\in\mathcal{T}_H} \gamma \frac{\max_{\kappa\in\mathcal{R}_{\mathcal{K}}} p_\kappa^2}{\min_{\kappa\in\mathcal{R}_{\mathcal{K}}} h_\kappa} H_{\mathcal{K}}^{-1} \|v - R_0^\top v_0\|^2_{L_2(\mathcal{K})}. \tag{4.8}$$

To estimate the term $\|v-\Pi_0(v)\|^2_{L_2(\mathcal{K})}$ on the right hand side, we recall the broken Sobolev–Poincaré inequality

$$\|v-\Pi_0(v)\|^2_{L_2(\mathcal{K})} \lesssim H_{\mathcal{K}}^2 \left(\sum_{\kappa\in\mathcal{R}_{\mathcal{K}}} \|\nabla v\|^2_{L_2(\kappa)} + \sum_{F\in\mathcal{F}(\mathcal{T}_h):F\subset\mathcal{K}} \|\sigma_h^{1/2}[\![v]\!]\|^2_{L_2(F)} \right),$$

cf. [17]. The lemma now follows by employing the above inequality in (4.8). ∎

Using the stable splitting given in Lemma 4.2 and the abstract framework for the analysis of Schwarz methods [27], we are now able to state the main result of this section.

THEOREM 4.3. *Let $C_\natural$ be defined as in Lemma 4.2, i.e.,*

$$C_\natural := \gamma \max_{\mathcal{K}\in\mathcal{T}_H} H_{\mathcal{K}} \frac{\max_{\kappa\in\mathcal{R}_{\mathcal{K}}} p_\kappa^2}{\min_{\kappa\in\mathcal{R}_{\mathcal{K}}} h_\kappa},$$

where $\mathcal{R}_{\mathcal{K}}$ denotes the set of elements from the fine mesh $\mathcal{T}_h$, which are employed to construct $\mathcal{K}\in\mathcal{T}_H$. Then, there exist constants $C_1, C_2 \geq 1$, independent of the local meshsize and the local polynomial degrees, such that

$$\kappa(P_{\mathtt{ad}}) \leq C_1 C_\natural.$$

Moreover, the error propagation operator (4.4) *of the multiplicative Schwarz operator satisfies*

$$\|E_{\mathtt{mu}}\|_{\mathtt{DG}} \leq 1 - \frac{1}{1+C_2C_\natural},$$

and therefore a simple Richardson iteration applied to the preconditioned system $P_{\mathtt{mu}}u = g_{\mathtt{mu}}$, with $g_{\mathtt{mu}}$ a suitable right hand side, converges.

Thereby, for quasi-uniform meshes $\mathcal{T}_h$ and $\mathcal{T}_H$ of granularity h and H, respectively, and uniform polynomial orders, i.e., $p_\kappa \equiv p$ for all $\kappa\in\mathcal{T}_h$, we deduce that

$$\kappa(P_{\mathtt{ad}}) \leq C_1\gamma p^2 \frac{H}{h}, \quad E_{\mathtt{mu}} \leq 1 - \frac{h}{1-C_2\gamma p^2 H}.$$

REMARK 4.4. *As shown in Theorem 4.3, the condition number of the preconditioned systems still depend on the penalty parameter γ appearing in the definition of the penalization function* (2.8)*. To obtain preconditioners that are scalable also with respect to the penalty parameter we refer to [9].*

5. Implementation. In this section we briefly outline the implementation of the above two-level preconditioners; for further details concerning the implementation of the coarse grid (composite) DGCFEM, we refer to [10]. We first rewrite (2.6) in the following (equivalent) matrix form: find $\mathbf{U}\in\mathbb{R}^n$ such that

$$\mathbf{AU} = \mathbf{F},$$

where $\mathbf{A}\in\mathbb{R}^{n\times n}$ is the matrix representation of the bilinear form $B_{\mathtt{DG}}(\cdot,\cdot)$, $\mathbf{F}\in\mathbb{R}^n$ is the vector representation of the linear functional $F_h(\cdot)$, and $n := \dim(V(\mathcal{T}_h,p))$. In order to construct the Schwarz preconditioners outlined in the previous section, we now proceed in the following steps:

Aggregation. The construction of the coarse finite element space $V(\mathcal{T}_H, q)$ from the fine finite element space $V(\mathcal{T}_h, p)$ is undertaken, based on employing the aggregation algorithm outlined in [10]. In particular, the idea is to construct the shape functions defined on the coarse space $V(\mathcal{T}_H, q)$ as a linear combination of the basis functions from $V(\mathcal{T}_h, p)$; i.e., setting $n_0 := \dim(V(\mathcal{T}_H, q))$, we write a coarse level basis function $\phi_{\mathtt{H},i}$, $i = 1, \ldots, n_0$, in the following manner

$$\phi_{\mathtt{H},i} := \sum_{j=1,\ldots,n} \Lambda_{i,j} \phi_{h,j} \ ,$$

where $\{\phi_{h,j}\}_{j=1}^{n}$ denotes the set of basis functions which span $V(\mathcal{T}_h, p)$ and

$$\Lambda = [\Lambda_{i,j}]_{i=1,\ldots,n_0; j=1,\ldots,n}$$

defines the matrix representation used to determine how a function defined within the DGCFEM space $V(\mathcal{T}_H, q)$ may be represented in the richer space $V(\mathcal{T}_h, p)$. As noted in [10], the general approach employed for the computation of the coefficients $\Lambda_{i,j}$ is based on solving a series of *small* linear systems; here, we employ an *LU* factorization with column pivoting.

Projection (fine to coarse). Exploiting the aggregation matrix Λ introduced above, the matrix representation of the projection operator $\widetilde{P}_0 : V(\mathcal{T}_h, p) \to V(\mathcal{T}_H, q)$ is defined in the following manner:

$$\widetilde{\mathbf{P}}_0 := \Lambda \frac{n_0}{n} \ .$$

We remark that the scaling factor n_0/n employed within the definition of $\widetilde{\mathbf{P}}_0$ is typically used in practice, since it is observed that this scaling leads to a slight reduction in the number of iterations required to attain convergence of the underlying conjugate gradient solver, for example. Finally, we note that the matrix Λ is in general quite sparse, since the basis functions of both the coarse and fine level spaces have small support; see [10] for details.

Prolongation (coarse to fine). The matrix representation of the prolongation operator $R_0^\top : V(\mathcal{T}_H, q) \to V(\mathcal{T}_h, p)$ is given by

$$\mathbf{R}_0^\top := \Lambda^\top \ .$$

Projection onto local spaces. Since the global space $V(\mathcal{T}_h, p)$ naturally contains the basis functions which form each of the local finite element spaces $V(\mathcal{T}_{h_i}, p)$, $1 \leq i \leq N$, the prolongation operators $R_i^\top : V(\mathcal{T}_{h_i}, p) \to V(\mathcal{T}_h, p)$ and the corresponding local projection operators $\widetilde{P}_i : V(\mathcal{T}_h, p) \to V(\mathcal{T}_{h_i}, p)$ are very simple to construct. Indeed, writing $\mathbf{R}_i^\top$, $i = 1, \ldots, N$, and $\widetilde{\mathbf{P}}_i$, $i = 1, \ldots, N$, to denote the respective matrix representations of the prolongation and projection operators, respectively, we note that for a given i, $i = 1, \ldots, N$, $\mathbf{R}_i^\top$ is a rectangular matrix of size $n \times n_i$, where $n_i := \dim(V(\mathcal{T}_{h_i}, p))$. Moreover, we note that $[\mathbf{R}_i^\top]_{k,j} = 1$, only when the indices j and k correspond to the same basis function which belongs to both $V(\mathcal{T}_{h_i}, p)$ and $V(\mathcal{T}_h, p)$, subject to the given ordering of the basis functions within each of these finite element spaces; otherwise $[\mathbf{R}_i^\top]_{k,j} = 0$. With this construction, it is straightforward to see that $\widetilde{\mathbf{P}}_i = (\mathbf{R}_i^\top)^\top$, $i = 1, \ldots, N$.

Construction of the preconditioner. Firstly, we write $\mathbf{A}_0 \in \mathbb{R}^{n_0 \times n_0}$ to denote the matrix representation of the bilinear form $B_{\mathtt{DG}_0}(\cdot,\cdot)$ arising in the coarse DGCFEM solver defined in (4.2). Similarly, for $i = 1, \ldots, N$, we also write $\mathbf{A}_i \in \mathbb{R}^{n_i \times n_i}$ to be the matrix representation of the local bilinear forms $B_{\mathtt{DG}_i}(\cdot,\cdot)$ defined in (4.1), respectively. With this notation, the matrix representation of the projection operators $P_i : V(\mathcal{T}_h, p) \to V(\mathcal{T}_h, p)$, $i = 0, 1, \ldots, N$, is given by

$$\mathbf{P}_i = \mathbf{R}_i^\top \mathbf{A}_i^{-1} \mathbf{R}_i \mathbf{A} \in \mathbb{R}^{n \times n},$$

where $\mathbf{R}_0 := \widetilde{\mathbf{P}}_0$. Thereby, the matrix representations $\mathbf{P}_{\mathtt{ad}}$ and $\mathbf{P}_{\mathtt{mu}}$ of the additive and multiplicative Schwarz operators $P_{\mathtt{ad}}$ and $P_{\mathtt{mu}}$, respectively, are given by

$$\mathbf{P}_{\mathtt{ad}} = \sum_{i=0}^{N} \mathbf{P}_i = \sum_{i=0}^{N} \mathbf{R}_i^\top \mathbf{A}_i^{-1} \mathbf{R}_i \mathbf{A} \equiv \mathbf{M}_{\mathtt{ad}} \mathbf{A}$$

and

$$\mathbf{P}_{\mathtt{mu}} = \mathbf{I} - (\mathbf{I} - \mathbf{P}_N)(\mathbf{I} - \mathbf{P}_{N-1}) \cdots (\mathbf{I} - \mathbf{P}_0) \equiv \mathbf{M}_{\mathtt{mu}} \mathbf{A},$$

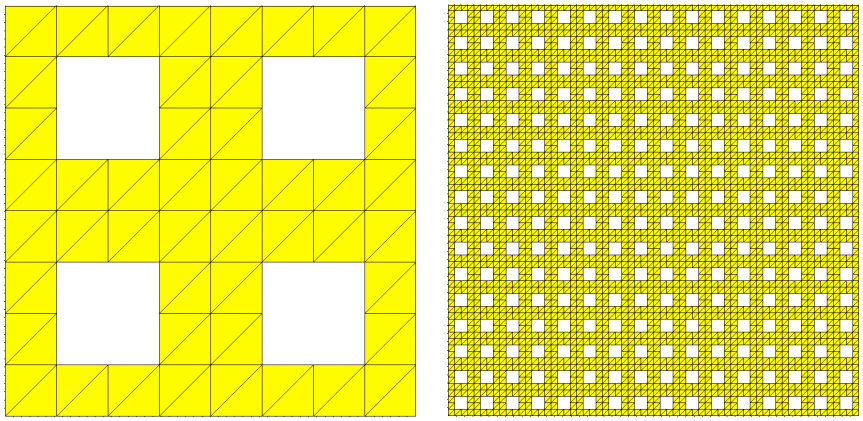
where $\mathbf{I}$ denotes the $n \times n$ identity matrix. Here, $\mathbf{M}_{\mathtt{ad}}$ and $\mathbf{M}_{\mathtt{mu}}$ are referred to as the additive and multiplicative Schwarz preconditioners, respectively.

From an implementation point of view, we simply need to compute the action of $\mathbf{M}_{\mathtt{ad}}$ or $\mathbf{M}_{\mathtt{mu}}$ on a generic vector when employing an iterative solver to evaluate the solution of the underlying set of linear equations. Algorithm 1 outlines the action of the additive Schwarz preconditioner; the multiplicative variant of the preconditioner may be constructed in an analogous fashion. We point out that the resulting system of linear equations stemming from the coarse grid solver $B_{\mathtt{DG}_0}(\cdot,\cdot)$ and the systems of linear equations arising from the local solvers $B_{\mathtt{DG}_i}(\cdot,\cdot)$, $i = 1, \ldots, N$, are solved based on employing the Multifrontal Massively Parallel Solver (MUMPS), see [1, 2, 3]. In order to take full advantage of MUMPS, the factorizations of all problems $B_{\mathtt{DG}_0}(\cdot,\cdot)$ and $B_{\mathtt{DG}_i}(\cdot,\cdot)$ are computed only once and used in all calls to the preconditioner. However, we point out that these may themselves be replaced by iterative solution methods.

Algorithm 1 Action of the additive Schwarz preconditioner $\mathbf{M}_{\mathtt{ad}}$ on a generic vector $\mathbf{x} \in \mathbb{R}^n$: `function` $\mathbf{z} = \mathbf{M}_{\mathtt{ad}}\mathbf{x}$.

Initialize the resulting vector to zero, i.e., set $\mathbf{z} = \mathbf{0}$.
for $i = 1 \to N$ **do** ▷ Loop over local problems
  Restrict the vector $\mathbf{x}$ to the local space, i.e., set $\mathbf{x}_i := \mathbf{R}_i \mathbf{x}$.
  Compute the solution $\mathbf{U}_i$ of the local problem $\mathbf{A}_i \mathbf{U}_i = \mathbf{x}_i$.
  Prolongate the vector $\mathbf{U}_i$ onto the global space, i.e., set $\mathbf{z}_i := \mathbf{R}_i^\top \mathbf{U}_i$.
  Set $\mathbf{z} = \mathbf{z} + \mathbf{z}_i$.
end for

Restrict the vector $\mathbf{x}$ onto coarse space, i.e., set $\mathbf{x}_0 := \widetilde{\mathbf{P}}_0 \mathbf{x}$. ▷ Coarse solver
Compute the solution $\mathbf{U}_0$ of the coarse level problem $\mathbf{A}_0 \mathbf{U}_0 = \mathbf{x}_0$.
Prolongate the vector $\mathbf{U}_0$ onto the global space, i.e., set $\mathbf{z}_0 := \mathbf{R}_0^\top \mathbf{U}_0$.
Set $\mathbf{z} = \mathbf{z} + \mathbf{z}_0$.



(a) (b)

FIG. 6.1. *Example 1: (a) Computational domain with 4 holes; (b) Computational domain with 256 holes.*

h^{-1} \ H^{-1}	2	4	8	16	32	64
4	19 (17.2)	-	-	-	-	-
8	32 (42.7)	27 (17.9)	-	-	-	-
16	58 (94.8)	47 (43.4)	29 (17.7)	-	-	-
32	93 (199.1)	74 (97.7)	48 (44.5)	31 (17.9)	-	-
64	134 (403.5)	121 (203.2)	80 (97.5)	50 (44.2)	31 (18.0)	-
128	192 (809.5)	185 (410.8)	137 (195.6)	80 (95.5)	50 (44.3)	31 (18.0)

TABLE 6.1

Example 1: Iteration counts and condition numbers for the preconditioner when $\Omega = (0,1)^2$, i.e., when Ω does not contain any holes, with $p = q = 1$.

6. Numerical experiments. In this section we present a series of computational examples to highlight the practical performance of the non-overlapping Schwarz preconditioners proposed in this article for problems where the underlying computational domain contains micro-structures. For simplicity, we restrict ourselves to the additive Schwarz operator $P_{\mathtt{ad}}$. Throughout this section we select the constant γ appearing in the discontinuity stabilization function σ_h equal to 10. All the numerical examples presented in this section have been computed using the AptoFEM package (`www.aptofem.com`).

6.1. Example 1: Two–dimensional domain with micro-structures. In this first example, we consider a computational domain Ω which contains (potentially) a large number of holes. Here, we let Ω be the unit square $(0,1)^2$ in two-dimensions, which has had a series of uniformly spaced square holes removed; here, we consider cases where up to 256 small square holes are removed. The size and the number of holes are chosen in each case in such a way that the area of the domain remains constant. In this way we can compare results from simulations with different numbers of holes. As an illustration, in Figure 6.1, we show the two cases where 4 and 256 holes have been introduced. Here, we select the right-hand side forcing function f and appropriate inhomogeneous boundary condition $u = g$ on $\partial\Omega$, so that the analytical solution to (2.1)–(2.2) is given by $u = \exp(x\,y)$.

h^{-1} \ H^{-1}	2	4	8	16	32	64
8	**32 (42.1)**	**27 (14.5)**	-	-	-	-
16	**58 (96.8)**	**47 (40.1)**	29 (17.5)	-	-	-
32	**93 (203.2)**	**74 (89.8)**	48 (44.1)	31 (17.8)	-	-
64	**134 (411.2)**	**121 (188.3)**	80 (95.4)	50 (44.2)	31 (17.9)	-
128	**192 (821.9)**	**185 (369.8)**	137 (194.3)	80 (95.2)	50 (44.2)	31 (17.9)

TABLE 6.2

Example 1: Iteration counts and condition numbers for the preconditioner when Ω contains 4 holes, with $p = q = 1$.

h^{-1} \ H^{-1}	2	4	8	16	32	64
16	**49 (69.6)**	**47 (40.6)**	**29 (14.9)**	-	-	-
32	**70 (152.2)**	**69 (93.6)**	**51 (37.3)**	31 (17.7)	-	-
64	**103 (314.6)**	**99 (196.9)**	**86 (86.1)**	50 (43.8)	31 (17.8)	-
128	**143 (636.3)**	**141 (399.7)**	**134 (180.9)**	80 (94.5)	50 (44.0)	31 (17.9)

TABLE 6.3

Example 1: Iteration counts and condition numbers for the preconditioner when Ω contains 16 holes, with $p = q = 1$.

h^{-1} \ H^{-1}	2	4	8	16	32	64
32	**52 (81.4)**	**52 (69.3)**	**49 (40.5)**	**30 (14.8)**	-	-
64	**75 (174.6)**	**75 (151.5)**	**74 (93.3)**	**51 (38.6)**	31 (17.7)	-
128	**106 (359.0)**	**106 (313.2)**	**106 (195.8)**	**87 (85.5)**	50 (43.8)	31 (17.8)

TABLE 6.4

Example 1: Iteration counts and condition numbers for the preconditioner when Ω contains 64 holes, with $p = q = 1$.

h^{-1} \ H^{-1}	2	4	8	16	32	64
64	**55 (83.8)**	**55 (81.3)**	**54 (69.2)**	**50 (40.4)**	**31 (14.7)**	-
128	**79 (178.6)**	**79 (174.5)**	**79 (151.4)**	**76 (93.2)**	**52 (38.2)**	31 (17.6)

TABLE 6.5

Example 1: Iteration counts and condition numbers for the preconditioner when Ω contains 256 holes, with $p = q = 1$.

In all the simulations presented below, we employ a fine level mesh $\mathcal{T}_h$ which is fine enough to precisely describe the computational domain Ω using 'standard' element shapes. On the other hand, employing a coarse level solver based on exploiting the DGCFEM outlined in Section 3, we are able to use very coarse meshes $\mathcal{T}_H$. In particular, in the following tables, cases highlighted in bold indicate when $\mathcal{T}_H$ is too coarse to precisely describe Ω using standard element shapes; instead general polygonal element domains, consisting of agglomerated elements are employed. Furthermore, the simulations have been performed, based on partitioning the fine level mesh $\mathcal{T}_h$ into 4 subdomains, i.e., $N = 4$. We employ coarse and fine finite element spaces $V(\mathcal{T}_H, q)$ and $V(\mathcal{T}_h, p)$, respectively, based on exploiting quasi-uniform coarse and fine meshes

h^{-1} \ H^{-1}	2	4	8	16	32	64
4	29 (36.9)	-	-	-	-	-
8	51 (95.6)	40 (39.5)	-	-	-	-
16	85 (201.0)	63 (99.8)	41 (39.8)	-	-	-
32	128 (395.5)	92 (198.6)	65 (100.0)	43 (39.9)	-	-
64	215 (763.0)	140 (382.5)	95 (205.6)	65 (100.0)	43 (39.8)	-
128	328 (996.5)	245 (704.0)	144 (414.6)	95 (188.5)	65 (100.1)	43 (32.7)

TABLE 6.6

Example 1: Iteration counts and condition numbers for the preconditioner when $\Omega = (0,1)^2$, i.e., when Ω does not contain any holes, with $p = q = 2$.

$\mathcal{T}_H$ and $\mathcal{T}_h$, respectively, of granularity H and h, respectively, with corresponding uniform polynomial degrees q and p, respectively. For reference, we first present in Table 6.1 the number of iterations needed for convergence (and in brackets the condition number of the preconditioned system) when Ω does not contain any holes, i.e., $\Omega = (0,1)^2$, cf. [6], and $p = q = 1$. Here, and throughout this article, we employ the conjugate gradient iterative solver, with an absolute tolerance of 10^{-6}, as the termination condition. As reported in [6], for example, the number of iterations required to achieve convergence remains roughly constant when the ratio of the coarse and fine mesh sizes is kept fixed. The condition number of the preconditioned system has been numerically evaluated exploiting the connections between the conjugate gradient and the Lanczos algorithms. Indeed, at each PCG iteration we can build a tridiagonal matrix with the property that its extremal eigenvalues are good estimates of P_{ad}'s extremal eigenvalues. Notice that no additional work is involved, since all the entries of the tridiagonal matrix are readily available during the PCG iteration, cf. [22, Sec. 10.2.5] for more details. Let us now consider the case when Ω contains a number of square holes. To this end, in Tables 6.2–6.5 we show the number of iterations required for convergence (and the condition number of the preconditioned system) when $p = q = 1$. As in the case when no holes are present in Ω we observe that the proposed preconditioner is robust as the number of holes increases, in the sense that the number of iterations required to achieve convergence remains roughly constant when the ratio of the coarse and fine mesh sizes is kept fixed. As an observational comment, we note that, as the number of holes increases, the number of iterations required to achieve convergence for a given coarse and fine mesh granularity H and h, respectively, decreases. For example, in the case when $h = 1/128$, $H = 1/2$, 192 iterations are required to achieve convergence and condition numbers when $\Omega = (0,1)^2$, while only 192, 143, 106, and 79 iterations are required when there are 4, 16, 64, and 256 holes present, respectively.

The analogous results are presented in Tables 6.6–6.10 for the case when $p = q = 2$; Table 6.6 shows the case when $\Omega = (0,1)^2$ (no holes), while Tables 6.7 –6.10 present the number of iterations required to achieve convergence in the case when Ω contains 4, 16, 64, and 256 holes, respectively. As for the case when discontinuous piecewise linear polynomials are employed, we again observe similar behaviour of the iteration counts: namely, that they remain roughly constant when the ratio H/h is kept fixed, while they decrease as the number of holes increases. We point out that while we have fixed $q = p$, setting $0 \leq q < p$ leads to the same behaviour in terms of the iteration counts; however, as noted in [6], cf. below also, a richer coarse space

h^{-1} \ H^{-1}	2	4	8	16	32	64
8	**60 (53.0)**	**41 (36.7)**	-	-	-	-
16	**98 (79.9)**	**64 (92.6)**	43 (38.9)	-	-	-
32	**156 (151.8)**	**92 (158.9)**	65 (98.5)	43 (36.1)	-	-
64	**227 (283.7)**	**145 (304.9)**	95 (189.4)	65 (89.3)	43 (33.0)	-
128	**317 (530.6)**	**246 (533.3)**	140 (382.9)	95 (188.3)	65 (89.3)	43 (32.8)

TABLE 6.7

Example 1: Iteration counts and condition numbers for the preconditioner when Ω contains 4 holes, with $p = q = 2$.

h^{-1} \ H^{-1}	2	4	8	16	32	64
16	**87 (100.0)**	**65 (63.6)**	**42 (35.0)**	-	-	-
32	**125 (196.0)**	**115 (88.3)**	**66 (89.2)**	43 (34.8)	-	-
64	**179 (366.4)**	**180 (147.7)**	**95 (139.4)**	65 (89.1)	43 (32.9)	-
128	**253 (681.4)**	**261 (243.6)**	**147 (263.9)**	95 (163.6)	65 (89.2)	43 (32.7)

TABLE 6.8

Example 1: Iteration counts and condition numbers for the preconditioner when Ω contains 16 holes, with $p = q = 2$.

h^{-1} \ H^{-1}	2	4	8	16	32	64
32	**99 (149.4)**	**97 (91.2)**	**69 (55.0)**	**44 (34.8)**	-	-
64	**137 (292.5)**	**137 (178.2)**	**117 (78.9)**	**66 (59.8)**	43 (31.1)	-
128	**193 (562.6)**	**193 (347.0)**	**183 (113.6)**	**96 (106.5)**	65 (64.8)	43 (31.0)

TABLE 6.9

Example 1: Iteration counts and condition numbers for the preconditioner when Ω contains 64 holes, with $p = q = 2$.

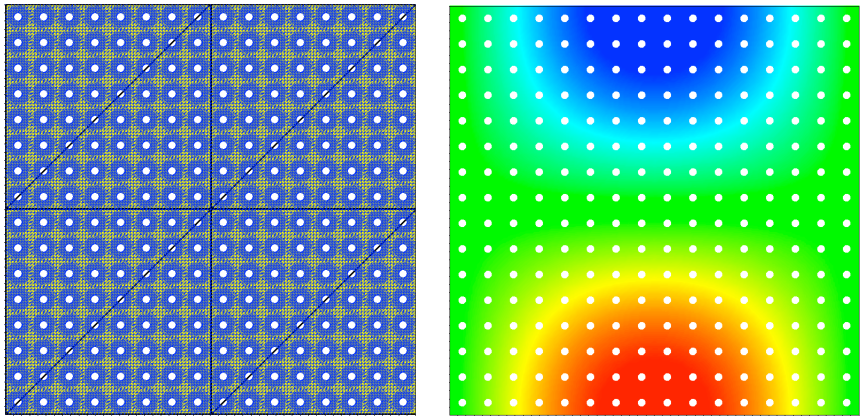
h^{-1} \ H^{-1}	2	4	8	16	32	64
64	**102 (181.0)**	**100 (148.4)**	**97 (89.4)**	**69 (37.0)**	**44 (29.1)**	-
128	**140 (359.0)**	**140 (290.5)**	**141 (176.7)**	**117 (66.9)**	**66 (33.3)**	43 (30.7)

TABLE 6.10

Example 1: Iteration counts and condition numbers for the preconditioner when Ω contains 256 holes, with $p = q = 2$.

$V(\mathcal{T}_H, q)$ typically leads to a reduced number of iterations. Finally, we point out that analogous behaviour of the preconditioner is also observed in the representative case when homogeneous Dirichlet boundary conditions are imposed on the boundary of Ω, with $f = 1$. Indeed, in this case we observe that the number of CG steps required to attain convergence for a given set of mesh parameters is roughly half of the corresponding iterations reported here; for brevity, these results are omitted.

6.2. Example 2: Two–dimensional domain with micro-structures II. In this second example, we consider the case when the computational domain Ω contains a large number of uniformly spaced circular holes; here, we consider the case where 256



(a) (b)

FIG. 6.2. *Example 2: (a) Initial composite finite element mesh. The colour blue denotes elements present in the fine level mesh (which consists of 85500 triangular elements); elements plotted in black form the coarse level mesh (containing 8 elements); finally, the domain Ω is shown in yellow. (b) Analytical solution.*

H^{-1}	$p=1,\ q=1$	$p=2,\ q=2$	$p=2,\ q=1$
2	115 (264.5)	198 (34.5)	198 (587.3)
4	115 (245.7)	201 (22.8)	201 (647.3)
8	115 (203.7)	199 (17.5)	201 (231.1)
16	111 (109.7)	164 (376.9)	195 (79.9)
32	75 (55.6)	77 (93.1)	120 (62.8)

TABLE 6.11

Example 2: Iteration counts and condition numbers for the preconditioner when Ω contains 256 circular holes.

	$p=1$	$p=2$
Iterations	931 (11793)	2283 (65799)

TABLE 6.12

Example 2: Iteration counts and condition numbers for (unpreconditioned) conjugate gradient with 256 circular holes.

small circular holes are removed from $(0,1)^2$, see Figure 6.2(a). In this example, we select the right-hand side forcing function f and appropriate inhomogeneous boundary condition $u=g$ on $\partial\Omega$, so that the analytical solution is given by $u=\sin(\pi x)\cos(\pi y)$, cf. Figure 6.2(b).

In this section, we fix the fine level mesh $\mathcal{T}_h$; here, $\mathcal{T}_h$ consists of 85500 triangular elements (blue elements depicted in Figure 6.2(a)). The mesh $\mathcal{T}_h$ has been selected to give a very accurate representation of the computational domain Ω. As in the previous example, we select the number of subpartition domains equal to $N=4$, and the conjugate gradient iterative solver (absolute) tolerance equal to 10^{-6}. To demonstrate the flexibility of employing the DGCFEM as the coarse level solver, we consider extremely coarse meshes $\mathcal{T}_H$, whose meshsize is far too large to accurately describe Ω using standard element shapes. Indeed, in Table 6.11 we present the iteration counts and condition numbers when $H=1/2$, $1/4$, $1/8$, $1/16$ and $1/32$. To

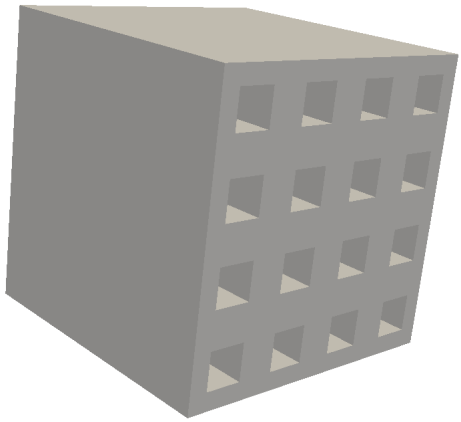


FIG. 6.3. *Example 3: Computational domain* Ω.

give an indication of the shape of the composite elements employed, in Figure 6.2(a) we plot (depicted in black) the coarsest mesh $\mathcal{T}_H$ when $H = 1/2$; here, a coarse element is formed by removing a series of holes from a coarse triangle. From Table 6.11, we observe that, initially, as the coarse mesh is refined, the number of iterations required to achieve convergence remains roughly constant. Indeed, the number of iterations only starts to decrease once the coarse finite element space $V(\mathcal{T}_H, q)$ is able to represent, at least on a coarse level, the geometric details present in Ω. This behaviour is observed for the three cases when $p = q = 1$, $p = q = 2$, and $p = 2$, $q = 1$; as indicated in the previous example, the latter case leads to an increase in the number of iterations when compared to the middle one. That said, even the finest coarse mesh $\mathcal{T}_H$, when $H = 1/32$ is still relatively coarse, when compared to the granularity of the fine mesh $\mathcal{T}_h$. For reference, in Table 6.12, we show the number of iterations required for convergence of the conjugate gradient solver when no preconditioning is employed, together with the condition number of the underlying matrix. When comparing these iteration counts with those presented in Table 6.11, we clearly observe that, even using a very inexpensive coarse solver, leads to a significant improvement when employing the proposed preconditioner.

6.3. Example 3: Three–dimensional domain with micro-structures. In this example, we consider a three–dimensional problem which (possibly) contains a number of holes. More precisely, we let Ω be the unit cube $(0,1)^3$ which has had a series of rectangular sections removed. As an illustration, in Figure 6.3 we show the case when 16 rectangular holes have been introduced; cf. [10]. We point out that the holes only go to the depth of half of the domain width. We select the right-hand side forcing function f and appropriate inhomogeneous boundary condition $u = g$ on $\partial\Omega$, so that the analytical solution to (2.1)–(2.2) is given by $u = \sin(\pi x)\cos(\pi y)\sin(\pi z)$.

As in the previous examples, the fine mesh $\mathcal{T}_h$ is always selected in such a manner that the underlying computational domain is precisely described using tetrahedral elements. Given the simple nature of the geometry, we first construct a uniform hexahedral mesh and subdivide each element into 6 tetrahedra. We select the number of subdomain partitions equal to $N = 8$, and the absolute tolerance for the conjugate gradient solver equal to 10^{-6}. In Tables 6.13, 6.14, & 6.15, we first consider the case when $\Omega = (0,1)^3$ (no holes) with $p = q = 1$, $p = q = 2$, and $p = 2$, $q = 1$,

h^{-1} \ H^{-1}	2	4	8	16
4	38 (32.1)	-	-	-
8	71 (87.1)	44 (35.6)	-	-
16	97 (203.0)	79 (91.4)	46 (35.1)	-
32	143 (429.7)	137 (207.4)	85 (91.4)	45 (35.2)

TABLE 6.13

Example 3: Iteration counts and condition numbers for the preconditioner when $\Omega = (0,1)^3$, i.e., when Ω does not contain any holes no holes in 3D, with $p = q = 1$.

h^{-1} \ H^{-1}	2	4	8	16
4	67	-	-	-
8	110	67	-	-
16	186	115	68	-
32	275	189	114	67

TABLE 6.14

Example 3: Iteration counts for the preconditioner when $\Omega = (0,1)^3$, i.e., when Ω does not contain any holes no holes in 3D, with $p = q = 2$.

h^{-1} \ H^{-1}	2	4	8	16
4	95	-	-	-
8	139	104	-	-
16	202	162	106	-
32	260	267	161	102

TABLE 6.15

Example 3: Iteration counts for the preconditioner when $\Omega = (0,1)^3$, i.e., when Ω does not contain any holes no holes in 3D, with $p = 2$, $q = 1$.

h^{-1} \ H^{-1}	2	4	8	16
8	**73 (76.8)**	**45 (30.7)**	-	-
16	**101 (183.9)**	**87 (81.6)**	47 (34.2)	-
32	**133 (417.2)**	**147 (195.5)**	87 (89.6)	46 (35.1)

TABLE 6.16

Example 3: Iteration counts and condition numbers for the preconditioner when Ω contains 4 holes, with $p = q = 1$.

h^{-1} \ H^{-1}	2	4	8	16
8	**135**	**69**	-	-
16	**210**	**118**	68	-
32	**300**	**216**	116	68

TABLE 6.17

Example 3: Iteration counts for the preconditioner when Ω contains 4 holes, with $p = q = 2$.

h^{-1} \ H^{-1}	2	4	8	16
8	**149**	**100**	-	-
16	**206**	**173**	107	-
32	**245**	**265**	166	106

TABLE 6.18

Example 3: Iteration counts for the preconditioner when Ω contains 4 holes, with $p = 2$, $q = 1$.

h^{-1} \ H^{-1}	2	4	8	16
16	**87 (175.7)**	**79 (69.0)**	**45 (29.8)**	-
32	**117 (393.8)**	**138 (168.1)**	**92 (80.2)**	46 (34.5)

TABLE 6.19

Example 3: Iteration counts and condition numbers for the preconditioner when Ω contains 16 holes, with $p = q = 1$.

h^{-1} \ H^{-1}	2	4	8	16
16	**185**	**135**	**70**	-
32	**276**	**226**	**118**	68

TABLE 6.20

Example 3: Iteration counts for the preconditioner when Ω contains 16 holes, with $p = q = 2$.

h^{-1} \ H^{-1}	2	4	8	16
16	**179**	**172**	**101**	-
32	**232**	**264**	**173**	106

TABLE 6.21

Example 3: Iteration counts for the preconditioner when Ω contains 16 holes, with $p = 2$, $q = 1$.

respectively. As noted in Example 1 above, the proposed preconditioner is robust, in the sense that the number of iterations required to achieve convergence remains roughly constant when the ratio of the coarse and fine mesh sizes is kept fixed. For the case when $p = 2$, we again observe that a richer coarse finite element space typically leads to an improvement in the number of iterations required to attain convergence.

We now consider the two cases when 4 and 16 holes have been introduced into the unit cube. Iteration counts for the same three cases considered above, namely $p = q = 1$, $p = q = 2$, and $p = 2$, $q = 1$ and condition numbers for the case $p = q = 1$, are presented in Tables 6.16–6.21. As before, cases highlighted in bold indicate when $\mathcal{T}_H$ is too coarse to precisely describe Ω using standard element shapes. As in the case when no holes are present in Ω we observe that the proposed preconditioner is robust as the number of holes increases, in the sense that the number of iterations required to achieve convergence remains roughly constant when the ratio of the coarse and fine mesh sizes is kept fixed. Moreover, we observe that, as the number of holes increases, the number of iterations required to achieve converge for a given coarse and fine mesh granularity H and h, respectively, decreases.

h^{-1} \ H^{-1}	2	4	8	16	32	64
4	17	-	-	-	-	-
8	20	20	-	-	-	-
16	22	23	20	-	-	-
32	25	26	24	19	-	-
64	28	29	29	24	18	-
128	31	30	32	29	23	18

TABLE 6.22

Example 4: GMRES iteration counts for the preconditioner when $\Omega = (0,1)^2$ (no holes) with $p = q = 1$.

6.4. Example 4: Convection-diffusion problem with micro-structures. In this final example, we consider the convection–diffusion problem

$$\begin{aligned} -\varepsilon\Delta u + \underline{a}(x)\cdot\nabla u &= f(x) \qquad \text{in } \Omega, \\ u &= g \qquad \text{on } \partial\Omega. \end{aligned} \tag{6.1}$$

Here, Ω is chosen in an analogous manner to the computational domains considered in Section 6.1. For the purposes of this section, we set $\underline{a} = (1,1)^\top$ and select f and g such that the analytical solution to (6.1) is given by

$$u(x,y) = x + y - xy + \frac{1}{1-\exp^{-1/\varepsilon}}\left[\exp^{-1/\varepsilon} - \exp^{-(1-x)(1-y)/\varepsilon}\right],$$

cf. [12]. We note that for $0 < \varepsilon \ll 1$, the solution exhibits boundary layers along $x = 1$ and $y = 1$. For brevity, we set $\varepsilon = 10^{-2}$; we shall comment on the effect of reducing the diffusion coefficient ε at the end of this section.

The discretization of (6.1) is based on employing the standard upwind DGFEM for the treatment of the advection operator, cf. [25], for example, together with the (symmetric) version of the IP DGFEM for numerical approximation of the diffusion terms, cf. Section 2. Clearly, the resulting matrix system is non-symmetric; thereby, we employ the GMRES method with an absolute termination tolerance of 10^{-6} with the additive Schwarz preconditioner. Results based on employing $N = 4$ subdomain partitions are presented in Tables 6.22, 6.23, 6.24, 6.25, & 6.26 for the cases when $\Omega = (0,1)^2$ (no holes), and when Ω contains 4, 16, 64, and 256 holes, respectively, for $p = q = 1$. Again, as in the previous examples, the number of iterations required to attain convergence remain roughly constant for a fixed value of the ratio of the coarse and fine mesh sizes H and h, respectively. Numerical experiments for $p = 2$ again indicate analogous behaviour; for brevity, these have been omitted.

7. Concluding remarks. In this article we have considered the application of Schwarz-type domain decomposition preconditioners for discontinuous Galerkin finite element approximations of elliptic partial differential equations posed on complicated domains, which are characterized by small details in the computational domain or microstructures. In particular, the coarse level solver employed within the proposed preconditioning strategy has been based on exploiting composite discontinuous Galerkin methods, which can easily handle general polygonal element domains consisting of agglomerated 'standard' elements. In this way, extremely coarse meshes may be defined, even for computational domains which contain small geometric details. The application of the preconditioner to a simple elliptic PDE, as well as a convection–dominated

h^{-1} \ H^{-1}	2	4	8	16	32	64
8	**20**	**20**	-	-	-	-
16	**22**	**23**	20	-	-	-
32	**25**	**26**	24	19	-	-
64	**28**	**29**	29	24	18	-
128	**31**	**30**	32	29	23	18

TABLE 6.23

Example 4: GMRES iteration counts and condition numbers for the preconditioner when Ω contains 4 holes and with $p = q = 1$.

h^{-1} \ H^{-1}	2	4	8	16	32	64
16	**33**	**31**	**25**	-	-	-
32	**34**	**34**	**30**	22	-	-
64	**42**	**42**	**39**	29	19	-
128	**52**	**52**	**49**	40	26	19

TABLE 6.24

Example 4: GMRES iteration counts and condition numbers for the preconditioner when Ω contains 16 holes, with $p = q = 1$.

h^{-1} \ H^{-1}	2	4	8	16	32	64
32	**37**	**37**	**34**	**24**	-	-
64	**42**	**42**	**41**	**33**	20	-
128	**51**	**51**	**51**	**43**	28	19

TABLE 6.25

Example 4: GMRES iteration counts and condition numbers for the preconditioner when Ω contains 64 holes, with $p = q = 1$.

h^{-1} \ H^{-1}	2	4	8	16	32	64
64	**35**	**35**	**35**	**32**	**21**	-
128	**44**	**44**	**44**	**43**	**31**	20

TABLE 6.26

Example 4: GMRES iteration counts and condition numbers for the preconditioner when Ω contains 256 holes, with $p = q = 1$.

diffusion problem clearly indicate the efficiency and robustness of the proposed solution strategy. Future work will consider the application of these techniques to more complicated problems; in particular, a key application area of interest is compressible fluid flows.

Acknowledgements. SG and PH acknowledge the financial support of the EPSRC under the grant EP/H005498.

REFERENCES

[1] P. R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Anal. Appl.*, 23(1):15–41, 2001.

[2] P. R. Amestoy, I. S. Duff, and J.-Y. L'Excellent. Multifrontal parallel distributed symmetricand unsymmetric solvers. *Comput. Methods Appl. Mech. Eng.*, 184:501–520, 2000.

[3] P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent, and S. Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32(2):136–156, 2006.

[4] P. F. Antonietti and B. Ayuso. Schwarz domain decomposition preconditioners for discontinuous Galerkin approximations of elliptic problems: non-overlapping case. *M2AN Math. Model. Numer. Anal.*, 41(1):21–54, 2007.

[5] P. F. Antonietti and B. Ayuso. Multiplicative Schwarz methods for discontinuous Galerkin approximations of elliptic problems. *M2AN Math. Model. Numer. Anal.*, 42(3):443–469, 2008.

[6] P. F. Antonietti and P. Houston. A class of domain decomposition preconditioners for *hp*-discontinuous Galerkin finite element methods. *J. Sci. Comp.*, 46(1):124–149, 2011.

[7] P. F. Antonietti and P. Houston. Preconditioning high-order discontinuous Galerkin discretizations of elliptic problems. In R. Bank, M. Holst, O. Widlund, and J. Xu, editors, *Domain Decomposition Methods in Science and Engineering XX. Lecture Notes in Computational Science and Engineering, Vol. 91*. Springer–Verlag, 2013.

[8] P.F. Antonietti and B. Ayuso. Two-level schwarz preconditioners for super penalty discontinuous Galerkin methods. *Commun. Comput. Phys.*, 5(2-4):398–412, 2009.

[9] P.F. Antonietti, B. Ayuso De Dios, S.C. Brenner, and L.-Y. Sung. Schwarz methods for a preconditioned WOPSIP method for elliptic problems. *Comp. Meth. Appl. Math.*, 12(3):241–272, 2012.

[10] P.F. Antonietti, S. Giani, and P. Houston. *hp*–Version composite discontinuous Galerkin methods for elliptic problems on complicated domains. *SIAM J. Sci. Comput.*, 35(3):A1417–A1439, 2013.

[11] P.F. Antonietti and P. Houston. Preconditioning high-order discontinuous galerkin discretizations of elliptic problems. *Lecture Notes in Computational Science and Engineering*, 91:231–238, 2013.

[12] P.F. Antonietti and E. Süli. Domain decomposition preconditioning for discontinuous Galerkin approximation of convection-diffusion problems. In M. Bercovier, M.J. Gander, R. Kornhuber, and O. Widlund, editors, *Proceedings of the 18th Domain Decomposition Conference. Lecture Notes in Computational Science and Engineering*, pages 259–266. Springer–Verlag, 2009.

[13] D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.*, 19(4):742–760, 1982.

[14] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39:1749–1779, 2001.

[15] A. T. Barker, S. C. Brenner, E.-H. Park, and Li-Y. Sung. Two-level additive Schwarz preconditioners for a weakly over-penalized symmetric interior penalty method. *J. Sci. Comp.*, 47:27–49, 2011.

[16] F. Bassi, L. Botti, A. Colombo, D.A. Di Pietro, and P. Tesini. On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations. *J. Comput. Phys.*, 231(1):45–65, 2012.

[17] S. C. Brenner. Poincaré-Friedrichs inequalities for piecewise H^1 functions. *SIAM J. Numer. Anal.*, 41(1):306–324, 2003.

[18] S. C. Brenner and K. Wang. Two-level additive Schwarz preconditioners for C^0 interior penalty methods. *Numer. Math.*, 102(2):231–255, 2005.

[19] S.C. Brenner. An additive analysis of multiplicative Schwarz methods. *Numer. Math.*, 123(1):1–19, 2013.

[20] A. Cangiani, E.H. Georgoulis, and P. Houston. *hp*-Version discontinuous Galerkin methods on polygonal and polyhedral meshes. *Submitted for publication*, 2013.

[21] X. Feng and O. A. Karakashian. Two-level additive Schwarz methods for a discontinuous Galerkin approximation of second order elliptic problems. *SIAM J. Numer. Anal.*, 39(4):1343–1365 (electronic), 2001.

[22] Gene H. Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[23] W. Hackbusch and S.A. Sauter. Composite finite elements for problems containing small geometric details. Part II: Implementation and numerical results. *Comput. Visual Sci.*, 1:15–25, 1997.

[24] W. Hackbusch and S.A. Sauter. Composite finite elements for the approximation of PDEs on domains with complicated micro-structures. *Numer. Math.*, 75:447–472, 1997.

[25] P. Houston, C. Schwab, and E. Süli. Discontinuous *hp*-finite element methods for advection–diffusion–reactio n problems. *SIAM J. Numer. Anal.*, 39:2133–2163, 2002.

[26] M. Rech, S. Sauter, and A. Smolianski. Two-scale composite finite element method for the dirichlet problem on complicated domains. *Numer. Math.*, 102(4):681–708, 2006.

[27] A. Toselli and O. Widlund. *Domain decomposition methods—algorithms and theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2005.