EFFICIENT ITERATIVE SOLVERS AND FAST INTEGRATION FOR HIGH-ORDER DISCONTINUOUS GALERKIN METHODS ON POLYTOPIC GRIDS

Doctoral Dissertation of

GIORGIO PENNESI

POLITECNICO DI MILANO
Department of Mathematics

Advisor:
Prof.ssa Paola Francesca Antonietti

Co-advisor:
Prof. Paul Houston

Chair of the Doctoral Programme:
Prof.ssa Irene Sabadini

Doctoral programme in
MATHEMATICAL MODELS AND METHODS IN ENGINEERING

2018 - CYCLE XXXI

Abstract

This thesis addresses the design, analysis and implementation of efficient solvers for the solution of the linear system of equations stemming from discontinuous Galerkin discretizations of second-order elliptic partial differential equations on polytopic meshes. In particular, we analyze the convergence properties of geometric V-cycle multigrid algorithms where the sequence of spaces which form the basis of the multigrid scheme are possibly non-nested and are obtained based on employing agglomeration algorithms with possible edge/face coarsening. We prove that the method converges uniformly with respect to the granularity of the grid and the polynomial approximation degree p, provided that the minimum number of smoothing steps, which depends on p, is chosen sufficiently large.

In order to improve the V-cycle solver we design and analyze a class of two-level non-overlapping Additive Schwarz preconditioners which are employed as a smoothing operator for the multigrid algorithm. The preconditioner is based on a coarse space which can possibly be chosen to be non-embedded with respect to the finer space. We investigate the dependence of the condition number of the preconditioned system with respect to the diffusion coefficient and the discretization parameters, i.e., the mesh size and the polynomial degree of the fine and coarse spaces. Several numerical tests confirm the theoretical bounds as well as demonstrating a considerable improvement of the iterative V-cycle algorithm in terms of the number of iterations needed to reduce the residual below a given tolerance.

We also investigate the implementation aspects of the proposed methods by presenting efficient quadrature rules for the numerical approximation of integrals of polynomial functions over general polygonal/polyhedral elements that do not require an explicit construction of a sub-tessellation into triangular/tetrahedral elements. The proposed "Quadrature free" method is based on recursive applications of Stokes' theorem on homogeneous functions; thereby, the underlying integral may be evaluated using only the values of the integrand and its derivatives at the vertices of the polytopic domain, and hence leads to an exact cubature rule whose quadrature points are the vertices of the polytope. We demonstrate the capabilities of the proposed approach by efficiently computing the stiffness and mass matrices arising from discontinuous Galerkin discretizations of second-order elliptic partial differential equations in both two- and three-dimensions.

TABLE OF CONTENTS

List of Tables

List of Figures

Introduction

In recent years the exploitation of computational meshes composed of general polygonal and polyhedral elements has become very popular in the field of numerical methods for partial differential equations. Indeed, the flexibility offered by polygonal/polyhedral elements allows for the design of efficient computational grids when the underlying problem is characterized by a strong complexity of the physical domain, such as, for example, in geophysical applications, fluid-structure interaction, or crack propagation problems. Indeed, for those types of applications, the construction of a standard grid with triangular/tetrahedral or quadrilateral/hexahedral elements represent one of the main issues in terms of computational costs for traditional approaches, whenever the underlying grid has to be aligned with the fractures/faults or the interfaces of different materials. However, if general polytopic elements can be employed, the process of mesh generation can be performed, for example, by first defining a possibly structured grid generated independently from the complexity of the domain; subsequently, the elements are cut according to the geometry of interest, i.e., a fracture network or interfaces between different materials. Moreover, the possibility to adopt computational meshes with hanging nodes is included in this framework by observing that, for example, a classical quadrilateral element with a hanging node on one of its edges can be treated as a pentagon with two aligned edges. Several conforming numerical discretization methods supporting polygonal/polyhedral meshes have been proposed within the current literature; we mention for example the Composite Finite Element Method [HS97b, HS97a, AGH13], the Mimetic Finite Difference (MFD) method [HSS97, BLS05b, BLS05a, BLS06, BLM14, AFS+16], the Polygonal Finite Element Method [ST04], the Extended Finite Element Method [TS08, FB10], the Virtual Element Method (VEM) [BBC+13, BBMR16a, BBMR16b, AdVMV14, AdVSV16] and the Hybrid High-Order (HHO) method [DEL14, DE15a, DE15b, DEL16]. The main issue in designing polygonal methods is represented by the definition of a suitable discrete polynomial space; in this context this is far from being a trivial task particularly for high-order approximations. A wide strand of literature has also focused on the non-conforming setting, where the possibility of defining local polynomial discrete spaces follows naturally with the flexibility provided by polytopic meshes. Here, we mention, for example, Hybridizable Discontinuous Galerkin methods [CDG08, CGL09, CGS10, CGW09], non-conforming VEM [AMV17, ALM16, CMS17] and the Gradient Schemes [DEH16]; Discontinuous Galerkin (DG) methods also represents a class of powerful non-conforming numerical schemes in which the exploitation of numerical grids characterized by general polytopic elements couples

very well with the possibility to build the underlying discrete space in the physical frame, thereby avoiding traditional polynomial spaces mapped from a reference element. This class of methods is the one considered in this thesis. Introduced in 1973 by Reed and Hill for the discretization of hyperbolic problems ([RH73]), DG methods have subsequently been proposed to deal with elliptic and parabolic problems; some of the most relevant earlier works include Baker [Bak77], Wheeler [Whe78] and Arnold [Arn82], whose contributions initiated the development of interior penalty DG methods. Since then, the scientific and industrial community has shown a growing interest in DG methods; see, for example, [CKS00, HW08, Riv08, DE11] and the references cited therein for an overview. Indeed, in addition to the mentioned advantages, the features of DG methods have been naturally enhanced by the recent development of High Performance Computing technologies as well as the growing request for high-order accuracy. In particular, since the discrete polynomial space can be defined locally on each mesh element, DG methods feature a high-level of intrinsic parallelism. Moreover, the local conservation properties and the possibility to use meshes with hanging nodes make DG methods interesting also from a practical point of view. The first attempt to define a DG setting on polytopic grids has been presented in [ABM09], where a bubble stabilization has been proposed for the Baumann-Oden formulation on a polytopic decomposition of the computational domain. Later then, different approaches based on interior penalty stabilization have been proposed. We mention for example the work of Lipnikov and coworkers [LVY14], where Darcy and Stokes flows are studied through DG (and MFD) discretizations on polytopic grids; here, the finite element spaces are defined on polygonal and polyhedral grids by introducing lifting operators mapping mimetic degrees of freedom to functional spaces. In [BBC$^+$12, BBCR12, BBC14] the discrete space is built based on a set of orthonormal basis functions generated in the physical frame; here, the value of the interior penalty stabilization parameter depends on the maximum number of faces that a polytopic element might possess. Antonietti, Giani and Houston have proposed in [AGH13, AGH14] a DG discretization for elliptic problems posed on computational domains which may contain a huge number of local geometrical features or microstructures; the approach presented there is based on the hypothesis that the number of faces of each element is bounded from above and that the stabilization parameter is chosen on each face to be proportional to the inverse of the diameter of the face itself. This method has been successively generalized by Cangiani et al. in [CGH14], where an efficient approach has been presented based on defining a local polynomial discrete space by making use of the bounding box of each element [GH14]; this technique, together with a careful choice of the discontinuity penalization parameter, allows for polytopic elements that can be characterized by faces of arbitrarily small measure. In particular, the DG approach considered in this thesis is a further extension of the previous setting and has been presented in [CDGH16, CDGH17, AHH$^+$17]; see also [ACC$^+$16, AFRV16] and [AM18, ABM18] for applications of this method to elastodynamics and elasto-acoustic problems. This approach is characterized by a generalization of the standard shape-regularity property to polytopic domains

which allows the elements to possess degenerating faces, as well as having an unbounded number of faces.

However, as shown in [AH11], the condition number of the linear system of equations stemming from the DG method is possibly prohibitively large and affected by the size of the partition and the polynomial degree employed for the discretization. For this reason, the development of fast solvers and preconditioners for the linear systems of equations stemming from (high-order) DG discretizations has been an intensive research area in recent years. A recent strand of the literature has focused on Schwarz domain decomposition methods, see, for example, [TW04] for a general abstract overview of these methods. In the DG setting where standard triangular/tetrahedral or quadrilateral/hexahedral grid are employed, one of the first contribution in terms of domain decomposition solvers has been presented for the solution of elliptic problems by Feng and Karakashian in [FK01, FK02], where bounds of order $\mathcal{O}(H/\delta)$ and $\mathcal{O}(H/h)$ have been obtained for the condition number of the preconditioned system in the framework of overlapping and non-overlapping Schwarz methods, respectively; here, H, h and δ represent the size of the coarse grid, the fine grid and the amount of overlap, respectively. Its extension to fourth order problems can be found in [FK05]. Similar results within the overlapping framework have been presented by Lasser and Toselli in [LT03] for advection-diffusion problems. In [AS09] Antonietti and Süli have discussed the acceleration properties of the Schwarz preconditioners within the *generalized minimal residual* (GMRES) Krylov space-based iterative solver for convection-diffusion equations. In particular, they have demonstrated through numerical computations that the classical Schwarz convergence theory cannot be applied to prove the convergence observed numerically. Further developments of this class of methods in the field of elliptic problems can be found in [AA07, AA08] where two-level methods with inexact local solvers for both symmetric and non-symmetric schemes are introduced and presented in a unified framework. In particular, for symmetric schemes it is shown that the condition number of the preconditioned system is of order $\mathcal{O}(H/h)$. Dryja and Sarkis proposed in [DS10] an additive Schwarz preconditioner for the solution of second order elliptic problem with highly discontinuous coefficients. Here, they have established that the condition number of the preconditioned system is independent of the jumps of the coefficients across the substructure boundaries and outside of a thin layer along the substructure boundaries. Further development of this algorithm, which is very well suited for parallel computations, can be found in [DKS14, DK16]. For low order DG schemes, the same authors designed and analyzed in [DGS07] a Balancing Domain Decomposition with Constraints (BDDC) algorithm for the same class of problems. The condition number of the resulting preconditioned system is shown to be of order $\mathcal{O}([1+\log(H/h)]^2)$. Concerning the high order DG setting, we mention the work of Antonietti and Houston [AH11], where additive and multiplicative Schwarz preconditioners have been introduced for efficiently solve the algebraic linear system of equations arising from the discretization of a symmetric, elliptic boundary value problem using hp–version DG methods. Optimal hp-spectral bounds of order $\mathcal{O}(\sigma p^2 H/h)$ have been derived in [AHS16] for a

class of domain decomposition preconditioners for DG discretizations, where σ is the coefficient of the interior penalty stabilization parameter, p the polynomial approximation degree and H, h the size of the coarse and fine meshes, respectively. We also mention the recent work presented by Karakashian and Collins in [KC17], where they have investigated the influence of the penalty terms, as well as the choice of coarse mesh spaces on the condition number of the linear system of equations preconditioned with additive Schwarz methods. Besides Schwarz preconditioners, a variety of two-level and multigrid/multilevel techniques have also been proposed both in the Geometric and Algebraic setting for the solution of DG discretizations. In particular, the availability of efficient geometric multilevel solvers is strongly related to the possibility of employing general shaped polytopic grids, as will be clarified later. One of the first works in this direction is the geometric V-cycle multigrid solver introduced in [GK03]. Here, the multilevel solver is employed to define a preconditioner which is uniformly convergent with respect to the granularity of the underlying partition and the number of levels employed in the algorithm. In the framework of space decomposition and subspace correction methods, cf. [Xu92] for a general abstract overview, we mention [DLVZ06] where preconditioning methods for DG approximations have been proposed for the solution of second order problems; in this approach the DG method is decomposed into a subspace plus a correction which can be handled by a standard smoothing procedure. In particular, they have considered three different auxiliary subspaces, namely, piecewise linear $\mathscr{C}^0$-conforming functions, Crouzeix-Raviart finite elements and piecewise constant functions defined over the finite element mesh. Several multigrid solvers have also been proposed by Brenner and collaborators. Starting from the theoretical framework developed in [Bre99] for non-conforming finite element methods, its extension to DG discretizations for second order problems have been introduced in [BZ05, BO07, BCS09, BCGS11]. There, the proposed multigrid iterative solvers are proved to converge uniformly with respect to the number of levels and the granularity of the underlying partition, provided the number of smoothing steps is chosen sufficiently large. With regards the multigrid setting designed for high-order DG discretization, a class of W-cycle algorithm has been recently introduced in [ASV15] for the solution of hp-DG methods for second-order elliptic problems. Starting from the classical framework in geometric multigrid analysis, based on employing a suitably defined smoother, together with an approximation property, the uniform convergence of the W-cycle scheme is established with respect to the discretization parameters and the number of levels, provided the number of smoothing steps is chosen of order p^2, where p is the polynomial degree of the approximation, see also [ASVZ17]. The efficiency of those methods has been further improved in [AHH$^+$17] for polygonal grids, because the flexibility of the element shape couples very well with the possibility of defining agglomerated meshes, which is the key ingredient for the development of multigrid algorithms. In [AHH$^+$17] a two-level scheme and W-cycle multigrid methods for high-order DG discretizations on polytopic grids have been analyzed. One iteration of the proposed methods consists of an iterative application of the smoothing Richardson operator and a recursive subspace correction step. In particular,

the latter is based on a nested sequence of discontinuous discrete polynomial spaces, where the underlying polytopic grids are defined by agglomeration. While being perfectly suited for multilevel schemes, the process of element agglomeration might feature itself some limitations. Indeed, agglomeration leads to coarser grids with an increasing number of faces and this might affect the overall efficiency of the solvers.

In this thesis, we aim at overcoming this issue by introducing and analyzing multigrid methods and domain decomposition solvers based on the possibility of employing non-nested agglomerated coarser meshes. In this way we assure, for example, that the number of faces of the agglomerates does not blows up as the number of levels of our multigrid method increases. With regards to the multilevel framework, the flexibility in the choice of the computational sub-grids leads to the definition of a non-nested multigrid method characterized by a sequence of non-nested multilevel discrete spaces and where the discrete bilinear forms are chosen differently on each level, cf. [BV90, Zha90, ZZ97]. The first non-nested multilevel method was introduced by Bank and Dupont in [BD81]; a generalized framework was developed by Bramble, Pasciak and Xu in [BPX91], and then widely used in the analysis of non-nested multigrid iterations. We mention for example the work of Scott and Zhang [SZ92], where non-nested multigrid methods are shown to be optimal-order solvers for systems of finite element methods arising from elliptic boundary problems. In [BP92] Bramble and Pasciak provided a general technique for defining and analyzing smoothing operators based on subspace decomposition to be used in multigrid non-nested algorithms. The same authors provided in [BP93] new estimates for multilevel algorithms based on non-nested grids that are carried out by designing a V-cycle solver characterized by a uniform rate of reduction per iteration independent of the mesh sizes and number of levels, even on non-convex domains which do not provide full elliptic regularity. They have shown the effectiveness of this approach for problems on domains with curved boundaries. Further extension of this work with different smoothers can be found in [BKP94]. An example of non-nested multigrid methods for fourth order elliptic boundary value problems which do not have full elliptic regularity has been proposed by Brenner in [Bre99]. Golapalakrishnan and Pasciak proposed in [GP00] a multigrid method for uniformly preconditioning linear systems arising from a mortar finite element discretization of second order elliptic boundary value problems. There, the computational domains is firstly partitioned into subdomains, each of which is independently triangulated in a multilevel fashion. Others non-nested multigrid methods for mortar element method have been proposed by Xu and collaborators in [XC01, XLC02] based on the general abstract Bramble Pasciak and Xu framework [BPX91]. The method of [BPX91], usually referred to as the BPX multigrid framework, is also able to generalize the multigrid framework that we will develop in this thesis, but the convergence analysis relies on the assumption that $\mathscr{A}_j(I^j_{j-1}u, I^j_{j-1}u) \leq \mathscr{A}_{j-1}(u,u)$, which might not be guaranteed in the DG setting, as we will see in Chapter 3. Here $\mathscr{A}_j(\cdot,\cdot)$ and $\mathscr{A}_{j-1}(\cdot,\cdot)$ are the bilinear forms on two consecutive levels j and $j-1$, respectively, and I^j_{j-1} is the prolongation operator whose definition in this setting is not trivial,

differently from the nested case. For this reason the convergence analysis will be presented based on employing the abstract setting proposed by Duan, Gao, Tan and Zhang in [DGTZ07], which permits the development of a full analysis of V-cycle multigrid methods in a non-nested framework and relaxing the hypothesis $\mathscr{A}_j(I^j_{j-1}u, I^j_{j-1}u) \leq \mathscr{A}_{j-1}(u,u)$. We also address the design and analysis of efficient space decomposition techniques by generalizing the framework presented in [AHS16], where a non-overlapping additive Schwarz preconditioner with sub-space correction step is analyzed for standard grids. Also in this case, we allow the coarser space to be possibly non-nested with respect to the finer one. In this setting, we show that the condition number of the preconditioned system can be bounded by $\mathscr{O}(\frac{\max\rho}{\min\rho}\frac{H}{h\mathbb{H}}\frac{p^{2+\eta}}{q})$, where ρ is the diffusion coefficient, H, $\mathbb{H}$ and h are the sizes of the coarse grid, the local solvers and the fine mesh, respectively, while p and q are the polynomial approximation degrees employed on the fine and the coarse spaces, respectively. Here, the parameter $\eta = 0,1$ if the coarse and the fine spaces are nested or non-nested, respectively. The non-nested coarse space leads to an increase in the condition number with respect to the nested case for high order computations, however, as we will show in Chapter 4, it guarantees that the number of faces of the coarse grid elements does not blow up.

An other key aspect concerning the development of efficient finite element discretizations on polygonal/polyhedral grids is the construction of quadrature formulae for the approximate computation of the terms appearing in the underlying weak formulation. Indeed, the design of efficient quadrature rules for the numerical computation of integrals over general shaped polytopes is far from being a trivial task. The classical and most widely employed technique for the integration over polytopes is the *Sub-Tessellation* method, cf. [MDB99, SMB00, GW08]; here, the domain of integration is subdivided into standard-shaped elements, such as triangular/quadrilateral elements in 2D or tetrahedral/hexahedral elements in 3D, whereby standard efficient quadrature rules are employed, cf. [SS67, MRW96, YR98], and also [YS16] and [LLD09], for an interpolation technique based on the same idea. On the one hand this technique is easy to implement, however, it is generally computationally expensive, particularly for high order polynomials, since the number of function evaluations may be very large. For this reason, the development of quadrature rules that avoid sub-tessellation is an active research field. Several approaches have been proposed; in particular, we mention [Ven06, HNS15, VB15, NBM09], for example. One interesting method in this direction is represented by the *Moment Fitting Equation* technique, firstly proposed by Lyness and Monegato in [LM77], for the construction of quadrature rules on polygons featuring the same symmetry as the regular hexagon. Generalizations to convex and non-convex polygons and polyhedra were proposed by Mousavi, Xiao and Sukumar in [MXS10]. Here, starting from an initial quadrature rule, given, for example, by the sub-tessellation method described above, an iterative node elimination algorithm is performed based on employing the least-squares Newton method [XG10] in order to minimise the number of quadrature points while retaining exact integration. Further improvements of the moment fitting equation algorithm can also be found in [MS11] and [SW13]. While this method is optimal with respect to the

number of function evaluations, the nodes and weights must be stored for every polygon, thus affecting memory efficiency. An alternative approach designed to overcome the limitations of the sub-tessellation approach is based on employing the generalized version of Stokes' theorem; here, the exploitation of Stokes' theorem reduces the integral over a polytope to an integration over its boundary; see [Tay96] for details. For the two–dimensional case, in [SV07], Sommariva and Vianello proposed a quadrature rule based on employing Green's theorem. In particular, if an x- or y-primitive of the integrand is available (as for bivariate polynomial functions), the integral over the polygon is reduced to a sum of line integrations over its edges. When the primitive is not known, this method does not directly require a sub-tessellation of the polygon, but a careful choice of the parameters in the proposed formula leads to a cubature rule that can be viewed as a particular sub-tessellation of the polygon itself. However, it is not possible to guarantee that all of the quadrature points lie inside the domain of integration. An alternative and very efficient formula has been proposed by Lasserre in [Las98] for the integration of *homogeneous functions* over convex polytopes. This technique has been recently extended to general convex and non-convex polytopes in [CLS15]. The essential idea here is to exploit the generalized Stokes' theorem together with Euler's homogeneous function theorem, cf. [SB96], in order to reduce the integration over a polytope only to boundary evaluations. The main difference with respect to the work presented in [SV07] is the possibility to apply the same idea recursively, leading to a quadrature formula which exactly evaluates integrals over a polygon/polyhedron by employing only point-evaluations of the integrand and its derivatives at the vertices of the polytope.

In this thesis we extend the approach of [CLS15] to the efficient computation of the volume/face integral terms appearing in the discrete weak formulation of second-order elliptic problems discretized by means of high-order DG methods. We point out that our approach is completely general and can be directly applied to other discretization schemes, such as VEM, HHO, Hybridizable DG, and MFD, for example. We show that our integration approach leads to a considerable improvement in the performance compared to classical quadrature algorithms based on sub-tessellation, in both two– and three–dimensions both from the computational time viewpoint and from FLOPs counting.

In the following we provide a brief description of the contents of each chapter of the thesis.

- In Chapter 1 we introduce a class of high-order Symmetric Interior Penalty DG methods on polytopic grids for the discretization of second-order elliptic partial differential equations. The method employed here has been developed by following the approach introduced in [CGH14]. The only difference here is related to the choice of the interior penalty discretization parameter, which in our case allows for high variations in the diffusion coefficient. We then recall the main theoretical results concerning this class of methods, i.e. trace and inverse inequalities as well as approximation properties if the underlying discrete space, the well-posedness of the discrete problem and hp-error bounds. The results

presented in this chapter form the basis for the theoretical analysis of the multigrid and domain decomposition solvers presented in Chapters 3 and 4, respectively.

- Chapter 2 is devoted to the development of new technique for the numerical computations of the integrals of polynomials over polygonal/polyhedral domains. We first describe in detail the method presented in [CLS15] for the exact integration of homogeneous functions over polytopic domains which is based on recursively employing Stokes' theorem. The advantages of employing this method with respect to classical quadrature approaches on polytopic grids are demonstrated by studying the computational costs and FLOPs counts of the different algorithms. We then extend this idea and propose a scheme to efficiently assemble the mass and stiffness matrices of the high-order DG method introduced in Chapter 1. The results of this chapter are original, and have been published in [AHP18b].

- In Chapter 3 we introduce and analyze V-cycle hp-multigrid methods for high order DG approximations of second order elliptic problems. Here, the sequence of spaces which represents the basis of the multigrid scheme are possibly non-nested and are obtained based on employing agglomeration algorithms with possible edge/face coarsening. We point out that our multigrid solver is a particular case of the more general abstract framework for multigrid methods on non-nested grids presented in [BPX91]. In particular, the key point of our algorithm is represented by the choice of the inter-grid transfer operators. The main theoretical result of this chapter is then the continuity of those inter-grid operators between non-nested discrete DG spaces. Exployting this property, we prove that our multigrid V-cycle algorithm converges uniformly with respect to the granularity of the grid h and the polynomial approximation degree p, provided that the number of smoothing steps, which depends on p, is chosen sufficiently large. We also point out that the classical nested framework is a particular case of our setting, so that the V-cycle multilevel solver presented here represents an extension of the W-cycle multigrid setting for nested polytopic grids presented in [AHH$^+$17]. Several numerical results are presented in order to validate the theoretical estimated. A further improvement of this algorithm is presented in a final section where we have developed a domain decomposition preconditioner which we have employed as a smoothing operator of the multigrid iteration. All the results presented in this chapter are original and have been published in [AP18].

- In Chapter 4 we design and analyzed a class of two-level non-overlapping additive Schwarz preconditioners for DG discretizations of second-order elliptic problems on polytopic grids. Given the DG discrete problem defined on a polytopic grid of granularity h, the preconditioner is designed by introducing two additional partitions employed to define the local solver operators and the coarse space correction. On the one hand, the partition employed to build the local solvers is related to a suitable space splitting of the DG space and so it is assumed to be nested with respect to the fine polytopic mesh of granularity h; On the

other hand no conditions are imposed on the coarse partition which can be non-nested with respect to the fine grid. We investigate the dependence of the condition number with respect to the discretization parameters, i.e., the diffusion coefficient and the size and the polynomial degree of the fine and coarse spaces. The preconditioner is then employed to accelerate the V-cycle multigrid solver developed in Chapter 3. The results presented in this Chapter are original and are contained in [AHP18a].

CHAPTER 1

THE HIGH-ORDER DISCONTINUOUS GALERKIN METHOD ON POLYTOPIC GRIDS

In this chapter, we introduce high-order Symmetric Interior Penalty Discontinuous Galerkin methods on polytopic grids for the discretization of second-order elliptic partial differential equations. The method employed here follows the efficient approach introduced in [CGH14], with suitable modifications to deal with strongly heterogeneous diffusion materials. For the sake of brevity, we use throughout the rest of this thesis the notation $x \lesssim y$ to mean $x \leq Cy$, where $C > 0$ is a constant independent from the discretization parameters. Similarly, we write $x \gtrsim y$ in lieu of $x \geq Cy$, while $x \eqsim y$ is used if both $x \lesssim y$ and $x \gtrsim y$ hold. This chapter is organized as follows. In Section 1.1 we introduce the model problem and its discretization based on employing a class of high order DG methods on grids characterized by polytopic shaped elements. In Section 1.2 we present the main theoretical aspects of this method which form the basis for the forthcoming analysis presented in Chapters 3 and 4.

1.1 Model problem and its DG discretization

Let $\Omega \subset \mathbb{R}^d$, $d = 2,3$ be a convex computational domain with Lipschitz boundary and let $f \in L^2(\Omega)$ be a given function. We consider the weak formulation of the following second-order elliptic problem, subject to homogeneous Dirichlet boundary conditions: find $u \in V = H_0^1(\Omega)$ such that

$$\mathscr{A}(u,v) = \int_\Omega \nabla u \cdot \nabla v \mathrm{d}x = \int_\Omega f v \mathrm{d}x \quad \forall\, v \in V. \tag{1.1}$$

Due to elliptic regularity, the unique solution $u \in V$ of problem (1.1) satisfies $u \in H^2(\Omega)$ and

$$\|u\|_{H^2(\Omega)} \lesssim \|f\|_{L^2(\Omega)}. \tag{1.2}$$

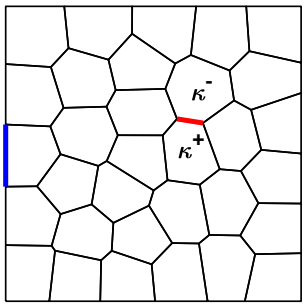


Figure 1.1: Example of polygonal mesh $\mathcal{T}_h$.

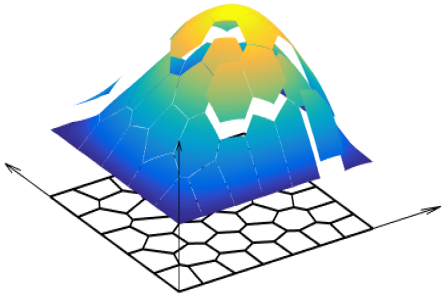


Figure 1.2: Example of $v_h \in V_h$ with $p = 2$.

In order to introduce the DG discretization of problem (1.1), we let $\mathcal{T}_h$ be a tessellation of Ω characterized by disjoint open polytopic elements κ of diameter h_κ, such that $\overline{\Omega} = \cup\overline{\kappa}$. We denote as $h = \max_{\kappa\in\mathcal{T}_h} h_\kappa$ the granularity of the grid $\mathcal{T}_h$. We then define the corresponding discontinuous finite element space V_h, defined as

$$V_h = \{v \in L^2(\Omega) : v|_\kappa \in \mathcal{P}_{p_\kappa}(\kappa), \kappa \in \mathcal{T}_h\}, \tag{1.3}$$

where $\mathcal{P}_{p_\kappa}(\kappa)$ denotes the space of polynomials of total degree at most $p_\kappa \geq 1$ for any $\kappa \in \mathcal{T}_h$, see Figure 1.2 for an example. The assumptions on the grid $\mathcal{T}_h$ are outlined in the following paragraph.

Remark 1.1. *We point out that the polynomial approximation degree p_κ can possibly be chosen differently on each element $\kappa \in \mathcal{T}_h$. In order to ease the analysis we will consider in this thesis $p_\kappa = p \ \forall \kappa \in \mathcal{T}_h$.*

1.1.1 Grid assumptions

We define the *faces* of the mesh $\mathcal{T}_h$ as the intersection of the $(d-1)$-dimensional facets of neighboring elements. This implies that, for $d = 2$, a *face* always consists of a line segment, whereas for $d = 3$, the *faces* of $\mathcal{T}_h$ are general shaped polygons. Thereby, for $d = 3$ we assume that each polygonal boundary facet of an element $\kappa \in \mathcal{T}_h$ can be subdivided into a set of co-planar $(d-1)$-dimensional simplices and we refer to them as *faces*. In order to introduce the DG formulation, it is helpful to distinguish between boundary and interior faces, denoted as $\mathcal{F}_h^B$ and $\mathcal{F}_h^I$, respectively. In particular, we observe that $\overline{F} \subset \partial\overline{\Omega}$ for $F \in \mathcal{F}_h^B$, while for any $F \in \mathcal{F}_h^I$, $\overline{F} \subset \partial\overline{\kappa}^\pm$, where $\kappa^\pm$ are two adjacent elements in $\mathcal{T}_h$, see Figure 1.1 for an example. Furthermore, we denote by $\mathcal{F}_h = \mathcal{F}_h^I \cup \mathcal{F}_h^B$ the set of all mesh faces of $\mathcal{T}_h$. With this notation, we assume that the sub-tessellation of element interfaces into $(d-1)$-dimensional simplices is given. Note that this decomposition is not needed in practice. Moreover, for the forthcoming analysis, we require that the following assumptions hold, cf. [CDG17, CDGH17].

Assumption 1.1. *Given $\kappa \in \mathcal{T}_h$ there exists a set of non-overlapping d-dimensional simplices $\kappa_\flat^F \subset \kappa$, for $F \subset \partial\kappa$, such that for any face $F \subset \partial\kappa$, we have that $\overline{F} = \partial\kappa \cap \partial\kappa_\flat^F$, $\bigcup_{F\subset\partial\kappa} \overline{\kappa_\flat^F} \subset \overline{\kappa}$, and the diameter h_κ of κ can be bounded by*

$$h_\kappa \leq C_1 \frac{d|\kappa_\flat^F|}{|F|} \quad \forall\, F \subset \partial\kappa,$$

where $|F|$ and $|\kappa_\flat^F|$ denote the Housdorff measure of F and $\kappa_\flat^F$, respectively, and where the constant $C_1 > 0$ is independent of the discretization parameters.

Assumption 1.1 allows us to employ very general polygonal and polyhedral elements which can possibly have an unbounded number of faces or faces of degenerating Housdorff measure. Indeed, if $\kappa \in \mathcal{T}_h$ is a polygonal/polyhedral element and $F \subset \partial\kappa$ is one of its faces, then Assumption 1.1 allows the size of F to be small compared to the diameter h_κ of κ, provided that the height of the related simplex T_l, with base F, is comparable to h_κ. Figure 1.3 shows two examples of polygonal elements that can be employed within our analysis and thereby satisfy Assumption 1.1. However, in Figure 1.4 we show an element which does not satisfy Assumption 1.1, since the shape of the polygon does not allow us to define a triangle $\kappa_\flat^F$ with base F whose height is comparable to h_κ. We refer to [CDG17] for more details. Moreover, Assumption 1.1 is needed in order to obtain the trace inequalities of Lemma 1.1 and Lemma 1.3 below.

Assumption 1.2. *For any $\kappa \in \mathcal{T}_h$, we assume that $h_\kappa^d \geq |\kappa| \gtrsim h_\kappa^d$, where $d = 2,3$ is the dimension of Ω.*

Assumption 1.3. *For every polytopic element $\kappa \in \mathcal{T}_h$, there exists a set of m_κ overlapping shape-regular simplices $\mathcal{K}_i$, $i = 1,\dots,m_\kappa$ such that*

$$dist(\kappa, \partial\mathcal{K}_i) \lesssim \frac{diam(\mathcal{K}_i)}{p^2}, \quad and \quad |\mathcal{K}_i| \gtrsim |\kappa|,$$

for all $i = 1,\dots,m_\kappa$.

Remark 1.2. *Assumption 1.2 and **A.3** are required for the inverse estimates of Lemma 4.5 and Theorem 1.1 below.*

Assumption 1.4. *We assume that there exists a covering $\mathcal{T}_h^\# = \{\mathcal{S}_\kappa\}_\kappa$ of $\mathcal{T}_h$ consisting of shape-regular d-dimensional simplices $\mathcal{S}_\kappa$, such that, for any $\kappa \in \mathcal{T}_h$, there exists $\mathcal{S}_\kappa \in \mathcal{T}_h^\#$ satisfying $\kappa \subset \mathcal{S}_\kappa$ and $h_{\mathcal{S}_\kappa} := diam(\mathcal{S}_\kappa) \lesssim h_\kappa$. We also assume that*

$$\max_{\kappa\in\mathcal{T}_h} card\{\kappa' \in \mathcal{T}_h : \kappa' \cap \mathcal{S}_\kappa \neq \emptyset, \mathcal{S}_\kappa \in \mathcal{T}_h^\# \text{ such that } \kappa \subset \mathcal{S}_\kappa\} \lesssim 1.$$

Remark 1.3. *Assumption 1.4 guarantees the validity of the approximation result and error estimates presented in Lemma 1.6 and Corollary 1.2, respectively.*

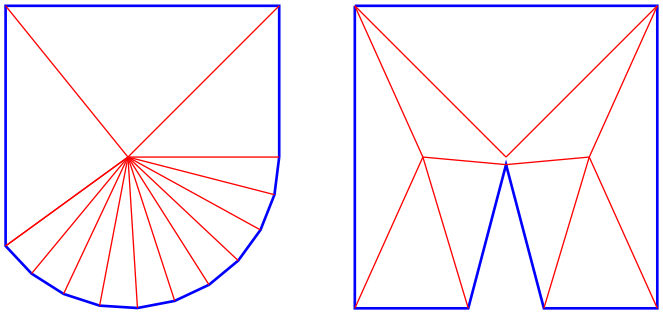


Figure 1.3: Two examples of polygonal elements that satisfy Assumption 1.1. In these cases all the triangles $\kappa_\flat^F$ (in red) defined as in Assumption 1.1 possess the height of size comparable to h_κ. Here $C_1 = \mathcal{O}(1)$.

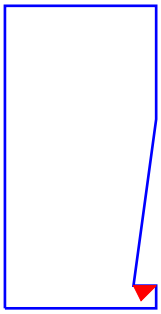


Figure 1.4: Example of an element that violates Assumption 1.1. Here $C_1 \gg 1$, $|F| \ll h_\kappa$ and $|\kappa_\flat^F| \ll \frac{|F|h_\kappa}{d}$.

1.1.2 DG formulation

In order to introduce the DG discretization of (1.1), we first need to define suitable jump and average operators across the faces $F \in \mathcal{F}_h$. We employ here the standard notation [ABCM02]. Let $\boldsymbol{\tau}$ and v be sufficiently smooth vector-valued and scalar functions, respectively. For each interior face $F \in \mathcal{F}_h^I$, such that F is shared by $\kappa^\pm \in \mathcal{T}_h$, let $\mathbf{n}^\pm$ denote the outward unit normal vector to $\partial\kappa^\pm$, respectively, and let $\boldsymbol{\tau}^\pm$ and $v^\pm$ be the traces of $\boldsymbol{\tau}$ and v on F from $\kappa^\pm$, respectively. The jump and weighted average operators across F are then defined as follows:

$$
\begin{aligned}
&[\![\boldsymbol{\tau}]\!] = \boldsymbol{\tau}^+ \cdot \mathbf{n}^+ + \boldsymbol{\tau}^- \cdot \mathbf{n}^-, && \{\{\boldsymbol{\tau}\}\}_\omega = \omega\boldsymbol{\tau}^+ + (1-\omega)\boldsymbol{\tau}^-, && F \in \mathcal{F}_h^I,\\
&[\![v]\!] = v^+\mathbf{n}^+ + v^-\mathbf{n}^-, && \{\{v\}\}_\omega = \omega v^+ + (1-\omega)v^-, && F \in \mathcal{F}_h^I.
\end{aligned} \tag{1.4}
$$

If $F \in \mathcal{F}_h^B$ is a boundary face, we set accordingly $\{\{\boldsymbol{\tau}\}\}_\omega = \boldsymbol{\tau}$, $[\![v]\!] = v\,\mathbf{n}$, cf. [ABCM02]. Here, $\omega \in [0,1]$ represents the weight employed for the definition of $\{\{\cdot\}\}_\omega$. Moreover, we write $\langle\cdot\rangle$ to denote the harmonic average operator defined as follows: let η be a sufficiently smooth function, then

$$
\langle\eta\rangle|_F = \begin{cases} \dfrac{2\eta^+\eta^-}{\eta^+ + \eta^-}, & F \in \mathcal{F}_h^I, \\ \eta, & F \in \mathcal{F}_h^B. \end{cases}
$$

With this notation, the bilinear form $\mathcal{A}_h(\cdot,\cdot) : V_h \times V_h \to \mathbb{R}$ corresponding to the symmetric interior penalty DG method is defined by

$$
\begin{aligned}
\mathcal{A}_h(u_h, v_h) = &\sum_{\kappa\in\mathcal{T}_h} \int_\kappa \Big[\nabla u_h \cdot \nabla v_h \; + \nabla u_h \cdot \mathcal{R}([\![v_h]\!]) + \nabla v_h \cdot \mathcal{R}([\![u_h]\!])\Big]\,\mathrm{d}\mathbf{x} \\
&+ \sum_{F\in\mathcal{F}_h} \int_F \sigma_h [\![u_h]\!] \cdot [\![v_h]\!]\,\mathrm{d}s,
\end{aligned} \tag{1.5}
$$

where the operator $\mathcal{R} : [L^1(\mathcal{F}_h)]^d \to [V_h]^d$ denotes the lifting operator defined as

$$
\int_\Omega \mathcal{R}(\mathbf{q}) \cdot \boldsymbol{\eta}\,\mathrm{d}\mathbf{x} = -\int_{\mathcal{F}_h} \mathbf{q} \cdot \{\{\boldsymbol{\eta}\}\}\,\mathrm{d}s \quad \forall\, \boldsymbol{\eta} \in [V_h]^d, \tag{1.6}
$$

cf. [ABCM02]. Here $\{\{\cdot\}\}$ denotes the weighted average operator with $\omega = \frac{1}{2}$ on each interior face $F \in \mathscr{F}_h^I$. In (1.5), according to [Dry03, DGS07], $\sigma_h \in L^\infty(\mathscr{F}_h)$ denotes the interior penalty stabilization function, which is defined by

$$\sigma_h|_F = C_\sigma \frac{p^2}{\langle h_\kappa \rangle}, \quad \forall\, F \in \mathscr{F}_h, \tag{1.7}$$

with $C_\sigma > 0$ independent of p, $|F|$, $|\kappa|$ and h_κ.

Then, based on the above definitions, the Symmetric Interior Penalty DG (SIPDG) discretization of (1.1) is given by:

$$\text{find } u_h \in V_h \text{ s.t. } \mathscr{A}_h(u_h, v_h) = \int_\Omega f v_h \, \mathrm{d}\mathbf{x} \quad \forall\, v_h \in V_h, \tag{1.8}$$

In the next section we recall the main theoretical results concerning formulation (1.8) that form the basis for the forthcoming analysis presented in Chapter 3 and 4.

1.2 Theoretical estimates

For the forthcoming analysis we first derive the following *trace* inequalities on polygonal/polyhedral elements.

Lemma 1.1. *Assume that the mesh $\mathscr{T}_h$ satisfies Assumption 1.1 and let $\kappa \in \mathscr{T}_h$, then the following bound holds*

$$\|v\|^2_{L^2(\partial\kappa)} \lesssim \frac{\epsilon}{h_\kappa}\|v\|^2_{L^2(\kappa)} + \frac{h_\kappa}{\epsilon}|v|^2_{H^1(\kappa)} \quad \forall v \in H^1(\kappa),$$

where h_κ is the diameter of κ and $\epsilon > 0$ is a positive number.

Proof. We follow the idea of [DE11, Proof of Lemma 1.49]. First of all, we observe that

$$\|v\|^2_{L^2(\partial\kappa)} = \sum_{F \subset \partial\kappa} \|v\|^2_{L^2(F)}. \tag{1.9}$$

For each face $F \subset \partial\kappa$ according to Assumption 1.1 let $\kappa_\flat^F \subset \kappa$ be a d-dimensional simplex sharing the face F. In $\kappa_\flat^F$ we define a function θ_F as follows:

$$\theta_F : \mathbf{x} \in \overline{\kappa_\flat^F} \mapsto \theta_F(\mathbf{x}) = \frac{|F|}{d|\kappa_\flat^F|}(\mathbf{x} - \mathbf{v}_F),$$

where $\mathbf{v}_F$ is the vertex of the simplex $\kappa_\flat^F$ opposite to the face F. We observe that:

$$\theta_F(\mathbf{x}) \cdot \mathbf{n}_F = \frac{|F|}{d|\kappa_\flat^F|}\tilde{h} = 1 \quad \forall \mathbf{x} \in F,$$

where $\mathbf{n}_F$ is the outward unit vector to the face F and $\tilde{h}$ is the height of the simplex with respect to the face F, thereby $\tilde{h} = \frac{d|\kappa_\flat^F|}{|F|}$. Moreover, it holds

$$\theta_F|_{F'} \cdot \mathbf{n}_{F'} = 0 \quad \forall \text{ face } F' \subset \partial\kappa_\flat^F, F' \neq F.$$

Then we have

$$\|v\|^2_{L^2(F)} = \int_F |v|^2 \mathrm{d}s = \int_{\partial \kappa_\flat^F} |v|^2 \theta_F \cdot \mathbf{n}_F \mathrm{d}s = \int_{\kappa_\flat^F} \nabla \cdot (|v|^2 \theta_F) d\mathbf{x}$$
$$= \int_{\kappa_\flat^F} 2v \nabla v \cdot \theta_F d\mathbf{x} + \int_{\kappa_\flat^F} |v|^2 \nabla \cdot \theta_F d\mathbf{x}.$$

We now observe that the following additional properties hold for θ_F:

$$\nabla \cdot \theta_F = \nabla \cdot \frac{|F|}{d|\kappa_\flat^F|}(\mathbf{x} - \mathbf{v}_F) = \frac{|F|}{d|\kappa_\flat^F|} \nabla \cdot \mathbf{x} = \frac{|F|}{|\kappa_\flat^F|},$$

and

$$\|\theta_F\|_{[L^\infty(\kappa_\flat^F)]^d} = \frac{|F|}{d|\kappa_\flat^F|} h_{\kappa_\flat^F} \leq \frac{|F|}{d|\kappa_\flat^F|} h_\kappa, \tag{1.10}$$

where in (1.10) the first equality follows from $\max_{\mathbf{x} \in \kappa_\flat^F} \{\mathbf{x} - \mathbf{v}_F\} = h_{\kappa_\flat^F}$, while the second inequality is a consequence of $h_{\kappa_\flat^F} \leq h_\kappa$. Hence

$$\|v\|^2_{L^2(F)} \leq 2\|\theta_F\|_{[L^\infty(\kappa_\flat^F)]^d} \|v \nabla v\|_{[L^1(\kappa_\flat^F)]^d} + \frac{|F|}{|\kappa_\flat^F|} \|v\|^2_{L^2(\kappa_\flat^F)}$$
$$\leq 2 \frac{|F|}{d|\kappa_\flat^F|} h_\kappa \|v\|_{L^2(\kappa_\flat^F)} |v|_{H^1(\kappa_\flat^F)} + \frac{|F|}{|\kappa_\flat^F|} \|v\|^2_{L^2(\kappa_\flat^F)},$$

Employing Assumption 1.1 gives

$$\|v\|^2_{L^2(F)} \leq 2C_1 \|v\|_{L^2(\kappa_\flat^F)} |v|_{H^1(\kappa_\flat^F)} + \frac{C_1 d}{h_\kappa} \|v\|^2_{L^2(\kappa_\flat^F)}.$$

Using Young's Inequality we get

$$\|v\|_{L^2(\kappa_\flat^F)} |v|_{H^1(\kappa_\flat^F)} \leq \frac{1}{2} \Big(\frac{\epsilon}{h_\kappa} \|v\|^2_{L^2(\kappa_\flat^F)} + \frac{h_\kappa}{\epsilon} |v|^2_{H^1(\kappa_\flat^F)} \Big),$$

where we have chosen $\epsilon \geq 0$. Using the previous inequality we have

$$\|v\|^2_{L^2(F)} \leq C_1 d \Big(\frac{\epsilon}{h_\kappa} \|v\|^2_{L^2(\kappa_\flat^F)} + \frac{h_\kappa}{\epsilon} |v|^2_{H^1(\kappa_\flat^F)} \Big). \tag{1.11}$$

We observe that (1.11) holds $\forall F \subset \partial\kappa$. Thereby, employing (1.9), we deduce that

$$\|v\|^2_{L^2(\partial\kappa)} = \sum_{F \subset \partial\kappa} \|v\|^2_{L^2(F)} \leq \sum_{F \subset \partial\kappa} Cd \Big(\frac{\epsilon}{h_\kappa} \|v\|^2_{L^2(\kappa_\flat^F)} + \frac{h_\kappa}{\epsilon} |v|^2_{H^1(\kappa_\flat^F)} \Big)$$
$$= C_1 d \Big(\frac{\epsilon}{h_\kappa} \sum_{F \subset \partial\kappa} \|v\|^2_{L^2(\kappa_\flat^F)} + \frac{h_\kappa}{\epsilon} \sum_{F \subset \partial\kappa} |v|^2_{H^1(\kappa_\flat^F)} \Big)$$
$$\leq C_1 d \Big(\frac{\epsilon}{h_\kappa} \|v\|^2_{L^2(\kappa)} + \frac{h_\kappa}{\epsilon} |v|^2_{H^1(\kappa)} \Big),$$

where in the last inequality we have used the fact that the simplices of the set $\{\kappa_\flat^F : F \subset \partial\kappa\}$ satisfy Assumption 1.1, in the sense that they are disjoint and $\cup_{F \subset \partial\kappa} \overline{\kappa_\flat^F} \subset \overline{\kappa}$. □

The following result is a *trace-inverse inequality* for polynomial functions on polytopic grids. This result has been proved in [CDG17], see also [AHH+17]; it is a direct consequence of Assumption 1.1, together with the analogous bound given for simplicial elements, which can be stated as follows.

Lemma 1.2. *Let $T \subset \mathbb{R}^d$, $d = 2,3$, be a simplex and let $F \subset \partial T$ be one of its faces. Then, the following bound holds*

$$\|v\|^2_{L^2(F)} \leq C_F p^2 \frac{|F|}{|T|} \|v\|^2_{L^2(T)} \quad \forall\, v \in \mathscr{P}_p(T),$$

where C_F is a positive constant independent of p, $|F|$ and $|T|$.

A detailed proof of Lemma 1.2 can be found in [WH03], for example. Then, the polygonal counterpart of Lemma 1.2 is given by the following Lemma.

Lemma 1.3. *Assume that the mesh $\mathscr{T}_h$ satisfies Assumption 1.1 and let $\kappa \in \mathscr{T}_h$. The following bound holds*

$$\|v\|^2_{L^2(\partial\kappa)} \leq C_{tr} \frac{p^2}{h_\kappa} \|v\|^2_{L^2(\kappa)} \quad \forall v \in \mathscr{P}_p(\kappa),$$

where $C_{tr} > 0$ is a constant independent of the discretizations parameters.

Proof. The proof is based on the application of Lemma 1.2 on each simplex $\kappa_\flat^F$ which satisfies Assumption 1.1. Thereby, for $v \in \mathscr{P}_p(\kappa)$, we have

$$\begin{aligned}\|v\|^2_{L^2(\partial\kappa)} &= \sum_{F\subset\partial\kappa} \|v\|^2_{L^2(F)} \leq \sum_{F\subset\partial\kappa} C_F p^2 \frac{|F|}{|T|} \|v\|^2_{L^2(\kappa_\flat^F)} \leq dC_1 \max_{F\subset\partial\kappa}\{C_F\} \frac{p^2}{h_\kappa} \sum_{F\subset\partial\kappa} \|v\|^2_{L^2(\kappa_\flat^F)} \\ &\leq C_{tr} \frac{p^2}{h_\kappa} \|v\|^2_{L^2(\kappa)},\end{aligned}$$

where $C_{tr} = dC_1 \max_{F\subset\partial\kappa}\{C_F\}$, here we have also employed the hypothesis $\bigcup_{F\subset\partial\kappa} \overline{\kappa_\flat^F} \subset \overline{\kappa}$ and $\kappa_\flat^F$ are disjoint simplices. □

Lemma 1.3 is a key result to prove the well-posedness of problem (1.8). Before showing that, we endow the discrete space V_h with the following α-DG norm

$$\|w\|_h^2 = \sum_{\kappa\in\mathscr{T}_h} \int_\kappa |\nabla w|^2 \,\mathrm{d}\mathbf{x} + \sum_{F\in\mathscr{F}_h} \int_F \sigma_h |[\![w]\!]|^2 \,\mathrm{d}s, \tag{1.12}$$

The well-posedness of problem (1.8) with respect to the norm (1.12) is then established in the following lemma.

Lemma 1.4. *The following continuity and coercivity bounds, respectively, hold:*

$$\mathscr{A}_h(u_h, v_h) \lesssim \|u_h\|_h \|v_h\|_h \quad \forall u_h, v_h \in V_h, \tag{1.13}$$

$$\mathscr{A}_h(u_h, u_h) \gtrsim \|u_h\|_h^2 \qquad \forall u_h \in V_h. \tag{1.14}$$

The second bound holds provided that C_σ appearing in (1.7) *is sufficiently large.*

Proof. For the proof of this Lemma we follows the idea of [CGH14] for its proof. We first consider the proof of the coercivity bound (1.14). For any $u_h \in V_h$ we obtain the following equality:

$$\mathscr{A}_h(u_h,u_h) = \|u_h\|_h^2 - 2\int_\Omega \nabla_h u_h \cdot \mathscr{R}(\llbracket u_h \rrbracket)\,\mathrm{d}\mathbf{x}, \tag{1.15}$$

where ∇_h is the piecewise gradient operator such that $\nabla_h v_h|_\kappa = \nabla v_h \ \forall v_h \in V_h$. Then, in order to obtain (1.14) we bound the second term on the right hand side of (1.15). By employing the Young inequality we have

$$\int_\Omega \nabla_h u_h \cdot \mathscr{R}(\llbracket u_h \rrbracket)\,\mathrm{d}\mathbf{x} \le \frac{1}{4\epsilon}\|\nabla_h u_h\|^2_{L^2(\Omega)} + \epsilon\|\mathscr{R}(\llbracket u_h \rrbracket)\|^2_{L^2(\Omega)}, \tag{1.16}$$

for any positive $\epsilon > 0$. The first term on the right hand side of (1.16) is the H^1-seminorm term of u_h appearing in the definition of $\|u_h\|_h$ given in (1.12). We then focus on the second term. From the definition of $\mathscr{R}$ given in Section 1.1.2 we have

$$\begin{aligned}\|\mathscr{R}(\llbracket u_h \rrbracket)\|^2_{L^2(\Omega)} &= \int_\Omega \mathscr{R}(\llbracket u_h \rrbracket)\cdot\mathscr{R}(\llbracket u_h \rrbracket)\,\mathrm{d}\mathbf{x} = -\int_{\mathscr{F}_h} \llbracket u_h \rrbracket \cdot \{\{\mathscr{R}(\llbracket u_h \rrbracket)\}\}\,\mathrm{d}s \\ &\le \|\sigma_h^{\frac{1}{2}}\llbracket u_h \rrbracket\|_{L^2(\mathscr{F}_h)}\ \ \|\sigma_h^{-\frac{1}{2}}\{\{\mathscr{R}(\llbracket u_h \rrbracket)\}\}\|_{L^2(\mathscr{F}_h)}.\end{aligned} \tag{1.17}$$

Moreover, by employing the trace inequality of Lemma 1.3, the second term to the right hand side of (1.17) can be bounded as

$$\begin{aligned}\|\sigma_h^{-\frac{1}{2}}\{\{\mathscr{R}(\llbracket u_h \rrbracket)\}\}\|^2_{L^2(\mathscr{F}_h)} &\le \sum_{\kappa\in\mathscr{T}_h}\|\sigma_h^{-\frac{1}{2}}\mathscr{R}(\llbracket u_h \rrbracket)\|^2_{L^2(\partial\kappa)} \\ &= \sum_{\kappa\in\mathscr{T}_h} C_\sigma^{-1}\frac{\langle h_\kappa\rangle}{p^2}\|\mathscr{R}(\llbracket u_h \rrbracket)\|^2_{L^2(\partial\kappa)} \\ &\le 4dC_\sigma^{-1}\sum_{\kappa\in\mathscr{T}_h}\frac{h_\kappa}{p^2}\|\mathscr{R}(\llbracket u_h \rrbracket)\|^2_{L^2(\partial\kappa)} \\ &\le 4dC_\sigma^{-1}C_{tr}\sum_{\kappa\in\mathscr{T}_h}\|\mathscr{R}(\llbracket u_h \rrbracket)\|^2_{L^2(\kappa)}.\end{aligned}$$

where we have also employed that $\langle h_\kappa\rangle \le 2h_{\kappa^\pm}$. By inserting the previous bound into (1.17) we obtain

$$\|\ \mathscr{R}(\llbracket u_h \rrbracket)\|_{L^2(\Omega)} \lesssim \sqrt{C_{tr}C_\sigma^{-1}}\|\sigma_h^{\frac{1}{2}}\llbracket u_h \rrbracket\|_{L^2(\mathscr{F}_h)}. \tag{1.18}$$

Inequality (1.18), together with (1.16) lead to

$$\int_\Omega \nabla_h u_h \cdot \mathscr{R}(\llbracket u_h \rrbracket)\,\mathrm{d}\mathbf{x} \lesssim \frac{1}{4\epsilon}\|\nabla_h u_h\|^2_{L^2(\Omega)} + \epsilon\frac{C_{tr}}{C_\sigma}\sum_{F\in\mathscr{F}_h}\|\sigma_h^{\frac{1}{2}}\llbracket u_h \rrbracket\|^2_{L^2(F)}. \tag{1.19}$$

Then, inserting the previous bound into (1.15) we deduce that

$$\begin{aligned}\mathscr{A}_h(u_h,u_h) &\gtrsim \|u_h\|_h - \frac{1}{2\epsilon}\|\nabla_h u_h\|^2_{L^2(\Omega)} - 2\epsilon\frac{C_{tr}}{C_\sigma}\sum_{F\in\mathscr{F}_h}\|\sigma_h^{\frac{1}{2}}\llbracket u_h \rrbracket\|^2_{L^2(F)} \\ &\gtrsim \Big(1-\frac{1}{2\epsilon}\Big)\|\ \nabla_h u_h\|^2_{L^2(\Omega)} + \Big(1-2\epsilon\frac{C_{tr}}{C_\sigma}\Big)\sum_{F\in\mathscr{F}_h}\|\sigma_h^{\frac{1}{2}}\llbracket u_h \rrbracket\|^2_{L^2(F)}.\end{aligned}$$

Hence, the bilinear from $\mathscr{A}_h(\cdot,\cdot)$ is coercive over $V_h \times V_h$ if $C_\sigma \gtrsim 2\epsilon C_{tr}$ for some $\epsilon > \frac{1}{2}$. We point out that $C_\sigma > 0$ is independent of p, $|F|$, $|\kappa|$ and h_κ. The continuity bound (1.13) can be carried out by applying the Cauchy-Schwarz inequality to each integral appearing in $\mathscr{A}_h(u_h, v_h)$ and by exploiting inequality (1.19). □

We also need to introduce the following inverse inequality, cf. [CDGH16, AHH$^+$17].

Lemma 1.5. *Assume that Assumptions 1.2 and **A.3** hold. Let $\kappa \in \mathscr{T}_h$, the following inverse estimate holds*

$$\|\nabla u_h\|^2_{L^2(\kappa)} \lesssim p^4 h_\kappa^{-2} \|u_h\|^2_{L^2(\kappa)} \qquad \forall \kappa \in \mathscr{T}_h.$$

Proof. We refer to [CDGH16, CDGH17] for the proof of this result. □

Exploiting the inverse estimate of Lemma 4.5, it is possible to obtain the following upper bound on the maximum eigenvalue of $\mathscr{A}_h$. We refer to [AH11] for a similar result on standard grids, and to [AHH$^+$17] for its extension to polygonal grids.

Theorem 1.1. *Let Assumptions 1.1, 1.2 and **A.3** be satisfied. Moreover we also assume that for any $\kappa \in \mathscr{T}_h$ it holds $h_\kappa \eqsim h_{\kappa'}$ $\forall \kappa' \in \mathscr{T}_h$ s.t. $\partial\kappa \cap \partial\kappa' \neq \emptyset$. Then*

$$\mathscr{A}_h(u_h, u_h) \lesssim \sum_{\kappa \in \mathscr{T}_h} \frac{p^4}{h_\kappa^2} \|u_h\|^2_{L^2(\kappa)} \quad \forall u_h \in V_h. \tag{1.20}$$

Remark 1.4. *We observe that if the mesh $\mathscr{T}_h$ is quasi uniform, i.e. $h \eqsim h_\kappa$ $\forall \kappa \in \mathscr{T}_h$, then the bound* (1.20) *can be written as*

$$\mathscr{A}_h(u_h, u_h) \lesssim \frac{p^4}{h^2} \|u_h\|^2_{L^2(\kappa)} \quad \forall u_h \in V_h.$$

Proof. Employing to the continuity bound of Lemma 1.4 we have

$$\mathscr{A}_h(u_h, u_h) \lesssim \|\nabla_h u_h\|^2_{L^2(\Omega)} + \|\sigma_h^{\frac{1}{2}} [\![u_h]\!]\|^2_{L^2(\mathscr{F}_h)}. \tag{1.21}$$

Here, the first term on the right hand side of (1.21) can be estimated by employing Lemma 4.5 as follows:

$$\|\nabla_h u_h\|^2_{L^2(\Omega)} = \sum_{\kappa \in \mathscr{T}_h} \|\nabla u_h\|^2_{L^2(\kappa)} \lesssim \sum_{\kappa \in \mathscr{T}_h} \frac{p^4}{h_\kappa^2} \|u_h\|^2_{L^2(\kappa)}. \tag{1.22}$$

The second term on the right hand side of (1.22) can be bounded by making use of the trace inequality of Lemma 1.3 as follows:

$$\begin{aligned} \|\sigma_h^{\frac{1}{2}} [\![u_h]\!]\|^2_{L^2(\mathscr{F}_h)} &\leq \sum_{\kappa \in \mathscr{T}_h} \|\sigma_h^{\frac{1}{2}} u_h\|^2_{L^2(\partial\kappa)} = \sum_{\kappa \in \mathscr{T}_h} C_\sigma \frac{p^2}{h_\kappa} \|u_h\|^2_{L^2(\partial\kappa)} \\ &\lesssim \sum_{\kappa \in \mathscr{T}_h} \frac{p^2}{h} \|u_h\|^2_{L^2(\partial\kappa)} \lesssim \sum_{\kappa \in \mathscr{T}_h} \frac{p^2}{h_\kappa} \frac{p^2}{h_\kappa} \|u_h\|^2_{L^2(\kappa)}. \end{aligned} \tag{1.23}$$

Inserting (1.22) and (1.23) into (1.21) gives the desired result. □

Next, we recall the following approximation result, which is an analogous bound presented in [CGH14, Theorem 5.2]. This result exploits the properties of the extension operator $\mathscr{E} : H^s(\Omega) \to H^s(\mathbb{R}^d)$, $s \in \mathbb{N}_0$, such that $\mathscr{E}v|_\Omega = v$ and $\|\mathscr{E}v\|_{H^s(\mathbb{R}^d)} \lesssim \|v\|_{H^s(\Omega)}$, introduced in [Ste70].

Lemma 1.6. *Let Assumption 1.4 be satisfied, and let $v \in L^2(\Omega)$ such that, for some $k \geq 0$, $v \in H^k(\kappa)$ and $\mathscr{E}v|_{\mathscr{S}_\kappa} \in H^k(\mathscr{S}_\kappa)$ for $\kappa \in \mathscr{T}_h$, with $\mathscr{S}_\kappa \in \mathscr{T}_h^\#$ as defined in Assumption 1.4. Then, there exists a local approximant operator $\Pi_{h,\kappa} : L^2(\kappa) \to \mathscr{P}_p(\kappa)$ such that*

$$\|v - \Pi_{h,\kappa} v\|_{H^q(\kappa)} \lesssim \frac{h_\kappa^{s-q}}{p^{k-q}} \|\mathscr{E}v\|_{H^k(\mathscr{S}_\kappa)} \quad for\ 0 \leq q \leq k,$$

$$\|v - \Pi_{h,\kappa} v\|_{L^2(\partial\kappa)} \lesssim \frac{h_\kappa^{s-\frac{1}{2}}}{p^{k-\frac{1}{2}}} \|\mathscr{E}v\|_{H^k(\mathscr{S}_\kappa)} \quad if\ k \geq 1,$$

where $s = \min\{p+1, k\}$ and $p \geq 1$.

Remark 1.5 (Global approximant)**.** *If Lemma 1.6 holds $\forall\ \kappa \in \mathscr{T}_h$ we can define the global approximant operator $\Pi_h : L^2(\Omega) \to V_h$ such that $\Pi_h|_\kappa = \Pi_{h,\kappa}\ \forall \kappa \in \mathscr{T}_h$. Moreover, if $v \in H^k(\Omega)$ and if we assume that $h \eqsim h_\kappa\ \forall \kappa \in \mathscr{T}_h$, then the following global bound holds:*

$$\|v - \Pi_h v\|_{H^q(\mathscr{T}_h)} \lesssim \frac{h^{s-q}}{p^{k-q}} \|v\|_{H^k(\Omega)}, \quad for \quad 0 \leq q \leq k.$$

The result presented in Lemma 1.6 leads to the following error bounds, which follows from the energy norm error bounds that have been proved in [CGH14], see also [CDG17] in the general case.

Theorem 1.2 (Error bound)**.** *Assume that Assumptions 1.1 and 1.4 hold. Moreover we also assume that for any $\kappa \in \mathscr{T}_h$ it holds $h_\kappa \eqsim h_{\kappa'}\ \forall \kappa' \in \mathscr{T}_h$ s.t. $\partial\kappa \cap \partial\kappa' \neq \emptyset$. Let $u_h \in V_h$ be the DG solution of problem* (1.8)*,* i.e.*,*

$$\mathscr{A}_h(u_h, v_h) = \int_\Omega f v_h \, \mathrm{d}\mathbf{x} \quad \forall v_h \in V_h.$$

If the solution u of (1.1) *is sufficiently regular, i.e. $u|_\kappa \in H^k(\kappa)\ \forall \kappa \in \mathscr{T}_h$ such that $\mathscr{E}u|_{\mathscr{S}_\kappa} \in H^k(\mathscr{S}_\kappa)$ with $k > \frac{3}{2}$, where $\mathscr{S}_\kappa$ is defined in Assumption 1.4, then*

$$\|u - u_h\|_h^2 \lesssim \sum_{\kappa \in \mathscr{T}_h} \frac{h_\kappa^{2s-2}}{p^{2k-3}} \|\mathscr{E}u\|_{H^k(\mathscr{S}_\kappa)}^2.$$

where $s = \min\{p+1, k\},\ p \geq 1$.

Remark 1.6. *We point out that in problem* (1.8) *the bilinear form* (1.5) *is defined only for discrete functions, so that it is not possible to exploit here the Galerkin orthogonality property. The proof of Theorem 1.2 can be carried out by introducing an inconsistent DG formulation and by making use of Strang's second lemma. We refer to [CGH14, CDGH17, AFRV16] for further details.*

Thanks to the error bound of Theorem 1.2 and under additional assumption on the regularity of the solution u and the polygonal mesh $\mathcal{T}_h$, it is possible to derive the following global bounds of the error, cf. Ref. [CGH14, CDGH16, AHH+17].

Corollary 1.1. *Assume that the hypothesis of Theorem 1.2 are satisfied. Moreover, we assume that $h_\kappa \eqsim h \ \forall \kappa \in \mathcal{T}_h$ and that $u \in H^k(\Omega)$ with $k \geq 2$, where u is the solution of* (1.1)*, then*

$$\|u - u_h\|_{h,\rho} \lesssim \frac{h^{s-1}}{p^{k-\frac{3}{2}}} \|u\|_{H^k(\Omega)},$$
$$\|u - u_h\|_{L^2(\Omega)} \lesssim \frac{h^{s}}{p^{k-1}} \|u\|_{H^k(\Omega)},$$

where $s = \min\{p+1, k\}$, $p \geq 1$.

Remark 1.7 (Optimal error bounds)**.** *We point out that the bounds in Corollary 1.1 are optimal in h and suboptimal in p of a factor $p^{\frac{1}{2}}$ and p for the DG-norm and the L^2-norm, respectively. Optimal error estimates with respect to p can be shown, for example, by using the projector of [GS05] for quadrilateral meshes providing the solution belongs to a suitable augmented Sobolev space. We also mention [SW10] where a continuous approximant has been built. The issue of proving optimal estimates as the ones in [GS05] on polytopic meshes is an open problem. In the following, we will write:*

$$\|u - u_h\|_{h} \lesssim \frac{h^{s-1}}{p^{k-1-\frac{\mu}{2}}} \|u\|_{H^k(\Omega)},$$
$$\|u - u_h\|_{L^2(\Omega)} \lesssim \frac{h^{s}}{p^{k-\mu}} \|u\|_{H^k(\Omega)},$$

where $s = \min\{p+1, k\}$, $p \geq 1$, and $\mu \in \{0, 1\}$ for optimal and suboptimal estimates, respectively.

CHAPTER 2

FAST NUMERICAL INTEGRATION ON POLYTOPIC MESHES

In this chapter we propose a new approach for the numerical evaluation of the integrals required to assemble the mass and stiffness matrices arising from the DG discretization of elliptic problems, where the underlying mesh is composed of polygonal/polyhedral elements. Starting from the idea proposed in [CLS15] for the integration of homogeneous functions, we develop a cubature method which does not require the definition of a set of nodes and weights on the domain of integration, and allows for the exact integration of polynomial functions based on evaluating the integrand and its derivatives at the vertices of the polytopal integration domain. This approach shows a remarkable gain in terms of CPU time with respect to classical quadrature rules, maintaining the same degree of accuracy. On the one hand, the number of computations is optimized, with respect to the polynomial degree of the integrand, and moreover less memory storage is required since a sub-tessellation and quadrature nodes and weights are not needed. The proposed technique is completely general and can be extended to several numerical methods based on discrete spaces defined on polygonal/polyhedral meshes, such as Virtual Element Methods, Mimetic Finite Differences, Hybrid High-Order Methods, Hybridizable DG Schemes, and Polygonal Finite Element Methods, for example. We stress that the proposed integration technique, which involves exact integration of bivariate and trivariate functions in two- and three–dimensions, respectively, has been observed to be numerically stable, at least for moderate polynomial degree.

The outline of this chapter is as follows: in Section 2.1 we recall the work introduced in [CLS15], and outline how this approach can be employed to efficiently compute the integral of d-variate polynomial functions over general polytopes in d-dimensions, $d = 1,2,3$. In Section 2.2 we outline the exploitation of the method presented in Section 2.1 for the assembly of the stiffness matrix appearing in the DG formulation, as well as the assembly of the mass matrix; the latter is

needed in order to apply the multigrid and space decomposition methods presented in Chapter 3 and 4. Several two– and three–dimensional numerical results are presented in Section 2.3 in order to show the efficiency of the proposed approach.

2.1 Integrating polynomials over general polygons/polyhedra

In this section we review the algorithm introduced by Chin, Lasserre, and Sukumar in [CLS15] for the integration of homogeneous functions over a polytopic domain. To this end, we consider the numerical computation of $\int_{\mathscr{P}} g(\mathbf{x})\mathrm{d}\mathbf{x}$, where

- $\mathscr{P} \subset \mathbb{R}^d$, $d = 2,3$, is a closed polytope, whose boundary $\partial\mathscr{P}$ is defined by m $(d-1)$–dimensional *faces* $\mathscr{F}_i$, $i = 1,\dots,m$. Each *face* $\mathscr{F}_i$ lies in a hyperplane $\mathscr{H}_i$ identified by a vector $\mathbf{a}_i \in \mathbb{R}^d$ and a scalar number b_i, such that

$$\mathbf{x} \in \mathscr{H}_i \iff \mathbf{a}_i \cdot \mathbf{x} = b_i, \quad i = 1,\dots,m. \tag{2.1}$$

  We observe that $\mathbf{a}_i$, $i = 1,\dots,m$, can be chosen as the unit outward normal vector to $\mathscr{F}_i$, $i = 1,\dots,m$, respectively, relative to $\mathscr{P}$, cf. Figs. 2.1 and 2.2.

- $g : \mathscr{P} \to \mathbb{R}$ is a *homogeneous function* of degree $q \in \mathbb{R}$, i.e., for all $\lambda > 0$, $g(\lambda\mathbf{x}) = \lambda^q g(\mathbf{x})$ for all $\mathbf{x} \in \mathscr{P}$.

We recall that Euler's homogeneous function theorem [SB96] states that, if g is a *homogeneous function* of degree $q \geq 0$, then the following identity holds:

$$q\ g(\mathbf{x}) = \nabla g(\mathbf{x}) \cdot \mathbf{x} \quad \forall \mathbf{x} \in \mathscr{P}. \tag{2.2}$$

Next we introduce the generalized Stokes' theorem, which can be stated as follows, cf. [Tay96]: given a generic vector field $\mathbf{X} : \mathscr{P} \to \mathbb{R}^d$, the following identity holds

$$\int_{\mathscr{P}} (\nabla \cdot \mathbf{X}(\mathbf{x})) g(\mathbf{x})\mathrm{d}\mathbf{x} + \int_{\mathscr{P}} \nabla g(\mathbf{x}) \cdot \mathbf{X}(\mathbf{x})\mathrm{d}\mathbf{x} = \int_{\partial\mathscr{P}} \mathbf{X}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) g(\mathbf{x})\mathrm{d}s, \tag{2.3}$$

where $\mathbf{n}$ is the unit outward normal vector to $\mathscr{P}$ and $\mathrm{d}s$ denotes the $(d-1)$-dimensional (surface) measure. Selecting $\mathbf{X} = \mathbf{x}$ in (2.3), and employing (2.2), gives

$$\int_{\mathscr{P}} g(\mathbf{x})\mathrm{d}\mathbf{x} = \frac{1}{d+q} \int_{\partial\mathscr{P}} \mathbf{x} \cdot \mathbf{n}(\mathbf{x}) g(\mathbf{x})\mathrm{d}s = \frac{1}{d+q} \sum_{i=1}^{m} b_i \int_{\mathscr{F}_i} g(\mathbf{x})\ \mathrm{d}s. \tag{2.4}$$

Equation (2.4) states that if g is *homogeneous*, then the integral of g over a polytope $\mathscr{P}$ can be evaluated by computing the integral of the same function over the boundary faces $\mathscr{F}_i \subset \partial\mathscr{P}$, $i = 1,\dots,m$. By applying Stokes' theorem recursively, we can further reduce each term $\int_{\mathscr{F}_i} g(\mathbf{x})\mathrm{d}s$, $i = 1,\dots,m$, to the integration over $\partial\mathscr{F}_i$, $i = 1,\dots,m$, respectively. To this end, Stokes' theorem needs to be applied on the hyperplane $\mathscr{H}_i$, $i = 1,\dots,m$, in which each $\mathscr{F}_i$, $i = 1,\dots,m$,

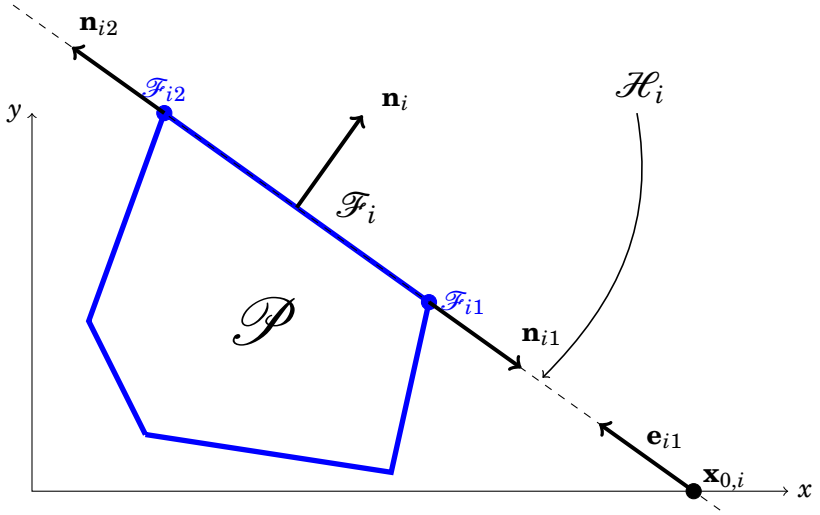


Figure 2.1: Example of a two–dimensional polytope $\mathscr{P}$ and its face $\mathscr{F}_i$. The hyperplane $\mathscr{H}_i$ is defined by the local origin $\mathbf{x}_{0,i}$ and the vector $\mathbf{e}_{i1}$.

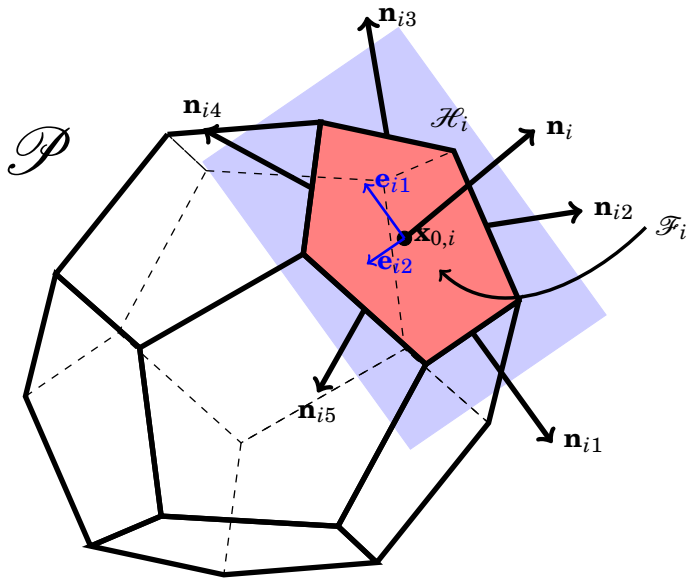


Figure 2.2: The dodecahedron $\mathscr{P}$ with pentagonal faces and the face $\mathscr{F}_i \subset \partial\mathscr{P}$ with unit outward normal vector $\mathbf{n}_i$. Here, $\mathscr{F}_i$ has five edges $\mathscr{F}_{ij}, j = 1,\ldots,5$, and five unit outward normal vectors $\mathbf{n}_{ij},\ j = 1,\ldots,5$, lying on the hyperplane $\mathscr{H}_i$. The hyperplane $\mathscr{H}_i$ is identified by the local origin $\mathbf{x}_{0,i}$ and the orthonormal vectors $\mathbf{e}_{i1}, \mathbf{e}_{i2}$.

lies, respectively. In order to proceed, let $\boldsymbol{\gamma} : \mathbb{R}^{d-1} \to \mathbb{R}^d$ be the function which expresses a generic point $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_{d-1})^\top \in \mathbb{R}^{d-1}$ as a point in $\mathbb{R}^d$ that lies on $\mathscr{H}_i,\ i = 1, \ldots, m$, i.e.,

$$\tilde{\mathbf{x}} \longmapsto \boldsymbol{\gamma}(\tilde{\mathbf{x}}) = \mathbf{x}_{0,i} + \sum_{n=1}^{d-1} \tilde{x}_n \mathbf{e}_{in}, \quad \text{with } \ \mathbf{e}_{in} \in \mathbb{R}^d, \ \ \mathbf{e}_{in} \cdot \mathbf{e}_{im} = \delta_{nm}.$$

Here, $\mathbf{x}_{0,i} \in \mathscr{H}_i,\ i = 1, \ldots, m$, is an arbitrary point which represents the origin of the coordinate system on $\mathscr{H}_i$, and $\{\mathbf{e}_{in}\}_{n=1}^{d-1}$ is an orthonormal basis on $\mathscr{H}_i,\ i = 1, \ldots, m$; see Figs. 2.1 and 2.2 for two– and three–dimensional examples, respectively. Notice that $\mathbf{x}_{0,i}$ does not have to lie inside $\mathscr{F}_i,\ i = 1, \ldots, m$. Let $\widetilde{\mathscr{F}_i} \subset \mathbb{R}^{d-1}$ such that $\boldsymbol{\gamma}(\widetilde{\mathscr{F}_i}) = \mathscr{F}_i,\ i = 1, \ldots, m$, then the following identity holds:

$$\int_{\mathscr{F}_i} g(\mathbf{x}) \mathrm{d}s = \int_{\widetilde{\mathscr{F}_i}} g(\boldsymbol{\gamma}(\tilde{\mathbf{x}})) \mathrm{d}\tilde{\mathbf{x}}, \quad i = 1, \ldots, m. \tag{2.5}$$

Before outlining the details regarding the recursive application of the Stokes' Theorem to (2.4), we first require the following lemma.

Lemma 2.1. *Let $\mathscr{F}_{ij} \subset \partial\mathscr{F}_i\ j = 1, \ldots, m_i$, be the vertices/edges of $\mathscr{F}_i,\ i = 1, \ldots, m$, for $d = 2, 3$, respectively, and let $\mathbf{n}_{ij}$ be the unit outward normal vectors to $\mathscr{F}_{ij}$ lying in $\mathscr{H}_i$. Moreover, let $\widetilde{\mathscr{F}_{ij}} \subset \partial\widetilde{\mathscr{F}_i}$ be the preimage of $\mathscr{F}_{ij}$ with respect to the map $\boldsymbol{\gamma}$, and $\tilde{\mathbf{n}}_{ij}$ be the corresponding unit outward normal vector. Then, the following holds*

$$\tilde{\mathbf{n}}_{ij} = \underline{\mathbf{E}}^\top \mathbf{n}_{ij}, \quad i = 1, \ldots, m, \ j = 1, \ldots, m_i,$$

where $\underline{\mathbf{E}} \in \mathbb{R}^{d \times (d-1)}$, whose columns are the vectors $\{\mathbf{e}_{in}\}_{n=1}^{d-1},\ i = 1, \ldots, m$.

Proof. We first note that employing the definition of $\boldsymbol{\gamma}$ we have that

$$\mathbf{x} = \boldsymbol{\gamma}(\tilde{\mathbf{x}}) = \mathbf{x}_{0,i} + \underline{\mathbf{E}}\tilde{\mathbf{x}}, \qquad \text{i.e.,} \qquad \mathbf{x} - \mathbf{x}_{0,i} = \underline{\mathbf{E}}\tilde{\mathbf{x}}. \tag{2.6}$$

The proof now follows immediately from simple linear algebra considerations; for full details, we refer to [AHP18b]. □

Given identity (2.5) and Lemma 2.1, we can prove the following result.

Proposition 2.1. *Let $\mathscr{F}_i,\ i = 1, \ldots, m$, be a face of the polytope $\mathscr{P}$, and let $\mathscr{F}_{ij},\ j = 1, \ldots, m_i$, be the planar (straight) faces (edges) such that $\partial\mathscr{F}_i = \cup_{j=1}^{m_i} \mathscr{F}_{ij}$ for some $m_i \in \mathbb{N}$. Then, for any homogeneous function g, of degree $q \geq 0$, the following identity holds*

$$\int_{\mathscr{F}_i} g(\mathbf{x}) \mathrm{d}s = \frac{1}{d-1+q} \Big(\sum_{j=1}^{m_i} d_{ij} \int_{\mathscr{F}_{ij}} g(\mathbf{x}) \mathrm{d}\nu \ + \ \int_{\mathscr{F}_i} \mathbf{x}_{0,i} \cdot \nabla g(\mathbf{x}) \mathrm{d}s \Big),$$

where d_{ij} denotes the Euclidean distance between $\mathscr{F}_{ij}$ and $\mathbf{x}_{0,i}$, $\mathbf{x}_{0,i} \in \mathscr{H}_i$, is arbitrary, $i = 1, \ldots, m$, and $\mathrm{d}\nu$ denotes the $(d-2)$-dimensional (surface) measure.

Proof. If we denote by $\nabla_i = \left[\frac{\partial}{\partial \tilde{x}_1}, \dots, \frac{\partial}{\partial \tilde{x}_{d-1}}\right]^\top$ the gradient operator on the hyperplane $\mathscr{H}_i$, $i = 1, \dots, m$, with respect to the coordinate system $(\tilde{x}_1, \dots, \tilde{x}_{d-1})$, then, upon application of Stokes' theorem, we have

$$\underbrace{\int_{\widetilde{\mathscr{F}}_i} (\nabla_i \cdot \widetilde{\mathbf{X}}) g(\boldsymbol{\gamma}(\tilde{\mathbf{x}})) \mathrm{d}\tilde{\mathbf{x}}}_{①} + \underbrace{\int_{\widetilde{\mathscr{F}}_i} \widetilde{\mathbf{X}} \cdot \nabla_i g(\boldsymbol{\gamma}(\tilde{\mathbf{x}})) \mathrm{d}\tilde{\mathbf{x}}}_{②} = \underbrace{\int_{\partial \widetilde{\mathscr{F}}_i} \widetilde{\mathbf{X}} \cdot \tilde{\mathbf{n}}\, g(\boldsymbol{\gamma}(\tilde{\mathbf{x}})) \mathrm{d}\nu(\tilde{\mathbf{x}})}_{③}, \tag{2.7}$$

where $\tilde{\mathbf{n}}$ is the unit outward normal vector of $\widetilde{\mathscr{F}}_i$ and $\widetilde{\mathbf{X}}$ is a vector field on $\mathbb{R}^{d-1}$. Next, we transform (2.7) back to the original coordinate system. To this end, denoting $\underline{\mathbf{E}} \in \mathbb{R}^{d \times (d-1)}$ to be the matrix whose columns are the vectors $\{\mathbf{e}_{in}\}_{n=1}^{d-1}$, we observe that, if we choose $\widetilde{\mathbf{X}} = \tilde{\mathbf{x}}$, then its divergence is $\nabla_i \cdot \widetilde{\mathbf{X}} = d - 1$. Exploiting (2.6), the term $\nabla_i g(\boldsymbol{\gamma}(\tilde{\mathbf{x}}))$ can be written as follows:

$$\nabla_i g(\boldsymbol{\gamma}(\tilde{\mathbf{x}})) = \begin{bmatrix} \frac{\partial \gamma_1}{\partial \tilde{x}_1} & \frac{\partial \gamma_2}{\partial \tilde{x}_1} & \cdots & \frac{\partial \gamma_d}{\partial \tilde{x}_1} \\ \frac{\partial \gamma_1}{\partial \tilde{x}_2} & \frac{\partial \gamma_2}{\partial \tilde{x}_2} & \cdots & \frac{\partial \gamma_d}{\partial \tilde{x}_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \gamma_1}{\partial \tilde{x}_{d-1}} & \frac{\partial \gamma_2}{\partial \tilde{x}_{d-1}} & \cdots & \frac{\partial \gamma_d}{\partial \tilde{x}_{d-1}} \end{bmatrix} \begin{bmatrix} \frac{\partial g}{\partial \tilde{x}_1} \\ \frac{\partial g}{\partial \tilde{x}_2} \\ \vdots \\ \frac{\partial g}{\partial \tilde{x}_d} \end{bmatrix} = (\underline{\mathbf{E}}^\top \nabla g)(\boldsymbol{\gamma}(\tilde{\mathbf{x}})). \tag{2.8}$$

Exploiting (2.6) and (2.8), we can write ① and ② as

$$① = (d-1) \int_{\widetilde{\mathscr{F}}_i} g(\boldsymbol{\gamma}(\tilde{\mathbf{x}})) \mathrm{d}\tilde{\mathbf{x}} = (d-1) \int_{\mathscr{F}_i} g(\mathbf{x}) \mathrm{d}s, \tag{2.9}$$

$$\begin{aligned} ② &= \int_{\widetilde{\mathscr{F}}_i} \tilde{\mathbf{x}}^\top \underline{\mathbf{E}}^\top \nabla g(\boldsymbol{\gamma}(\tilde{\mathbf{x}})) \mathrm{d}\tilde{\mathbf{x}} = \int_{\mathscr{F}_i} (\mathbf{x} - \mathbf{x}_{0,i}) \cdot \nabla g(\mathbf{x}) \mathrm{d}s \\ &= q \int_{\mathscr{F}_i} g(\mathbf{x}) \mathrm{d}s - \int_{\mathscr{F}_i} \mathbf{x}_{0,i} \cdot \nabla g(\mathbf{x}) \mathrm{d}s, \end{aligned} \tag{2.10}$$

respectively. Employing Lemma 2.1, together with (2.6), we have that

$$\begin{aligned} ③ &= \sum_{j=1}^{m_i} \int_{\widetilde{\mathscr{F}_{ij}}} \tilde{\mathbf{x}}^\top \tilde{\mathbf{n}}_{ij} g(\boldsymbol{\gamma}(\tilde{\mathbf{x}})) \mathrm{d}\nu(\tilde{\mathbf{x}}) = \sum_{j=1}^{m_i} \int_{\mathscr{F}_{ij}} (\mathbf{x} - \mathbf{x}_{0,i})^\top \underline{\mathbf{E}\mathbf{E}}^\top \mathbf{n}_{ij} g(\mathbf{x}) \mathrm{d}\nu(\mathbf{x}) \\ &= \sum_{j=1}^{m_i} \int_{\mathscr{F}_{ij}} (\mathbf{x} - \mathbf{x}_{0,i}) \cdot \mathbf{n}_{ij} g(\mathbf{x}) \mathrm{d}\nu. \end{aligned} \tag{2.11}$$

We observe that the term $(\mathbf{x} - \mathbf{x}_{0,i}) \cdot \mathbf{n}_{ij}$ is constant for any $\mathbf{x} \in \mathscr{F}_{ij}$, and that it represents the Euclidean distance between $\mathscr{F}_{ij}$ and $\mathbf{x}_{0,i}$; thereby, we define $d_{ij} = (\mathbf{x} - \mathbf{x}_{0,i}) \cdot \mathbf{n}_{ij}$. From the above identities (2.9), (2.10) and (2.11) we deduce the statement of the Proposition. □

Using Proposition 2.1, together with equation (2.4), we obtain the following identity

$$\int_{\mathscr{P}} g(\mathbf{x}) \mathrm{d}\mathbf{x} = \frac{1}{d+q} \sum_{i=1}^{m} \frac{b_i}{d-1+q} \Big(\sum_{j=1}^{m_i} d_{ij} \int_{\mathscr{F}_{ij}} g(\mathbf{x}) \mathrm{d}\nu + \int_{\mathscr{F}_i} \mathbf{x}_{0,i} \cdot \nabla g(\mathbf{x}) \mathrm{d}s \Big), \tag{2.12}$$

where we recall that $\partial \mathscr{P} = \cup_{i=1}^{m} \mathscr{F}_i$ and $\partial \mathscr{F}_i = \cup_{j=1}^{m_i} \mathscr{F}_{ij}$, for $i = 1, \dots, m$.

Remark 2.1. *If $d = 2$, then $\mathscr{F}_{ij}$ is a point and* (2.12) *states that the integral of g on $\mathscr{P}$ can be computed by vertex-evaluations of the integrand plus a line integration of the partial derivative of g. If $d = 3$ we can apply Stokes' Theorem recursively to $\int_{\mathscr{F}_{ij}} g(\mathbf{x})\mathrm{d}v$. Proceeding as before, we get*

$$\int_{\mathscr{F}_{ij}} g(\mathbf{x})\mathrm{d}v = \frac{1}{d-2+q}\Big(\sum_{k=1}^{m_{ij}} d_{ijk}\int_{\mathscr{F}_{ijk}} g(\mathbf{x})\mathrm{d}\xi \;+\; \int_{\mathscr{F}_{ij}} \mathbf{x}_{0,ij}\cdot\nabla g(\mathbf{x})\mathrm{d}v\Big),$$

where $\partial\mathscr{F}_{ij} = \cup_{k=1}^{m_{ij}}\mathscr{F}_{ijk}$, $\mathbf{x}_{0,ij}$ is an arbitrarily chosen origin for $\mathscr{F}_{ij}$, and d_{ijk} is the Euclidean distance between $\mathscr{F}_{ijk}$ and $\mathbf{x}_{0,ij}$.

In view of the application of Proposition 2.1 to finite element methods, we are interested in the integration of a particular class of *homogeneous functions*, namely *polynomial homogeneous functions* of the form

$$g(\mathbf{x}) = x_1^{k_1}x_2^{k_2}\cdots x_d^{k_d}, \quad \text{where } k_n \in \mathbb{N}_0 \text{ for } n = 1,\dots,d.$$

In this case, g is a *homogeneous function* of degree $q = k_1 + \cdots + k_d$, and the partial derivative $\frac{\partial g}{\partial x_n}$ is a *homogeneous function* of degree $q-1$. With this in mind, it is possible to recursively apply formula (2.12) to the terms involving the integration of the derivatives of g. To this end, we write $\mathscr{E} \subset \mathbb{R}^d$, $d = 2,3$, be a N-polytopic domain of integration, with $N = 1,\dots,d$, and let $\partial\mathscr{E} = \cup_{i=1}^{m}\mathscr{E}_i$, where each $\mathscr{E}_i \subset \mathbb{R}^d$ is a $(N-1)$-polytopic domain. When $N = d$, $d = 2,3$, $\mathscr{E}_i$, $i = 1,\dots,m$, can be an edge or a face, respectively; see Table 2.1 for details. We define the function

$$\mathscr{I}(N,\mathscr{E},k_1,\dots,k_d) = \int_{\mathscr{E}} x_1^{k_1}\dots x_d^{k_d}\ \mathrm{d}s_N(x_1,\dots,x_d), \tag{2.13}$$

which returns the integral of the polynomial $x_1^{k_1}\dots x_d^{k_d}$ over $\mathscr{E}$, where $\mathrm{d}s_N$ is the N–dimensional (surface) measure, $N = 1,2,\dots,d$. According to Proposition 2.1, the recursive definition of the function $\mathscr{I}(\cdot,\cdot,\dots,\cdot)$ is given in Algorithm 1.

Remark 2.2. *With a slight abuse of notation, when $1 \le N \le d-1$, in Algorithm 1 (and for the purposes of the following discussion), the point $\mathbf{x}_0 = (x_{0,1},\dots,x_{0,d})^\top$ denotes an arbitrarily chosen origin for the coordinate system which defines the N-polytope $\mathscr{E}$ and d_i represents the Euclidean distance between the $(N-1)$-polytopes $\mathscr{E}_i$, which form the boundary of $\mathscr{E}$, and $\mathbf{x}_0$, $i = 1,\dots,m$. Furthermore, in Algorithm 1, b_i, $i = 1,\dots,m$, is the same constant appearing in* (2.1)*. Here it can be evaluated as $b_i = \mathbf{n}_i\cdot\mathbf{v}$, where $\mathbf{v}$ is a vertex of $\mathscr{E}_i$ and $\mathbf{n}_i$ is the unit outward normal vector, $i = 1,\dots,m$.*

Remark 2.3. *We point out that in* (2.12)*, cf. also* (2.13)*, the shape of the underlying polytope can be general: indeed, nonconvex simply-connected domains $\mathscr{E}$ are admissible.*

Algorithm 1 $\mathscr{I}(N,\mathscr{E},k_1,\ldots,k_d)=\int_{\mathscr{E}} x_1^{k_1}\ldots x_d^{k_d}\ \mathrm{d}s_N(x_1,\ldots,x_d)$

if $N=0$ ($\mathscr{E}=(v_1,\ldots,v_d)\in\mathbb{R}^d$ is a point)

$$\textbf{return}\quad \mathscr{I}(N,\mathscr{E},k_1,\ldots,k_d)=v_1^{k_1}\cdots v_d^{k_d};$$

else if $1\le N\le d-1$ ($\mathscr{E}$ is a point if $d=1$ or an edge if $d=2$ or a face if $d=3$)

$$\begin{aligned}\mathscr{I}(N,\mathscr{E},k_1,\ldots,k_d)=\frac{1}{N+\sum_{n=1}^d k_n}\Big(&\sum_{i=1}^m d_i\ \mathscr{I}(N-1,\mathscr{E}_i,k_1,\ldots,k_d)\\ &+x_{0,1}\ k_1\ \mathscr{I}(N,\mathscr{E},k_1-1,k_2,\ldots,k_d)\\ &+\cdots+\ x_{0,d}\ k_d\ \mathscr{I}(N,\mathscr{E},k_1,\ldots,k_d-1)\Big);\end{aligned}$$

else if $N=d$ ($\mathscr{E}$ is an interval if $d=1$ or a polygon if $d=2$ or a polyhedron if $d=3$)

$$\mathscr{I}(N,\mathscr{E},k_1,\ldots,k_d)=\frac{1}{N+\sum_{n=1}^d k_n}\Big(\sum_{i=1}^m b_i\ \mathscr{I}(N-1,\mathscr{E}_i,k_1,\ldots,k_d)\Big).$$

end if

Table 2.1: Polytopic domains of integration $\mathscr{E}$ considered in Algorithm 1 as a function of the dimension d.

	$N=3$	$N=2$	$N=1$	$N=0$
$d=3$	$\mathscr{E}=\mathscr{P}$ is a polyhedron	$\mathscr{E}=\mathscr{F}_i\subset\partial\mathscr{P}$ is a polygon	$\mathscr{E}=\mathscr{F}_{ij}\subset\partial\mathscr{F}_i$ is an edge	$\mathscr{E}=\mathscr{F}_{ijk}\subset\partial\mathscr{F}_{ij}$ is a point
$d=2$		$\mathscr{E}=\mathscr{P}$ is a polygon	$\mathscr{E}=\mathscr{F}_i\subset\partial\mathscr{P}$ is an edge	$\mathscr{E}=\mathscr{F}_{ij}\subset\partial\mathscr{F}_i$ is a point
$d=1$			$\mathscr{E}=\mathscr{P}$ is an interval	$\mathscr{E}=\mathscr{F}_i\subset\partial\mathscr{P}$ is a point

2.1.1 Integration of bivariate polynomials over polygonal domains

In order to test the performance of the method proposed in Algorithm 1, we consider the integration of bivariate *homogeneous functions* on a given polygon $\mathscr{P}\subset\mathbb{R}^2$ based on using the three different techniques:

T.1 Recursive algorithm described in Section 2.1, based on formula (2.13)

$$\int_{\mathscr{P}} x^k\ y^l\,\mathrm{d}\mathbf{x}=\mathscr{I}(2,\mathscr{P},k,l),$$

cf. Algorithm 1.

T.2 Use of the formula (2.4) together with numerical integration employed for the evaluation of the edge integrals with known one–dimensional Gaussian quadrature rules, as recently proposed in [CLS17];

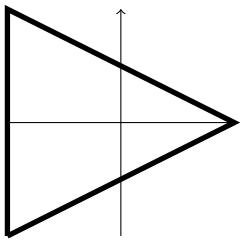


Figure 2.3: Triangle ($\mathscr{P}_1$).

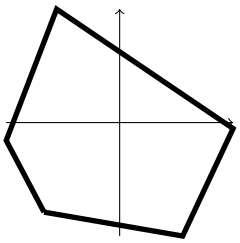


Figure 2.4: Irregular (non-convex) polygon with 5 faces ($\mathscr{P}_2$).

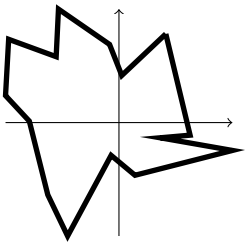


Figure 2.5: Irregular polygon with 15 faces ($\mathscr{P}_3$).

Table 2.2: Coordinates of the polygons of Figs. 2.3, 2.4, and 2.5.

	vertex	x-cordinates	y-cordinates
$\mathscr{P}_1$	1	−1.000000000000000	−1.000000000000000
	2	1.000000000000000	0.000000000000000
	3	−1.000000000000000	1.000000000000000
$\mathscr{P}_2$	1	−0.666666666666667	−0.789473684210526
	2	0.555555555555556	−1.000000000000000
	3	1.000000000000000	−0.052631578947368
	4	−0.555555555555556	1.000000000000000
	5	−1.000000000000000	−0.157894736842105
$\mathscr{P}_3$	1	0.413048522141662	0.781696234443715
	2	0.024879797655533	0.415324992429711
	3	−0.082799691823524	0.688810136531751
	4	−0.533191422779328	1.000000000000000
	5	−0.553573605852999	0.580958514816226
	6	−0.972432940212767	0.734117068746903
	7	−1.000000000000000	0.238078507228890
	8	−0.789986179147920	0.012425068086110
	9	−0.627452906935866	−0.636532897516109
	10	−0.452662174765764	−1.000000000000000
	11	−0.069106265580153	−0.289054989277619
	12	0.141448047807069	−0.464417038155806
	13	1.000000000000000	−0.245698820584615
	14	0.363704451489016	−0.134079689960635
	15	0.627086024018283	−0.110940423607648

T.3 Sub-tessellation technique: the domain of integration $\mathscr{P}$ is firstly decomposed into triangles where standard efficient quadrature rules are then employed.

We test the three different approaches for integrating bivariate polynomials of different polynomial degrees on the triangle depicted in Fig. 2.3 and the two irregular polygons shown in Figs. 2.4 and 2.5, cf. Table 2.2 for the list of coordinates for each integration domain; the exact values of the integrals are given in Table 2.3. In Table 2.4 we show the average CPU-time taken to evaluate the underlying integral using each method. We point out that, for each integrand and each integration domain $\mathscr{P}$, the relative errors between the output of the three different approaches are of the order of machine precision; that is, all three algorithms return

Table 2.3: The approximated values of the integral over the three polygons in Figs. 2.3, 2.4 and 2.5 obtained with approach **T.1**.

	$\mathscr{P}_1$	$\mathscr{P}_2$	$\mathscr{P}_3$
$\int_{\mathscr{E}} x^5 y^5$	0	-0.0020324991	-0.002589861
$\int_{\mathscr{E}} x^{10} y^{10}$	0.0111339078	$7.4274779926 \times 10^{-5}$	$1.5738050178 \times 10^{-4}$
$\int_{\mathscr{E}} x^{20} y^{20}$	0.0030396808	$6.0738145408 \times 10^{-8}$	$1.3793481020 \times 10^{-6}$
$\int_{\mathscr{E}} x^{40} y^{40}$	$7.9534562047 \times 10^{-4}$	$2.2238524572 \times 10^{-12}$	$4.2588831784 \times 10^{-10}$
$\int_{\mathscr{E}} x^{10} y^5$	0	$-2.0911953867 \times 10^{-4}$	0.0014996521
$\int_{\mathscr{E}} x^{20} y^5$	0	$-1.3797380205 \times 10^{-5}$	$7.0356275077 \times 10^{-4}$
$\int_{\mathscr{E}} x^{40} y^5$	0	$-7.9203571311 \times 10^{-7}$	$2.5065856538 \times 10^{-4}$
$\int_{\mathscr{E}} x^5 y^{20}$	-0.005890191	$8.08469022058 \times 10^{-5}$	$-1.330384913 \times 10^{-4}$
$\int_{\mathscr{E}} x^5 y^{40}$	-0.001868889	$4.37593748009 \times 10^{-5}$	$-3.963064075 \times 10^{-5}$

Table 2.4: CPU times as a function of the integrand and the integration domain $\mathscr{P}$ for the three approaches **T.1**, **T.2** and **T.3**.

	$\mathscr{P}_1$			$\mathscr{P}_2$			$\mathscr{P}_3$		
	T.1	**T.2**	**T.3**	**T.1**	**T.2**	**T.3**	**T.1**	**T.2**	**T.3**
$x^5 y^5$	0.054	0.159	0.616	0.083	0.244	0.973	0.227	0.678	2.856
$x^{10} y^{10}$	0.078	0.221	1.359	0.123	0.328	2.321	0.351	0.939	7.301
$x^{20} y^{20}$	0.124	0.344	4.060	0.207	0.540	7.399	0.580	1.498	22.70
$x^{40} y^{40}$	0.208	0.578	14.79	0.377	0.934	27.24	1.073	2.671	86.63
$x^{10} y^5$	0.064	0.191	0.999	0.081	0.296	1.699	0.237	0.833	5.125
$x^{20} y^5$	0.078	0.240	1.955	0.089	0.412	3.690	0.274	1.093	10.99
$x^{40} y^5$	0.107	0.363	4.975	0.085	0.616	9.504	0.332	1.680	29.40
$x^5 y^{20}$	0.052	0.244	1.971	0.085	0.412	3.662	0.243	1.117	11.07
$x^5 y^{40}$	0.051	0.365	5.009	0.082	0.597	9.295	0.272	1.673	29.17

the exact integral up to roundoff error. For completeness, we note that the times for **T.1** include the computation of b_i, $\mathbf{n}_i$, and d_{ij}, $j = 1,\ldots,m_i$, $i = 1,\ldots,m$. For **T.2** we take into account the evaluation of b_i, $\mathbf{n}_i$, $i = 1,\ldots,m$, and the one-time computation of the one-dimensional quadrature defined on $(-1,1)$, consisting of $\mathscr{N}$ nodes and weights, employed for the line integrations. Here, we select $\mathscr{N} = \left\lceil \frac{k+l}{2} \right\rceil + 1$, in order to guarantee the exact integration of $x^k y^l$. The CPU times for **T.3** include the one-time computation of the $\mathscr{N}^2$ nodes and weights on the reference triangle, where $\mathscr{N}$ is selected as in **T.2**, the time required for sub-tessellation, as well as the time needed for numerical integration on each sub-triangle. The results shown in Table 2.4 illustrate that the sub-tessellation approach **T.3** is the slowest while the proposed method **T.1** is the fastest for all of the considered cases; in particular, we highlight that, even for just a single domain of integration, the former method is between one- to two-orders of magnitude slower than the latter approach proposed in this thesis. Moreover, when the integration domain consists of a triangle, our algorithm **T.1** still outperforms classical quadrature rules, cf. **T.3**, even though in this case no sub-tessellation is undertaken. When comparing **T.1** and **T.2**, we observe that the former algorithm is again superior in terms of CPU time in comparison with the latter

approach; this difference seems to grow when the exponents k and l of the integrand function $x^k y^l$ are very different. This is because in **T.1** we have made an optimal selection of the points $\mathbf{x}_{0,i} = (x_{0i,1}, x_{0i,2})^\top$, $i = 1,\dots,m$, appearing in (2.12). Indeed, performing the geometric reduction of the edges of the domain of integration, we then choose $x_{0i,1} = 0$ or $x_{0i,2} = 0$, $i = 1,\dots,m$, if the exponents of the integrand function $x^k y^l$ are $k \geq l$ or $k < l$, respectively. The choice $x_{0i,1} = 0$ or $x_{0i,2} = 0$, $i = 1,\dots,m$, allows us to avoid the recursive calls to the function $\mathscr{I}(\cdot,\cdot,\dots,\cdot)$ related to the x– or y–partial derivatives, respectively. In this way the approach **T.1** is able to exploit the form of the integrand in order to optimize the evaluation of the corresponding integral. To explore this issue further, in the following section we consider the computational complexity of **T.1** in both the cases when an optimal and non-optimal selection of the points $\mathbf{x}_{0,i}$, $i = 1,\dots,m$, is made.

2.1.2 Computational complexity of the quadrature free method

The computational complexity of Algorithm 1, which is employed in **T.1**, depends in general on the number of recursive calls of the function $\mathscr{I}(\cdot,\cdot,\dots,\cdot)$. In particular, using the short-hand notation introduced in Remark 2.2, the selection of the points $\mathbf{x}_0 = (x_{0,1},\dots,x_{0,d})^\top$, which are used to define the origin of the coordinate system of each N-polytope $\mathscr{E}$ which defines the facets of $\mathscr{P}$ is crucial. In general, any $(d-1)$-dimensional hyperplane in $\mathbb{R}^d$ possesses a non-empty intersection with some axis of the Cartesian reference system, which means that it is always possible to choose $(d-1)$ components of $\mathbf{x}_0$ as zero. Without loss of generality we select $x_{0,r} = {}^{b_i}\!/_{n_{i,r}}$ and $x_{0,s} = 0$ for $s \neq r$, where b_i and $\mathbf{n}_i$ are as defined in Remark 2.2, and $r \in \{1,\dots,d\}$ is chosen so that $k_r = \min\{k_1,\dots,k_d\}$.

Remark 2.4. *In general, if $\mathscr{E} \subset \mathscr{H}$ is a N-polytopic domain in $\mathbb{R}^d$, then at most N components of $\mathbf{x}_0 \in \mathscr{H}$ can be selected to be zero.*

In this way, the selection of k_r essentially fixes the number of recursive calls of $\mathscr{I}(\cdot,\cdot,\dots,\cdot)$ in Algorithm 1. More precisley, we write $|\mathscr{I}(N,\mathscr{E},k_1,\dots,k_d)|_{Fl}$ to denote the number of FLOPs to perform $\mathscr{I}(N,\mathscr{E},k_1,\dots,k_d)$, and let C_N be the number of FLOPs required by $\mathscr{I}(N,\mathscr{E},k_1,\dots,k_d)$, without considering the recursive calls of $\mathscr{I}(\cdot,\cdot,\dots,\cdot)$ to itself. With this in mind, let us consider the following two examples:

- Set $d = 2$ and assume $k_1 \leq k_2$, so that we can choose $x_{0,1} \neq 0$ and $x_{0,2} = 0$ on each of the edges of $\mathscr{P}$. Then, according to Algorithm 1 we have

$$|\mathscr{I}(2,\mathscr{E},k_1,k_2)|_{Fl} = C_2 + \sum_{i=1}^{m} \underbrace{|\mathscr{I}(1,\mathscr{E}_i,k_1,k_2)|_{Fl}}_{\text{(i)}},$$

and

$$\begin{aligned}
\textcircled{i} &= C_1 + \sum_{j=1}^{2} |\mathscr{I}(0, \mathbf{v}_{ij}, k_1, k_2)|_{Fl} + |\mathscr{I}(1, \mathscr{E}_i, k_1 - 1, k_2)|_{Fl} \\
&= C_1 + 2C_0 + C_1 + \sum_{j=1}^{2} |\mathscr{I}(0, \mathbf{v}_{ij}, k_1 - 1, k_2)|_{Fl} + |\mathscr{I}(1, \mathscr{E}_i, k_1 - 2, k_2)|_{Fl} \\
&= \cdots = k_1(C_1 + 2C_0),
\end{aligned}$$

where we have denoted the vertices of the edge $\mathscr{E}_i$ as $\mathbf{v}_{i1}$ and $\mathbf{v}_{i2}$. Hence,

$$|\mathscr{I}(2, \mathscr{P}, k_1, k_2)|_{Fl} = C_2 + mk_1(C_1 + 2C_0) \sim \mathscr{O}(k_1).$$

In general, for $d = 2$ we deduce that

$$|\mathscr{I}(2, \mathscr{P}, k_1, k_2)|_{Fl} \sim \mathscr{O}(\min\{k_1, k_2\}). \tag{2.14}$$

- Set $d = 3$ and assume $k_1 = \min\{k_1, k_2, k_3\}$, so that we may select $x_{0,1} \neq 0$ and $x_{0,2} = x_{0,3} = 0$ on each of the faces of $\mathscr{P}$. Thereby, employing Algorithm 1 we deduce that

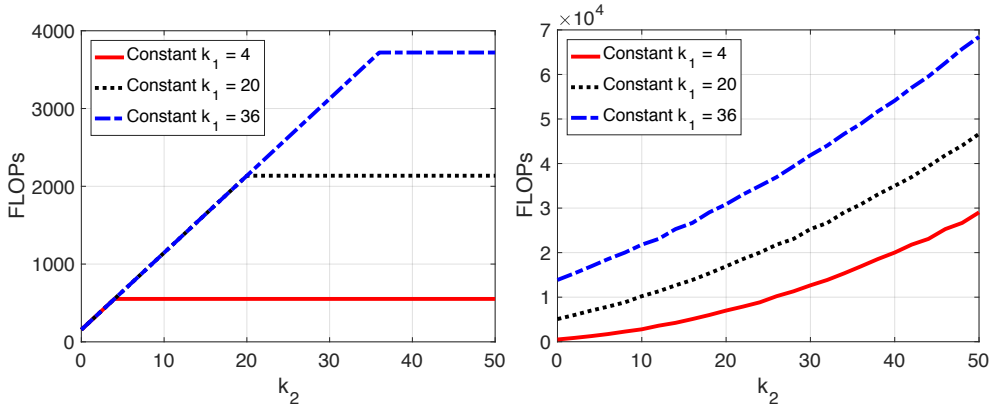
$$|\mathscr{I}(3, \mathscr{E}, k_1, k_2, k_3)|_{Fl} = C_3 + \sum_{i=1}^{m} \underbrace{|\mathscr{I}(2, \mathscr{E}_i, k_1, k_2, k_3)|_{Fl}}_{\textcircled{i}},$$

  where, for each $i = 1, \ldots, m$,

$$\begin{aligned}
\textcircled{i} &= C_2 + \sum_{j=1}^{m_i} |\mathscr{I}(1, \mathscr{E}_{ij}, k_1, k_2, k_3)|_{Fl} + |\mathscr{I}(2, \mathscr{E}_i, k_1 - 1, k_2, k_3)|_{Fl} \\
&= 2C_2 + \sum_{j=1}^{m_i} |\mathscr{I}(1, \mathscr{E}_{ij}, k_1, k_2, k_3)|_{Fl} \\
&\quad + \sum_{j=1}^{m_i} (|\mathscr{I}(1, \mathscr{E}_{ij}, k_1 - 1, k_2, k_3)|_{Fl}) + |\mathscr{I}(2, \mathscr{E}_i, k_1 - 2, k_2, k_3)|_{Fl} \\
&= \cdots = k_1 C_2 + \sum_{k=1}^{k_1} \Big(\sum_{j=1}^{m_i} |\mathscr{I}(1, \mathscr{E}_{ij}, k, k_2, k_3)|_{Fl} \Big).
\end{aligned}$$

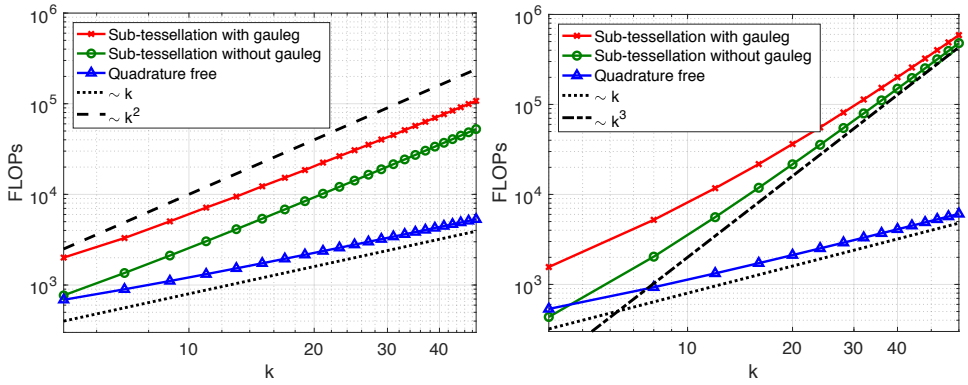
  Here, the computational complexity of $\mathscr{I}(1, \mathscr{E}_{ij}, k, k_2, k_3)$ depends on the choice of $\mathbf{x}_0 \equiv \mathbf{x}_{0,ij}$ which defines the origin of the coordinate system for $\mathscr{E}_{ij}$, $j = 1, \ldots, m_i$, $i = 1, \ldots, m$. According to Remark 2.4, two components of $\mathbf{x}_{0,ij}$ can possibly be different from zero, which implies that the complexity of Algorithm 1 increases exponentially when $d = 3$. However, it is possible to modify Algorithm 1 in order to avoid the double recursive calls which cause this exponential complexity. In particular, in Section 2.1.3 we propose an alternative algorithm which exploits the same idea of Algorithm 1 and allows us overcome this issue.

In order to confirm (2.14), we use the tool [Qia15] to measure the number of FLOPs required to exactly compute $\int_{\mathscr{P}} x^{k_1} y^{k_2} \mathrm{d}\mathbf{x}$; moreover, comparisons will also be made with **T.3**. To simplify



(a) Quadrature free method **T.1** (b) Sub-tessellation method **T.3**

Figure 2.6: Comparison of the number of FLOPs required to integrate $x^{k_1}y^{k_2}$, based on fixing k_1 and varying $k_2 \in \{0,\dots,50\}$



(a) Excludes function evaluations. (b) Total cost including function evaluations.

Figure 2.7: Comparison of the number of FLOPs required to evaluate $\int_{\mathscr{P}} x^k y^k \mathrm{d}\mathbf{x}$ as k increases employing both the quadrature free and sub-tessellation methods

the presentation, the polygon $\mathscr{P}$ is selected to be the triangle with vertices $(-1,0.3)$, $(1,-1)$, and $(0.3,1)$; thereby, **T.3** does not require the computation of a sub-tessellation. In Fig. 2.6, we plot the number of FLOPs needed to evaluate $\int_{\mathscr{P}} x^{k_1}y^{k_2}\mathrm{d}\mathbf{x}$ by fixing k_1 and varying $k_2 \in \{0,\dots,50\}$ employing both **T.1** and **T.3**. In particular, Fig. 2.6(a) shows that the number of FLOPs required by the quadrature free method **T.1** growths linearly with respect to k_2 when $k_1 > k_2$ and becomes constant as k_2 increases when $k_1 \leq k_2$. Fig. 2.7 confirms the asymptotic behaviour of the two algorithms in the case when $k_1 = k_2$; here, the number of FLOPs required by the sub-tessellation method is reported in both the case when the cost of the evaluation of the quadrature nodes and weights employing the function `gauleg`, cf. [PTVF07], for example, is included/excluded. In particular, we show results both in the case when the cost of the function evaluations is excluded, cf. Fig. 2.7(a), as well as the total number of FLOPs required by each algorithm to exactly evaluate

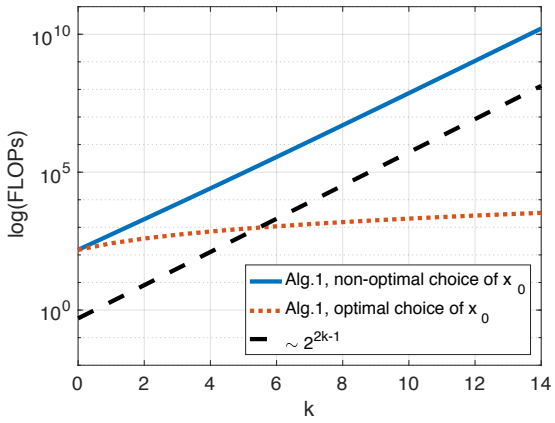


Figure 2.8: Number of FLOPs required to evaluate $\int_{\mathscr{P}} x^k y^k \mathrm{d}\mathbf{x}$ as k increases, based on employing the quadrature free method, with a sub-optimal choice of $\mathbf{x}_0$.

$\int_{\mathscr{P}} x^k y^k \mathrm{d}\mathbf{x}$, cf. Fig. 2.7(b). As expected, the computational complexity of **T.3** grows like $\mathscr{O}(k^2)$ and $\mathscr{O}(k^3)$ in these two latter cases, respectively, while the cost of the quadrature free method is always $\mathscr{O}(k)$ as k increases.

So, the numerical results presented for the proposed quadrature free method have assumed that the points $\mathbf{x}_0$, which are used to define the origin of the coordinate system of each N-polytope $\mathscr{E}$ which defines the facets of $\mathscr{P}$, has been chosen in an optimal manner to ensure that the number of recursive calls of $\mathscr{I}(\cdot,\cdot,\ldots,\cdot)$, cf. Algorithm 1, is minimized. Indeed, a sub-optimal choice of these points leads to an exponential growth in the number of recursive calls of the function $\mathscr{I}(\cdot,\cdot,\ldots,\cdot)$ in Algorithm 1. For example, if $d=2$ the non-optimal choice of $\mathbf{x}_0$ implies that each call of $\mathscr{I}(\cdot,\cdot,\ldots,\cdot)$ with $N=1$ leads to a double recursive call of $\mathscr{I}(\cdot,\cdot,\ldots,\cdot)$, up to when a zero exponent k_1 or k_2 appears as input. In particular, if $k_1=k_2=k$, it is possible to show that the number of FLOPs required by the quadrature free method grows as $\mathscr{O}(2^{2k-1})$, as k increases, cf. Fig. 2.8. In the following section, we present an alternative implementation of the quadrature free algorithm which avoids this exponential growth, irrespective of the selection of the points $\mathbf{x}_0$.

2.1.3 Integration of families of monomial functions

In the context of employing the quadrature free approach within a finite element method, in practice we are not interested in integrating a single monomial function, but instead an entire family of monomials, which, for example, form a basis for the space of polynomials of a given degree over a given polytopic element κ which belongs to the underlying computational mesh. For example, when $d=2$, let us consider the evaluation of

$$\int_{\kappa} x^{k_1} y^{k_2} \mathrm{d}\mathbf{x} \quad \forall\ k_1,k_2 \geq 0, \quad k_1+k_2 \leq p. \tag{2.15}$$

We note that even when employing the Approach **T.1** with an optimal choice of the points $\mathbf{x}_0$, the total number of FLOPs required for the computation of (2.15) is approximately $\mathscr{O}(p^3)$, as p

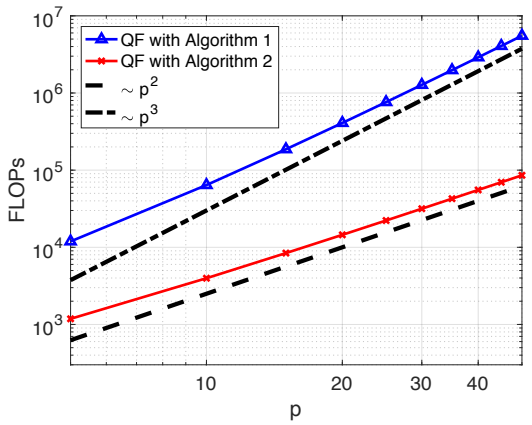


Figure 2.9: Number of FLOPs required to evaluate $\{\int_\kappa x^{k_1} y^{k_2} \mathrm{d}\mathbf{x} \ \forall\ k_1, k_2 \geq 0,\ k_1 + k_2 \leq p\}$ based on employing Algorithm 1 (with an optimal selection of the points $\mathbf{x}_0$), and Algorithm 2.

increases.

To improve the dependence on p we propose an alternative approach, cf. Algorithm 2; this based on the observation that, using the notation of Algorithm 1, if the values of $\mathscr{I}(N-1, \mathscr{E}_j, k_1, \ldots, k_d)$, $j = 1, \ldots, m$, $\mathscr{I}(N, \mathscr{E}, k_1 - 1, \ldots, k_d) \ldots \mathscr{I}(N, \mathscr{E}, k_1, \ldots, k_d - 1)$, for $1 \leq N \leq d-1$, in Algorithm 1, have already been computed, then the computation of $\mathscr{I}(N, \mathscr{E}, k_1, \ldots, k_d)$ is extremely cheap. Indeed, since we must store the integrals of all the monomials on κ anyway, we can start by computing and storing $\int_\kappa x^{k_1} y^{k_2} \mathrm{d}x_1 \mathrm{d}x_2$ related to the lower degrees k_1, k_2 and $N = 1$, then exploit these values in order to compute the integrals with higher degrees k_1, k_2 and higher dimension N of the integration domain $\mathscr{E}$. This leads to an algorithm, whereby the number of FLOPs required to compute and store $\{\int_\kappa x_1^{k_1} \ldots x_d^{k_d} \mathrm{d}s_d(x_1, \ldots, x_d),\ k_1, \ldots, k_d \geq 0,\ k_1 + k_2 + \ldots + k_d \leq p\}$ is of order $\mathscr{O}(p^d)$, as p increases, irrespective of the selection of choice of the points $\mathbf{x}_0$. In Fig. 2.9 we now compare these two approaches for $d = 2$, when the underlying element is selected to be the triangular region employed in the previous section. Here, we compare Algorithm 1, with an optimal selection of the points $\mathbf{x}_0$, with Algorithm 2, where in the latter case the points $\mathbf{x}_0$ are simply selected to be equal to the first vertex defining each edge; here, we clearly observe the predicted increase in FLOPs of $\mathscr{O}(p^3)$ and $\mathscr{O}(p^2)$, as p increases, for each of the two algorithms, respectively.

2.2 Elemental stiffness and mass matrices

We now consider problem (1.8) and we fix $\rho = 1$ in order to ease the forthcoming analysis. By fixing a basis $\{\phi_i\}_{i=1}^{N_h}$, N_h denoting the dimension of the discrete space V_h, (1.8) can be rewritten as: find $\mathbf{U} \in \mathbb{R}^{N_h}$ such that

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

Algorithm 2 Algorithm for integrating all monomials up to order p

$\partial\mathscr{E} = \{\mathscr{E}_1,\dots,\mathscr{E}_m\}$ where $\mathscr{E}_i \subset \partial\mathscr{E}$;
$F = FaceIntegrals(d-1,\mathscr{E}_1,\dots,\mathscr{E}_m,k_1,\dots,k_d)$;
for $a_1 = 0:k_1,\dots,a_d = 0:k_d;\ k_1+k_2+\dots+k_d \le p$ **do**
  $V(a_1,\dots,a_d) = \frac{1}{d+\sum_{n=1}^{d} a_n}\sum_{i=1}^{m} b_i F(a_1,\dots,a_d,i)$;
end for
procedure $F = FaceIntegrals(N,\mathscr{E}_1,\dots,\mathscr{E}_m,k_1,\dots,k_d)$
  $F(-1:k_1,\dots,-1:k_d,1:m) = 0$;
  for i=1:m **do**
    choose $\mathbf{x}_0$ as the first vertex of $\mathscr{E}_i$;
    $\partial\mathscr{E}_i = \{\mathscr{E}_{i1},\dots,\mathscr{E}_{im_i}\}$ where $\mathscr{E}_{ij} \subset \partial\mathscr{E}_i,\ j = 1,\dots,m_i$;
    if N-1>0
      $E = FaceIntegrals(N-1,\mathscr{E}_{i1},\dots,\mathscr{E}_{im_i},k_1,\dots,k_d)$;
    else if N-1=0 ($\mathscr{E}_{ij} = (v_1,\dots,v_d) \in \mathbb{R}^d$ is a point)
      $E(a_1,\dots,a_d,j) = v_1^{a_1}\dots v_d^{a_d} \quad \forall\ 0 \le a_n \le k_n,\ j = 1,\dots,m_i$;
    end if
    for $a_1 = 0:k_1,\dots,a_d = 0:k_d;\ k_1+k_2+\dots+k_d \le p$ **do**

$$F(a_1,\dots,a_d,i) = \frac{1}{N+\sum_{n=1}^{d} a_n}\Big(\sum_{j=1}^{m_i} d_{ij}E(a_1,\dots,a_d,j) + x_{0,1}k_1F(a_1-1,\dots,a_d,i) + \dots + x_{0,d}k_dF(a_1,\dots,a_d-1,i)\Big);$$

    end for
  end for
end procedure

where $\mathbf{f}_i = \int_\Omega f\phi_i \mathrm{d}\mathbf{x}\ \forall i = 1,\dots,N_h$, $\mathbf{A}$ is the stiffness matrix, given by $\mathbf{A}_{ij} = \mathscr{A}_h(\phi_j,\phi_i)\ \forall i,j = 1,\dots,N_h$, and $\mathbf{U}$ contains the expansion coefficients of $u_h \in V_h$ with respect to the chosen basis. Moreover, we also consider the assembling of the mass matrix $\mathbf{M}$, which is needed for the definition of the solvers presented in Chapters 3. Then, we are interested in computing the following matrices:

$$\mathbf{M}_{i,j} = \int_\Omega \phi_i\phi_j \mathrm{d}\mathbf{x}, \qquad \mathbf{V}_{i,j} = \int_\Omega \nabla\phi_i \cdot \nabla\phi_j \ \mathrm{d}\mathbf{x}, \tag{2.16}$$

$$\mathbf{S}_{i,j} = \sum_{F\in\mathscr{F}_h}\int_F \sigma_{h,1}[\![\phi_i]\!]\cdot[\![\phi_j]\!]\mathrm{d}s, \qquad \mathbf{I}_{i,j} = \sum_{F\in\mathscr{F}_h}\int_F \{\{\nabla\phi_i\}\}\cdot[\![\phi_j]\!]\mathrm{d}s, \tag{2.17}$$

for $i,j = 1,\dots,N_h$, where as before N_h denotes the dimension of the DG space V_h. In particular, the stiffness matrix related to the dG approximation of problem (1.8) is defined as $\mathbf{A} = \mathbf{V} - \mathbf{I}^\top - \mathbf{I} + \mathbf{S}$. In this section, we outline the application of Algorithm 2 for the efficient computation of the matrices appearing in (2.16) and (2.17).

2.2.1 Shape functions for the discrete space V_h

To construct the discrete space V_h we exploit the approach presented in [CGH14], based on employing polynomial spaces defined over the bounding box of each element. More precisely, given an element $\kappa \in \mathscr{T}_h$, we first construct the Cartesian bounding box B_κ, such that $\overline{\kappa} \subset \overline{B_\kappa}$. Given B_κ, $\kappa \in \mathscr{T}_h$, it is easy to define a linear map between B_κ and the reference element $\hat{B} = (-1,1)^d$ as follow: $\mathbf{F}_\kappa : \hat{B} \to B_\kappa$ such that $\mathbf{F}_\kappa : \hat{\mathbf{x}} \in \hat{B} \longmapsto \mathbf{F}_\kappa(\hat{\mathbf{x}}) = \mathbf{J}_\kappa \hat{\mathbf{x}} + \mathbf{t}_\kappa$, where $\mathbf{J}_\kappa \in \mathbb{R}^{d\times d}$ is the Jacobi matrix of the transformation which describes the stretching in each direction, and $\mathbf{t}_\kappa \in \mathbb{R}^d$ is the translation between the point $\mathbf{0} \in \hat{B}$ and the baricenter of the bounded box B_κ, see Fig. 2.10. We note that since $\mathbf{F}_\kappa$ affinely maps one bounding box to another (without rotation), the Jacobi matrix $\mathbf{J}_\kappa$ is diagonal.

Employing the map $\mathbf{F}_\kappa$, $\kappa \in \mathscr{T}_h$, we may define a standard polynomial space $\mathscr{P}_p(B_\kappa)$ on B_κ spanned by a set of basis functions $\{\phi_{i,\kappa}\}$ for $i = 1,\dots,N_{p_\kappa} = \dim(\mathscr{P}_p(B_\kappa))$. More precisely, we denote by $\{\mathscr{L}_n(x)\}_{n=0}^{\infty}$ the family of one–dimensional and L^2–orthonormal Legendre polynomials, defined over $L^2(-1,1)$, i.e.,

$$\mathscr{L}_n(x) = \frac{L_n(x)}{\|L_n\|_{L^2(-1,1)}}, \quad \text{with } L_n(x) = \frac{1}{2^n n!}\frac{\mathrm{d}}{\mathrm{d}x}\big[(x^2-1)^n\big],$$

cf. [QSS07, Gri05]. We then define the basis functions for the polynomial space $\mathscr{P}_p(\hat{B})$ as follows: writing $I = (i_1, i_2, \dots, i_d)$ to denote the multi-index used to identify each basis function $\{\hat{\phi}_I\}_{0\le|I|\le p}$, where $|I| = i_1 + \cdots + i_d$, we have that

$$\hat{\phi}_I(\hat{\mathbf{x}}) = \hat{\phi}_I(\hat{x}_1,\dots,\hat{x}_d) = \mathscr{L}_{i_1}(\hat{x}_1)\mathscr{L}_{i_2}(\hat{x}_2)\cdots\mathscr{L}_{i_d}(\hat{x}_d).$$

Then, the basis functions for the polynomial space $\mathscr{P}_{p_\kappa}(\kappa)$ are defined by using the map $\mathbf{F}_\kappa$, namely:

$$\phi_{I,\kappa}(\mathbf{x}) = \hat{\phi}_I(\mathbf{F}_\kappa^{-1}(\mathbf{x})) \quad \forall \mathbf{x} \in \kappa \subset B_\kappa \quad \forall I : 0 \le |I| \le p_\kappa. \tag{2.18}$$

The set $\{\phi_{I,\kappa} :\ 0 \le |I| \le p_\kappa,\ \kappa \in \mathscr{T}_h\}$ forms a basis for the space V_h. On each element $\kappa \in \mathscr{T}_h$ we introduce a bijective relation between the set of multi-indices $\{I = (i_1,\dots,i_d) : 0 \le |I| \le p_\kappa\}$ and the set $\{1,2,\dots,N_{p_\kappa}\}$.

2.2.2 Volume integrals over polytopic mesh elements

In the following we describe the application of Algorithm 2 to compute the entries in the local volume matrices

$$\mathbf{M}^\kappa_{i,j} = \int_\Omega \phi_{i,\kappa}\phi_{j,\kappa}\mathrm{d}\mathbf{x}, \quad \mathbf{V}^\kappa_{i,j} = \int_\Omega \nabla\phi_{i,\kappa}\cdot\nabla\phi_{j,\kappa}\mathrm{d}\mathbf{x} \quad i,j = 1,\dots,N_{p_\kappa}, \tag{2.19}$$

respectively, for all $\kappa \in \mathscr{T}_h$. For simplicity of presentation, we restrict ourselves to two–dimensions, though we emphasize that the three–dimensional case is analogous, cf. Section 2.3.2 below. Since the basis functions are supported only on one element, employing the transformation $\mathbf{F}_\kappa$, we have

$$\mathbf{M}^\kappa_{i,j} = \int_\kappa \phi_{i,\kappa}(x,y)\phi_{j,\kappa}(x,y)\mathrm{d}\mathbf{x} = \int_{\hat{\kappa}} \hat{\phi}_i(\hat{x},\hat{y})\hat{\phi}_j(\hat{x},\hat{y})|\mathbf{J}_\kappa|\mathrm{d}\hat{\mathbf{x}},\ i,j = 1,\dots,N_{p_\kappa},$$

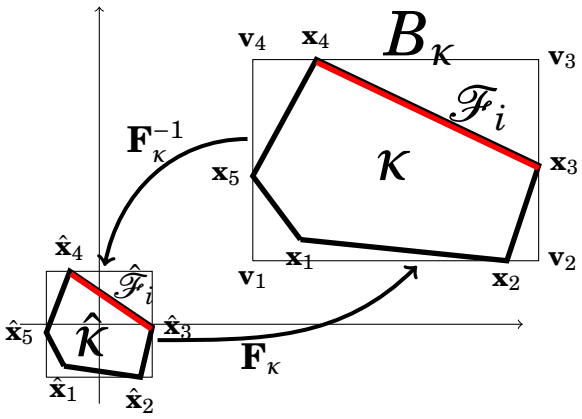


Figure 2.10: Example of polygonal element $\kappa \in \mathscr{T}_h$, the relative bounded box B_κ, the map $\mathbf{F}_\kappa$ and $\hat{\kappa} = \mathbf{F}_\kappa^{-1}(\kappa)$.

where in the last integral $\hat{\kappa} = \mathbf{F}_\kappa^{-1}(\kappa) \subset \hat{B}$, see Fig. 2.10. Here, the Jacobian of the transformation $\mathbf{F}_\kappa$ is given by $|\mathbf{J}_\kappa| = (\mathbf{J}_\kappa)_{1,1}(\mathbf{J}_\kappa)_{2,2}$, which is constant, due to the definition of the map. In order to employ the *homogeneous function* integration method described in the previous section, we need to identify the coefficients of the homogeneous polynomial expansion for the function $\hat{\phi}_i(\hat{x},\hat{y})\hat{\phi}_j(\hat{x},\hat{y})$. We observe that $\hat{\phi}_i(\hat{x},\hat{y}) = \mathscr{L}_{i_1}(\hat{x})\mathscr{L}_{i_2}(\hat{y})$, and each one–dimensional Legendre polynomial can be expanded as

$$\mathscr{L}_{i_1}(\hat{x}) = \sum_{m=0}^{i_1} C_{i_1,m}\, \hat{x}^m, \qquad \mathscr{L}_{i_2}(\hat{y}) = \sum_{n=0}^{i_2} C_{i_2,n}\, \hat{y}^n. \tag{2.20}$$

Therefore, we have

$$\begin{aligned}
\mathbf{M}^\kappa_{i,j} &= \int_{\hat{\kappa}} \Big(\sum_{m=0}^{i_1} C_{i_1,m}\hat{x}^m\Big)\Big(\sum_{n=0}^{i_2} C_{i_2,n}\hat{y}^n\Big)\Big(\sum_{s=0}^{j_1} C_{j_1,s}\hat{x}^s\Big)\Big(\sum_{r=0}^{j_2} C_{j_2,r}\hat{y}^r\Big)|\mathbf{J}_\kappa|\mathrm{d}\hat{\mathbf{x}} \\
&= \int_{\hat{\kappa}} \Big(\sum_{k=0}^{i_1+j_1} \mathscr{C}_{i_1,j_1,k}\hat{x}^k\Big)\Big(\sum_{l=0}^{i_2+j_2} \mathscr{C}_{i_2,j_2,l}\hat{y}^l\Big)|\mathbf{J}_\kappa|\mathrm{d}\hat{\mathbf{x}} \\
&= \sum_{k=0}^{i_1+j_1}\sum_{l=0}^{i_2+j_2} \mathscr{C}_{i_1,j_1,k}\ \mathscr{C}_{i_2,j_2,l}\ |\mathbf{J}_\kappa| \int_{\hat{\kappa}} \hat{x}^k\hat{y}^l \mathrm{d}\hat{\mathbf{x}}.
\end{aligned}$$

Here, we have written

$$\mathscr{C}_{i,j,k} = \sum_{n+m=k} \big(C_{i,n}\, C_{j,m}\big), \qquad \text{for } 0 \le i,j \le p_\kappa, \quad 0 \le k \le i+j. \tag{2.21}$$

Notice that the coefficients $\mathscr{C}_{i,j,k}$ can be evaluated, once and for all, independently of the polygonal element κ. We now consider the general element of the volume matrix $\mathbf{V}_{i,j}$, cf. (2.19). Proceeding as before, let I, J be the two multi-indices corresponding respectively to i and j, we have

$$\mathbf{V}^\kappa_{i,j} = \int_\kappa \nabla\phi_i \cdot \nabla\phi_j \,\mathrm{d}\mathbf{x} = \underbrace{\int_\kappa \frac{\partial\phi_{I,\kappa}}{\partial x}\,\frac{\partial\phi_{J,\kappa}}{\partial x}\,\mathrm{d}\mathbf{x}}_{①} + \underbrace{\int_\kappa \frac{\partial\phi_{I,\kappa}}{\partial y}\,\frac{\partial\phi_{J,\kappa}}{\partial y}\,\mathrm{d}\mathbf{x}}_{②}. \tag{2.22}$$

Proceeding as before, we apply a change of variables to the terms ① and ② with respect to the map $\mathbf{F}_\kappa$; thereby, we obtain

$$① = \int_{\hat{\kappa}} \frac{\partial \phi_{I,\kappa}}{\partial x}(\mathbf{F}_\kappa(\hat{\mathbf{x}}))\frac{\partial \phi_{J,\kappa}}{\partial x}(\mathbf{F}_\kappa(\hat{\mathbf{x}}))|\mathbf{J}_\kappa| \mathrm{d}\hat{\mathbf{x}},$$

$$② = \int_{\hat{\kappa}} \frac{\partial \phi_{I,\kappa}}{\partial y}(\mathbf{F}_\kappa(\hat{\mathbf{x}}))\frac{\partial \phi_{J,\kappa}}{\partial y}(\mathbf{F}_\kappa(\hat{\mathbf{x}}))|\mathbf{J}_\kappa| \mathrm{d}\hat{\mathbf{x}}.$$

From the definition of $\mathbf{F}_\kappa$, the inverse map is given by $\mathbf{F}_\kappa^{-1}(\mathbf{x}) = \mathbf{J}_\kappa^{-1}(\mathbf{x} - \mathbf{t}_\kappa)$. Then, using the definition (2.18) of the basis functions, we have the following characterization of the partial derivatives appearing in the terms ① and ②:

$$\frac{\partial}{\partial x}\phi_{I,\kappa}(\mathbf{x}) = \frac{\partial \hat{\phi}_I}{\partial \hat{x}}(\mathbf{F}_\kappa^{-1}(\mathbf{x}))\ (\mathbf{J}_\kappa^{-1})_{1,1}, \quad \frac{\partial}{\partial y}\phi_{I,\kappa}(\mathbf{x}) = \frac{\partial \hat{\phi}_I}{\partial \hat{y}}(\mathbf{F}_\kappa^{-1}(\mathbf{x}))\ (\mathbf{J}_\kappa^{-1})_{2,2}$$

where we have used that $(\mathbf{J}_\kappa^{-1})_{2,1} = (\mathbf{J}_\kappa^{-1})_{1,2} = 0$ since $\mathbf{J}_\kappa$ is diagonal. Then, ① can be written as:

$$① = \int_{\hat{\kappa}} \frac{\partial \hat{\phi}_I}{\partial \hat{x}}(\hat{\mathbf{x}})\frac{\partial \hat{\phi}_J}{\partial \hat{x}}(\hat{\mathbf{x}})\ (\mathbf{J}_\kappa^{-1})_{1,1}^2|\mathbf{J}_\kappa| \mathrm{d}\hat{\mathbf{x}}.$$

Since $(\mathbf{J}_\kappa^{-1})_{1,1}^2|\mathbf{J}_\kappa|$ is constant, the integrand function of term ① is a polynomial. Thereby, we have the following relation:

$$\left.\begin{aligned} \frac{\partial \hat{\phi}_I}{\partial \hat{x}}(\hat{\mathbf{x}}) &= \mathscr{L}'_{i_1}(\hat{x})\ \mathscr{L}_{i_2}(\hat{y}), \\ \frac{\partial \hat{\phi}_J}{\partial \hat{x}}(\hat{\mathbf{x}}) &= \mathscr{L}'_{j_1}(\hat{x})\ \mathscr{L}_{j_2}(\hat{y}), \end{aligned}\right\} \Rightarrow \frac{\partial \hat{\phi}_I}{\partial \hat{x}}(\hat{\mathbf{x}})\frac{\partial \hat{\phi}_J}{\partial \hat{x}}(\hat{\mathbf{x}}) = \mathscr{L}'_{i_1}(\hat{x})\ \mathscr{L}_{i_2}(\hat{y})\ \mathscr{L}'_{j_1}(\hat{x})\ \mathscr{L}_{j_2}(\hat{y}).$$

From the expansion (2.20) of the Legendre polynomials, we note that

$$\mathscr{L}'_0(\hat{x}) = 0, \qquad \mathscr{L}'_i(\hat{x}) = \sum_{m=0}^{i-1}(m+1)C_{i,m+1}\ \hat{x}^m = \sum_{m=0}^{i-1} C'_{i,m}\ \hat{x}^m, \quad \text{for } i > 0; \tag{2.23}$$

where the indices $C'_{i,m} = (m+1)C_{i,m+1}$ are the coefficients for the expansion of $\mathscr{L}'_i(\cdot)$. We deduce that ① $= 0$ if $i_1 = 0$ or $j_1 = 0$, and

$$① = \sum_{k=0}^{i_1+j_1-2}\ \sum_{l=0}^{i_2+j_2} \mathscr{C}'_{i_1,j_1,k}\ \mathscr{C}_{i_2,j_2,l}\ (\mathbf{J}_\kappa^{-1})_{1,1}^2|\mathbf{J}_\kappa| \int_{\hat{\kappa}} \hat{x}^k \hat{y}^l \mathrm{d}\hat{\mathbf{x}}, \quad i_1, j_1 > 0,$$

where $\mathscr{C}_{i_2,j_2,l}$ is defined in (2.21), and

$$\mathscr{C}'_{i,j,k} = \sum_{n+m=k} C'_{i,n}\ C'_{j,m}, \quad 1 \le i,j \le p_\kappa, \text{ for } 0 \le k \le i+j-2,$$

with $C'_{i,n} = (n+1)C_{i,n+1}$, $C'_{j,m} = (m+1)C_{j,m+1}$, cf. (2.23), is the expansion of the derivatives of the Legendre polynomials which is computable independently of the element κ, $\kappa \in \mathscr{T}_h$. Analogously, we deduce the following expression for the second term of equation (2.22):

$$② = \sum_{k=0}^{i_1+j_1}\ \sum_{l=0}^{i_2+j_2-2} \mathscr{C}_{i_1,j_1,k}\ \mathscr{C}'_{i_2,j_2,l}\ (\mathbf{J}_\kappa^{-1})_{2,2}^2|\mathbf{J}_\kappa| \int_{\hat{\kappa}} \hat{x}^k \hat{y}^l \mathrm{d}\hat{\mathbf{x}}.$$

2.2.3 Interface integrals over polytopic mesh elements

With regards the interface integrals appearing in equation (1.5), we describe the method by expanding the jump and average operators and computing each term separately, working, for simplicity, again in two–dimensions. Firstly, we discuss how to transform the integral over a physical face $F \subset \partial\kappa$ to the corresponding integral over the face $\hat{F} = \mathbf{F}_\kappa^{-1}(F) \subset \partial\hat{\kappa}$ on the reference rectangular element $\hat{\kappa}$. To this end, let $F \subset \partial\kappa$ be a face of the polygon κ, $\kappa \in \mathcal{T}_h$, and let $\mathbf{x}_1 = (x_1, y_1)$ and $\mathbf{x}_2 = (x_2, y_2)$ denote the vertices of the face, based on counter clock-wise ordering of the polygon vertices. The face $\hat{F} = \mathbf{F}_\kappa^{-1}(F)$ is identified by the two vertices $\hat{\mathbf{x}}_1 = \mathbf{F}_\kappa^{-1}(\mathbf{x})$ and $\hat{\mathbf{x}}_2 = \mathbf{F}_\kappa^{-1}(\mathbf{x}_2)$. For a general integrable function $g : \kappa \to \mathbb{R}$ we have

$$\int_F g(x,y)\mathrm{d}s(x,y) = \int_{\hat{F}} g(\mathbf{F}_\kappa(\hat{x},\hat{y}))\ \mathrm{d}s(\mathbf{F}_\kappa(\hat{x},\hat{y})),$$

where $\mathrm{d}s(\mathbf{F}_\kappa(\hat{x},\hat{y})) = \mathscr{J}_F\ \mathrm{d}\hat{s}$ and $\mathscr{J}_F$ is defined as $\mathscr{J}_F = \|\mathbf{J}_\kappa^{-\top}\hat{\mathbf{n}}_{\hat{F}}\|\,|\mathbf{J}_\kappa|$, where $\hat{\mathbf{n}}_{\hat{F}}$ is the unit outward normal vector to $\hat{F}$.

We next describe how to compute the interface integrals. From the definition of the jump and average operators, cf. Section 1.1.2, on each edge $F \in \mathscr{F}_h^I$ shared by the elements $\kappa^\pm$ we need to assemble

$$\begin{aligned}
\mathbf{S}_{i,j}^{+/+} &= \int_F \sigma_{h,1}\ \phi_{i,\kappa^+}\ \phi_{j,\kappa^+}\ \mathrm{d}s, \qquad \mathbf{I}_{i,j}^{+/+} = \frac{1}{2}\int_F (\nabla\phi_{i,\kappa^+}\cdot\mathbf{n}^+)\ \phi_{j,\kappa^+}\ \mathrm{d}s,\\
\mathbf{S}_{i,j}^{-/-} &= \int_F \sigma_{h,1}\ \phi_{i,\kappa^-}\ \phi_{j,\kappa^-}\ \mathrm{d}s, \qquad \mathbf{I}_{i,j}^{-/-} = \frac{1}{2}\int_F (\nabla\phi_{i,\kappa^-}\cdot\mathbf{n}^-)\ \phi_{j,\kappa^-}\ \mathrm{d}s,\\
\mathbf{S}_{i,j}^{+/-} &= -\int_F \sigma_{h,1}\ \phi_{i,\kappa^+}\ \phi_{j,\kappa^-}\ \mathrm{d}s, \quad \mathbf{I}_{i,j}^{+/-} = -\frac{1}{2}\int_F (\nabla\phi_{i,\kappa^+}\cdot\mathbf{n}^+)\ \phi_{j,\kappa^-}\ \mathrm{d}s,\\
\mathbf{S}_{i,j}^{-/+} &= -\int_F \sigma_{h,1}\ \phi_{i,\kappa^-}\ \phi_{j,\kappa^+}\ \mathrm{d}s, \quad \mathbf{I}_{i,j}^{-/+} = -\frac{1}{2}\int_F (\nabla\phi_{i,\kappa^-}\cdot\mathbf{n}^-)\ \phi_{j,\kappa^+}\ \mathrm{d}s,
\end{aligned}$$

for $i,j = 1,\ldots,N_{p_{\kappa^\pm}}$. Analogously, on the boundary face $F \in \mathscr{F}_h^B$ belonging to $\kappa^+ \in \mathcal{T}_h$ we only have to compute

$$\mathbf{S}_{i,j}^{+/+} = \int_F \sigma_{h,1}\ \phi_{i,\kappa^+}\ \phi_{j,\kappa^+}\ \mathrm{d}s, \qquad \mathbf{I}_{i,j}^{+/+} = \int_F (\nabla\phi_{i,\kappa^+}\cdot\mathbf{n}^+)\ \phi_{j,\kappa^+}\ \mathrm{d}s,$$

for $i,j = 1,\ldots,N_{p_{\kappa^+}}$. We next show how to efficiently compute a term of the form

$$\mathbf{S}_{i,j}^{+/+} = \int_F \sigma_{h,1}\ \phi_{I,\kappa^+}(x,y)\ \phi_{J,\kappa^+}(x,y)\mathrm{d}s,$$

where I, J are the suitable multi-indices associated to $i,j = 1,\ldots,N_{p_{\kappa^+}}$, respectively. Proceeding as before, we have

$$\begin{aligned}
\mathbf{S}_{i,j}^{+/+} &= \int_F \sigma_{h,1}\phi_{I,\kappa^+}(x,y)\phi_{J,\kappa^+}(x,y)\mathrm{d}s(x,y) = \int_{\hat{F}} \sigma_{h,1}\hat{\phi}_I(\hat{x},\hat{y})\hat{\phi}_J(\hat{x},\hat{y})\mathscr{J}_F\mathrm{d}\hat{s}\\
&= \sum_{k=0}^{i_1+j_1}\sum_{l=0}^{i_2+j_2} \sigma_{h,1}\ \mathscr{C}_{i_1,j_1,k}\ \mathscr{C}_{i_2,j_2,l}\ \mathscr{J}_F \int_{\hat{F}} \hat{x}^k\hat{y}^l\,\mathrm{d}\hat{s}.
\end{aligned}$$

Analogously, we have

$$\begin{aligned}\mathbf{S}_{i,j}^{+/-} &= -\int_F \sigma_{h,1}\ \phi_{I,\kappa^+}(x,y)\phi_{J,\kappa^-}(x,y)\mathrm{d}s(x,y)\\ &= -\int_{\mathbf{F}_{\kappa^+}^{-1}(F)} \sigma_{h,1}\underbrace{\phi_{I,\kappa^+}(\mathbf{F}_{\kappa^+}(\hat{\mathbf{x}}))}_{\text{(a)}}\underbrace{\phi_{J,\kappa^-}(\mathbf{F}_{\kappa^+}(\hat{\mathbf{x}}))}_{\text{(b)}}\mathscr{J}_{F^+}\mathrm{d}s.\end{aligned} \tag{2.24}$$

For the term (a), we directly apply the definition of the basis function, and obtain

$$\text{(a)} = \phi_{I,\kappa^+}(\mathbf{F}_{\kappa^+}(\hat{\mathbf{x}})) = \hat{\phi}_I(\mathbf{F}_{\kappa^+}^{-1}(\mathbf{F}_{\kappa^+}(\hat{\mathbf{x}}))) = \hat{\phi}_I(\hat{\mathbf{x}}) = \sum_{k=0}^{i_1}\sum_{l=0}^{i_2} C_{i_1,k}\ C_{i_2,l}\ \hat{x}^k\hat{y}^l, \tag{2.25}$$

while for the term (b) we have

$$\text{(b)} = \phi_{J,\kappa^-}(\mathbf{F}_{\kappa^+}(\hat{\mathbf{x}})) = \hat{\phi}_J(\mathbf{F}_{\kappa^-}^{-1}(\mathbf{F}_{\kappa^+}(\hat{\mathbf{x}}))).$$

In order to obtain a homogeneous polynomial expansion for (b) we have to write explicitly the composite map $\tilde{\mathbf{F}}(\hat{\mathbf{x}}) = \mathbf{F}_{\kappa^-}^{-1}(\mathbf{F}_{\kappa^+}(\hat{\mathbf{x}}))$. That is

$$\tilde{\mathbf{F}}(\hat{\mathbf{x}}) = \mathbf{J}_{\kappa^-}^{-1}(\mathbf{J}_{\kappa^+}\hat{\mathbf{x}} + \mathbf{t}_{\kappa^+}) - \mathbf{J}_{\kappa^-}^{-1}\mathbf{t}_{\kappa^-} = \underbrace{\mathbf{J}_{\kappa^-}^{-1}\mathbf{J}_{\kappa^+}}_{\tilde{\mathbf{J}}}\hat{\mathbf{x}} + \underbrace{\mathbf{J}_{\kappa^-}^{-1}(\mathbf{t}_{\kappa^+} - \mathbf{t}_{\kappa^-})}_{\tilde{\mathbf{t}}},$$

where the matrix $\tilde{\mathbf{J}}$ is diagonal since $\mathbf{J}_{\kappa^-}^{-1}$ and $\mathbf{J}_{\kappa^+}$ are diagonal. We then have

$$\begin{aligned}\text{(b)} &= \hat{\phi}_J(\hat{\mathbf{F}}(\hat{\mathbf{x}})) = \hat{\phi}_J(\tilde{\mathbf{J}}\hat{\mathbf{x}} + \tilde{\mathbf{t}}) = \hat{\phi}_J(\tilde{\mathbf{J}}_{1,1}\hat{x} + \tilde{\mathbf{t}}_1, \tilde{\mathbf{J}}_{2,2}\hat{y} + \tilde{\mathbf{t}}_2)\\ &= \sum_{k=0}^{j_1}\sum_{l=0}^{j_2} C_{j_1,k}\ C_{j_2,l}\ (\tilde{\mathbf{J}}_{1,1}\hat{x} + \tilde{\mathbf{t}}_1)^k(\tilde{\mathbf{J}}_{2,2}\hat{y} + \tilde{\mathbf{t}}_2)^l.\end{aligned} \tag{2.26}$$

Combining (2.25) and (2.26), and denoting by $\hat{F}^+ = \mathbf{F}_{\kappa^+}^{-1}(F)$, cf. Fig. 2.11, from (2.24) we obtain

$$\mathbf{S}_{i,j}^{+/-} = -\sum_{k=0}^{i_1+j_1}\sum_{l=0}^{i_2+j_2} \tilde{\mathscr{X}}_{i_1,j_1,k}\ \tilde{\mathscr{Y}}_{i_2,j_2,l}\ \mathscr{J}_{F^+}\int_{\hat{F}^+}\hat{x}^k\hat{y}^l\mathrm{d}\hat{s},$$

where $\tilde{\mathscr{X}}$ and $\tilde{\mathscr{Y}}$ are defined as

$$\left.\begin{aligned}\tilde{\mathscr{X}}_{i,j,k} &= \sum_{n+m=k}\left(C_{i,n}\ \tilde{X}_{j,m}\right)\\ \tilde{\mathscr{Y}}_{i,j,k} &= \sum_{n+m=k}\left(C_{i,n}\ \tilde{Y}_{j,m}\right)\end{aligned}\right\}\quad \text{for } 0\le i\le p_{\kappa^+},\ 0\le j\le p_{\kappa^-},\ 0\le k\le i+j.$$

Here, as before, $C_{i,n}$ are the coefficients of the homogeneous function expansion of the Legendre polynomials in $(-1,1)$, while $\tilde{X}_{j,m}$ and $\tilde{Y}_{j,m}$ are defined by

$$\left.\begin{aligned}\tilde{X}_{j,m} &= \sum_{r=m}^{j} C_{j,r}\binom{r}{m}(\tilde{\mathbf{J}}_{1,1})^m\ (\tilde{\mathbf{t}}_1)^{r-m}\\ \tilde{Y}_{j,m} &= \sum_{r=m}^{j} C_{j,r}\binom{r}{m}(\tilde{\mathbf{J}}_{2,2})^m\ (\tilde{\mathbf{t}}_2)^{r-m}\end{aligned}\right\}\quad \text{for } 0\le m\le p_{\kappa^-},\ m\le j\le p_{\kappa^-};$$

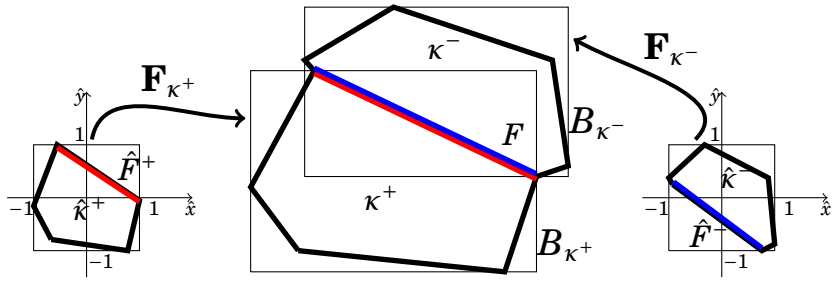


Figure 2.11: Example of a polygonal elements $\kappa^\pm \in \mathcal{T}_h$, together with the bounded boxes $B_{\kappa^\pm}$, and the local maps $\mathbf{F}_{\kappa^\pm} : \hat{\kappa} \to \kappa^\pm$ for the common face $F \subset \kappa^\pm$.

here, we have exploited the Newton-binomial expansion of the terms $(\tilde{\mathbf{J}}_{1,1}\hat{x} + \tilde{\mathbf{t}}_1)^k$ and $(\tilde{\mathbf{J}}_{2,2}\hat{y} + \tilde{\mathbf{t}}_2)^l$ appearing in equation (2.26).

Similar considerations allow us to compute

$$\begin{aligned}
\mathbf{I}^{+/+}_{i,j} &= \frac{1}{2}\Big(\int_F \frac{\partial \phi_{I,\kappa^+}}{\partial x}(x,y)\phi_{J,\kappa^+}(x,y)n^+_x + \int_F \frac{\partial \phi_{I,\kappa^+}}{\partial y}(x,y)\phi_{J,\kappa^+}(x,y)n^+_y\Big) \\
&= \frac{1}{2}\Big(\int_{\hat{F}} (\mathbf{J}^{-1}_{\kappa^+})_{1,1}\frac{\partial \hat{\phi}_I}{\partial \hat{x}}\hat{\phi}_J n^+_x \mathscr{J}_F \mathrm{d}\hat{s} + \int_{\hat{F}} (\mathbf{J}^{-1}_{\kappa^+})_{2,2}\frac{\partial \hat{\phi}_I}{\partial \hat{y}}\hat{\phi}_J n^+_y \mathscr{J}_F \mathrm{d}\hat{s}\Big) \\
&= \frac{1}{2}\mathscr{J}_F\Big((\mathbf{J}^{-1}_{\kappa^+})_{1,1} n^+_x \sum_{k=0}^{i_1+j_1-1}\sum_{l=0}^{i_2+j_2} \mathscr{C}''_{i_1,j_1,k}\mathscr{C}_{i_2,j_2,l}\int_{\hat{F}} \hat{x}^k\hat{y}^l \mathrm{d}\hat{s} \\
&\quad + (\mathbf{J}^{-1}_{\kappa^+})_{2,2} n^+_y \sum_{k=0}^{i_1+j_1}\sum_{l=0}^{i_2+j_2-1} \mathscr{C}_{i_1,j_1,k}\,\mathscr{C}''_{i_2,j_2,l}\int_{\hat{F}} \hat{x}^k\hat{y}^l \mathrm{d}\hat{s}\Big),
\end{aligned}$$

where $\mathscr{C}''_{i,j,k}$ are defined as

$$\begin{cases} \mathscr{C}''_{0,j,k} = 0 & \forall j,\ \forall k, \\ \mathscr{C}''_{i,j,k} = \sum_{n+m=k} C'_{i,n}C_{j,m}, & 1 \le i \le p_{\kappa^+}, 0 \le j \le p_{\kappa^+}, 0 \le k \le i+j-1, \end{cases}$$

and $\mathbf{n}^+ = [n^+_x, n^+_y]^\top$ is the unit outward normal vector to the physical face F from κ^+. Similarly,

$$\begin{aligned}
\mathbf{I}^{+/-}_{i,j} &= -\frac{1}{2}\int_F (\nabla\phi_{I,\kappa^+}\cdot\mathbf{n}^+)\phi_{J,\kappa^-}\mathrm{d}s \\
&= -\frac{1}{2}\Big(\int_F \frac{\partial \phi_{I,\kappa^+}}{\partial x}\phi_{J,\kappa^-}n^+_x \mathrm{d}s + \int_F \frac{\partial \phi_{I,\kappa^+}}{\partial y}\phi_{J,\kappa^-}n^+_y \mathrm{d}s\Big) \\
&= -\frac{1}{2}\mathscr{J}_F\Big((\mathbf{J}^{-1}_{\kappa})_{1,1}\sum_{k=0}^{i_1+j_1-1}\sum_{l=0}^{i_2+j_2} \tilde{\mathscr{X}}'_{i_1,j_1,k}\tilde{\mathscr{Y}}_{i_2,j_2,l}\int_{\hat{F}^+} \hat{x}^k\hat{y}^l \mathrm{d}\hat{s} \\
&\quad + (\mathbf{J}^{-1}_{\kappa})_{2,2}\sum_{k=0}^{i_1+j_1}\sum_{l=0}^{i_2+j_2-1} \tilde{\mathscr{X}}_{i_1,j_1,k}\,\tilde{\mathscr{Y}}'_{i_2,j_2,l}\int_{\hat{F}^+} \hat{x}^k\hat{y}^l \mathrm{d}\hat{s}\Big),
\end{aligned}$$

where we have also introduced $\tilde{\mathscr{X}}'$ and $\tilde{\mathscr{Y}}'$ defined as

$$\left.\begin{aligned} \tilde{\mathscr{X}}'_{i,j,k} &= \sum_{n+m=k}\big(C'_{i,n}\,\tilde{X}_{j,m}\big), \\ \tilde{\mathscr{Y}}'_{i,j,k} &= \sum_{n+m=k}\big(C'_{i,n}\,\tilde{Y}_{j,m}\big), \end{aligned}\right\} \text{ for } 1 \le i \le p_{\kappa^+},\ 0 \le j \le p_{\kappa^-},\ 0 \le k \le i+j-1.$$

Remark 2.5. *The coefficients $\tilde{X}$ and $\tilde{Y}$ depend on the maps $\mathbf{F}_{\kappa^+}$ and $\mathbf{F}_{\kappa^-}$, as well as $\tilde{\mathscr{X}}$, $\tilde{\mathscr{X}}'$, $\tilde{\mathscr{Y}}$ and $\tilde{\mathscr{Y}}'$; thereby, they must be computed for each element κ in the mesh $\mathscr{T}_h$.*

Remark 2.6. *With regards the computation of the forcing term*

$$\mathbf{f}_i = \int_\Omega f(\mathbf{x})\phi_i(\mathbf{x})\mathrm{d}\mathbf{x}, \ \forall i = 1,\dots,N_h, \tag{2.27}$$

we point out that the quadrature method proposed in this chapter allows us to exactly evaluate (2.27) *when f is a constant or a polynomial function. If f is a general function, an explicit polynomial approximation of f is required.*

2.3 Numerical experiments

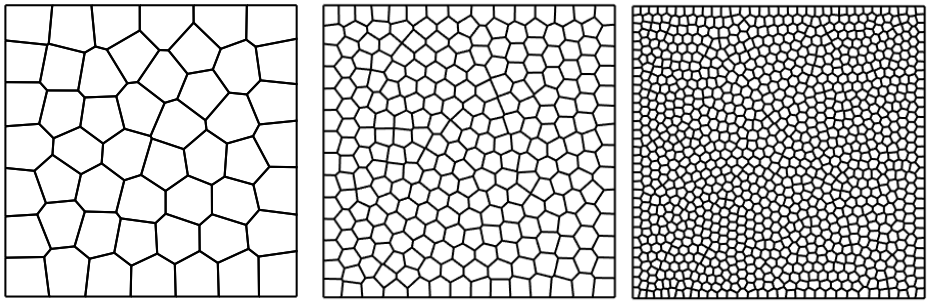
We present some two– and three–dimensional numerical experiments to test the practical performance of the proposed approach. Here, the results are compared with standard assembly algorithms based on employing efficient quadrature rules on a sub-tessellation.

2.3.1 Two–dimensional test cases

We test the performance of the algorithm outlined in Section 2.2 for the computation of the elemental mass and stiffness matrices resulting from the DG discretization (3.4) on Voronoi decompositions as shown in Fig. 2.12. In particular, we compare the CPU-time needed to assemble the local and global elemental matrices using Algorithm 2, cf. Section 2.2, with *Quadrature Integration* over polygonal domains, based on the sub-tessellation method on polygons and Gaussian line integration for the related interface terms. More precisely, given $\kappa \in \mathscr{T}_h$, the sub-tessellation scheme on κ is performed by constructing a non-overlapping sub-tessellation $\kappa_{\mathscr{S}} = \{\tau_\kappa\}$ consisting of standard triangular elements; in particular, as, for our tests, we consider Voronoi numerical grids, we exploit the convexity of κ and define $\kappa_{\mathscr{S}}$ by connecting the centre of mass of κ with its vertices. As an example, if we consider computing the elemental mass matrix $\mathbf{M}^\kappa_{i,j}$, we have that

$$\mathbf{M}^\kappa_{i,j} = \int_\kappa \phi_i\phi_j\mathrm{d}\mathbf{x} \approx \sum_{\tau_\kappa\in\kappa_{\mathscr{S}}}\sum_{r=1}^{q_{\tau_\kappa}} \phi_i(\mathbf{F}_{\tau_\kappa}(\boldsymbol{\xi}_r))\phi_j(\mathbf{F}_{\tau_\kappa}(\boldsymbol{\xi}_r))|\mathbf{J}_{\tau_\kappa}|\omega_r,$$

where $\mathbf{F}_{\tau_\kappa} : \hat{\tau} \to \tau_\kappa$ is the mapping from the reference simplex $\hat{\tau}$ to τ_κ, with Jacobian $|\mathbf{J}_{\tau_\kappa}|$, and $\{(\boldsymbol{\xi}_r,\omega_r)\}_{r=1}^{q_\kappa}$ denotes the quadrature rule defined on $\hat{\tau}$. The construction of quadrature rules on $\hat{\tau}$ may be computed based on employing the Duffy transformation, whereby the reference tensor-product element $(-1,1)^2$ is mapped to the reference simplex. As the algorithm outlined in Section 2.2 does not require the definition of quadrature nodes and weights, in the following we will refer to it as the *Quadrature Free Method*. Consider the problem (1.8) introduced in Section 1.1.2 with $d = 2$ and $\Omega = (0,1)^2$, where we select the set of basis functions $\{\phi_i\}_{i=1}^{N_h}$ for V_h as described in Section 2.2. In order to quantify the performance of the proposed approach,



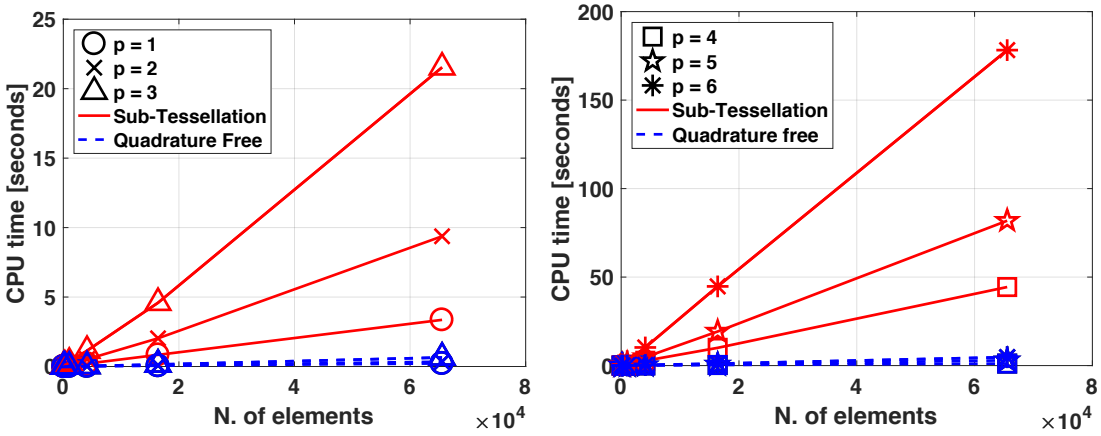
(a) 50 elements. (b) 250 elements. (c) 1000 elements.

Figure 2.12: Example of a Voronoi mesh on $\Omega = (0,1)^2$.

we consider a series of numerical tests obtained by varying the polynomial degree $p_\kappa = p$ for all $\kappa \in \mathcal{T}_h$, between 1 and 6 and by employing a series of uniform polygonal meshes of different granularity, cf. Fig. 2.12. The numerical grids are constructed based on employing `PolyMesher`, cf. [TPPM12]. Here, we are interested in the CPU time needed to assemble the matrices (2.16) and (2.17).

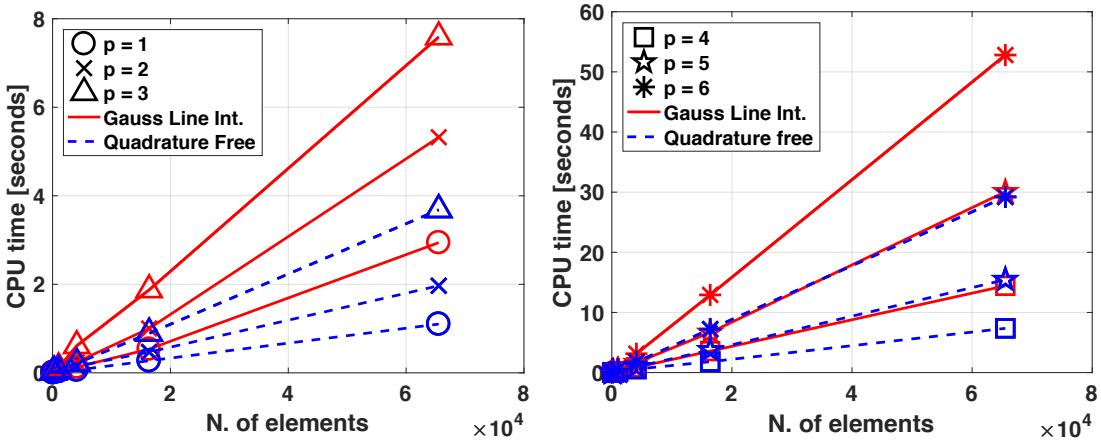
In the first test case, we consider the CPU time needed to assemble the matrices $\mathbf{M}$ and $\mathbf{V}$. As pointed out in Section 2.2, these matrices are block diagonal and each block consists of an integral over each polygonal element $\kappa \in \mathcal{T}_h$. In Fig. 2.13 we present the comparison between the CPU times needed to assemble the global matrices $\mathbf{M}$ and $\mathbf{V}$ based on employing the quadrature free method and quadrature integration (based on sub-tessellation) when varying the number of elements $N_e \in \{64, 256, 1024, 4094, 16384, 65536\}$ and the polynomial degree $p \in \{1,2,3\}$ (left), and $p \in \{4,5,6\}$ (right). Clearly, our approach outperforms the classical sub-tessellation method leading to substantial gains in efficiency. For a more detailed comparison, we have presented in Fig. 2.15(a) the logarithmic-scaled graphs of each computation: from the results of Fig. 2.15(a) we observe that the CPU time grows linearly with the number of elements.

We have repeated the same set of numerical experiments measuring the CPU times needed to assemble the face terms appearing in the matrices $\mathbf{S}$ and $\mathbf{I}$; these results are reported in Fig. 2.14. Here, the domains of integration of the integrals involved are the edges of the polygonal elements, which are simply line segments in the plane $\mathbf{R}^2$. We compare the quadrature free method described in Section 2.2.3 with classical Gaussian line integration, where the integrating function is pointwise evaluated on the physical quadrature nodes lying on each face. The graphs in Fig. 2.14(a) and 2.14(b) show the comparison between the CPU time of the two different approaches. Here, we again observe that significant computational savings are made when the proposed quadrature free method is employed, though the increase in efficiency is less than that attained for the computation of the volume integrals. In Fig. 2.15(b) we plot the logarithmic-scaled CPU time with respect to the number of mesh elements; again the CPU time grows linearly with the number of elements.



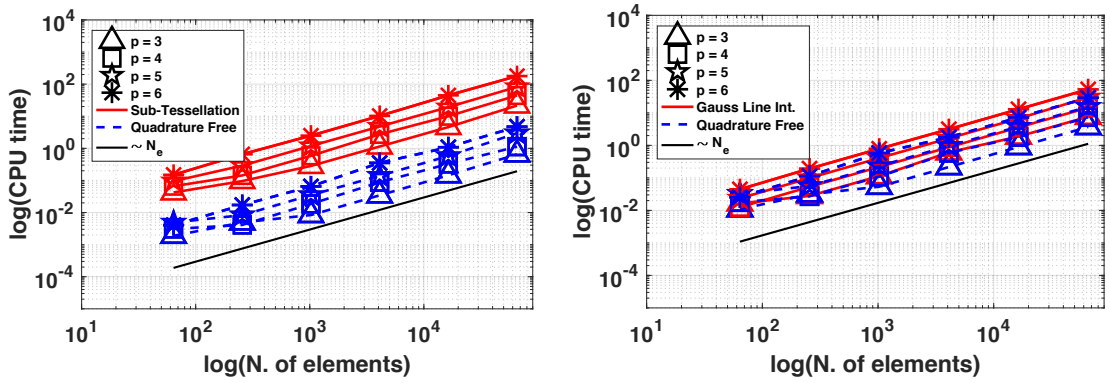
(a) Comparison for $p \in \{1,2,3\}$. (b) Comparison for $p \in \{4,5,6\}$.

Figure 2.13: Comparison of the CPU time needed to assemble the global matrices $\mathbf{M}$ and $\mathbf{V}$ for a two–dimensional problem by using the proposed quadrature free method and the classical sub-tessellation scheme. For each algorithm, each line is obtained by fixing the polynomial approximation degree $p \in \{1,2,3\}$ (left) and $p \in \{4,5,6\}$ (right), and measuring the CPU time by varying the number of elements in the underlying mesh.



(a) Comparison for $p \in \{1,2,3\}$. (b) Comparison for $p \in \{4,5,6\}$.

Figure 2.14: Comparison of the CPU time needed to assemble the global matrices $\mathbf{S}$ and $\mathbf{I}$ for a two–dimensional problem by using the proposed quadrature free method and the classical Gauss line integration scheme. For each approach, each line is obtained by fixing the polynomial approximation degree $p \in \{1,2,3\}$ (left) and $p \in \{4,5,6\}$ (right), and measuring the CPU time by varying the number of elements in the underlying mesh.



(a) CPU time comparison needed to assemble **M** and **V** in log-log scale.

(b) CPU time comparison needed to assemble **S** and **I** in log-log scale.

Figure 2.15: Comparison between the CPU time needed by the two methods to assemble the global matrices **M** and **V** (left) and **S** and **I** (right) for a three–dimensional problem, versus the number of elements and for different choices of $p = 3, \dots, 6$ (log-log scale).

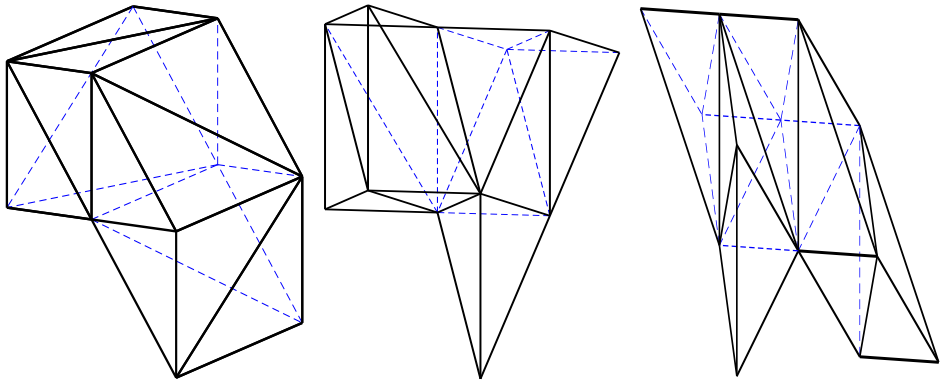
Referring to Figs. 2.13 and 2.14, we observe that the cost of assembly of the matrices **M** and **V**, which involve volume integrals over each element κ in the computational mesh $\mathcal{T}_h$, is more expensive than the time it takes to assemble the face-based matrices **S** and **I**, when the quadrature is employed. This is, of course, due to the greater number of function evaluations required to compute **M** and **V** on the underlying sub-tessellation; note that in two–dimensions, a sub-tesellation of the faces is not necessary, since they simply consist of line segments. However, the opposite behaviour is observed when the quadrature free method is employed; in this case, the volume integrals can be very efficiently computed since the coefficients $\mathscr{C}_{i,j,k}$ and $\mathscr{C}'_{i,j,k}$ only need to be computed once, cf. Section 2.2.2. On the other hand, computing the face integrals present in **S** and **I** requires the evaluation of the coefficients $\tilde{X}_{b,m}$, $\tilde{\mathscr{X}}_{a,b,k}$, $\tilde{\mathscr{X}}'_{a,b,k}$, $\tilde{Y}_{b,m}$, $\tilde{\mathscr{Y}}_{a,b,k}$, and $\tilde{\mathscr{Y}}'_{a,b,k}$, cf. Section 2.2.3, which must be computed for each face $F \in \mathcal{F}_h$.

2.3.2 Three–dimensional test cases

We now consider the diffusion-reaction problem (3.4) with $d = 3$ and $\Omega = (0,1)^3$. The polyhedral grids employed for this test case are defined by agglomeration: starting from a fine partition $\mathcal{T}_{fine}$ of Ω consisting of N_{fine} disjoint tetrahedrons $\{\kappa_f^i\}_{i=1}^{N_{fine}}$, such that $\overline{\Omega} = \cup_{i=1}^{N_{fine}} \overline{\kappa}_f^i$, a coarse mesh $\mathcal{T}_h$ of Ω consisting of disjoint polyhedral elements κ can be defined such that

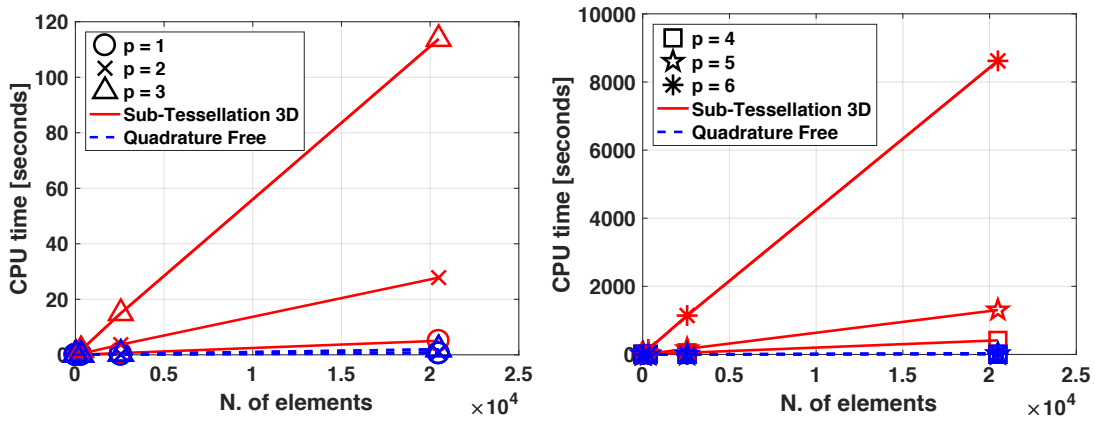
$$\kappa = \cup_{\kappa'_f \in \mathcal{S}_\kappa} \kappa'_f \quad \forall \kappa \in \mathcal{T}_h, \tag{2.28}$$

where $\mathcal{S}_\kappa \subset \mathcal{T}_h^{fine}$ denotes the set of fine elements which forms κ. Here, the agglomeration of fine tetrahedral elements is performed based on employing the *METIS* library for graph partitioning, cf., for example, [KK98, KK09]. With this definition each polyhedral element is typically non-convex. For simplicity, we have considered only the case of simply connected elements. In this



(a) Element κ_1. (b) Element κ_2. (c) Element κ_3.

Figure 2.16: Example of polyhedral elements $\kappa \in \mathcal{T}_h$ obtained by agglomeration of tetrahedra. κ_1 has 18 triangular *faces*, κ_2 has 20 triangular *faces* and κ_3 has 22 triangular *faces*.

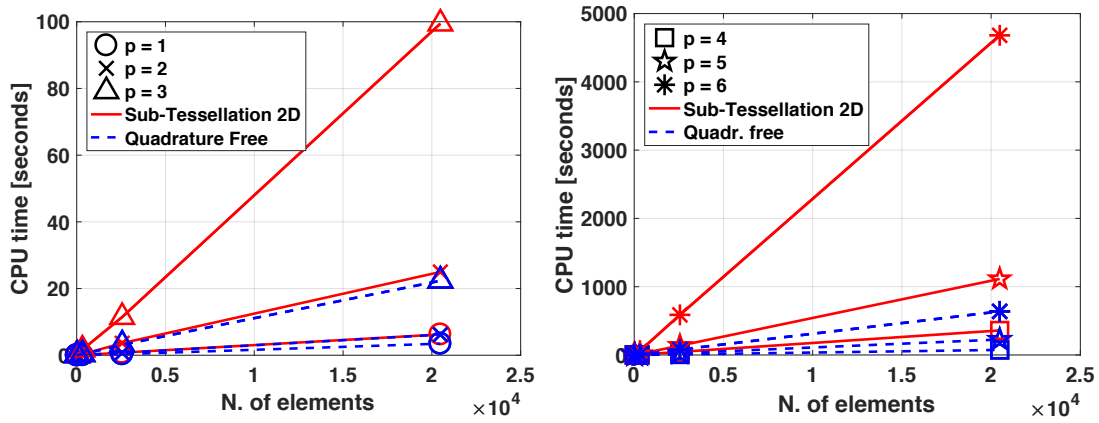


(a) Comparison for $p \in \{1,2,3\}$. (b) Comparison for $p \in \{4,5,6\}$.

Figure 2.17: Comparison of the CPU time needed to assemble the global matrices $\mathbf{M}$ and $\mathbf{V}$ for a three–dimensional problem by using the proposed quadrature free method and the classical sub-tessellation method. For each approach, each line is obtained by fixing the polynomial approximation degree $p \in \{1,2,3\}$ (left) and $p \in \{4,5,6\}$ (right), and measuring the CPU time by varying the number of elements of the underlying mesh.

particular case, the faces of the mesh $\mathcal{T}_h$ are the triangular intersections of two–dimensional facets of neighbouring elements. Fig. 2.16 shows three examples of the polyhedral elements resulting from agglomeration.

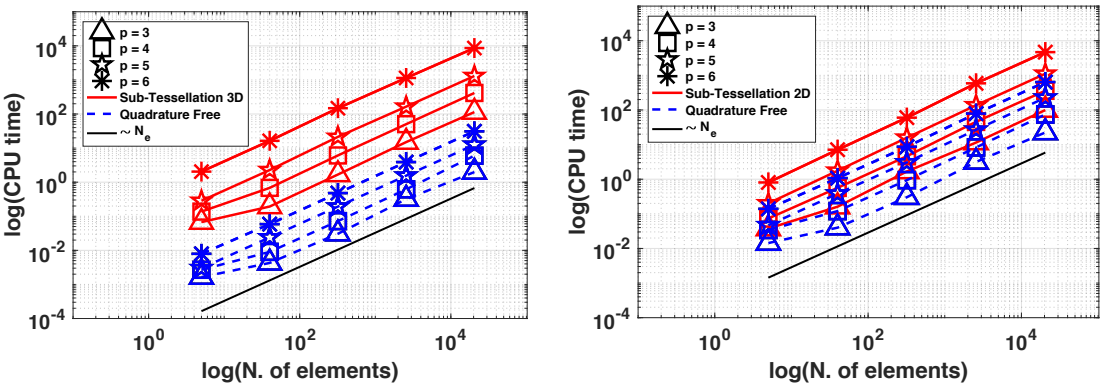
We perform a similar set of experiments as the ones outlined in Section 2.3.2 for the two–dimensional case. Again, we compare the CPU time required by the proposed quadrature free method with the quadrature integration/sub-tessellation approach to assemble the stiffness and mass matrices resulting from the DG discretization of problem (3.4). Numerical integration over a polyhedral domain is required to assemble the matrices $\mathbf{M}$ and $\mathbf{V}$, cf. (2.16), whereas for the



(a) Comparison for $p \in \{1,2,3\}$.

(b) Comparison for $p \in \{4,5,6\}$.

Figure 2.18: Comparison of the CPU time needed to assemble the global matrices $\mathbf{S}$ and $\mathbf{I}$ for a three–dimensional problem by using the proposed quadrature free method and the classical sub-tessellation method. For each approach, each line is obtained by fixing the polynomial approximation degree $p \in \{1,2,3\}$ (left) and $p \in \{4,5,6\}$ (right), and measuring the CPU time by varying the number of elements of the underlying mesh.



(a) CPU time comparison needed to assemble $\mathbf{M}$ and $\mathbf{V}$ in log-log scale.

(b) CPU time comparison needed to assemble $\mathbf{S}$ and $\mathbf{I}$ in log-log scale.

Figure 2.19: Comparison between the CPU time needed by the two method to assemble the global matrices $\mathbf{M}$ and $\mathbf{V}$ (left) and $\mathbf{S}$ and $\mathbf{I}$ (right) for a three–dimensional problem, versus the number of elements and for different choices of $p = 3,\dots,6$ (log-log scale).

computation of $\mathbf{S}$ and $\mathbf{I}$, cf. (2.17), a cubature rule over polygonal faces (here triangules shaped) is needed. In general, for three–dimensional problems the quadrature integration approach consists in the application of the sub-tessellation method both for volume and face integrals. Moreover, as in this case a sub-tessellation into tetrahedral domains is already given by the definition of the polyhedral mesh, the quadrature integration for volume integrals on a general agglomerated polyhedral element $\kappa = \cup_{\kappa'_f \in \mathscr{S}_\kappa} \kappa'_f$ is realized by applying an exact quadrature rule on each tetrahedron $\kappa'_f \in \mathscr{S}_\kappa$. The comparison of the CPU times for the two methods outlined here are presented for a set of agglomerated polyhedral grids where we vary the number of elements $N_e \in \{5, 40, 320, 2560, 20480\}$, and the polynomial degree $p \in \{1, 2, 3, 4, 5, 6\}$. For each agglomerated polyhedral grid $\mathscr{T}_h$ we have chosen the corresponding fine tetrahedral grid $\mathscr{T}_{fine}$ such that the cardinality of the set $\mathscr{S}_\kappa$ appearing in (2.28) is $|\mathscr{S}_\kappa| \sim 10 \ \forall \kappa \in \mathscr{T}_h$. The results are shown in Fig. 2.17 for the computation of the matrices $\mathbf{M}$ and $\mathbf{V}$, and in Fig. 2.18 for the computation of matrices $\mathbf{S}$ and $\mathbf{I}$. Here, we observe analogous behaviour to the two–dimensional case: the quadrature free method substantially outperforms quadrature integration both for the computation of the volume and face integrals. We also have reported in Fig. 2.19 the logarithmic-scaled graphs of each computation, showing that, as expected, the gain in terms of CPU time attained by exploiting the proposed method is more evident here, with respect to the two–dimensional case, also for the face integrals. With regards the three–dimensional tests presented in this section, we note that more substantial gains in terms of CPU time, with respect to classical approaches, can be obtained if the underlying grid is composed of pure (not agglomerated) polyhedral elements: firstly, this is because a sub-partition should be defined on the fly for each element, and secondly, as faces are not only triangles but possibly polygons of arbitrary shape, a sub-tessellation is needed also for surface integrals.

CHAPTER 3

V-CYCLE MULTIGRID ALGORITHMS ON NON-NESTED POLYTOPIC GRIDS

In this chapter we extend the W-cycle multigrid convergence analysis on nested polygonal/polyhedral grids of [AHH+17] to V-cycle algorithms with non-nested meshes. We focus on the solution of the linear systems of equations stemming from high-order DG discretizations of second-order elliptic partial differential equations on polytopic meshes. Here, the possibility of employing non-nested polytopic meshes allows us to choose the sequence of grids within the multigrid method based on employing agglomeration procedures together with edge-coarsening. The key aspect of our method is the *projection operator* which is defined as the L^2-projection between two consecutive (non-nested) partitions. By following the general framework introduced in [BPX91] for non-nested multigrid methods, we prove that our non-nested multigrid method converges uniformly with respect of the number of degree of freedom and the number of multigrid levels, provided that the number of smoothing steps is chosen sufficiently large. More precisely, we prove that the convergence rate is independent of the granularity of the underlying (fine) grid, the polynomial approximation degree p, and the number of levels, provided that the number of smoothing steps is chosen of order $p^{2+\mu}$, $\mu \in \{0,1\}$. We have also proposed a further improvement of the method by considering a Schwarz-type smoother. We demonstrate through several numerical experiments the effectiveness of the proposed algorithm, also for geometries with curved boundaries, where the coarse grid does not precisely fit the geometry.

The chapter is organized as follows. In Section 3.1 we recall the model problem and its DG discretization; furthermore, we introduce the ingredients needed for the forthcoming multigrid analysis. In Section 3.2 we define the multilevel BPX framework for the V-cycle multigrid solver based on employing non-nested grids, and present the convergence analysis of our algorithm. The main theoretical results are validated through a series of numerical experiments in Section 3.3.

In Section 3.4 we propose an improved version of the algorithm, obtained by enhancing the smoothing operator based on a domain decomposition preconditioner.

3.1 Model problem

We consider the weak formulation of the Poisson problem subject to homogeneous Dirichlet boundary conditions: find $u \in V = H^2(\Omega) \cap H_0^1(\Omega)$ such that

$$\mathscr{A}(u,v) = \int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}\mathbf{x} = \int_\Omega f v \, \mathrm{d}\mathbf{x} \qquad \forall v \in V. \tag{3.1}$$

In view of the forthcoming multigrid analysis, let $\{\mathscr{T}_j\}_{j=1}^J$ be a sequence of tessellations of the domain Ω, each of which is characterized by disjoint open polytopic elements κ of diameter h_κ, such that $\overline{\Omega} = \bigcup_{\kappa \in \mathscr{T}_j} \bar{\kappa}$, $j = 1, \ldots, J$. We also assume that each tessellation $\mathscr{T}_j$ satisfies the grid Assumptions 1.1– 1.4 of Section 1.1.1. Here, the mesh size of each grid $\mathscr{T}_j$ is denoted by $h_j = \max_{\kappa \in \mathscr{T}_j} h_\kappa$, $j = 1, \ldots, J$. To each $\mathscr{T}_j$ we associate the corresponding discontinuous finite element space V_j, defined as

$$V_j = \{v \in L^2(\Omega) : v|_\kappa \in \mathscr{P}_{p_j}(\kappa), \kappa \in \mathscr{T}_j\},$$

where $\mathscr{P}_{p_j}(\kappa)$ denotes the space of polynomials of total degree at most $p_j \geq 1$ on $\kappa \in \mathscr{T}_j$.

A suitable choice of $\{\mathscr{T}_j\}_{j=1}^J$ and $\{V_j\}_{j=1}^J$ leads to non-nested hp-multigrid schemes. This method is based on employing a set of non-nested polytopic partitions $\{\mathscr{T}_j\}_{j=1}^J$, such that the coarse level $\mathscr{T}_{j-1}$ is independent from $\mathscr{T}_j$, with the only constraint

$$h_{j-1} \lesssim h_j \leq h_{j-1} \qquad \forall\ j = 2, \ldots, J. \tag{3.2}$$

We also assume that the polynomial degree varies from one level to another such that

$$p_{j-1} \leq p_j \lesssim p_{j-1} \qquad \forall\ j = 2, \ldots, J. \tag{3.3}$$

With this notation, we choose $\mathscr{T}_J = \mathscr{T}_h$ and $V_J = V_h$, cf. Section 1.1, and we refer to them respectively as the grid and the discrete space of the *finest* level. Moreover, the bilinear form $\mathscr{A}_j(\cdot,\cdot) : V_j \times V_j \to \mathbb{R}$ corresponding to the symmetric interior penalty DG method on the j-th level is defined according to (1.5), i.e.,

$$\begin{aligned}\mathscr{A}_j(u,v) = &\sum_{\kappa \in \mathscr{T}_j} \int_\kappa \Big[\nabla u \cdot \nabla v \ + \mathscr{R}_j([\![u]\!]) \cdot \nabla v + \mathscr{R}_j([\![v]\!]) \cdot \nabla u\Big] \mathrm{d}\mathbf{x} \\ &+ \sum_{F \in \mathscr{F}_j} \int_F \sigma_j [\![u]\!] \cdot [\![v]\!] \, \mathrm{d}s,\end{aligned}$$

where $\mathscr{R}_j : [L^1(\mathscr{F}_j)]^d \to [V_j]^d$ is the lifting operator on the level j defined as in (1.6). $\sigma_j \in L^\infty(\mathscr{F}_j)$ denotes the interior penalty stabilization function, which is defined for the level j as in (1.7) with $\rho = 1$.

The goal of this chapter is to develop non-nested V-cycle multigrid schemes to solve the following problem posed on the finest level V_J: find $u_J \in V_J$ such that

$$\mathscr{A}_J(u_J, v_J) = \int_\Omega f v_J \, \mathrm{d}\mathbf{x} \quad \forall v_J \in V_J. \tag{3.4}$$

Remark 3.1. *For the rest of this chapter we denote the energy norm on the space V_j as $\|\cdot\|_j = \|\cdot\|_{h_j}$, $j = 1,\dots,J$. Moreover, for any $v \in V_j$, $j = 1,\dots,J$, we denote*

$$\|\nabla_j v\|^2_{L^2(\mathscr{T}_j)} = \sum_{\kappa \in \mathscr{T}_j} \|\nabla v\|^2_{L^2(\kappa)},$$
$$\|v\|^2_{L^2(\mathscr{F}_j)} = \sum_{F \in \mathscr{F}_j} \|v\|^2_{L^2(F)}$$

where ∇_j is the piecewise gradient operator on the space V_j, $j = 1,\dots,J$.

3.2 The BPX-framework for V-cycle multigrid algorithms

The analysis presented in this section is based on the general multigrid theoretical framework of [BPX91] for multigrid methods with non-nested spaces and non-inherited bilinear forms. In order to develop a geometric multigrid, the discretization at each level V_j follows the one already presented in [ASV15], where a W-cycle multigrid methods based on nested subspaces is considered.

First, we introduce the operators $A_j : V_j \to V_j$, defined as

$$(A_j w, v)_{L^2(\Omega)} = \mathscr{A}_j(w, v) \quad \forall w, v \in V_j, \quad j = 1,\dots,J, \tag{3.5}$$

and we denote by $\Lambda_j \in \mathbb{R}$ the maximum eigenvalue of A_j, $j = 2,\dots,J$. Moreover, let Id_j be the identity operator on the level V_j. The smoothing scheme, which is chosen to be the Richardson iteration, is given by

$$B_j = \Lambda_j \mathrm{Id}_j \quad j = 2,\dots,J.$$

The key ingredient in the construction of our proposed multigrid schemes is the inter-grid transfer operators. The prolongation operator connecting the coarser space V_{j-1} to the finer space V_j is denoted by I^j_{j-1}, $j = 2,\dots,J$. Since the two spaces are non-nested, i.e., $V_{j-1} \not\subset V_j$, it cannot be chosen as the natural injection operator. The most natural way to define the prolongation operator is the L^2-projection, i.e., $I^j_{j-1} : v_H \in V_{j-1} \longmapsto I^j_{j-1} v_H \in V_j$ is given by

$$(I^j_{j-1} v_H, w_h)_{L^2(\Omega)} = (v_H, w_h)_{L^2(\Omega)} \ \forall w_h \in V_j \quad j = 2,\dots,J. \tag{3.6}$$

The restriction operator $I^{j-1}_j : w_h \in V_j \longmapsto I^{j-1}_j w_h \in V_{j-1}$ is defined as the adjoint of I^j_{j-1} with respect to the $L^2(\Omega)$-inner product, i.e.,

$$(I^{j-1}_j w_h, v_H)_{L^2(\Omega)} = (w_h, I^j_{j-1} v_H)_{L^2(\Omega)} \ \ \forall v_H \in V_{j-1}.$$

Algorithm 3 Multigrid V-cycle iteration for the solution of problem (3.7)

Initialize $u_0 \in V_J$;
for $k = 0, 1, \dots$ **do**
  $u_{k+1} = \mathsf{MG}_{\mathcal{V}}(J, f_J, u_k, m_1, m_2)$;
  $u_k = u_{k+1}$;
end for

For our analysis, we also need to introduce the operator $P_j^{j-1} : V_j \to V_{j-1}$ given by

$$\mathscr{A}_{j-1}(P_j^{j-1} w_h, v_H) = \mathscr{A}_j(w_h, I_{j-1}^j v_H) \quad \forall v_H \in V_{j-1}, w_h \in V_j.$$

According to (3.5), problem (3.4) can be written in the following equivalent form: find $u_J \in V_J$ such that

$$A_J u_J = f_J, \tag{3.7}$$

where $f_J \in V_J$ is defined as $(f_J, v)_{L^2(\Omega)} = \int_\Omega f v \, dx \ \forall v \in V_J$. Given an initial guess $u_0 \in V_J$, and choosing the parameters $m_1, m_2 \in \mathbb{N}$, the multigrid V-cycle iteration algorithm for the approximation of u_J is outlined in Algorithm 3. In particular, $\mathsf{MG}_{\mathcal{V}}(J, f_J, u_k, m_1, m_2)$ represents one step of the iterative procedure to compute the approximate solution obtained after one iteration of our non-nested V-cycle scheme, which is defined by induction: if we consider the general problem of finding $z \in V_j$ such that

$$A_j z = g, \tag{3.8}$$

with $j \in \{2, \dots, J\}$ and $g \in L^2(\Omega)$, then $\mathsf{MG}_{\mathcal{V}}(j, g, z_0, m_1, m_2)$ represents the approximate solution of (3.8) obtained after one iteration of the non-nested V-cycle scheme with initial guess $z_0 \in V_j$ and m_1, m_2 pre-smoothing and post-smoothing steps, respectively. The recursive procedure is outlined in Algorithm 4, where we also observe that on level $j = 1$ the problem is solved exactly.

3.2.1 Abstract convergence analysis

We first define the following norms on each discrete space V_j

$$|\!|\!| v |\!|\!|_{s,j} = \sqrt{(A_j^s v, v)_{L^2(\Omega)}} \qquad \forall\, s \in \mathbb{R}, \quad \forall v \in V_j, \quad \forall j = 1, \dots, J.$$

Next, we introduce the following generalized Cauchy–Schwarz inequality, referring to [ASV15] for its proof.

Lemma 3.1. *For any $v, w \in V_j$ and $s \in \mathbb{R}$, it holds that*

$$\mathscr{A}_j(v, w) \le |\!|\!| v |\!|\!|_{1+s,j} |\!|\!| w |\!|\!|_{1-s,j}.$$

To analyze the convergence of Algorithm 4, for any $j = 2, \dots, J$, we set $G_j = \mathrm{Id}_j - B_j^{-1} A_j$ and define G_j^* as its adjoint with respect to $\mathscr{A}_j(\cdot, \cdot)$. Following [DGTZ07], we make three standard assumptions in order to prove convergence of Algorithm 3:

Algorithm 4 One iteration of the Multigrid V-cycle scheme on the level $j \geq 2$

if j=1 **then**

$\mathsf{MG}_{\mathcal{V}}(1, g, z_0, m_1, m_2) = A_1^{-1} g.$

else

Pre-smoothing:

for $i = 1, \ldots, m_1$ **do**

$z^{(i)} = z^{(i-1)} + B_j^{-1}(g - A_j z^{(i-1)});$

end for

Coarse grid correction:

$r_{j-1} = I_j^{j-1}(g - A_j z^{(m_1)});$

$e_{j-1} = \mathsf{MG}_{\mathcal{V}}(j-1, r_{j-1}, 0, m_1, m_2);$

$z^{(m_1+1)} = z^{(m_1)} + I_{j-1}^{j} e_{j-1};$

Post-smoothing:

for $i = m_1 + 2, \ldots, m_1 + m_2 + 1$ **do**

$z^{(i)} = z^{(i-1)} + B_j^{-1}(g - A_j z^{(i-1)});$

end for

$\mathsf{MG}_{\mathcal{V}}(j, g, z_0, m_1, m_2) = z^{(m_1+m_2+1)}.$

end if

A.1 Stability estimate: $\exists\, C_Q > 0$ such that

$$\|\!|(\mathrm{Id}_j - I_{j-1}^{j} P_j^{j-1}) v_h |\!\|_{1,j} \leq C_Q \|\!| v_h |\!\|_{1,j} \qquad \forall v_h \in V_j, \quad j = 2, \ldots, J.$$

A.2 Regularity-approximation property: $\exists\, C_A > 0$ such that

$$\left| \mathscr{A}_j((\mathrm{Id}_j - I_{j-1}^{j} P_j^{j-1}) v_h, v_h) \right| \leq C_A \frac{\|\!| v_h |\!\|_{2,j}^2}{\Lambda_j} \qquad \forall v_h \in V_j, \quad j = 2, \ldots, J.$$

A.3 Smoothing property: $\exists\, C_S > 0$ such that

$$\frac{\|v_h\|_{L^2(\Omega)}}{\Lambda_j} \leq C_S \left(\mathbb{S} v_h, v_h \right)_{L^2(\Omega)} \qquad \forall v_h \in V_j, \quad j = 2, \ldots, J,$$

where $\mathbb{S} = \left(\mathrm{Id}_j - G_j^* G_j\right) A_j^{-1}$.

The convergence analysis of the V-cycle method is stated in the following theorem which gives an estimate for the error propagation operator related to the j-th level iteration with m_1 and m_2 pre- and post-smoothing steps, respectively. The error propagation operator is defined as

$$\begin{cases} \mathbb{E}_{1,m_1,m_2} v & = 0, & j = 1, \\ \mathbb{E}_{j,m_1,m_2} v & = \mathbb{G}_{j,m_2}^* (\mathrm{Id}_j - I_{j-1}^{j} P_j^{j-1} + I_{j-1}^{j} \mathbb{E}_{j-1,m_1,m_2} P_j^{j-1}) \mathbb{G}_{j,m_1} v, & j > 1, \end{cases}$$

where $\mathbb{G}_{j,m} = (G_j)^m$ and $\mathbb{G}_{j,m}^* = (G_j^*)^m$, $m \geq 1$.

Theorem 3.1. *If Assumptions **A.1**, **A.2** and **A.3** hold and $m > 2C_AC_S$, then*

$$\left|\mathscr{A}_j(\mathbb{E}_{j,m,m}u,u)\right| \leq \delta_j \mathscr{A}_j(u,u) \qquad \forall u \in V_j, \quad j = 2,\dots,J,$$

where $\delta_j = \frac{C_AC_R}{m-C_AC_S} < 1$.

We refer to [DGTZ07] for the proof of Theorem 3.1 in an abstract setting. In the following, we prove the validity of Assumptions **A.1**, **A.2** and **A.3** for our algorithm. We start with a two-level approach, i.e., $J = 2$, and consider the two-level method for the solution of (3.4), based on two spaces $V_{J-1} \not\subset V_J$. The extension of the theory to the V-cycle method will be given at the end of this section.

3.2.2 Validity of Assumption A.1

In order to verify Assumption **A.1** for the two-level method we first show a stability result for the prolongation operator I^J_{J-1}. In the following, we also consider the L^2-projection operator on the space V_J defined as

$$\Pi^J_{L^2} : L^2(\Omega) \to V_J, \text{ such that } (\Pi^J_{L^2}w, v_J)_{L^2(\Omega)} = (w, v_J)_{L^2(\Omega)} \quad \forall v_J \in V_J.$$

Remark 3.2. *From the definition of I^J_{J-1} given in* (3.6)*, we note that $I^J_{J-1}(v_{J-1}) = \Pi^J_{L^2}(v_{J-1})$ for any discrete function $v_{J-1} \in V_{J-1} \subset L^2(\Omega)$.*

Moreover, we need the following approximation result which shows that any $v_j \in V_j$, $j = J-1, J$, can be approximated by an H^1-function, cf. [AHS16]. Let $\mathscr{G}_j : V_j \to [V_j]^d$ be the discrete gradient operator defined as

$$\mathscr{G}_j(v_h) = \nabla_j v + \mathscr{R}_j([\![v]\!]) \quad \forall v \in V_j, \ \forall\ j = 1,\dots,J,$$

We then consider the following problem: $\forall v_j \in V_j$, find $\mathscr{H}(v_j) \in H^1_0(\Omega)$ such that

$$\int_\Omega \nabla\mathscr{H}(v_j)\cdot\nabla w \ \mathrm{d}\mathbf{x} = \int_\Omega \mathscr{G}_j(v_j)\cdot\nabla w \ \mathrm{d}\mathbf{x} \quad \forall w \in H^1_0(\Omega). \tag{3.9}$$

It is shown in [AHS16] that $\mathscr{H}(\cdot)$ possesses good approximation properties in terms of providing an H^1-conforming approximant of the discontinuous function v_j, in particular the following result holds.

Theorem 3.2. *Let Ω be a bounded convex polygonal / polyhedral domain in $\mathbb{R}^d$, $d = 2,3$. Given $v_j \in V_j$, we write $\mathscr{H}(v_j) \in H^1_0(\Omega)$ to be the approximation defined in* (3.9)*. Then, the following bounds hold:*

$$\|v_j - \mathscr{H}(v_j)\|_{L^2(\Omega)} \lesssim \frac{h_j}{p_j}\|\sigma_j^{\frac{1}{2}}[\![v_j]\!]\|_{L^2(\mathscr{F}_h)}, \qquad |\mathscr{H}(v_j)|_{H^1(\Omega)} \lesssim \|v_j\|_j. \tag{3.10}$$

We make use of the previous result in order to show the following stability result of the prolongation operator.

Lemma 3.2. *There exists a positive constant* $\mathsf{C}_{\mathsf{stab}} = \mathsf{C}_{\mathsf{stab}}(p_J)$*, independent of the mesh size such that*

$$\|I_{J-1}^{J} v_H\|_J \leq \mathsf{C}_{\mathsf{stab}}(p_J)\, \|v_H\|_{J-1} \quad \forall v_H \in V_{J-1},$$

where $\mathsf{C}_{\mathsf{stab}}(p_J) = \mathcal{O}(p_J)$*.*

Proof. Let $v_H \in V_{J-1}$, then by the definition of the DG norm (1.12), we need to estimate:

$$\|I_{J-1}^{J} v_H\|_J^2 = \underbrace{\sum_{\kappa \in \mathcal{T}_J} \|\nabla (I_{J-1}^{J} v_H)\|_{L^2(\kappa)}^2}_{\|\nabla_J (I_{J-1}^{J} v_H)\|_{L^2(\mathcal{T}_J)}^2} + \underbrace{\sum_{F \in \mathcal{F}_J} \|\sigma_J^{\frac{1}{2}} [\![I_{J-1}^{J} v_H]\!]\|_{L^2(F)}^2}_{\|\sigma_J^{\frac{1}{2}} [\![I_{J-1}^{J} v_H]\!]\|_{L^2(\mathcal{F}_J)}^2}, \tag{3.11}$$

where $\|\nabla_J(\cdot)\|_{L^2(\mathcal{T}_J)}$ and $\|\cdot\|_{L^2(\mathcal{F}_J)}$ are defined in Remark 3.1. We next bound each of the two terms on the right hand side of (3.11) separately. For the first term, we let $\widetilde{v_H} = \mathscr{H}(v_H) \in H_0^1(\Omega)$ be the conforming approximant of v_H defined as in (3.9). Then,

$$\begin{aligned} \|\nabla_J (I_{J-1}^{J} v_H)\|_{L^2(\mathcal{T}_J)}^2 \leq\ & \|\nabla_J (I_{J-1}^{J} v_H - \Pi_J(\widetilde{v_H}))\|_{L^2(\mathcal{T}_J)}^2 \\ & + \|\nabla_J (\widetilde{v_H} - \Pi_J(\widetilde{v_H}))\|_{L^2(\mathcal{T}_J)}^2 + |\widetilde{v_H}|_{H^1(\Omega)}^2, \end{aligned} \tag{3.12}$$

where we have added and subtracted the terms $\nabla_J(\Pi_J(\widetilde{v_H}))$ and $\nabla \widetilde{v_H}$. Here, Π_J is the approximant operator on the space V_J, defined as in Lemma 1.6. The second term in the right hand side of (3.12) can be estimated using the interpolation bounds of Lemma 1.6, the Poincaré inequality for $\widetilde{v_H} \in H_0^1(\Omega)$ and the second bound of (3.10); thereby, we get

$$\|\nabla_J (\widetilde{v_H} - \Pi_J(\widetilde{v_H}))\|_{L^2(\mathcal{T}_J)}^2 \lesssim \|\widetilde{v_H}\|_{H^1(\Omega)}^2 \lesssim \|v_H\|_{J-1}^2.$$

In order to estimate the first term on the right hand side in (3.12) we observe that, since $I_{J-1}^{J} v_H - \Pi_J(\widetilde{v_H}) \in V_J$, it is possible to make use of the inverse inequality stated in Lemma 4.5; this leads to the following bound:

$$\|\nabla_J (I_{J-1}^{J} v_H - \Pi_J(\widetilde{v_H}))\|_{L^2(\mathcal{T}_J)}^2 \lesssim p_J^4 h_J^{-2} \|I_{J-1}^{J} v_H - \Pi_J(\widetilde{v_H})\|_{L^2(\Omega)}^2. \tag{3.13}$$

By adding and subtracting $\widetilde{v_H}$ to $\|I_{J-1}^{J} v_H - \Pi_J(\widetilde{v_H})\|_{L^2(\Omega)}^2$ we obtain

$$\|I_{J-1}^{J} v_H - \Pi_J(\widetilde{v_H})\|_{L^2(\Omega)}^2 \lesssim \|I_{J-1}^{J} v_H - \widetilde{v_H}\|_{L^2(\Omega)}^2 + \|\widetilde{v_H} - \Pi_J(\widetilde{v_H})\|_{L^2(\Omega)}^2. \tag{3.14}$$

Using Lemma 1.6 and the Poincaré inequality (since $\widetilde{v_H} \in H_0^1(\Omega)$) we have

$$\|\widetilde{v_H} - \Pi_J(\widetilde{v_H})\|_{L^2(\Omega)}^2 \lesssim \frac{h_J^2}{p_J^2} \|\widetilde{v_H}\|_{H^1(\Omega)}^2 \lesssim \frac{h_J^2}{p_J^2} \|v_H\|_{J-1}^2.$$

The term $\|I_{J-1}^{J} v_H - \widetilde{v_H}\|_{L^2(\Omega)}^2$ can be estimated as follows

$$\|I_{J-1}^{J} v_H - \widetilde{v_H}\|_{L^2(\Omega)}^2 \lesssim \|I_{J-1}^{J} v_H - \Pi_{L^2}^{J}(\widetilde{v_H})\|_{L^2(\Omega)}^2 + \|\widetilde{v_H} - \boldsymbol{Q}_J(\widetilde{v_H})\|_{L^2(\Omega)}^2.$$

From Remark 3.2, the continuity of $\Pi^J_{L^2}$ with respect to the L^2-norm, Lemma 1.6, the bound (3.10) and the Poincaré inequality we have

$$\begin{aligned}\|I^J_{J-1}v_H - \widetilde{v_H}\|^2_{L^2(\Omega)} &\lesssim \|\Pi^J_{L^2}(v_H - \widetilde{v_H})\|^2_{L^2(\Omega)} + \|\widetilde{v_H} - \boldsymbol{Q}_J(\widetilde{v_H})\|^2_{L^2(\Omega)} \\ &\lesssim \|v_H - \widetilde{v_H}\|^2_{L^2(\Omega)} + \|\widetilde{v_H} - \Pi_J(\widetilde{v_H})\|^2_{L^2(\Omega)} \\ &\lesssim \frac{h_J^2}{p_J^2}\|\sigma_J^{\frac{1}{2}}[\![v_H]\!]\|^2_{L^2(\mathscr{F}_J)} + \frac{h_J^2}{p_J^2}\|\widetilde{v_H}\|^2_{H^1(\Omega)} \\ &\lesssim \frac{h_J^2}{p_J^2}\|v_H\|^2_{J-1}.\end{aligned}$$

From the previous estimates and inequality (3.14), we obtain

$$\|I^J_{J-1}v_H - \Pi_J(\widetilde{v_H})\|^2_{L^2(\Omega)} \lesssim \frac{h_J^2}{p_J^2}\|v_H\|^2_{J-1}. \tag{3.15}$$

The above estimate, together with (3.13), (3.12) and the second bound of (3.10) lead to

$$\|\nabla_J(I^J_{J-1}v_H)\|^2_{L^2(\mathscr{T}_J)} \lesssim p_J^2\ \|v_H\|^2_{J-1}. \tag{3.16}$$

Next we bound the second term on the right hand side in (3.11). By the definition of the jump term and remembering that $[\![\widetilde{v_H}]\!] = 0\ \forall F \in \mathscr{F}_J$ since $\widetilde{v_H} \in H^1_0(\Omega)$, we get

$$\|\sigma_J^{\frac{1}{2}}[\![I^J_{J-1}v_H]\!]\|^2_{L^2(\mathscr{F}_J)} \lesssim \frac{p_J^2}{h_J}\sum_{\kappa\in\mathscr{T}_J}\Big(\|I^J_{J-1}v_H - \Pi_J(\widetilde{v_H})\|^2_{L^2(\partial\kappa)} + \|\Pi_J(\widetilde{v_H}) - \widetilde{v_H}\|^2_{L^2(\partial\kappa)}\Big), \tag{3.17}$$

where we also used the definition of σ_J. Now, we first observe that we can use the trace inequality of Lemma 1.3 in order to obtain

$$\|I^J_{J-1}v_H - \Pi_J(\widetilde{v_H})\|^2_{L^2(\partial\kappa)} \lesssim \frac{p_J^2}{h_J}\|I^J_{J-1}v_H - \Pi_J(\widetilde{v_H})\|^2_{L^2(\kappa)}. \tag{3.18}$$

To bound the second term on the right hand side in (3.17), we first exploit the continuous trace inequality on polygons of Lemma 1.1 with $\epsilon = p_J$; thereby, we get

$$\|\Pi_J(\widetilde{v_H}) - \widetilde{v_H}\|^2_{L^2(\partial\kappa)} \lesssim \frac{p_J}{h_J}\|\Pi_J(\widetilde{v_H}) - \widetilde{v_H}\|^2_{L^2(\kappa)} + \frac{h_J}{p_J}|\Pi_J(\widetilde{v_H}) - \widetilde{v_H}|^2_{H^1(\kappa)}.$$

Then, by summing over $\kappa \in \mathscr{T}_J$, using the approximation property of Lemma 1.6 and the Poincaré inequality, we obtain

$$\begin{aligned}\sum_{\kappa\in\mathscr{T}_J}\|\Pi_J(\widetilde{v_H}) - \widetilde{v_H}\|^2_{L^2(\partial\kappa)} &\lesssim \frac{p_J}{h_J}\frac{h_J^2}{p_J^2}\|\widetilde{v_H}\|^2_{H^1(\Omega)} + \frac{h_J}{p_J}\|\widetilde{v_H}\|^2_{H^1(\Omega)} \\ &\lesssim \frac{h_J}{p_J}|\widetilde{v_H}|^2_{H^1(\Omega)}.\end{aligned}$$

From the previous inequality and the bound (3.18), (3.17) becomes:

$$\begin{aligned}\|\sigma_J^{\frac{1}{2}}[\![I_{J-1}^J v_H]\!]\|_{L^2(\mathscr{F}_J)}^2 &\lesssim \frac{p_J^4}{h_J^2}\|I_{J-1}^J v_H - \Pi_J(\widetilde{v_H})\|_{L^2(\Omega)}^2 + p_J|\widetilde{v_H}|_{H^1(\Omega)}^2 \\ &\lesssim p_J^2\|v_H\|_{J-1}^2,\end{aligned}$$

where we also used inequality (3.15). This estimate together with (3.16) leads to

$$\|I_{J-1}^J v_H\|_J \le \mathsf{C}_{\mathsf{stab}}(p_J)\ \|v_H\|_{J-1} \quad \forall v_H \in V_{J-1}.$$

where $\mathsf{C}_{\mathsf{stab}}(p_J) = \mathscr{O}(p_J)$. □

We can use the previous result in order to prove that Assumption **A.1** holds. We first observe that also the operator P_J^{J-1} satisfies a similar stability estimate as the one of I_{J-1}^J, that is

$$\begin{aligned}\|P_J^{J-1}v_h\|_{DG,J-1}^2 &\lesssim \mathscr{A}_{J-1}(P_J^{J-1}v_h, P_J^{J-1}v_h) = \mathscr{A}_J(v_h, I_{J-1}^J P_J^{J-1}v_h) \\ &\lesssim \|v_h\|_J\|I_{J-1}^J P_J^{J-1}v_h\|_J \lesssim \mathsf{C}_{\mathsf{stab}}(p_J)\ \|v_h\|_J\|P_J^{J-1}v_h\|_J,\end{aligned}$$

from which it follows

$$\|P_J^{J-1}v_h\|_{J-1} \lesssim \mathsf{C}_{\mathsf{stab}}(p_J)\ \|v_h\|_J.$$

Proposition 3.1. *Assumption **A.1** holds with $C_Q \lesssim p_J^2$.*

Proof. Let $v_H \in V_{J-1}$, employing Lemma 1.4 we have

$$\mathscr{A}_J(I_{J-1}^J v_H, I_{J-1}^J v_H) \lesssim \|I_{J-1}^J v_H\|_J^2 \lesssim p_J^2\ \|v_H\|_{J-1}^2 \lesssim p_J^2\ \mathscr{A}_{J-1}(v_H, v_H).$$

Similarly, it holds

$$\mathscr{A}_{J-1}(P_J^{J-1}v_h, P_J^{J-1}v_h) \lesssim p_J^2\ \mathscr{A}_J(v_h, v_h) \quad \forall v_h \in V_J. \tag{3.19}$$

Let $v_h \in V_J$ and set $v_H = P_J^{J-1}v_h$, then the following inequality holds:

$$\mathscr{A}_J(I_{J-1}^J P_J^{J-1}v_h, I_{J-1}^J P_J^{J-1}v_h) \lesssim p_J^2\ \mathscr{A}_{J-1}(P_J^{J-1}v_h, P_J^{J-1}v_h). \tag{3.20}$$

By adding and subtracting v_h to both arguments of $\mathscr{A}_J(\cdot,\cdot)$ on the left hand side of (3.20), and using (3.19) we obtain

$$\underbrace{\mathscr{A}_J((\mathrm{Id}_J - I_{J-1}^J P_J^{J-1})v_h, (\mathrm{Id}_J - I_{J-1}^J P_J^{J-1})v_h)}_{=\|(\mathrm{Id}_J - I_{J-1}^J P_J^{J-1})v_h\|_{1,J}^2} \lesssim \underbrace{\left(p_J^2\left(p_J^2 - 2\right) + 1\right)}_{\le p_J^4}\mathscr{A}_J(v_h, v_h),$$

which concludes the proof. □

3.2.3 Validity of Assumption A.2

We follow the analysis presented in [DGTZ07] in order to show the validity of Assumption **A.2**. We first show two preliminary results making use of the properties in Section 1.2.

Lemma 3.3. *Let Assumptions 1.1 - 1.4 hold and assumptions* (3.2) *and* (3.3) *hold. Let Π_j be the projection operator on V_j as defined in Lemma 1.6, for $j = J, J-1$. Then*

$$\|\Pi_J w - I_{J-1}^J \Pi_{J-1} w\|_{L^2(\Omega)} \lesssim \frac{h_J^2}{p_J^2} \|w\|_{H^2(\Omega)} \quad \forall w \in H^2(\Omega).$$

Proof. Using the triangle inequality, Remark 3.2 and the approximation estimates of Lemma 1.6 we have:

$$\begin{aligned}
\|\Pi_J w - I_{J-1}^J \Pi_{J-1} w\|_{L^2(\Omega)} &\leq \|\Pi_J w - w\|_{L^2(\Omega)} + \|w - \Pi_{L^2}^J w\|_{L^2(\Omega)} + \|\Pi_{L^2}^J w - I_{J-1}^J \Pi_{J-1} w\|_{L^2(\Omega)} \\
&= \|\Pi_J w - w\|_{L^2(\Omega)} + \min_{z_h \in V_J} \|w - z_h\|_{L^2(\Omega)} + \|\Pi_{L^2}^J (w - \Pi_{J-1} w)\|_{L^2(\Omega)} \\
&\leq \|\Pi_J w - w\|_{L^2(\Omega)} + \| w - \Pi_J w\|_{L^2(\Omega)} + \|w - \Pi_{J-1} w\|_{L^2(\Omega)} \\
&\lesssim \frac{h_J^2}{p_J^2} \|w\|_{H^2(\Omega)} + \frac{h_{J-1}^2}{p_{J-1}^2} \|w\|_{H^2(\Omega)} \lesssim \frac{h_J^2}{p_J^2} \|w\|_{H^2(\Omega)},
\end{aligned}$$

where in the last inequality we have used hypotheses (3.2) and (3.3). □

Lemma 3.4. *Let Assumptions 1.1 - 1.4 hold. Given $g \in L^2(\Omega)$, denote by $w_j \in V_j$ the solution of $\mathscr{A}_j(w_j, v) = (g, v)_{L^2(\Omega)}$ $\forall v \in V_j$ with $j = J-1, J$. Then the following inequality holds:*

$$\|w_J - I_{J-1}^J w_{J-1}\|_{L^2(\Omega)} + \|w_{J-1} - P_J^{J-1} w_J\|_{L^2(\Omega)} \lesssim \frac{h_J^2}{p_J^{2-\mu}} \|g\|_{L^2(\Omega)}.$$

Proof. Consider the unique solution $w \in V$ of the problem

$$\mathscr{A}(w, v) = (g, v)_{L^2(\Omega)} \quad \forall v \in V.$$

Using Corollary 1.2, we have

$$\|w - w_j\|_{L^2(\Omega)} \lesssim \frac{h_j^2}{p_j^{2-\mu}} \|w\|_{H^2(\Omega)}, \quad j = J-1, J. \tag{3.21}$$

Using the triangle inequality and Remark 3.2 we have:

$$\begin{aligned}
\|w_J - I_{J-1}^J w_{J-1}\|_{L^2(\Omega)} \leq{}& \|w_J - w\|_{L^2(\Omega)} + \|w - \Pi_J w\|_{L^2(\Omega)} \\
&+ \|\Pi_J w - I_{J-1}^J \Pi_{J-1} w\|_{L^2(\Omega)} + \|I_{J-1}^J \Pi_{J-1} w - \Pi_{L^2}^J w\|_{L^2(\Omega)} \\
&+ \|\Pi_{L^2}^J w - I_{J-1}^J w_{J-1}\|_{L^2(\Omega)} \\
={}& \|w_J - w\|_{L^2(\Omega)} + \|w - \Pi_J w\|_{L^2(\Omega)} + \|\Pi_J w - I_{J-1}^J \Pi_{J-1} w\|_{L^2(\Omega)} \\
&+ \|\Pi_{L^2}^J (\Pi_{J-1} w - w)\|_{L^2(\Omega)} + \|\Pi_{L^2}^J (w - w_{J-1})\|_{L^2(\Omega)} \\
\leq{}& \|w_J - w\|_{L^2(\Omega)} + \|w - \Pi_J w\|_{L^2(\Omega)} + \|\Pi_J w - I_{J-1}^J \Pi_{J-1} w\|_{L^2(\Omega)} \\
&+ \|\Pi_{J-1} w - w\|_{L^2(\Omega)} + \|w - w_{J-1}\|_{L^2(\Omega)}.
\end{aligned}$$

Using (3.21), Lemma 1.6 and Lemma 3.3, gives

$$\|w_J - I_{J-1}^J w_{J-1}\|_{L^2(\Omega)} \lesssim \frac{h_J^2}{p_J^{2-\mu}}\|w\|_{H^2(\Omega)} + \frac{h_J^2}{p_J^2}\|w\|_{H^2(\Omega)} + \frac{h_J^2}{p_J^2}\|w\|_{H^2(\Omega)} + \frac{h_{J-1}^2}{p_{J-1}^2}\|w\|_{H^2(\Omega)} + \frac{h_{J-1}^2}{p_{J-1}^{2-\mu}}\|w\|_{H^2(\Omega)}.$$

From the elliptic regularity assumption (1.2) and hypotheses (3.2) and (3.3), we can write

$$\|w_J - I_{J-1}^J w_{J-1}\|_{L^2(\Omega)} \lesssim \frac{h_J^2}{p_J^{2-\mu}}\|g\|_{L^2(\Omega)}. \tag{3.22}$$

Now, let $z_j \in V_j$ be the solution of:

$$\mathscr{A}_j(z_j, q) = (w_{J-1} - P_J^{J-1} w_J, \Pi_{L^2}^J)_{L^2(\Omega)} \quad \forall \Pi_{L^2}^J \in V_j, \quad j = J-1, J.$$

Using (3.22) we get the following estimate:

$$\|z_{J-1} - I_{J-1}^J z_{J-1}\|_{L^2(\Omega)} \lesssim \frac{h_J^2}{p_J^{2-\mu}}\|w_{J-1} - P_J^{J-1} w_J\|_{L^2(\Omega)}.$$

Thereby, we have:

$$\begin{aligned}
\|w_{J-1} - P_J^{J-1} w_J\|_{L^2(\Omega)}^2 &= \mathscr{A}_{J-1}(z_{J-1}, w_{J-1} - P_J^{J-1} w_J) \\
&= \mathscr{A}_{J-1}(z_{J-1}, w_{J-1}) - \mathscr{A}_J(I_{J-1}^J z_{J-1}, w_J) \\
&= (z_{J-1}, g) - (I_{J-1}^J z_{J-1}, g) = (g, z_{J-1} - I_{J-1}^J z_{J-1}) \\
&\lesssim \|g\|_{L^2(\Omega)} \frac{h_J^2}{p_J^{2-\mu}}\|w_{J-1} - P_J^{J-1} w_J\|_{L^2(\Omega)},
\end{aligned}$$

from which, together with (3.22), the thesis follows. □

We are now ready to show the following standard approximation result, needed to prove the validity of Assumption **A.2**.

Lemma 3.5. *Let Assumptions 1.1 - 1.4 hold. Then,*

$$\|(\mathrm{Id}_J - I_{J-1}^J P_J^{J-1}) v_J\|_{L^2(\Omega)} \lesssim \frac{h_J^2}{p_J^{2-\mu}}|||v_J|||_{2,J} \quad \forall v_J \in V_J.$$

Proof. For any $v_J \in V_J$ we have

$$\|(\mathrm{Id}_J - I_{J-1}^J P_J^{J-1}) v_J\|_{L^2(\Omega)} = \sup_{0 \neq \phi \in L^2(\Omega)} \frac{\left(\phi, (\mathrm{Id}_J - I_{J-1}^J P_J^{J-1}) v_J\right)_{L^2(\Omega)}}{\|\phi\|_{L^2(\Omega)}}. \tag{3.23}$$

Next, consider the solution z_j of the following problems

$$\mathscr{A}_j(z_j, v_j) = \left(\phi, v_j\right) \quad \forall v_j \in V_j, \text{ for } j = J, J-1.$$

Using the definition of P_J^{J-1}, Lemma 3.4 and Lemma 3.1 we have:

$$\begin{aligned}\left(\phi,(\mathrm{Id}_J - I_{J-1}^J P_J^{J-1})v_J\right)_{L^2(\Omega)} &= \mathscr{A}_J(z_J,v_J) - \mathscr{A}_{J-1}(P_J^{J-1}z_J,P_J^{J-1}v_J)\\ &= \mathscr{A}_J(z_J - I_{J-1}^J z_{J-1},v_J) + \mathscr{A}_J(I_{J-1}^J(z_{J-1}-P_J^{J-1}z_J),v_J)\\ &\le |||v_J|||_{2,J}\Big(\|z_J - I_{J-1}^J z_{J-1}\|_{L^2(\Omega)} + \|z_{J-1}-P_J^{J-1}z_J\|_{L^2(\Omega)}\Big)\\ &\lesssim |||v_J|||_{2,J}\frac{h_J^2}{p_J^{2-\mu}}\|\phi\|_{L^2(\Omega)}.\end{aligned}$$

Using the last inequality together with (3.23) we get the thesis. □

Exploiting Lemma 3.5,we now deduce the following result.

Proposition 3.2. *The regularity-approximation property **A.2** holds with $C_A \lesssim p_J^{2+\mu}$, $\mu = 0,1$.*

Proof. Theorem 1.1 gives the following bound of the maximum eigenvalue of A_J: $\Lambda_J \lesssim \frac{p_J^4}{h_J^2}$. Using Lemma 3.5, the above bound on Λ_J, and the symmetry of $\mathscr{A}_J(\cdot,\cdot)$ we have, for all $v \in V_J$:

$$\begin{aligned}\mathscr{A}_J((\mathrm{Id}_J - I_{J-1}^J P_J^{J-1})v,v) \le |||v|||_{2,J}|||(\mathrm{Id}_J - I_{J-1}^J P_J^{J-1})v|||_{0,J} &\lesssim \frac{h_J^2}{p_J^{2-\mu}}|||v|||_{2,J}^2\\ &\lesssim p_J^{2+\mu}\frac{|||v|||_{2,J}^2}{\Lambda_J},\end{aligned}$$

and the proof is complete. □

3.2.4 Validity of Assumption A.3

Proposition 3.3. *Assumption **A.3** holds with $C_S = \mathcal{O}(1)$.*

Proof. We have:

$$\mathbb{S} = \left(\mathrm{Id}_J - G_J^* G_J\right)A_J^{-1} = \Big(\frac{2}{\Lambda_J}A_J - \frac{1}{\Lambda_J^2}A_J A_J\Big)A_J^{-1} = \frac{1}{\Lambda_J}\Big(\mathrm{Id}_J + \Big(\mathrm{Id}_J - \frac{1}{\Lambda_J}A_J\Big)\Big),$$

and so

$$\left(\mathbb{S}u,u\right)_{L^2(\Omega)} = \frac{\|u_h\|_{L^2(\Omega)}}{\Lambda_J} + \Big(\Big(\mathrm{Id}_J - \frac{1}{\Lambda_J}A_J\Big)u,u\Big)_{L^2(\Omega)}.$$

We now prove that $\left(\mathrm{Id}_J - \frac{1}{\Lambda_J}A_J\right)$ is a positive definite operator. By contradiction, let us suppose that there exists a function $\overline{u} \in V_J$, $\overline{u} \neq 0$, such that

$$\Lambda_J(\overline{u},\overline{u})_{L^2(\Omega)} < \mathscr{A}_J(\overline{u},\overline{u}). \tag{3.24}$$

By Lemma 1.4 and the symmetry of the bilinear form $\mathscr{A}_J(\cdot,\cdot)$, the eigenfunctions $\{\phi_k^J\}_{k=1}^{N_J}$ satisfy

$$\mathscr{A}_J(\phi_k^J,v) = \lambda_k^J(\phi_k^J,v)_{L^2(\Omega)} \qquad \forall v \in V_J,$$

where $0 < \lambda_1^J \leq \lambda_2^J \leq \cdots \leq \lambda_{N_J}^J = \Lambda_J$. The set of eigenfunctions is an orthonormal basis for the space V_J, i.e. $(\phi_i^J, \phi_j^J)_{L^2(\Omega)} = \delta_{ij}$, where δ_{ij} is the Kronecker symbol, and satisfies $\mathscr{A}_J(\phi_i^J, \phi_j^J) = \lambda_i^J \delta_{ij}$. Since $\{\phi_k^J\}_{k=1}^{N_J}$ is a basis of the space V_J, we can write $\overline{u} = \sum_{k=1}^{N_J} c_k \phi_k^J$, so that (3.24) becomes

$$\Lambda_J \sum_{i,j=1}^{N_J} c_j (\phi_j^J, \phi_i^J)_{L^2(\Omega)} c_i < \sum_{i,j=1}^{N_J} c_j \mathscr{A}_J(\phi_j^J, \phi_i^J) c_i = \sum_{i,j=1}^{N_J} c_j \lambda_i^J (\phi_i^J, \phi_j^J)_{L^2(\Omega)} c_i,$$

$$\Rightarrow \quad \Lambda_J \sum_{i=1}^{N_J} c_i^2 < \sum_{i,j=1}^{N_J} c_i^2 \lambda_i^J,$$

which is a contradiction. We then deduce that $\left(\mathrm{Id}_J - \frac{1}{\Lambda_J} A_J\right)$ is a positive definite operator. □

Remark 3.3. *We observe that, as we need to satisfy the condition $m > 2C_A C_S$ of Theorem 3.1, we can guarantee the convergence of the method based on employing a number of smoothing steps such that $m \gtrsim p_J^{2+\mu}$, which is in agreement with the corresponding result derived for the W-cycle algorithms in [ASV15] and [AHH+17] in the case of nested grids.*

Remark 3.4. *The analysis of this section can be generalized to the full V-cycle algorithm with $J > 2$ as follows: Assumption **A.3** is verified with $C_S = \mathcal{O}(1)$ also on the arbitrary levels $j, j-1$, because each level j satisfies Assumption **A.3** with constant $C_S^j = \mathcal{O}(1)$. Assumptions **A.2** and **A.1** are satisfied with $C_A = \max_j\{C_A^j\}$ and $C_Q = \max_j\{C_Q^j\}$, respectively, where C_A^j and C_Q^j are the same as the ones defined in the previous analysis but on the level j.*

3.3 Numerical results

In this section we present several numerical results to test the theoretical convergence estimates provided in Theorem 3.1 and to demonstrate the capability of our algorithm in practical cases. We focus on a two dimensional Poisson problem posed on the unit square $\Omega = (0,1)^2$. For the simulations, we consider the sets of polygonal grids shown in Figure 3.1. Each polygonal mesh is generated by using the software package `PolyMesher` [TPPM12]. In particular the finest grids (Level 4) of Figure 3.1 consist of 512 (Set 1), 1024 (Set 2), 2048 (Set 3) and 4096 (Set 4) elements. Starting from the number of elements of each initial mesh, a sequence of non-nested partitions is generated: each coarse mesh is built independently from the finer one, with the only constrain that the number of element is approximately 1/4 of the corresponding finer one.

First of all, we verify the estimate of Lemma 3.2, by numerically evaluating $\mathsf{C}_{\mathsf{stab}}(p)$, where p is the polynomial approximation degree. To this end we consider three pairs of non-nested grids, where the number of elements of the coarser grid is equal to the number in the finer mesh divided by 4: for each pair, we compute the value of $\mathsf{C}_{\mathsf{stab}}(p)$ as a function of p. Figure 3.2 shows that, as expected, $\mathsf{C}_{\mathsf{stab}}(p)$ depends linearly on p and is independent of the mesh-size h.

Remark 3.5. *From the implementation point of view, we point out that the assembly of the prolongation and projection matrices requires the knowledge of the intersections between elements of two consecutive levels. To this end, our computations make use of the tool PolygonClipper [Hol].*

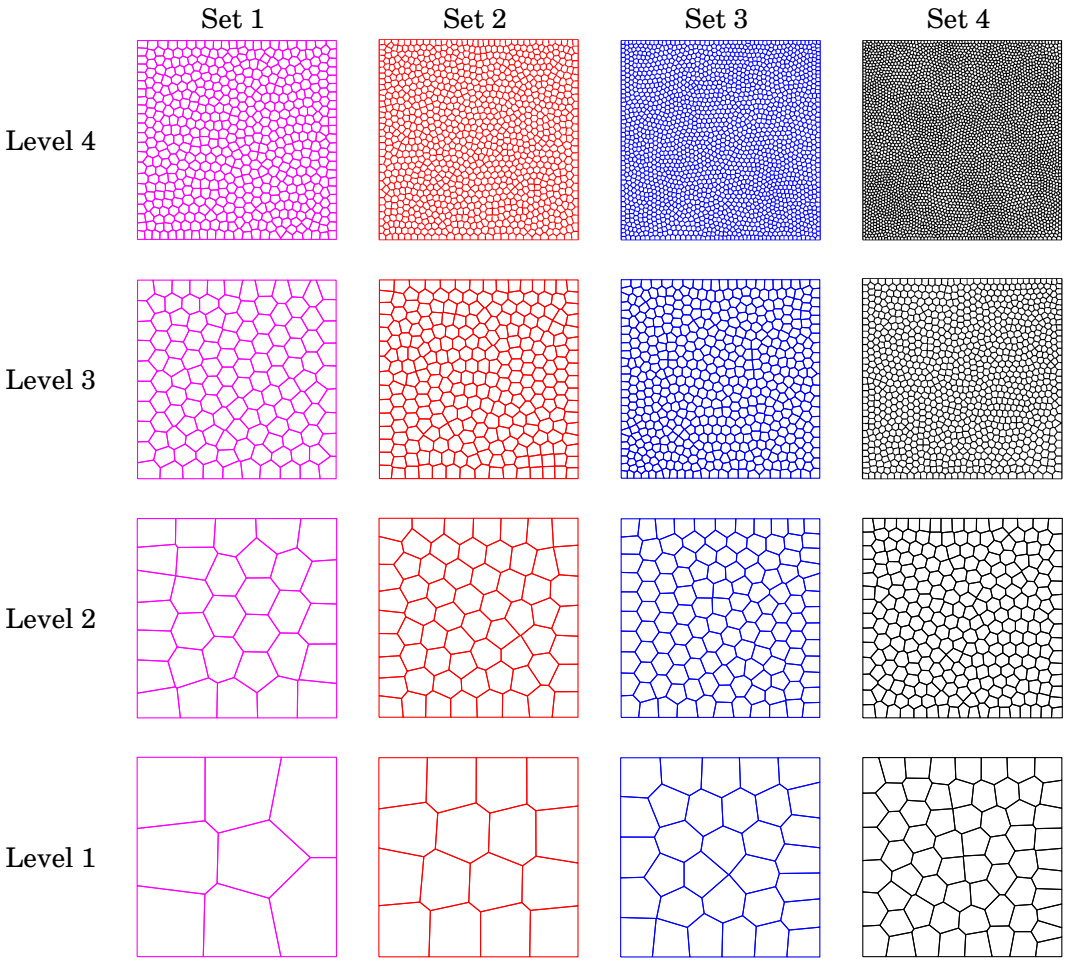


Figure 3.1: Sets of non-nested grids employed for numerical simulations.

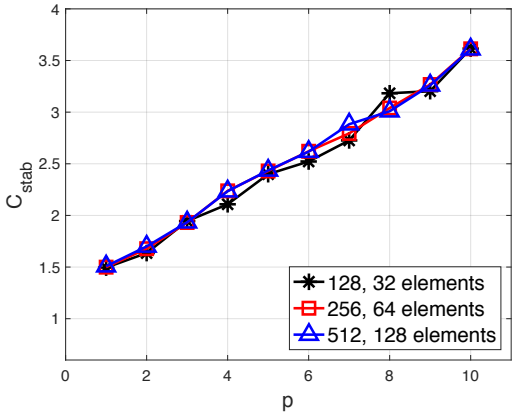


Figure 3.2: Estimates of $C_{stab}(p)$ in Lemma 3.2 as a function of p for three pairs of non-nested Voronoi meshes as shown in Figure 3.1.

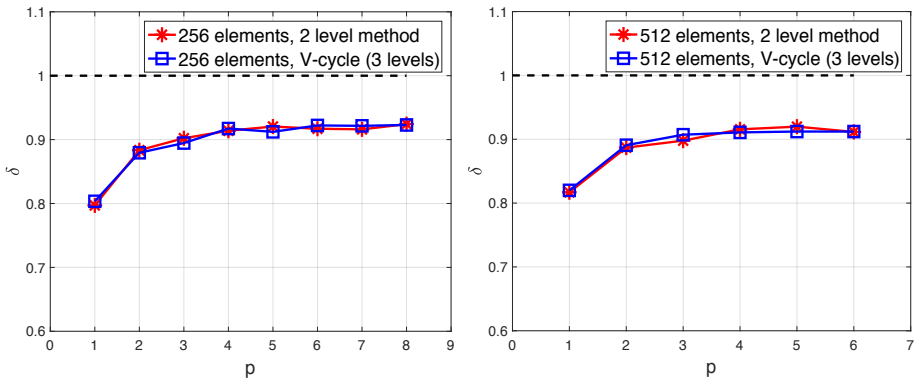


Figure 3.3: Estimates of δ_2 and δ_3 in Theorem 3.1 as a function of p, with $m_1 = m_2 = 3p^2$ and two polygonal grids of 256 (left) and 512 (right) elements.

We now consider the grids shown in Set 1 and Set 2 of Figure 3.1 and numerically evaluate the constant δ_j in Theorem 3.1 based on selecting the Richardson smoother with $m_1 = m_2 = m = 3p^2$, cf. Figure 3.3. Here, we observe that δ_2 and δ_3 are asymptotically constant, as the polynomial degree p increases showing that our two-level and V-cycle algorithms are uniformly convergent also with respect to p provided that $m \gtrsim p^2$, i.e. $\mu = 0$ also for polygonal meshes.

Next, we investigate the performance of the V-cycle algorithm with non-nested partitions presented in Sect. 3.2. We compute the iteration counts needed by our V-cycle algorithm to reduce the relative residual error below a given tolerance of 10^{-6}, by varying the polynomial degree and the granularity of the finest grid. In Table 3.1 we report the computed convergence factor

$$\theta_J = \exp\left(\frac{1}{N_{it,J}} \ln \frac{\|\mathbf{r}_{N_{it,J}}\|}{\|\mathbf{r}_0\|}\right),$$

where $N_{it,J}$ is the iteration counts needed to reduce the residual below the given tolerance by the h-version of the V-cycle scheme with J levels, where $J = 2,3,4$, while $\mathbf{r}_{N_{it,J}}$ and $\mathbf{r}_0$ are the final and initial residual vectors, respectively. Here, the polynomial approximation degree on each level is chosen as $p_j = 1$, $j = 1,\ldots,J$, while we vary the number of elements of the finest grid and the number of smoothing steps ($m_1 = m_2 = m$). According to Theorem 3.1, the convergence factor is independent from the spatial discretization step h. Indeed, for a fixed $J \in \{2,3,4\}$ and a fixed number of smoothing steps m, the convergence factor is roughly constant. In particular, this means that the number of iterations needed by our V-cycle method to reduce the residual below a given tolerance is independent of the granularity of the underlying grid. As expected, the convergence factor is reduced by increasing the number of smoothing step.

We have repeated the same set of experiments employing $p_j = 2$, $p_j = 3$ and $p_j = 4$; the results are reported in Table 3.2– 3.4 together with the corresponding iteration counts (in parenthesis), respectively. First, a comparison between Table 3.1 and Table 3.3 shows that the convergence factor increases as p grows if the number of smoothing steps is kept fixed. We also observe that, for constant number of smoothing steps we observe that the convergence factor depends on

Table 3.1: Convergence factor θ_J (and iteration counts) of the V-cycle multigrid method as a function of m ($C_\sigma^j \equiv C_\sigma = 10$, $p = 1$).

	Set 1			Set 2		
	2 levels	3 levels	4 levels	2 levels	3 levels	4 levels
$m = 3$	0.52 (22)	0.60 (27)	0.65 (32)	0.55 (24)	0.59 (27)	0.72 (42)
$m = 5$	0.41 (16)	0.48 (19)	0.52 (21)	0.41 (16)	0.46 (18)	0.51 (21)
$m = 8$	0.30 (12)	0.37 (14)	0.40 (16)	0.31 (12)	0.36 (14)	0.40 (15)
	Set 3			Set 4		
	2 levels	3 levels	4 levels	2 levels	3 levels	4 levels
$m = 3$	0.54 (23)	0.62 (30)	0.77 (54)	0.55 (23)	0.67 (34)	0.80 (62)
$m = 5$	0.42 (16)	0.50 (20)	0.53 (22)	0.42 (16)	0.49 (20)	0.53 (22)
$m = 8$	0.31 (12)	0.39 (15)	0.42 (16)	0.33 (13)	0.38 (15)	0.41 (16)

Table 3.2: Convergence factor θ_J (and iteration counts) of the V-cycle multigrid method as a function of m ($C_\sigma^j \equiv C_\sigma = 10$, $p = 2$).

	Set 1			Set 2		
	2 levels	3 levels	4 levels	2 levels	3 levels	4 levels
$m = 3$	0.86 (91)	0.87 (102)	0.88 (106)	0.86 (91)	0.87 (98)	0.87 (103)
$m = 5$	0.79 (58)	0.81 (65)	0.81 (68)	0.79 (59)	0.80 (63)	0.81 (66)
$m = 8$	0.70 (40)	0.73 (44)	0.74 (46)	0.71 (41)	0.72 (43)	0.73 (45)
	Set 3			Set 4		
	2 levels	3 levels	4 levels	2 levels	3 levels	4 levels
$m = 3$	0.86 (91)	0.88 (105)	0.88 (110)	0.85 (88)	0.87 (102)	0.88 (105)
$m = 5$	0.79 (58)	0.81 (66)	0.82 (70)	0.78 (56)	0.81 (64)	0.81 (67)
$m = 8$	0.70 (39)	0.73 (45)	0.75 (47)	0.69 (38)	0.72 (43)	0.73 (45)

Table 3.3: Convergence factor θ_J (and iteration counts) of the V-cycle multigrid method as a function of m ($C_\sigma^j \equiv C_\sigma = 10$, $p = 3$).

	Set 1			Set 2		
	2 levels	3 levels	4 levels	2 levels	3 levels	4 levels
$m = 3$	0.93 (182)	0.94 (224)	0.94 (239)	0.92 (158)	0.93 (195)	0.94 (213)
$m = 5$	0.89 (114)	0.90 (138)	0.91 (147)	0.87 (102)	0.89 (121)	0.90 (132)
$m = 8$	0.83 (76)	0.86 (90)	0.86 (94)	0.82 (70)	0.84 (79)	0.85 (87)
	Set 3			Set 4		
	2 levels	3 levels	4 levels	2 levels	3 levels	4 levels
$m = 3$	0.92 (172)	0.94 (209)	0.94 (233)	0.92 (157)	0.93 (200)	0.94 (211)
$m = 5$	0.88 (108)	0.90 (129)	0.91 (143)	0.87 (99)	0.89 (123)	0.90 (129)
$m = 8$	0.82 (72)	0.85 (85)	0.86 (92)	0.81 (66)	0.84 (80)	0.85 (83)

the value of p_J, i.e. the polynomial degree of approximation employed on the finest level. This is in accord with Theorem 3.1, where the independence of all the discretization parameters is guarantee only if the number of smoothing step is not chosen sufficiently large. Indeed, according to Theorem 3.1, in order to attain uniform convergence (also with respect to p) the number of smoothing steps m must satisfy $m > 2C_A C_S \gtrsim p^{2+\mu}$, cf. also Figure 3.3. In practice we observe that $\mu = 0$ independently of the kinds of elements.

Table 3.4: Convergence factor θ_J (and iteration counts) of the V-cycle multigrid method as a function of m ($C_\sigma^j \equiv C_\sigma = 10$, $p = 4$).

	Set 1			Set 2		
	2 levels	3 levels	4 levels	2 levels	3 levels	4 levels
$m = 3$	0.96 (343)	0.97 (391)	0.97 (404)	0.95 (256)	0.96 (322)	0.96 (350)
$m = 5$	0.94 (212)	0.94 (239)	0.94 (244)	0.92 (158)	0.93 (201)	0.94 (216)
$m = 8$	0.90 (138)	0.91 (153)	0.91 (154)	0.88 (104)	0.90 (132)	0.91 (139)
	Set 3			Set 4		
	2 levels	3 levels	4 levels	2 levels	3 levels	4 levels
$m = 3$	0.96 (329)	0.96 (367)	0.96 (388)	0.95 (293)	0.96 (348)	0.96 (359)
$m = 5$	0.93 (203)	0.94 (224)	0.94 (234)	0.93 (182)	0.94 (213)	0.94 (217)
$m = 8$	0.90 (132)	0.91 (143)	0.91 (147)	0.89 (119)	0.90 (137)	0.90 (137)

3.4 Additive Schwarz smoother

In order to improve the performance of our V-cycle algorithm, in this section we define a domain decomposition preconditioner that can be used as a smoothing operator in place of the Richardson smoother. To this end, let $\mathcal{T}_j$ and $\mathcal{T}_{j-1}$ be a pair of consecutive (non-nested) coarse/fine meshes, respectively, satisfying the grid assumptions given in Sect. 1.1.1. We next introduce the *local* and *coarse* solvers, that are the key ingredients in the definition of the smoother on the space V_j, $j = 2,\dots,J$.

Local Solvers. Let us consider the finest mesh $\mathcal{T}_j$ with cardinality N_j, then for each element $\kappa_i \in \mathcal{T}_j$, we define a local space $V_j^i = \mathcal{P}_{p_j}(\kappa_i)$ $\forall i = 1,...,N_j$, where N_j is the number of elements in $\mathcal{T}_j$; for each local space, the associated local bilinear form is defined by

$$\mathcal{A}_j^i : V_j^i \times V_j^i \to \mathbb{R}, \quad \mathcal{A}_j^i(u_i, v_i) = \mathcal{A}_j(R_i^T u_i, R_i^T v_i) \quad \forall u_i, v_i \in V^i,$$

where $R_i^T : V_j^i \to V_j$ denotes the classical extension by-zero operator from the local space V_j^i to the global one V_j.

Coarse Solver. The natural choice in our context is to define the coarse space V_j^0 to be exactly the same one used for the *Coarse grid correction* step of the V-cycle algorithm introduced in Sect. 3.2, that is

$$V_j^0 = V_{j-1} \equiv \{v \in L^2(\Omega) : v|_\kappa \in \mathcal{P}_{p_{j-1}}(\kappa), \kappa \in \mathcal{T}_{j-1}\}.$$

The bilinear form on V_j^0 is then given by

$$\mathcal{A}_j^0 : V_j^0 \times V_j^0 \to \mathbb{R}, \quad \mathcal{A}_j^0(u_0, v_0) = \mathcal{A}_{j-1}(u_0, v_0) \quad \forall u_0, v_0 \in V_j^0.$$

Here, we define the injection operator from V_j^0 to V_j as the prolongation operator introduced in Sect. 3.2, that is $R_0^T : V_j^0 \to V_j$, $R_0^T = I_{j-1}^j$. By introducing the projection operators $P_i = R_i^T \tilde{P}_i : V_j \to V_j$, $i = 0, 1, \dots, N_j$, where

$$\tilde{P}_i : V_j \to V_j^i, \quad \mathcal{A}_j^i(\tilde{P}_i v_h, w_i) = \mathcal{A}_j(v_h, R_i^T w_i) \quad \forall w_i \in V_j^i, \quad i = 1,\dots,N_j,$$
$$\tilde{P}_0 : V_j \to V_j^0, \quad \mathcal{A}_j^0(\tilde{P}_0 v_h, w_0) = \mathcal{A}_j(v_h, R_0^T w_0) \quad \forall w_0 \in V_j^0,$$

Algorithm 5 One iteration of Multigrid V-cycle scheme with AS-smoother

Pre-smoothing:
if j=1 **then**
$\mathsf{MG}_{\mathscr{AS}}(1,g,z_0,m_1,m_2) = A_1^{-1}g.$
else
Pre-smoothing:
$z^{(m_1)} = ASPCG(A_j,z_0,g,m_1);$

Coarse grid correction:
$r_{j-1} = I_j^{j-1}(g - A_j z^{(m_1)});$
$e_{j-1} = \mathsf{MG}_{\mathscr{AS}}(j-1,r_{j-1},0,m_1,m_2);$
$z^{(m_1+1)} = z^{(m_1)} + I_{j-1}^{j} e_{j-1};$

Post-smoothing:
$z^{(m_1+m_1+1)} = ASPCG(A_j,z^{(m_1+1)},g,m_2);$

$\mathsf{MG}_{\mathscr{AS}}(j,g,z_0,m_1,m_2) = z^{(m_1+m_2+1)}.$
end if

the additive Schwarz operator is defined by $P_{ad} = B_{ad}^{-1}A_j$, where $B_{ad}^{-1} = \sum_{i=0}^{N_j}(R_i^T(A_j^i)^{-1}R_i)$ is the preconditioner. Then, the *Additive Schwarz* smoothing operator with m steps consists in performing m iterations of the *Preconditioned Conjugate Gradient* method using B_{ad} as preconditioner. In Algorithm 5 we show the V-cycle multigrid method using P_{ad} as a smoother. Here, $\mathsf{MG}_{\mathscr{AS}}(j,g,z_0,m_1,m_2)$ denotes the approximate solution of $A_j z = g$ obtained after one iteration, with initial guess z_0 and m_1, m_2 pre- and post-smoothing steps, respectively. The smoothing step is given by the algorithm $ASPCG$, i.e., $z = ASPCG(A,z_0,g,m)$ represents the output of m steps of *Preconditioned Conjugate Gradient* method applied to the linear system of equations $Ax = g$, by using B_{as} as preconditioner and starting from the initial guess z_0.

The computed convergence factor and iteration counts based on employing Algorithm 5 are reported in Tables 3.5–3.9, for the corresponding V-cycle algorithm with $J = 2,3,4$ levels. The simulations are similar to the ones described in the previous section: here we used the grids of Set 2, 3 and 4, cf. Figure 3.1, and we varied the polynomial degree $p \in \{1,\dots,5\}$. First, we observe that, also in this case, the iteration counts seem to be independent of the number of elements in the underlying mesh for a fixed number of smoothing steps m. Moreover, the results show that a minimal number of smoothing steps is not needed to attain the convergence as p increases. Finally, Table 3.10 shows the computed convergence factor, where different polynomial approximation degrees are employed on different levels. Also in this case we observe that the iteration counts seem to be independent of the granularity of the underlying grid.

Table 3.5: Convergence factor (and iteration counts) of the V-cycle multigrid method with the Additive Schwarz smoother as a function of m ($C_\sigma^j \equiv C_\sigma = 10$, $p = 1$).

	set 1			set 2			set 3		
	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.
$m = 3$	0.160 (8)	0.160 (8)	0.160 (8)	0.150 (8)	0.150 (8)	0.150 (8)	0.140 (7)	0.140 (7)	0.140 (7)
$m = 5$	0.023 (4)	0.023 (4)	0.023 (4)	0.021 (4)	0.021 (4)	0.021 (4)	0.024 (4)	0.024 (4)	0.024 (4)
$m = 8$	0.001 (3)	0.001 (3)	0.001 (3)	0.001 (3)	0.001 (3)	0.001 (3)	0.001 (3)	0.001 (3)	0.001 (3)

Table 3.6: Convergence factor (and Iteration counts) of the V-cycle multigrid method with the Additive Schwarz smoother as a function of m ($C_\sigma^j \equiv C_\sigma = 10$, $p = 2$).

	set 1			set 2			set 3		
	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.
$m = 3$	0.38 (15)	0.39 (15)	0.39 (15)	0.36 (14)	0.36 (14)	0.36 (14)	0.30 (12)	0.31 (12)	0.31 (12)
$m = 5$	0.16 (8)	0.16 (8)	0.16 (8)	0.12 (7)	0.12 (7)	0.12 (7)	0.12 (7)	0.12 (7)	0.12 (7)
$m = 8$	0.02 (4)	0.02 (4)	0.02 (4)	0.02 (4)	0.02 (4)	0.02 (4)	0.02 (4)	0.02 (4)	0.02 (4)

Table 3.7: Convergence factor (and iteration counts) of the V-cycle multigrid method with the Additive Schwarz smoother as a function of m ($C_\sigma^j \equiv C_\sigma = 10$, $p = 3$).

	set 1			set 2			set 3		
	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.
$m = 3$	0.47 (19)	0.46 (18)	0.46 (18)	0.45 (18)	0.46 (18)	0.46 (18)	0.50 (20)	0.50 (20)	0.49 (20)
$m = 5$	0.23 (10)	0.23 (10)	0.23 (10)	0.19 (9)	0.19 (9)	0.19 (9)	0.24 (10)	0.24 (10)	0.24 (10)
$m = 8$	0.05 (5)	0.05 (5)	0.05 (5)	0.04 (5)	0.04 (5)	0.04 (5)	0.06 (5)	0.06 (5)	0.06 (5)

Table 3.8: Convergence factor (and iteration counts) of the V-cycle multigrid method with the Additive Schwarz smoother as a function of m ($C_\sigma^j \equiv C_\sigma = 10$, $p = 4$).

	set 1			set 2			set 3		
	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.
$m = 3$	0.53 (22)	0.55 (23)	0.55 (23)	0.56 (24)	0.58 (26)	0.58 (26)	0.56 (24)	0.57 (25)	0.57 (25)
$m = 5$	0.30 (12)	0.31 (12)	0.31 (12)	0.28 (11)	0.28 (11)	0.28 (11)	0.31 (12)	0.31 (12)	0.31 (12)
$m = 8$	0.09 (6)	0.09 (6)	0.09 (6)	0.08 (6)	0.08 (6)	0.08 (6)	0.09 (6)	0.09 (6)	0.09 (6)

Table 3.9: Convergence factor (and iteration counts) of the V-cycle multigrid method with the Additive Schwarz smoother as a function of m ($C_\sigma^j \equiv C_\sigma = 10$, $p = 5$).

	set 1			set 2			set 3		
	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.
$m = 3$	0.59 (27)	0.61 (28)	0.61 (28)	0.59 (27)	0.60 (27)	0.60 (27)	0.58 (26)	0.59 (26)	0.59 (26)
$m = 5$	0.36 (14)	0.37 (14)	0.37 (14)	0.36 (14)	0.37 (14)	0.37 (14)	0.33 (13)	0.34 (13)	0.34 (13)
$m = 8$	0.15 (8)	0.15 (8)	0.15 (8)	0.13 (7)	0.13 (7)	0.13 (7)	0.13 (7)	0.13 (7)	0.13 (7)

3.4.1 Applications to domains with curved boundaries

In this section we consider two examples where the coarser grid does not conform to the boundary. Indeed, in these cases the agglomeration process with edge-coarsening might lead to coarse meshes whose boundary does not fit the geometry, cf. Figure 3.4 for an example.

In the following we present two examples showing that the convergence properties of our multigrid method seem to not deteriorate for such problems and that our approach seems to

Table 3.10: Convergence factor (and iteration counts) of the hp-version of the V-cycle multigrid method with the Additive Schwarz smoother as a function of m. Here the polynomial degree on each space is $p_j = j$ for $j = 1,2,3,4$.

	set 1			set 2			set 3		
	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.
$m = 3$	0.66 (34)	0.67 (35)	0.67 (34)	0.64 (31)	0.65 (32)	0.65 (32)	0.69 (37)	0.69 (38)	0.70 (38)
$m = 5$	0.39 (15)	0.39 (15)	0.39 (15)	0.36 (14)	0.36 (14)	0.36 (14)	0.41 (16)	0.41 (16)	0.41 (16)
$m = 8$	0.20 (9)	0.20 (9)	0.20 (9)	0.18 (8)	0.18 (8)	0.18 (8)	0.20 (9)	0.20 (9)	0.20 (9)

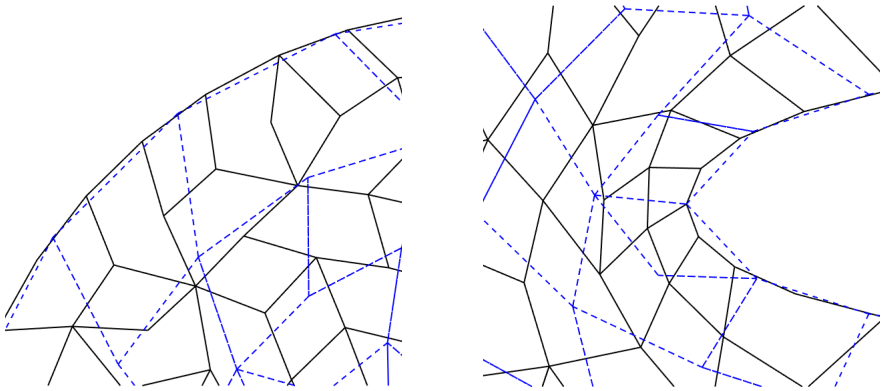


Figure 3.4: Examples of fine $\mathcal{T}_h$ (–) and coarse $\mathcal{T}_H$ (– –) grids for a domain with a curved boundary.

Table 3.11: Convergence factor of the h-version of the V-cycle multigrid method with the Additive Schwarz smoother as a function of m (circular crown test case, $p = 1$).

	set 1			set 2			set 3		
	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.
$m = 3$	0.268	0.268	0.268	0.274	0.257	0.257	0.325	0.325	0.325
$m = 5$	0.098	0.098	0.098	0.093	0.093	0.093	0.086	0.086	0.086
$m = 8$	0.013	0.013	0.013	0.011	0.011	0.011	0.010	0.010	0.010

Table 3.12: Convergence factor of the h-version of the V-cycle multigrid method with the Additive Schwarz smoother as a function of m (circular crown test case, $p = 2$).

	set 1			set 2			set 3		
	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.
$m = 3$	0.578	0.598	0.598	0.585	0.592	0.592	0.582	0.584	0.583
$m = 5$	0.340	0.340	0.340	0.362	0.367	0.367	0.325	0.332	0.332
$m = 8$	0.105	0.105	0.105	0.125	0.125	0.125	0.121	0.121	0.121

Table 3.13: Convergence factor of the h-version of the V-cycle multigrid method with the Additive Schwarz smoother as a function of m (airfoil profile test case, $p = 1$).

	set 1			set 2			set 3		
	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.
$m = 3$	0.312	0.318	0.318	0.325	0.315	0.315	0.320	0.334	0.334
$m = 5$	0.121	0.124	0.124	0.105	0.107	0.107	0.115	0.124	0.124
$m = 8$	0.020	0.020	0.020	0.022	0.022	0.022	0.021	0.021	0.021

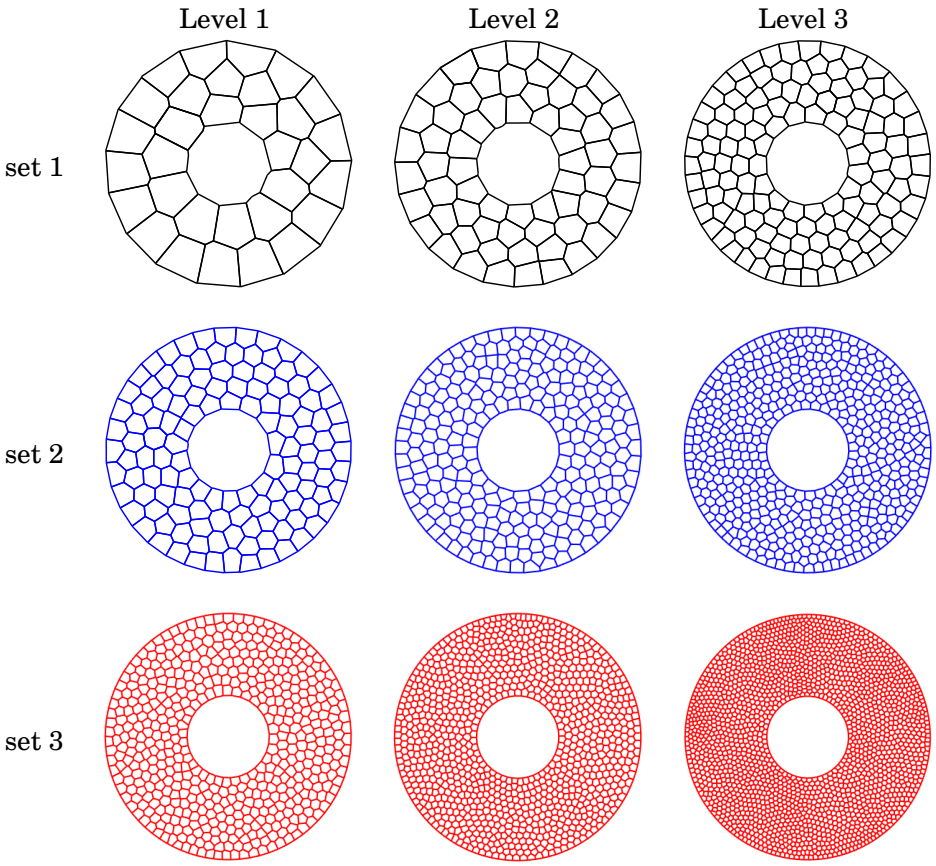


Figure 3.5: Circular crown test case: for any set of grids the first three levels of non-nested meshes are shown.

Table 3.14: Convergence factor of the h-version of the V-cycle multigrid method with the Additive Schwarz smoother as a function of m (airfoil profile test case, $p = 2$).

	set 1			set 2			set 3		
	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.
$m = 3$	0.866	0.848	0.848	0.842	0.848	0.843	0.865	0.864	0.865
$m = 5$	0.621	0.630	0.630	0.629	0.636	0.637	0.655	0.661	0.660
$m = 8$	0.331	0.332	0.334	0.353	0.354	0.355	0.374	0.374	0.376

be competitive in practical cases. The results of this section have been obtained with the AS smoother, cf. Section 3.4. First, we consider problem (3.1) with a constant forcing term $f = 1$, and choose the computational domain to be a circular crown $\Omega = \Omega_1 \setminus \Omega_2$, where Ω_1 and Ω_2 are two concentric circles of radii $r_1 = 2$ and $r_2 = \frac{2}{3}$, respectively. We have tested the V-cycle method by defining three sequences of uniform Voronoi grids (set 1, set 2, set 3) depicted in Figure 3.5, where, for each set of grids, the first three levels of refinement are shown. Here, each polygonal mesh at different levels is defined independently from the previous one with the only constraint that the cardinality of each coarser grid is approximately $\frac{1}{4}$ of that of the finer level. Tables 3.11 and 3.12 show the computed convergence factors for $p = 1$ and $p = 2$, respectively, by choosing

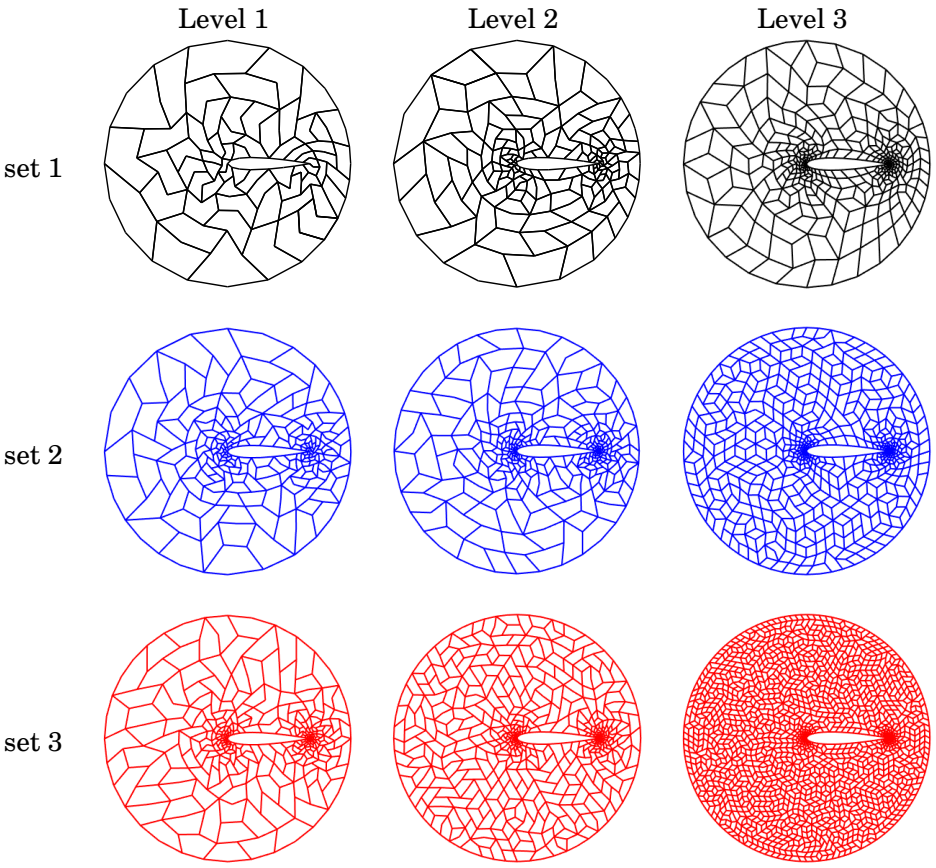


Figure 3.6: Airfoil profile test case: for any set of grids the first three levels of non-nested grids are shown.

$m = 3,5,8$ smoothing steps. As expected, since m is fixed, the results confirm that the convergence rate depends on p but it is independent of the granularity of the underlying grid, as well as the number of levels employed.

Next, we consider the airfoil geometry of [CXZ98], which is characterized by a more complicated geometry $\Omega = \Omega_1 \setminus \Omega_2$, where Ω_1 is the circle of radius $r_1 = \frac{3}{2}$, and Ω_2 is the airfoil profile NACA0015 [LB74]. As before, we consider three sequences of non-nested polygonal meshes (set 1, set 2, set 3), cf. Figure 3.6. The grids have been obtained by firstly defining a non-uniform triangular mesh on Ω with the tool *DistMesh* [PS04], and then by agglomerating based on employing *METIS* [KK09]. The results for $p = 1$ and $p = 2$ are shown in Tables 3.13 and 3.14, respectively. Also in this case we observe that, by fixing the number of smoothing steps m and the polynomial degree p, the convergence factor seems to be independent of the mesh size. Moreover, the performance of the method seems to not deteriorate even if the underlying mesh is characterized by elements of different size, which suggests that our algorithm seems to be well suited for the solution of problems characterized by a local refinement or applications with mesh adaptation.

CHAPTER 4

AN AGGLOMERATION-BASED, MASSIVELY PARALLEL NON-OVERLAPPING ADDITIVE SCHWARZ PRECONDITIONER

In this chapter, we introduce a two-level Additive Schwarz method based on a coarse space correction. The method presented here differs from the smoothing operator introduced in Section 3.4 due to a different choice of the coarse solver operator. In this chapter we will consider the following second order elliptic problem. Let $\Omega \subset \mathbb{R}^d$, $d = 2,3$ be a convex computational domain with Lipschitz boundary and let $f \in L^2(\Omega)$ be a given function. We consider the weak formulation of the following second-order elliptic problem, with homogeneous Dirichlet boundary conditions: find $u \in V = H_0^1(\Omega)$ such that

$$\mathscr{A}(u,v) = \int_\Omega \rho \nabla u \cdot \nabla v \, \mathrm{d}\mathbf{x} = \int_\Omega f v \, \mathrm{d}\mathbf{x} \quad \forall\, v \in V. \tag{4.1}$$

Here, $\rho \in L^\infty(\Omega)$ denotes the diffusion coefficient, which we suppose to be such that $0 < \rho_0 \leq \rho$. In particular, we can assume $\rho_0 = 1$, since (4.1) can always be scaled by $1/\rho_0$. Given a polytopic grid $\mathscr{T}_h$ satisfying the mesh assumptions stated in Section 1.1.1, and the corresponding DG finite element space V_h defined as in (1.3), the bilinear form $\mathscr{A}_h(\cdot,\cdot) : V_h \times V_h \to \mathbb{R}$ corresponding to the symmetric interior penalty DG approximation of (4.1) is given by

$$\begin{aligned} \mathscr{A}_h(u_h,v_h) = &\int_\Omega \Big[\rho \nabla_h u_h \cdot \nabla_h v_h \; + \rho \nabla_h u_h \cdot \mathscr{R}_\rho(\llbracket v_h \rrbracket) + \rho \nabla_h v_h \cdot \mathscr{R}_\rho(\llbracket u_h \rrbracket) \Big] \mathrm{d}\mathbf{x} \\ &+ \int_{\mathscr{F}_h} \sigma_{h,\rho} \llbracket u_h \rrbracket \cdot \llbracket v_h \rrbracket \, \mathrm{d}s, \end{aligned} \tag{4.2}$$

where ∇_h denotes the piecewise gradient operator on $\mathscr{T}_h$. Here, $\mathscr{R}_\rho : [L^1(\mathscr{F}_h)]^d \to [V_h]^d$ denotes the lifting operator defined by

$$\int_\Omega \mathscr{R}_\rho(\mathbf{q}) \cdot \boldsymbol{\eta} \, \mathrm{d}\mathbf{x} = -\int_{\mathscr{F}_h} \mathbf{q} \cdot \{\{\boldsymbol{\eta}\}\}_\omega \, \mathrm{d}s \quad \forall\, \boldsymbol{\eta} \in [V_h]^d. \tag{4.3}$$

Furthermore, the parameter ω is related to the weighted average operator (1.4) and it is defined as $\omega = \frac{\rho_{\kappa^-}}{\rho_{\kappa^+} + \rho_{\kappa^-}}$ on each internal face $F \in \mathscr{F}_h^I$, $F \subset \partial\kappa^{\pm}$.

Remark 4.1 (Diffusion coefficient ρ)**.** *Here, we make the hypothesis that the diffusion coefficient ρ is piecewise constant on each polytopic element $\kappa \in \mathscr{T}_h$ and we write $\rho_\kappa = \rho|_\kappa$ to denote its restriction to κ. We refer to [GL10] for the more general case when ρ violates this assumption.*

In (4.2), according to [Dry03, DGS07], $\sigma_{h,\rho} \in L^\infty(\mathscr{F}_h)$ denotes the interior penalty stabilization function, which is defined by

$$\sigma_{h,\rho}|_F = C_\sigma \langle \rho_\kappa \rangle \frac{p^2}{\langle h_\kappa \rangle} \quad \forall\, F \in \mathscr{F}_h, \tag{4.4}$$

with $C_\sigma > 0$ independent of ρ, p, $|F|$, $|\kappa|$ and h_κ. Then, based on the above definitions, the Symmetric Interior Penalty DG (SIPDG) discretization of (4.1) is given by:

$$\text{find } u_h \in V_h \text{ s.t. } \mathscr{A}_h(u_h, v_h) = \int_\Omega f v_h \, \mathbf{dx} \quad \forall\, v_h \in V_h. \tag{4.5}$$

4.1 Non-overlapping Additive Schwarz preconditioner

The additive Schwarz preconditioner requires the introduction of two additional partitions (besides $\mathscr{T}_h$): a partition $\mathscr{T}_\mathbb{H}$ composed of disjoint polyhedral subdomains where local solvers are applied in parallel and a non-overlapping partition $\mathscr{T}_H$ employed for the coarse space correction. In particular, we make the following choices:

- $\mathscr{T}_\mathbb{H} = \{\Omega_1, \ldots, \Omega_{N_\mathbb{H}}\}$ of size $\mathbb{H} = \max_{1 \le i \le N_\mathbb{H}} \{diam(\Omega_i)\}$ such that each subdomain Ω_i is the union of some elements $\kappa \in \mathscr{T}_h$; we assume that $\mathbb{H} \eqsim diam(\Omega_i)$ $\forall i = 1, \ldots, N_\mathbb{H}$; we also assume that a colouring properly holds, i.e., there exists $N_\mathbb{S}$ such that

  $$\max_{i=1,\ldots,N_\mathbb{H}} card\{\Omega_j \text{ s.t. } \partial\Omega_i \cap \partial\Omega_j \neq \emptyset\} \le N_\mathbb{S}, \tag{4.6}$$

  i.e., $N_\mathbb{S}$ represents the maximum number of neighbors that an element $\Omega_i \in \mathscr{T}_\mathbb{H}$ might possess.

- $\mathscr{T}_H = \{\mathscr{D}_1, \ldots, \mathscr{D}_{N_H}\}$ is the coarse space of size $H = \max_{1 \le j \le N_H} \{diam(\mathscr{D}_j)\}$ such that $H \eqsim diam(\mathscr{D}_j)$ $\forall j = 1, \ldots, N_H$;

We remark that the grids $\mathscr{T}_H$ and $\mathscr{T}_\mathbb{H}$ are possibly non-nested, see Figure 4.1 for an example.

Remark 4.2. *Since $\mathscr{T}_\mathbb{H}$ are defined as the agglomeration of fine grid elements $\kappa \in \mathscr{T}_h$, we have the relation $\mathscr{T}_h \subseteq \mathscr{T}_\mathbb{H}$, i.e. for all $\kappa \in \mathscr{T}_h$ there exists $\mathscr{K} \in \mathscr{T}_\mathbb{H}$ such that $\kappa \subseteq \mathscr{K}$.. However, we point out that no further assumptions are needed between $\mathscr{T}_\mathbb{H}$ and $\mathscr{T}_H$ for the definition of our method. Classical Additive Schwarz methods have typically been defined based on the assumption that $\mathscr{T}_h \subseteq \mathscr{T}_H \subseteq \mathscr{T}_\mathbb{H}$. In this chapter we take a different approach: firstly, we* assume *that the granularity*

of $\mathscr{T}_{\mathbb{H}}$ is finer than $\mathscr{T}_H$; indeed, we are particularly interested in the massively parallel case whereby $\mathscr{T}_{\mathbb{H}} = \mathscr{T}_h$, cf. [DK16]. Secondly, we also permit the use of non-nested coarse and fine partitions, i.e., when $\mathscr{T}_h \nsubseteq \mathscr{T}_H$.

The main ingredients of the additive Schwarz method are defined as follows.

Local Solvers. Let us consider the subdomain partition $\mathscr{T}_{\mathbb{H}}$ with cardinality $N_{\mathbb{H}}$, then for each sub-domain $\Omega_i \in \mathscr{T}_{\mathbb{H}}$ we define a local space V_i as the restriction of the DG finite element space V_h to Ω_i, i.e., for $i = 1,\dots,N_{\mathbb{H}}$,

$$V_i = V_h|_{\Omega_i} \equiv \{v_h \in V_h : v_h|_{\Omega_j} = 0 \ \forall \ j \neq i\}, \quad i = 1,\dots,N_{\mathbb{H}}.$$

For each local space V_i, the associated local bilinear form is defined by

$$\mathscr{A}_i : V_i \times V_i \to \mathbb{R}, \quad \mathscr{A}_i(u_i, v_i) = \mathscr{A}_h(R_i^\top u_i, R_i^\top v_i) \quad \forall u_i, v_i \in V_i,$$

where $R_i^\top : V_i \to V_h$ denotes the classical extension by-zero operator from the local space V_i to the global space V_h. The restriction operator $R_i : V_h \to V_i$, $i = 1,\dots,N_{\mathbb{H}}$, is defined as the transpose of $R_i^\top$ with respect to the $L^2(\Omega_i)$ inner product.

Coarse Solver. For $1 \leq q \leq p$, the coarse solver is defined based on exploiting the partition $\mathscr{T}_H$. To this end, let V_0 be the DG finite element space defined on $\mathscr{T}_H$ given by

$$V_0 \equiv V_H = \{v_H \in L^2(\Omega) : v_H|_{\mathscr{D}_k} \in \mathscr{P}_q(\mathscr{D}_k),\ k = 1,\dots,N_H\}, \quad 0 \leq q \leq p.$$

Further, let $R_0^\top$ be the L^2-projection operator from V_0 to V_h, defined as:

$$R_0^\top : v_0 \in V_0 \longmapsto R_0^\top v_0 \in V_h : \quad \int_\Omega R_0^\top v_0 w_h \, \mathrm{d}\mathbf{x} = \int_\Omega v_0 w_h \, \mathrm{d}\mathbf{x} \quad \forall w_h \in V_h.$$

In this way $R_0^\top$ is well defined also when $\mathscr{T}_H$ and $\mathscr{T}_h$ are non-nested. Then, the bilinear form associated to V_0 is defined by

$$\mathscr{A}_0 : V_0 \times V_0 \to \mathbb{R}, \quad \mathscr{A}_0(u_0, v_0) = \mathscr{A}_h(R_0^\top u_0, R_0^\top v_0) \quad \forall u_0, v_0 \in V_0. \tag{4.7}$$

Remark 4.3 (Implementation)**.** *From the implementation point of view, we point out that the operator $R_0^\top$ can be computed based on employing the optimized numerical integration method described in Chapter 2, cf. also [AHP18b].*

Remark 4.4 (Nested spaces)**.** *When $V_0 \equiv V_H \subseteq V_h$, i.e., when the coarse and fine grids $\mathscr{T}_H$ and $\mathscr{T}_h$, respectively, are nested, then the action of $R_0^\top$ on a coarse function coincides with the action of the natural injection operator. Indeed, by contradiction, if $\exists\, \overline{v}_0 \in V_0$ such that $R_0^\top \overline{v}_0 \neq \overline{v}_0$, then, by employing the definition of $R_0^\top$, we have*

$$0 < \|R_0^\top \overline{v}_0 - \overline{v}_0\|_{L^2(\Omega)} = \min_{w_h \in V_h} \|w_h - \overline{v}_0\|_{L^2(\Omega)} \leq \|\overline{v}_0 - \overline{v}_0\|_{L^2(\Omega)} = 0,$$

which is a contradiction and hence $R_0^\top v_0 = v_0$ for all $v_0 \in V_0$ when $V_0 \subseteq V_h$.

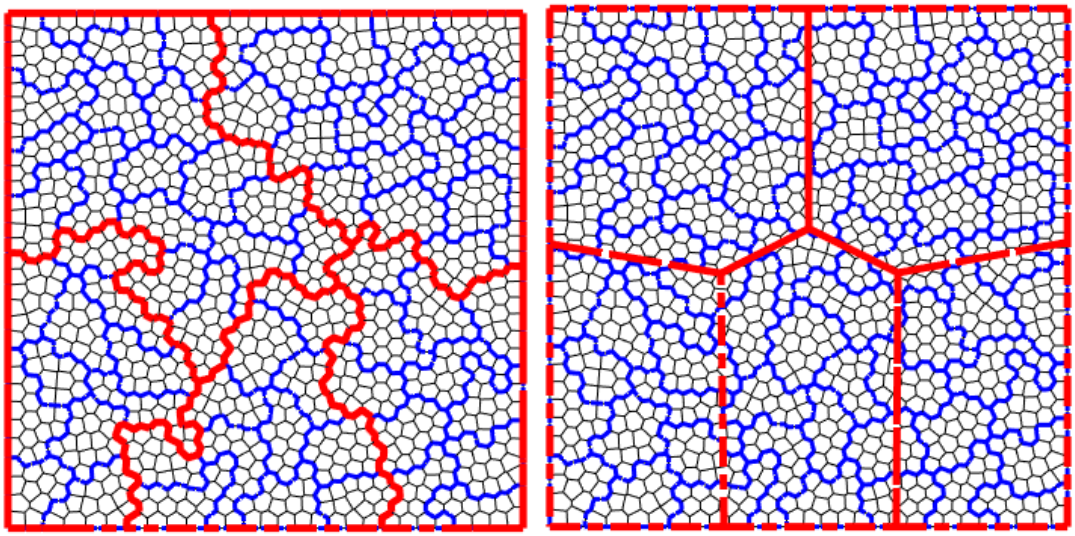


Figure 4.1: Example of polygonal $\mathscr{T}_h$ (black), $\mathscr{T}_{\mathbb{H}}$ (blue) and $\mathscr{T}_H$ (red), when the coarse and fine grids are nested, i.e., $\mathscr{T}_h \subseteq \mathscr{T}_H$, (left) and non-nested, i.e., $\mathscr{T}_h \nsubseteq \mathscr{T}_H$ (right).

By introducing the projection operators $P_i = R_i^\top \tilde{P}_i : V_i \to V_h,\ i = 0,1,\dots,N_h$, where

$$\widetilde{P_i} : V_h \to V_i, \quad \mathscr{A}_i(\tilde{P}_i v_h, w_i) = \mathscr{A}_h(v_h, R_i^\top w_i) \quad \forall w_i \in V_i, \quad i = 1,\dots,N_{\mathbb{H}},$$
$$\widetilde{P_0} : V_h \to V_0, \quad \mathscr{A}_0(\tilde{P}_0 v_h, w_0) = \mathscr{A}_h(v_h, R_0^\top w_0) \quad \forall w_0 \in V_0,$$

the additive Schwarz operator is defined by $P_{ad} = \sum_{i=0}^{N_{\mathbb{H}}} P_i = B_{ad}^{-1} A_h$, where $B_{ad}^{-1} = \sum_{i=0}^{N_{\mathbb{H}}} (R_i^\top (A_i)^{-1} R_i)$ is the preconditioner. An upper bound of the condition number of P_{as} can be derived by following the abstract analysis presented, for example, in [TW04].

4.2 Preliminary results

We first present some preliminary results that will be employed for the analysis presented in Section 4.3. For the sake of the analysis, we assume that the grids $\mathscr{T}_h$ and $\mathscr{T}_H$ are nested. The extension of the theoretical analysis to the much general case $\mathscr{T}_h \nsubseteq \mathscr{T}_H$ is presented in Section 4.4 for the particular case $\mathscr{T}_{\mathbb{H}} = \mathscr{T}_h$ and $\rho = 1$. Here, we introduce the following energy norm, which is an extension of (1.12):

$$\|w\|_{h,\rho}^2 = \sum_{\kappa \in \mathscr{T}_h} \int_\kappa \rho |\nabla w|^2 \, \mathrm{d}\mathbf{x} + \sum_{F \in \mathscr{F}_h} \int_F \sigma_{h,\rho} |[\![w]\!]|^2 \, \mathrm{d}s. \tag{4.8}$$

The well-posedness of problem (4.5) with respect to the norm (4.8) is then established in the following lemma, which is an extension of Lemma 1.4.

Lemma 4.1. *The following continuity and coercivity bounds, respectively, hold:*

$$\mathscr{A}_h(u_h,v_h) \lesssim \|u_h\|_{h,\rho}\|v_h\|_{h,\rho} \quad \forall u_h,v_h \in V_h,$$
$$\mathscr{A}_h(u_h,u_h) \gtrsim \|u_h\|^2_{h,\rho} \quad \forall u_h \in V_h.$$

The second bound holds provided that C_σ appearing in (4.4) *is sufficiently large.*

A key aspect of our analysis is based on employing the conforming approximant defined in (3.9). In particular, to ensure that the preconditioner is scalable also with the jumps of the diffusion coefficient, here we define the conforming approximant in a slightly different manner, in order to obtain an approximation of discontinuous discrete functions $v_h \in V_h$ on each local domain $\mathscr{D}_j \in \mathscr{T}_H$, $j = 1,\dots,N_H$. To this end, we first need to define the following local grids generated from $\mathscr{T}_h$ and $\mathscr{T}_H$; for $j = 1,\dots,N_H$:

$$\mathscr{T}_{h,j} = \{\kappa \in \mathscr{T}_h \text{ s.t. } \kappa \subset \mathscr{D}_j\}, \text{ for some } \mathscr{D}_j \in \mathscr{T}_H, \tag{4.9}$$
$$\mathscr{F}^I_{h,j} = \{F \in \mathscr{F}_h \text{ s.t. } F \subset \mathscr{D}_j\}, \text{ for some } \mathscr{D}_j \in \mathscr{T}_H,$$
$$\mathscr{F}^B_{h,j} = \{F \in \mathscr{F}_h \text{ s.t. } F \subset \partial\mathscr{D}_j\}, \text{ for some } \mathscr{D}_j \in \mathscr{T}_H,$$
$$\mathscr{F}_{h,j} = \mathscr{F}^I_{h,j} \cup \mathscr{F}^B_{h,j}, \tag{4.10}$$

Remark 4.5. *Note that since the grids $\mathscr{T}_h$ and $\mathscr{T}_H$ are nested, i.e., $\mathscr{T}_{h,j} \subseteq \mathscr{T}_h$, $j = 1,\dots,N_H$, $\mathscr{T}_{h,j}$ also satisfies Assumptions 1.1 and 1.2, for all $j = 1,\dots,N_H$.*

The local conforming approximant employed here is then defined as follows.

Definition 4.1. Let $\mathscr{D}_j \in \mathscr{T}_H$ and $\mathscr{G}_h(v_h) = \nabla_h v_h + \mathscr{R}_1(\llbracket v_h \rrbracket)$ be the discrete gradient operator of $v_h \in V_h$. Here, $\mathscr{R}_1 : L^1(\mathscr{F}_h) \to [V_h]^d$ is the lifting operator with $\rho = 1$ and $\omega = \frac{1}{2}$ in its definition given in (4.3). Then, $\widetilde{v}_{h,j}$ is defined as the solution of the following problem: find $\widetilde{v}_{h,j} \in H^1_0(\Omega)$ such that

$$\int_\Omega \nabla\widetilde{v}_{h,j} \cdot \nabla w \, \mathrm{d}\mathbf{x} = \int_\Omega \mathscr{G}_h(v_h \mathbb{1}_{\{\mathscr{D}_j\}}) \cdot \nabla w \, \mathrm{d}\mathbf{x} \quad \forall w \in H^1_0(\Omega), \tag{4.11}$$

where $\mathbb{1}_{\{\mathscr{D}_j\}}$ is the characteristic function on $\mathscr{D}_j$, that is $\mathbb{1}_{\{\mathscr{D}_j\}}(\mathbf{x}) = 1$ if $\mathbf{x} \in \mathscr{D}_j$ and $\mathbb{1}_{\{\mathscr{D}_j\}}(\mathbf{x}) = 0$ elsewhere.

Following the analysis presented in [AHS16] we prove the following approximation properties.

Theorem 4.1. *Let $\mathscr{D}_j \in \mathscr{T}_H$ $j = 1,\dots,N_H$. Given $v_h \in V_h$ we write $\widetilde{v}_{h,j} \in H^1_0(\Omega)$, $j = 1,\dots,N_H$ to be the conforming approximant given in Definition 4.1. Then, the following approximation and stability results hold:*

$$\|v_h - \widetilde{v}_{h,j}\|_{L^2(\mathscr{D}_j)} \lesssim \frac{h}{p}\|\sigma^{1/2}_{h,1}\llbracket v_h \mathbb{1}_{\{\mathscr{D}_j\}} \rrbracket\|_{L^2(\mathscr{F}_{h,j})}, \qquad j = 1,\dots,N_H, \tag{4.12}$$

$$|\widetilde{v}_{h,j}|_{H^1(\Omega)} \lesssim \|v_h \mathbb{1}_{\{\mathscr{D}_j\}}\|_{h,1}, \qquad j = 1,\dots,N_H, \tag{4.13}$$

where $\mathscr{F}_{h,j}$ is defined in (4.10)*, $j = 1,\dots,N_H$, and $\|\cdot\|_{h,1}$ is as in* (4.8) *with $\rho = 1$.*

Proof. Let $\mathscr{D}_j \in \mathscr{T}_H$, since Ω is convex, there exists $z \in H^2(\Omega) \cap H_0^1(\Omega)$ such that

$$\begin{cases} -\Delta z = v_h \mathbb{1}_{\{\mathscr{D}_j\}} - \widetilde{v}_{h,j} & \text{in } \Omega, \\ z = 0 & \text{on } \partial\Omega, \end{cases}$$

Here, z satisfies

$$\|z\|_{H^2(\Omega)} \lesssim \|v_h \mathbb{1}_{\{\mathscr{D}_j\}} - \widetilde{v}_{h,j}\|_{L^2(\Omega)}. \tag{4.14}$$

Employing integration by parts we obtain

$$\begin{aligned} \|v_h \mathbb{1}_{\{\mathscr{D}_j\}} - \widetilde{v}_{h,j}\|^2_{L^2(\Omega)} &= \int_\Omega (v_h \mathbb{1}_{\{\mathscr{D}_j\}} - \widetilde{v}_{h,j})^2 \mathrm{d}\mathbf{x} = -\int_\Omega (v_h \mathbb{1}_{\{\mathscr{D}_j\}} - \widetilde{v}_{h,j})\, \Delta z \, \mathrm{d}\mathbf{x} \\ &= \int_\Omega (\nabla_h (v_h \mathbb{1}_{\{\mathscr{D}_j\}}) - \nabla \widetilde{v}_{h,j}) \cdot \nabla z \, \mathrm{d}\mathbf{x} \; - \; \int_{\mathscr{F}_{h,j}} \nabla z \cdot [\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!] \mathrm{d}s \\ &= \underbrace{\int_\Omega (\mathscr{G}_h (v_h \mathbb{1}_{\{\mathscr{D}_j\}}) - \nabla \widetilde{v}_{h,j}) \cdot \nabla z \, \mathrm{d}\mathbf{x}}_{=0} \; - \; \int_{\mathscr{F}_{h,j}} \nabla z \cdot [\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!] \mathrm{d}s - \int_\Omega \mathscr{R}_1([\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!]) \cdot \nabla z \mathrm{d}\mathbf{x}, \end{aligned}$$

where we have also employed the fact that $[\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!]|_F = \mathbf{0}$ if $F \notin \mathscr{F}_{h,j}$, $[\![\widetilde{v}_{h,j}]\!]|_F = \mathbf{0}$ for all $F \in \mathscr{F}_h$, since $\widetilde{v}_{h,j} \in H_0^1(\Omega)$, and that $\mathscr{G}_h(v_h \mathbb{1}_{\{\mathscr{D}_j\}}) = \nabla_h (v_h \mathbb{1}_{\{\mathscr{D}_j\}}) + \mathscr{R}_1([\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!])$. Using the definitions of $\widetilde{v}_{h,j}$ and $\mathscr{R}_1$, cf. (4.11) and Definition 4.1, respectively, for any $z_h \in V_h$, we have

$$\begin{aligned} \|v_h \mathbb{1}_{\{\mathscr{D}_j\}} - \widetilde{v}_{h,j}\|^2_{L^2(\Omega)} &= -\int_{\mathscr{F}_{h,j}} \nabla z \cdot [\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!] \mathrm{d}s \; - \; \int_\Omega \mathscr{R}_1([\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!]) \cdot \nabla z \mathrm{d}\mathbf{x} \\ &= -\int_\Omega \mathscr{R}_1([\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!]) \cdot (\nabla z - \nabla_h z_h) \mathrm{d}\mathbf{x} + \int_{\mathscr{F}_h} [\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!] \cdot \{\{\nabla_h z_h\}\} \mathrm{d}s - \int_{\mathscr{F}_{h,j}} \nabla z \cdot [\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!] \mathrm{d}s \\ &= -\int_\Omega \mathscr{R}_1([\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!]) \cdot (\nabla z - \nabla_h z_h) \mathrm{d}\mathbf{x} + \int_{\mathscr{F}_{h,j}} ([\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!] \cdot \{\{\nabla_h z_h\}\} - \nabla z \cdot [\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!]) \mathrm{d}s. \end{aligned}$$

Given that $z \in H^2(\Omega)$ and $\{\{\nabla z\}\}_{\mathscr{V}_2}|_F = \nabla z|_F$, we deduce that

$$\begin{aligned} \|v_h \mathbb{1}_{\{\mathscr{D}_j\}} - \widetilde{v}_{h,j}\|^2_{L^2(\Omega)} &= -\int_\Omega \mathscr{R}_1([\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!]) \cdot (\nabla z - \nabla_h z_h) \mathrm{d}\mathbf{x} - \int_{\mathscr{F}_{h,j}} [\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!] \cdot \{\{\nabla z - \nabla z_h\}\} \mathrm{d}s \\ &\lesssim \|\mathscr{R}_1([\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!])\|_{L^2(\Omega)} \|\nabla z - \nabla_h z_h\|_{L^2(\Omega)} \; + \; \|[\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!]\|_{L^2(\mathscr{F}_{h,j})} \|\{\{\nabla z - \nabla_h z_h\}\}\|_{L^2(\mathscr{F}_{h,j})}. \end{aligned} \tag{4.15}$$

The first term on the right hand side of (4.15) can be written as follows.

$$\begin{aligned} \|\mathscr{R}_1([\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!])\|^2_{L^2(\Omega)} &= \int_\Omega \mathscr{R}_1([\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!]) \cdot \mathscr{R}_1([\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!]) \mathrm{d}\mathbf{x} \\ &= -\int_{\mathscr{F}_h} [\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!] \cdot \{\{\mathscr{R}_1([\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!])\}\} \mathrm{d}s \\ &= -\int_{\mathscr{F}_{h,j}} [\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!] \cdot \{\{\mathscr{R}_1([\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!])\}\} \mathrm{d}s, \end{aligned} \tag{4.16}$$

where the last inequality follows from $[\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!]|_F = \mathbf{0}$ if $F \notin \mathscr{F}_{h,j}$. Then, from (4.16) and the Cauchy-Schwarz inequality we obtain the following bound:

$$\|\mathscr{R}_1([\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!])\|^2_{L^2(\Omega)} \leq \|\sigma_{h,1}^{\frac{1}{2}} [\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!]\|_{L^2(\mathscr{F}_{h,j})} \, \|\sigma_{h,1}^{-\frac{1}{2}} \{\{\mathscr{R}_1([\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!])\}\}\|_{L^2(\mathscr{F}_{h,j})} \tag{4.17}$$

Here, the second term on the right hand side of (4.17) can be bounded by employing Lemma 1.3 as follows:

$$\begin{aligned}\|\sigma_{h,1}^{-\frac{1}{2}}\{\{\mathscr{R}_1([\![v_h\mathbb{1}_{\{\mathscr{D}_j\}}]\!])\}\}\|^2_{L^2(\mathscr{F}_{h,j})} &\leq \sum_{\kappa\in\mathscr{T}_{h,j}}\|\sigma_{h,1}^{-\frac{1}{2}}\mathscr{R}_1([\![v_h\mathbb{1}_{\{\mathscr{D}_j\}}]\!])\|^2_{L^2(\partial\kappa)}\\ &= C_\sigma^{-1}\sum_{\kappa\in\mathscr{T}_{h,j}}\frac{\langle h_\kappa\rangle}{p^2}\|\mathscr{R}_1([\![v_h\mathbb{1}_{\{\mathscr{D}_j\}}]\!])\|^2_{L^2(\partial\kappa)}\\ &\leq C_\sigma^{-1}C_{tr}\sum_{\kappa\in\mathscr{T}_{h,j}}\|\mathscr{R}_1([\![v_h\mathbb{1}_{\{\mathscr{D}_j\}}]\!])\|^2_{L^2(\kappa)}\\ &\lesssim \|\mathscr{R}_1([\![v_h\mathbb{1}_{\{\mathscr{D}_j\}}]\!])\|^2_{L^2(\mathscr{D}_j)},\end{aligned}\tag{4.18}$$

where we also have employed that $\langle h_\kappa\rangle \leq 2h_\kappa$. By inserting (4.18) into (4.17) we obtain

$$\|\mathscr{R}_1([\![v_h\mathbb{1}_{\{\mathscr{D}_j\}}]\!])\|^2_{L^2(\Omega)} \lesssim \|\sigma_{h,1}^{\frac{1}{2}}[\![v_h\mathbb{1}_{\{\mathscr{D}_j\}}]\!]\|^2_{L^2(\mathscr{F}_{h,j})}.\tag{4.19}$$

Then, by choosing $z_h = \Pi_h z$ in (4.15) and by employing (4.19), Lemma 1.6, cf., also, Remark 1.5 and the bound (4.14) we obtain

$$\begin{aligned}\|v_h\mathbb{1}_{\{\mathscr{D}_j\}} - \widetilde{v}_{h,j}\|^2_{L^2(\Omega)} &\lesssim \frac{h}{p}\|z\|_{H^2(\Omega)}\|\sigma_{h,1}^{\frac{1}{2}}[\![v_h\mathbb{1}_{\{\mathscr{D}_j\}}]\!]\|_{L^2(\mathscr{F}_{h,j})}\\ &\lesssim \frac{h}{p}\|v_h\mathbb{1}_{\{\mathscr{D}_j\}} - \widetilde{v}_{h,j}\|_{L^2(\Omega)}\|\sigma_{h,1}^{\frac{1}{2}}[\![v_h\mathbb{1}_{\{\mathscr{D}_j\}}]\!]\|_{L^2(\mathscr{F}_{h,j})}.\end{aligned}\tag{4.20}$$

Then the bound (4.12) follows from (4.20) by observing that $\|v_h - \widetilde{v}_{h,j}\|_{L^2(\mathscr{D}_j)} \leq \|v_h\mathbb{1}_{\{\mathscr{D}_j\}} - \widetilde{v}_{h,j}\|_{L^2(\Omega)}$. In order to obtain (4.13) we first select $w = \widetilde{v}_{h,j}$ in (4.11) and by making use of the Cauchy-Schwarz inequality we obtain:

$$|\widetilde{v}_{h,j}|^2_{H^1(\Omega)} \lesssim \|\mathscr{G}_h(v_h\mathbb{1}_{\{\mathscr{D}_j\}})\|_{L^2(\Omega)}|\widetilde{v}_{h,j}|_{H^1(\Omega)}.$$

Then, from the definition of $\mathscr{G}_h$ given in Definition 4.1 we have:

$$\|\mathscr{G}_h(v_h\mathbb{1}_{\{\mathscr{D}_j\}})\|^2_{L^2(\Omega)} \lesssim \|\nabla_h v_h\|^2_{L^2(\mathscr{T}_{h,j})} + \|\mathscr{R}_1([\![v_h\mathbb{1}_{\{\mathscr{D}_j\}}]\!])\|^2_{L^2(\mathscr{D}_j)}.\tag{4.21}$$

The bound (4.13) is then obtained by inserting (4.19) into (4.21). □

We are now ready to investigate the relationship between the spaces V_h, $V_{\mathbb{H}}$, and V_H introduced above. The following result concerns the approximation of a function $v_h \in V_h$ with a coarse function $v_H \in V_H$; this represents an extension of the analogous result presented in [AHS16, Lemma 5.1].

Lemma 4.2. *For any $v_h \in V_h$ there exists a coarse function $v_H \in V_H$ such that*

$$\|v_h - R_0^\top v_H\|_{L^2(\mathscr{D}_j)} \lesssim \frac{H}{q}\|v_h\mathbb{1}_{\{\mathscr{D}_j\}}\|_{h,1},\tag{4.22}$$

$$|v_h - R_0^\top v_H|_{H^1(\mathscr{T}_{h,j})} \lesssim \|v_h\mathbb{1}_{\{\mathscr{D}_j\}}\|_{h,1},\tag{4.23}$$

for $j = 1,\dots,N_H$, where $\mathscr{T}_{h,j}$ and $\mathscr{F}_{h,j}$ are as defined in (4.9) *and* (4.10)*, respectively.*

Proof. Let $v_h \in V_h$ and let v_H be defined as

$$v_H|_{\mathscr{D}_j} = (\Pi_H(\widetilde{v}_{h,j}))|_{\mathscr{D}_j},\ j = 1,\dots,N_H,$$

with $\widetilde{v}_{h,j}$ as defined in Definition 4.1 while Π_H denotes the global variant of the hp-approximant introduced in Lemma 1.6, cf., also, Remark 1.5, defined on the coarse space V_H. Then, noting Remark 4.4, application of the triangle inequality gives

$$\begin{aligned}\|v_h - R_0^\top v_H\|_{L^2(\mathscr{D}_j)} &= \|v_h - v_H|_{\mathscr{D}_j}\|_{L^2(\mathscr{D}_j)} \\ &\lesssim \|v_h - \widetilde{v}_{h,j}\|_{L^2(\mathscr{D}_j)} + \|\widetilde{v}_{h,j} - \Pi_H(\widetilde{v}_{h,j})\|_{L^2(\mathscr{D}_j)} \\ &\lesssim \|v_h - \widetilde{v}_{h,j}\|_{L^2(\mathscr{D}_j)} + \|\widetilde{v}_{h,j} - \Pi_H(\widetilde{v}_{h,j})\|_{L^2(\Omega)}.\end{aligned}$$

Employing Lemma 1.6 together with Assumption 1.2, cf. Remark 1.5, gives

$$\|v_h - R_0^\top v_H\|_{L^2(\mathscr{D}_j)} \lesssim \|v_h - \widetilde{v}_{h,j}\|_{L^2(\mathscr{D}_j)} + \frac{H}{q}\|\widetilde{v}_{h,j}\|_{H^1(\Omega)}.$$

By applying the Poincaré inequality to $\widetilde{v}_{h,j} \in H_0^1(\Omega)$ and exploiting the bounds of Theorem 4.1, inequality (4.22) immediately follows by observing that $h \leq H$ and $q \leq p$. In order to obtain (4.23) we proceed as follows:

$$\begin{aligned}|v_h - R_0^\top v_H|_{H^1(\mathscr{T}_{h,j})} &\lesssim |v_h|_{H^1(\mathscr{T}_{h,j})} + |R_0^\top v_H|_{H^1(\mathscr{T}_{h,j})} \\ &= |v_h|_{H^1(\mathscr{T}_{h,j})} + |v_{H,j}|_{H^1(\mathscr{T}_{h,j})}.\end{aligned} \tag{4.24}$$

Moreover, by employing the triangle inequality and by observing that $v_{H,j} \in \mathscr{P}_q(\mathscr{D}_j) \subset H^1(\mathscr{D}_j)$ we have

$$\begin{aligned}|v_{H,j}|_{H^1(\mathscr{T}_{h,j})} &\lesssim |\Pi_H(\widetilde{v}_{h,j}) - \widetilde{v}_{h,j}|_{H^1(\mathscr{D}_j)} + |\widetilde{v}_{h,j}|_{H^1(\mathscr{D}_j)} \\ &\lesssim |\Pi_H(\widetilde{v}_{h,j}) - \widetilde{v}_{h,j}|_{H^1(\Omega)} + |\widetilde{v}_{h,j}|_{H^1(\Omega)} \\ &\lesssim |\widetilde{v}_{h,j}|_{H^1(\Omega)},\end{aligned}$$

where we also have employed the bound of Lemma 1.6 and the Poincaré inequality. The previous result together with (4.24) and the estimates of Theorem 4.1 lead to (4.23). □

Before proceeding with the analysis of P_{ad} we also need the following result regarding the properties of the non-overlapping decomposition $V_i,\ i = 1,\dots,N_H$, introduced in Section 4.1.

Lemma 4.3. *For each $v_h \in V_h$ there exists a unique decomposition $v_h = \sum_{i=1}^{N_\mathbb{H}} R_i^\top v_i$, with $v_i \in V_i$ $i = 1,\dots,N_\mathbb{H}$, such that*

$$\mathscr{A}_h(v_h, v_h) = \sum_{i=1}^{N_\mathbb{H}} \mathscr{A}_i(v_i, v_i) + \sum_{i,j=1, i\neq j}^{N_\mathbb{H}} \mathscr{A}_h(R_i^\top v_i, R_j^\top v_j),$$

and

$$\Big|\sum_{i,j=1, i\neq j}^{N_\mathbb{H}} \mathscr{A}_h(R_i^\top v_i, R_j^\top v_j)\Big| \lesssim \|\sqrt{\rho}\,\nabla_h v_h\|^2_{L^2(\mathscr{T}_h)} + \sum_{i=1}^{N_\mathbb{H}} \|\sigma^{\frac{1}{2}}_{h,\rho} v_h\|^2_{L^2(\partial\Omega_i)}.$$

Proof. Let $v_h \in V_h$ and let $v_i = R_i v_h,\ i = 1,\dots,N_{\mathbb{H}}$ where R_i as defined in Section 4.1, then

$$\mathscr{A}_h(R_i^\top v_i, R_j^\top v_j) = 0 \quad \text{if } \partial\Omega_i \cap \partial\Omega_j = \emptyset,$$

and

$$\Big|\sum_{i,j=1,i\neq j}^{N_{\mathbb{H}}} \mathscr{A}_i(R_i^\top v_i, R_j^\top v_j)\Big| \lesssim \sum_{i,j=1,i\neq j}^{N_{\mathbb{H}}} |\mathscr{A}_i(R_i^\top v_i, R_j^\top v_j)| \quad \forall\ v_i \in V_i, v_j \in V_j,\ i,j = 1,\dots,N_{\mathbb{H}}.$$

Now let $i \neq j$ such that $\partial\Omega_i \cap \partial\Omega_j \neq \emptyset$ and write $\check{v}_i = R_i^\top v_i$ and $\check{v}_j = R_j^\top v_j$, then

$$\mathscr{A}_h(\check{v}_i, \check{v}_j) = \int_\Omega \Big[\rho\nabla\check{v}_i \cdot \mathscr{R}_\rho([\![\check{v}_j]\!]) + \rho\nabla\check{v}_j \cdot \mathscr{R}_\rho([\![\check{v}_i]\!])\Big]\mathrm{d}\mathbf{x} + \int_{\mathscr{F}_h} \sigma_{h,\rho}[\![\check{v}_i]\!]\cdot[\![\check{v}_j]\!]\mathrm{d}s, \tag{4.25}$$

By employing the definition of $\mathscr{R}_\rho$ given in (4.3), the first term on the right hand side of (4.25) can be written as

$$\int_\Omega \rho\nabla_h\check{v}_i \cdot \mathscr{R}_\rho([\![\check{v}_j]\!])\mathrm{d}\mathbf{x} = -\int_{\mathscr{F}_h} [\![\check{v}_j]\!]\cdot\{\{\rho\nabla_h\check{v}_i\}\}\mathrm{d}s.$$

By observing that $[\![\check{v}_j]\!] = \mathbf{0}$ on the faces $F \in \mathscr{F}_h$ such that $F \cap \overline{\Omega_j} = \emptyset$, and also $\{\{\rho\nabla_h\check{v}_i\}\} = \mathbf{0}$ for any $F \cap \overline{\Omega_i} = \emptyset$, we have

$$\begin{aligned}
\int_\Omega \rho\nabla_h\check{v}_i \cdot \mathscr{R}_\rho([\![\check{v}_j]\!])\mathrm{d}\mathbf{x} &= -\int_{\partial\Omega_i\cap\partial\Omega_j} [\![\check{v}_j]\!]\cdot\{\{\rho\nabla_h\check{v}_i\}\}\mathrm{d}s \\
&\lesssim \|\sigma_{h,\rho}^{\frac{1}{2}}[\![\check{v}_j]\!]\|^2_{L^2(\partial\Omega_i\cap\partial\Omega_j)} + \|\sigma_{h,\rho}^{-\frac{1}{2}}\{\{\rho\nabla_h\check{v}_i\}\}\|^2_{L^2(\partial\Omega_i\cap\partial\Omega_j)} \\
&\leq \|\sigma_{h,\rho}^{\frac{1}{2}}[\![\check{v}_j]\!]\|^2_{L^2(\partial\Omega_j)} + \|\sigma_{h,\rho}^{-\frac{1}{2}}\{\{\rho\nabla_h\check{v}_i\}\}\|^2_{L^2(\partial\Omega_i)}. \qquad (4.26)
\end{aligned}$$

Here, the second term on the right hand side of (4.26) can be bounded by employing Lemma 1.3 as follows:

$$\begin{aligned}
\|\sigma_{h,\rho}^{-\frac{1}{2}}\{\{\rho\nabla_h\check{v}_i\}\}\|^2_{L^2(\partial\Omega_i)} &\leq \sum_{\kappa\subset\Omega_i} \|\sigma_{h,\rho}^{-\frac{1}{2}}\langle\rho_\kappa\rangle\nabla_h\check{v}_i\|^2_{L^2(\partial\kappa)} \\
&= C_\sigma^{-1}\sum_{\kappa\subset\Omega_i}\frac{\langle h_\kappa\rangle}{\langle\rho_\kappa\rangle p^2}\langle\rho_\kappa\rangle^2\|\nabla_h\check{v}_i)\|^2_{L^2(\partial\kappa)} \\
&\leq C_\sigma^{-1}C_{tr}\sum_{\kappa\subset\Omega_i}\|\sqrt{\rho_\kappa}\nabla_h\check{v}_i)\|^2_{L^2(\kappa)} \\
&\lesssim \|\sqrt{\rho}\nabla_h\check{v}_i\|^2_{L^2(\Omega_i)}, \qquad (4.27)
\end{aligned}$$

where we also have employed that $\langle\rho_\kappa\rangle \leq 2\rho_\kappa$ and $\langle h_\kappa\rangle \leq 2h_\kappa$. Inserting (4.27) into (4.26) we obtain

$$\int_\Omega \rho\nabla\check{v}_i \cdot \mathscr{R}_\rho([\![\check{v}_j]\!])\mathrm{d}\mathbf{x} \lesssim \|\sqrt{\rho}\ \nabla_h\check{v}_i\|^2_{L^2(\Omega_i)} + \|\sigma_{h,\rho}^{\frac{1}{2}}[\![\check{v}_j]\!]\|^2_{L^2(\partial\Omega_j)}. \tag{4.28}$$

Similarly, we obtain

$$\int_\Omega \rho\nabla\check{v}_j \cdot \mathscr{R}_\rho([\![\check{v}_i]\!])\mathrm{d}\mathbf{x} \lesssim \|\sqrt{\rho}\ \nabla_h\check{v}_j\|^2_{L^2(\Omega_j)} + \|\sigma_{h,\rho}^{\frac{1}{2}}[\![\check{v}_i]\!]\|^2_{L^2(\partial\Omega_i)}, \tag{4.29}$$

and

$$\int_{\mathscr{F}_h} \sigma_{h,\rho}[\![\check{v}_i]\!]\cdot[\![\check{v}_j]\!]\mathrm{d}s \lesssim \|\sigma_{h,\rho}^{\frac{1}{2}}[\![\check{v}_i]\!]\|^2_{L^2(\partial\Omega_i)} + \|\sigma_{h,\rho}^{\frac{1}{2}}[\![\check{v}_j]\!]\|^2_{L^2(\partial\Omega_j)}. \tag{4.30}$$

Inserting (4.28), (4.29) and (4.30) into (4.25) we obtain

$$\mathscr{A}_h(\check{v}_i,\check{v}_j) \lesssim \|\sqrt{\rho}\,\nabla_h\check{v}_i\|^2_{L^2(\Omega_i)} + \|\sqrt{\rho}\,\nabla_h\check{v}_j\|^2_{L^2(\Omega_j)} + \|\sigma_{h,\rho}^{\frac{1}{2}}[\![\check{v}_i]\!]\|^2_{L^2(\partial\Omega_i)} + \|\sigma_{h,\rho}^{\frac{1}{2}}[\![\check{v}_j]\!]\|^2_{L^2(\partial\Omega_j)}.$$

The result follows by summing over $i,j=1,\dots,N_{\mathbb{H}},\ i\neq j$, and by exploiting (4.6). □

For the forthcoming analysis we also require an extension of the trace inequality introduced by Feng and Karakashian in [FK01] and presented by Smears in [Sme18, Lemma 5], to which we refer for the proof.

Lemma 4.4 (Trace inverse inequality)**.** *Let $\mathscr{T}_h$ and $\mathscr{T}_{\mathbb{H}}$ be a couple of nested polygonal grids. We assume that $\mathscr{T}_{\mathbb{H}}$ is obtained by agglomeration of elements of $\mathscr{T}_h$ and that both $\mathscr{T}_h$ and $\mathscr{T}_{\mathbb{H}}$ satisfy Assumption 1.1. Moreover, we assume that for all $\Omega_i\in\mathscr{T}_{\mathbb{H}},\ i=1,\dots,N_{\mathbb{H}}$, there exists $\mathbf{x}_{0,i}\in\Omega_i$ such that $(\mathbf{x}_{0,i}-\mathbf{x})\cdot\mathbf{n}_i \gtrsim \mathbb{H}$ for all $\mathbf{x}\in\partial\Omega_i$, where $\mathbf{n}_i$ is the outward normal vector to $\partial\Omega_i$. Then, for any $v_h\in V_h$, writing $\mathscr{F}_h(\Omega_i)=\{F\in\mathscr{F}_h$ such that $F\subset\Omega_i, F\not\subset\partial\Omega_i\}$, the following bound holds*

$$\|v_h\|^2_{L^2(\partial\Omega_i)} \lesssim \|\nabla_h v_h\|_{L^2(\Omega_i)}\|v_h\|_{L^2(\Omega_i)} + \frac{1}{\mathbb{H}}\|v_h\|^2_{L^2(\Omega_i)} + \Big(\sum_{\mathscr{F}_h(\Omega_i)}\|\sigma_{h,1}^{\frac{1}{2}}[\![v_h]\!]\|^2_{L^2(F)}\Big)^{\frac{1}{2}}\|v_h\|_{L^2(\Omega_i)}.$$

Proof. Since there exists $\mathbf{x}_{0,i}\in\Omega_i$ such that $(\mathbf{x}-\mathbf{x}_{0,i})\cdot\mathbf{n}_i\gtrsim\mathbb{H}\ \forall\mathbf{x}\in\partial\Omega_i$, we have

$$\|v_h\|^2_{L^2(\partial\Omega_i)} = \int_{\partial\Omega_i}|v_h|^2\mathrm{d}\mathbf{x} \lesssim \frac{1}{\mathbb{H}}\int_{\partial\Omega_i}|v_h|^2(\mathbf{x}-\mathbf{x}_{0,i})\cdot\mathbf{n}_i\mathrm{d}s. \tag{4.31}$$

By employing integration by parts, the following equalities hold

$$\begin{aligned}
&\sum_{\kappa\subset\Omega_i} -\int_\kappa(\mathbf{x}-\mathbf{x}_{0,i})\cdot\nabla(|v_h|^2)\mathrm{d}\mathbf{x}\\
&\qquad=\sum_{\kappa\subset\Omega_i}\Big[\int_\kappa\nabla\cdot(\mathbf{x}-\mathbf{x}_{0,i})\,|v_h|^2\mathrm{d}\mathbf{x} - \int_{\partial\kappa}(\mathbf{x}-\mathbf{x}_{0,i})\cdot\mathbf{n}\,|v_h|^2\mathrm{d}s\Big]\\
&\qquad=\sum_{\kappa\subset\Omega_i}\Big[\int_\kappa\nabla\cdot(\mathbf{x}-\mathbf{x}_{0,i})\,|v_h|^2\mathrm{d}\mathbf{x}\Big] - \int_{\partial\Omega_i}(\mathbf{x}-\mathbf{x}_{0,i})\cdot\mathbf{n}_i\,|v_h|^2\mathrm{d}s\\
&\qquad-\sum_{F\subset\Omega_i,F\not\subset\partial\Omega_i}\int_F[\![|v_h|^2]\!]\cdot\{\{(\mathbf{x}-\mathbf{x}_{0,i})\}\}\mathrm{d}s,
\end{aligned}$$

where the last equality follows from $\mathbf{x}-\mathbf{x}_{0,i} = \{\{(\mathbf{x}-\mathbf{x}_{0,i})\}\}$. Then, by observing that $[\![|v_h|^2]\!] = 2[\![v_h]\!]\{\{v_h\}\}$, we have

$$\begin{aligned}
&\int_{\partial\Omega_i} (\mathbf{x}-\mathbf{x}_{0,i})\cdot\mathbf{n}_i\ |v_h|^2 \mathrm{d}s \\
&\quad = \sum_{\kappa\subset\Omega_i}\int_\kappa \Big[\nabla\cdot(\mathbf{x}-\mathbf{x}_{0,i})\ |v_h|^2\mathrm{d}\mathbf{x}\ +\ (\mathbf{x}-\mathbf{x}_{0,i})\cdot\nabla(|v_h|^2)\Big] \\
&\quad - \sum_{F\subset\Omega_i, F\not\subset\partial\Omega_i}\int_F [\![|v_h|^2]\!]\cdot\{\{(\mathbf{x}-\mathbf{x}_{0,i})\}\}\mathrm{d}s \\
&\quad = \sum_{\kappa\subset\Omega_i}\int_\kappa \Big[-d\ |v_h|^2\mathrm{d}\mathbf{x}\ +\ (\mathbf{x}-\mathbf{x}_{0,i})\cdot 2v_h\nabla v_h\mathrm{d}\mathbf{x}\Big] \\
&\quad - \sum_{F\subset\Omega_i, F\not\subset\partial\Omega_i}\int_F 2[\![v_h]\!]\{\{v_h\}\}\cdot\{\{(\mathbf{x}-\mathbf{x}_{0,i})\}\}\mathrm{d}s.
\end{aligned}$$

Then, by employing Hölder and Cauchy-Schwarz inequalities we obtain

$$\begin{aligned}
&\int_{\partial\Omega_i} (\mathbf{x}_{0,i}-\mathbf{x})\cdot\mathbf{n}_i\ |v_h|^2\mathrm{d}s \\
&\qquad \lesssim \|v_h\|^2_{L^2(\Omega_i)} + \|\mathbf{x}-\mathbf{x}_{0,i}\|_{[L^\infty(\Omega_i)]^d}\|v\nabla_h v_h\|_{[L^1(\Omega_i)]^d} \\
&\qquad +\ \|\mathbf{x}-\mathbf{x}_{0,i}\|_{[L^\infty(\mathscr{F}_h(\Omega_i))]^d}\|[\![v_h]\!]\{\{v_h\}\}\|_{[L^1(\mathscr{F}_h(\Omega_i))]^d} \\
&\qquad \lesssim \|v_h\|^2_{L^2(\Omega_i)} + \mathbb{H}\ \|v_h\|_{L^2(\Omega_i)}\ \|\nabla_h v_h\|_{L^2(\Omega_i)} \\
&\qquad + \mathbb{H}\ \|\sigma_{h,1}^{\frac{1}{2}}[\![v_h]\!]\|_{L^2(\mathscr{F}_h(\Omega_i))}\ \|\sigma_{h,1}^{-\frac{1}{2}}\{\{v_h\}\}\|_{L^2(\mathscr{F}_h(\Omega_i))} \\
&\qquad \lesssim \|v_h\|^2_{L^2(\Omega_i)} + \mathbb{H}\ \|v_h\|_{L^2(\Omega_i)}\ \|\nabla_h v_h\|_{L^2(\Omega_i)} \\
&\qquad + \mathbb{H}\ \|\sigma_{h,1}^{\frac{1}{2}}[\![v_h]\!]\|_{L^2(\mathscr{F}_h(\Omega_i))}\ \|v_h\|_{L^2(\Omega_i)},
\end{aligned}$$

where we have employed 1.3 in the last inequality. By inserting the previous bound into (4.31) we obtain the thesis. □

4.3 Condition number estimates

In this section we derive an upper bound of the condition number of P_{ad} by following the analysis presented in [TW04], see also [Lio87, Lio88, Lio89]. To this end, we show that the following three Assumptions are satisfied.

Assumption 4.1 (Local stability)**.** *There exists* $0<\alpha<2$ *such that*

$$\mathscr{A}_h(R_i^\top v_i, R_i^\top v_i) \le \alpha\mathscr{A}_i(v_i,v_i) \quad \forall v_i\in V_i\ i=0,1,\dots,N_H.$$

We point out that in our case Assumption 4.1 immediately follows with $\omega=1$ from the definition of $\mathscr{A}_i(\cdot,\cdot)$ given in Section 4.1.

Assumption 4.2 (Strengthened Cauchy-Schwarz inequality)**.** *There exists constants* $0 \leq \mathscr{E}_{ij} \leq 1$*, for* $1 \leq i, j \leq N_{\mathbb{H}}$*, such that*

$$|\mathscr{A}_h(R_i^\top v_i, R_j^\top v_j)| \leq \mathscr{E}_{ij}\mathscr{A}_h(R_i^\top v_i, R_i^\top v_i)^{\frac{1}{2}}\mathscr{A}_h(R_j^\top v_j, R_j^\top v_j)^{\frac{1}{2}},$$

for all $v_i \in V_i$ *and for all* $v_j \in V_j$*. Moreover, by writing* $\mathbf{E} \in \mathbb{R}^{N_{\mathbb{H}} \times N_{\mathbb{H}}}$ *such that* $\mathbf{E}_{ij} = \mathscr{E}_{ij}$*, then the spectral radius* $\Theta(\mathbf{E}) \lesssim N_{\mathbb{S}}$*, where* $N_{\mathbb{S}}$ *is the maximum number of adjacent partitions that any* $\Omega_i \in \mathscr{T}_{\mathbb{H}}$ *might possess.*

In our case, Assumption 4.2 follows since we have required that each subdomain $\Omega_i \in \mathscr{T}_{\mathbb{H}}$ can possess only a finite number of neighbors, cf. (4.6). In particular, by observing that if $\overline{\partial\Omega_i} \cap \overline{\partial\Omega_j} = \emptyset$ then $\mathscr{A}_h(R_i^\top v_i, R_j^\top v_j) = 0 \ \forall v_i \in V_i$ and $\forall v_j \in V_j$, we can define

$$\mathscr{E}_{ij} = \begin{cases} 0 & \text{if } \overline{\partial\Omega_i} \cap \overline{\partial\Omega_j} = \emptyset, \\ 1 & \text{otherwise.} \end{cases}$$

Then $\Theta(\mathbf{E})$ is uniformly bounded by by $(N_{\mathbb{S}} + 1)$, where $N_{\mathbb{S}}$ is the maximum number of neighbors that each subdomain may possess, cf. (4.6). This result ensures that a stable (in the sense of the energy norm) decomposition can be found for the local spaces and the coarse one.

Assumption 4.3 (Stable decomposition)**.** *Each* $v_h \in V_h$ *admits a decomposition of the form* $v_h = \sum_{i=0}^{N_H} R_i^\top v_i$*,* $v_i \in V_i$*,* $i = 0, 1, \ldots, N_H$*, such that*

$$\sum_{i=0}^{N_H} \mathscr{A}_i(u_i, u_i) \leq C_\sharp^2 \mathscr{A}_h(u_h, u_h).$$

Following [TW04, Theorem 2.7] the upper bound for the condition number of P_{ad} is stated in the following theorem.

Theorem 4.2. *Given that Assumptions 4.1–4.3 hold, then the condition number of the additive Schwarz operator is given by*

$$K(P_{ad}) \lesssim C_\sharp^2 \alpha(\Theta(\mathscr{E}) + 1),$$

where α*,* $\mathscr{E}$*, and* $C_\sharp$ *are as defined in Assumptions 4.1, 4.2 and 4.3, respectively.*

Next we prove that Assumption 4.3 holds.

Theorem 4.3. *Let* $v_h \in V_h$*, and assume that the grid* $\mathscr{T}_h$ *satisfies Assumptions 1.1 and 1.2. We also assume that* $\mathscr{T}_{\mathbb{H}}$ *is obtained by agglomeration of elements of* $\mathscr{T}_h$*,* $\mathscr{T}_H$ *is obtained by agglomeration of elements of* $\mathscr{T}_{\mathbb{H}}$ *and that both* $\mathscr{T}_{\mathbb{H}}$ *and* $\mathscr{T}_H$ *satisfy Assumptions 1.1 and 1.2. Then Assumption 4.3 holds with*

$$C_\sharp^2 \eqsim \Big[\max_{j=1,\ldots,N_H}\Big(\frac{\overline{\rho}_j}{\underline{\langle\rho\rangle}_j}\Big)\Big]\Big(\frac{p^2}{q}\frac{H}{h} - \frac{p^2}{q^2}\frac{H^2}{h\mathbb{H}}\Big),$$

where

$$\underline{\langle\rho\rangle}_j = \min_{\mathbf{x}\in\mathscr{D}_j}(\langle\rho\rangle(\mathbf{x})), \qquad \overline{\rho}_j = \max_{\mathbf{x}\in\mathscr{D}_j}(\rho(\mathbf{x})).$$

Proof. Let $v_h \in V_h$ and let us choose $v_0 = v_H$ where $v_H \in V_H$ is defined in Lemma 4.2. Then, employing Lemma 4.3, $v_h - R_0^\top v_0$ can be uniquely decomposed as

$$v_h - R_0^\top v_0 = \sum_{i=1}^{N_{\mathbb{H}}} R_i^\top v_i,$$

where $v_i = R_i(v_h - R_0^\top v_0)$, $i = 1,\dots,N_{\mathbb{H}}$, and

$$\mathscr{A}_h(v_h - R_0^\top v_0, v_h - R_0^\top v_0) = \sum_{i=1}^{N_{\mathbb{H}}} \mathscr{A}_i(v_i, v_i) + \sum_{i,j=1, i\neq j}^{N_{\mathbb{H}}} \mathscr{A}_h(R_i^\top v_i, R_j^\top v_j). \tag{4.32}$$

Adding $\mathscr{A}_0(v_0, v_0)$ to both sides of (4.32) we obtain the following inequality:

$$\begin{aligned} \left|\sum_{i=0}^{N_{\mathbb{H}}} \mathscr{A}_i(v_i, v_i)\right| &\leq \left|\mathscr{A}_h(v_h - R_0^\top v_0, v_h - R_0^\top v_0)\right| + \left|\mathscr{A}_0(v_0, v_0)\right| + \left|\sum_{i,j=1, i\neq j}^{N_{\mathbb{H}}} \mathscr{A}_h(R_i^\top v_i, R_j^\top v_j)\right| \\ &\equiv I + II + III. \end{aligned} \tag{4.33}$$

From the definition of $\mathscr{A}_0(\cdot,\cdot)$, cf. (4.7), we note that

$$\begin{aligned} II &\leq |\mathscr{A}_h(R_0^\top v_0 - v_h, R_0^\top v_0)| + |\mathscr{A}_h(v_h, R_0^\top v_0)| \\ &\leq |\mathscr{A}_h(R_0^\top v_0 - v_h, R_0^\top v_0 - v_h)| + 2|\mathscr{A}_h(R_0^\top v_0 - v_h, v_h)| + |\mathscr{A}_h(v_h, v_h)|. \end{aligned} \tag{4.34}$$

Employing the continuity of $\mathscr{A}_h$, cf. Chapter 4.1, and Young's inequality we obtain

$$|\mathscr{A}_h(R_0^\top v_0 - v_h, v_h)| \lesssim \|v_h - R_0^\top v_0\|_{h,\rho} \|v_h\|_{h,\rho} \lesssim \|v_h - R_0^\top v_0\|_{h,\rho}^2 + \|v_h\|_{h,\rho}^2.$$

Then, by inserting the above bound into (4.34) and employing the continuity and the coercivity of $\mathscr{A}_h$, cf. Chapter 4.1, we obtain

$$I + II \lesssim \|v_h - R_0^\top v_0\|_{h,\rho}^2 + \mathscr{A}_h(v_h, v_h). \tag{4.35}$$

In particular, we observe that, by the definition of $\|\cdot\|_{h,\rho}$, we have

$$\|v_h - R_0^\top v_0\|_{h,\rho}^2 = \|\sqrt{\rho}\nabla_h(v_h - R_0^\top v_0)\|_{L^2(\mathscr{T}_h)}^2 + \|\sigma_{h,\rho}^{\frac{1}{2}} [\![v_h - R_0^\top v_0]\!]\|_{L^2(\mathscr{F}_h)}^2. \tag{4.36}$$

Writing $\mathscr{F}_H$ to denote the set of faces of $\mathscr{T}_H$, and observing that $\mathscr{F}_H \subseteq \mathscr{F}_h$ since $\mathscr{T}_H \subseteq \mathscr{T}_h$, the second term on the right hand side of (4.36) can be bounded as follows

$$\begin{aligned} \|\sigma_{h,\rho}^{\frac{1}{2}} [\![v_h - R_0^\top v_0]\!]\|_{L^2(\mathscr{F}_h)}^2 &= \|\sigma_{h,\rho}^{\frac{1}{2}} [\![v_h - R_0^\top v_0]\!]\|_{L^2(\mathscr{F}_h \setminus \mathscr{F}_H)}^2 \; + \; \|\sigma_{h,\rho}^{\frac{1}{2}} [\![v_h - R_0^\top v_0]\!]\|_{L^2(\mathscr{F}_H)}^2 \\ &= \|\sigma_{h,\rho}^{\frac{1}{2}} [\![v_h]\!]\|_{L^2(\mathscr{F}_h \setminus \mathscr{F}_H)}^2 \; + \; \|\sigma_{h,\rho}^{\frac{1}{2}} [\![v_h - R_0^\top v_0]\!]\|_{L^2(\mathscr{F}_H)}^2 \\ &\lesssim \|v_h\|_{h,\rho}^2 \; + \; \sum_{j=1}^{N_H} \|\sigma_{h,\rho}^{\frac{1}{2}} (v_h - R_0^\top v_0)\|_{L^2(\partial\mathscr{D}_j)}^2 \\ &\lesssim \|v_h\|_{h,\rho}^2 \; + \; \sum_{i=1}^{N_{\mathbb{H}}} \|\sigma_{h,\rho}^{\frac{1}{2}} (v_h - R_0^\top v_0)\|_{L^2(\partial\Omega_i)}^2, \end{aligned} \tag{4.37}$$

where we have employed that $[\![R_0^\top v_0]\!] = \mathbf{0}$ on each face $F \in \mathscr{F}_h \setminus \mathscr{F}_H$ and the fact that $\mathscr{T}_{\mathbb{H}} \subseteq \mathscr{T}_H$, cf. Remark 4.2. Hence, inserting (4.37) into (4.36) and employing Lemma 4.1, the bound (4.35) becomes

$$I + II \lesssim \|\sqrt{\rho}\,\nabla_h(v_h - R_0^\top v_0)\|^2_{L^2(\mathscr{T}_h)} + \sum_{i=1}^{N_{\mathbb{H}}} \|\sigma^{1/2}_{h,\rho}(v_h - R_0^\top v_0)\|^2_{L^2(\partial\Omega_i)} + \Big|\mathscr{A}_h(v_h, v_h)\Big|.$$

From Lemma 4.3 we get

$$III \lesssim \|\sqrt{\rho}\,\nabla_h(v_h - R_0^\top v_0)\|^2_{L^2(\mathscr{T}_h)} + \sum_{i=1}^{N_{\mathbb{H}}} \|\sigma^{1/2}_{h,\rho}(v_h - R_0^\top v_0)\|^2_{L^2(\partial\Omega_i)}. \tag{4.38}$$

Then, (4.33) may be bounded as follows

$$\begin{aligned}\Big|\sum_{i=0}^{N_{\mathbb{H}}} \mathscr{A}_i(v_i, v_i)\Big| &\lesssim \Big|\mathscr{A}_h(v_h, v_h)\Big| + \|\sqrt{\rho}\,\nabla_h(v_h - R_0^\top v_0)\|^2_{L^2(\mathscr{T}_h)} + \sum_{i=1}^{N_{\mathbb{H}}} \|\sigma^{1/2}_{h,\rho}(v_h - R_0^\top v_0)\|^2_{L^2(\partial\Omega_i)} \\ &\equiv IV + V + VI.\end{aligned} \tag{4.39}$$

Exploiting Lemma 4.2 we have

$$\begin{aligned} V &= \sum_{j=1}^{N_H} \|\sqrt{\rho}\nabla_h(v_h - R_0^\top v_0)\|^2_{L^2(\mathscr{D}_j)} \\ &\lesssim \sum_{j=1}^{N_H} \overline{\rho}_j \|\nabla_h(v_h - R_0^\top v_0)\|^2_{L^2(\mathscr{D}_j)} \\ &\lesssim \sum_{j=1}^{N_H} \overline{\rho}_j \Big[\|\nabla_h v_h\|^2_{L^2(\mathscr{T}_{h,j})} + \|\sigma^{\frac{1}{2}}_{h,1}[\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!]\|^2_{L^2(\mathscr{F}_{h,j})}\Big] \\ &\lesssim \sum_{j=1}^{N_H} \frac{\overline{\rho}_j}{\underline{\langle\rho\rangle}_j} \Big[\|\sqrt{\rho}\,\nabla_h v_h\|^2_{L^2(\mathscr{T}_{h,j})} + \|\sigma^{\frac{1}{2}}_{h,\rho}[\![v_h \mathbb{1}_{\{\mathscr{D}_j\}}]\!]\|^2_{L^2(\mathscr{F}_{h,j})}\Big] \\ &\lesssim \max_{j=1,\dots,N_H} \Big(\frac{\overline{\rho}_j}{\underline{\langle\rho\rangle}_j}\Big) \mathscr{A}_h(v_h, v_h), \end{aligned} \tag{4.40}$$

where we have also employed the coercivity bound of Lemma 4.1 in the last inequality. A bound for ③ can be obtained exploiting the inverse trace inequality of Lemma 4.4. To this end, we first observe that

$$VI \lesssim \sum_{i=1}^{N_{\mathbb{H}}} \frac{p^2 \max_{\{\kappa\subset\Omega_i\}} \rho_\kappa}{h} \|v_h - R_0^\top v_0\|^2_{L^2(\partial\Omega_i)}, \tag{4.41}$$

where we have also employed the definition of $\sigma_{h,\rho}$ and the fact that $\langle\rho_\kappa\rangle|_F \le 2\rho_{\kappa^\pm}$ for any $F \subset \partial\Omega_i$, $F \subset \partial\kappa^\pm$, for some $\kappa^\pm \in \mathscr{T}_h$, which implies that $\langle\rho_\kappa\rangle|_F \le 2\max_{\{\kappa\subset\Omega_i\}}\rho_\kappa$ for all $F \in \mathscr{F}_h$ such that $F \subset \partial\Omega_i$. Then, by applying Lemma 4.4 to each $\Omega_i \in \mathscr{T}_{\mathbb{H}}$, $i = 1,\dots,N_{\mathbb{H}}$, from (4.41) we obtain the

following bound:

$$
\begin{aligned}
VI \lesssim \sum_{i=1}^{N_{\mathbb{H}}} \frac{p^2 \max_{\{\kappa\subset\Omega_i\}}\rho_\kappa}{h}\Big[& \|\nabla_h(v_h - R_0^\top v_0)\|_{L^2(\Omega_i)}\|v_h - R_0^\top v_0\|_{L^2(\Omega_i)} \\
& + \frac{1}{\mathbb{H}}\|v_h - R_0^\top v_0\|^2_{L^2(\Omega_i)} \\
& + \Big(\sum_{F\in\mathscr{F}_h(\Omega_i)} \|\sigma_{h,1}^{\frac{1}{2}}[\![v_h - R_0^\top v_0]\!]\|^2_{L^2(F)}\Big)^{1/2}\|v_h - R_0^\top v_0\|_{L^2(\Omega_i)}\Big].
\end{aligned}
$$

Since $\mathscr{T}_{\mathbb{H}} \subseteq \mathscr{T}_H$, we denote by $\mathscr{I}_j = \{k : 1 \le k \le N_{\mathbb{H}},\ \Omega_k \in \mathscr{T}_{\mathbb{H}} \text{ and } \Omega_k \subset \mathscr{D}_j\}$ the indices that correspond to the subdomains inside $\mathscr{D}_j \in \mathscr{T}_H$, for all $j = 1,\dots,N_H$. Hence, $\mathscr{I}_j \cap \mathscr{I}_k = \emptyset$ for any $j \neq k$, $1 \le j,k \le N_H$, and $\cup_{j=1}^{N_H}\mathscr{I}_j = \{1,\dots,N_{\mathbb{H}}\}$. Then,

$$
\begin{aligned}
VI &\lesssim \sum_{j=1}^{N_H}\sum_{i\in\mathscr{I}_j} \frac{p^2 \max_{\{\kappa\subset\Omega_i\}}\rho_\kappa}{h}\Big[\|\nabla_h(v_h - R_0^\top v_0)\|_{L^2(\Omega_i)}\|v_h - R_0^\top v_0\|_{L^2(\Omega_i)} \\
&\quad + \frac{1}{\mathbb{H}}\|v_h - R_0^\top v_0\|^2_{L^2(\Omega_i)} \\
&\quad + \Big(\sum_{F\in\mathscr{F}_h(\Omega_i)} \|\sigma_{h,1}^{1/2}[\![v_h - R_0^\top v_0]\!]\|^2_{L^2(F)}\Big)^{1/2}\|v_h - R_0^\top v_0\|_{L^2(\Omega_i)}\Big] \\
&\lesssim \sum_{j=1}^{N_H}\frac{p^2\overline{\rho}_j}{h}\Big[\sum_{i\in\mathscr{I}_j}\|\nabla_h(v_h - R_0^\top v_0)\|_{L^2(\Omega_i)}\|v_h - R_0^\top v_0\|_{L^2(\Omega_i)} \\
&\quad + \frac{1}{\mathbb{H}}\sum_{i\in\mathscr{I}_j}\|v_h - R_0^\top v_0\|^2_{L^2(\Omega_i)} \\
&\quad + \sum_{i\in\mathscr{I}_j}\Big(\sum_{F\in\mathscr{F}_h(\Omega_i)} \|\sigma_{h,1}^{1/2}[\![v_h - R_0^\top v_0]\!]\|^2_{L^2(F)}\Big)^{1/2}\|v_h - R_0^\top v_0\|_{L^2(\Omega_i)}\Big]. \qquad (4.42)
\end{aligned}
$$

We now proceed by bounding each term present in the bracket in (4.42); to this end, exploiting the discrete Cauchy-Schwarz inequality, we get

$$
\begin{aligned}
&\sum_{i\in\mathscr{I}_j}\|\nabla_h(v_h - R_0^\top v_0)\|_{L^2(\Omega_i)}\|v_h - R_0^\top v_0\|_{L^2(\Omega_i)} \\
&\qquad \le \Big(\sum_{i\in\mathscr{I}_j}\|\nabla_h(v_h - R_0^\top v_0)\|^2_{L^2(\Omega_i)}\Big)^{1/2}\Big(\sum_{i\in\mathscr{I}_j}\|v_h - R_0^\top v_0\|^2_{L^2(\Omega_i)}\Big)^{1/2} \\
&\qquad = |v_h - R_0^\top v_0|_{H^1(\mathscr{T}_{h,j})}\|v_h - R_0^\top v_0\|_{L^2(\mathscr{D}_j)}.
\end{aligned}
$$

Similarly, noting that $\mathscr{F}_h(\Omega_i)$ is the set of faces $F \in \mathscr{F}_h$ strictly contained in Ω_i, then $\cup_{i\in\mathscr{I}_j}\mathscr{F}_h(\Omega_i) \subset \mathscr{F}_{h,j}^I$, we deduce that

$$
\begin{aligned}
&\sum_{i\in\mathscr{I}_j}\Big(\sum_{F\in\mathscr{F}_h(\Omega_i)} \|\sigma_{h,1}^{1/2}[\![v_h - R_0^\top v_0]\!]\|^2_{L^2(F)}\Big)^{1/2}\|v_h - R_0^\top v_0\|_{L^2(\Omega_i)} \\
&\qquad \le \Big(\sum_{i\in\mathscr{I}_j}\|\sigma_{h,1}^{1/2}[\![v_h - R_0^\top v_0]\!]\|^2_{L^2(\mathscr{F}_h(\Omega_i))}\Big)^{1/2}\Big(\sum_{i\in\mathscr{I}_j}\|v_h - R_0^\top v_0\|^2_{L^2(\Omega_i)}\Big)^{1/2} \\
&\qquad \le \Big(\sum_{F\in\mathscr{F}_{h,j}^I}\|\sigma_{h,1}^{1/2}[\![v_h - R_0^\top v_0]\!]\|^2_{L^2(F)}\Big)^{1/2}\|v_h - R_0^\top v_0\|_{L^2(\mathscr{D}_j)}.
\end{aligned}
$$

Noting that $\sum_{i\in\mathscr{I}_j}\|v_h-R_0^\top v_0\|^2_{L^2(\Omega_i)}=\|v_h-R_0^\top v_0\|^2_{L^2(\mathscr{D}_j)}$, gives

$$VI\lesssim\sum_{j=1}^{N_H}\frac{p^2\overline{\rho}_j}{h}\Big[|v_h-R_0^\top v_0|_{H^1(\mathscr{T}_{h,j})}\|v_h-R_0^\top v_0\|_{L^2(\mathscr{D}_j)}+\frac{1}{\mathbb{H}}\|v_h-R_0^\top v_0\|^2_{L^2(\mathscr{D}_j)}$$
$$+\Big(\sum_{F\in\mathscr{F}^I_{h,j}}\|\sigma^{1/2}_{h,1}[\![v_h-R_0^\top v_0]\!]\|^2_{L^2(F)}\Big)^{1/2}\|v_h-R_0^\top v_0\|_{L^2(\mathscr{D}_j)}\Big]. \tag{4.43}$$

The last term on the right-hand side of (4.43) can be bounded by

$$\Big(\sum_{F\in\mathscr{F}^I_{h,j}}\|\sigma^{1/2}_{h,1}[\![v_h-R_0^\top v_0]\!]\|^2_{L^2(F)}\Big)^{1/2}\|v_h-R_0^\top v_0\|_{L^2(\mathscr{D}_j)}$$
$$=\Big(\sum_{F\in\mathscr{F}^I_{h,j}}\|\sigma^{1/2}_{h,1}[\![v_h]\!]\|^2_{L^2(F)}\Big)^{1/2}\|v_h-R_0^\top v_0\|_{L^2(\mathscr{D}_j)}$$
$$\le\|\sigma^{1/2}_{h,1}[\![v_h\mathbb{1}_{\{\mathscr{D}_j\}}]\!]\|_{L^2(\mathscr{F}_{h,j})}\|v_h-R_0^\top v_0\|_{L^2(\mathscr{D}_j)};$$

here we note that $[\![R_0^\top v_0]\!]|_F=\mathbf{0}$ on each $F\in\mathscr{F}^I_{h,j}$, since $\mathscr{T}_h$ and $\mathscr{T}_H$ are nested. Then, by employing the above estimate together with Lemma 4.2, we deduce that

$$VI\lesssim\sum_{j=1}^{N_H}\Big[\frac{p^2\overline{\rho}_j}{h}\Big(\frac{H}{q}+\frac{1}{q^2}\frac{H^2}{\mathbb{H}}\Big)\Big(\|\nabla_h v_h\|^2_{L^2(\mathscr{T}_{h,j})}+\|\sigma^{1/2}_{h,1}[\![v_h\mathbb{1}_{\{\mathscr{D}_j\}}]\!]\|^2_{L^2(\mathscr{F}_{h,j})}\Big)\Big]$$
$$\lesssim\sum_{j=1}^{N_H}\frac{\overline{\rho}_j}{\underline{\langle\rho\rangle}_j}\Big(\frac{p^2}{q}\frac{H}{h}+\frac{p^2}{q^2}\frac{H^2}{h\mathbb{H}}\Big)\Big(\|\sqrt{\rho}\,\nabla_h v_h\|^2_{L^2(\mathscr{T}_{h,j})}+\|\sigma^{1/2}_{h,\rho}[\![v_h\mathbb{1}_{\{\mathscr{D}_j\}}]\!]\|^2_{L^2(\mathscr{F}_{h,j})}\Big)$$
$$\lesssim\max_{j=1,\ldots,N_H}\Big(\frac{\overline{\rho}_j}{\underline{\langle\rho\rangle}_j}\Big)\Big(\frac{p^2}{q}\frac{H}{h}+\frac{p^2}{q^2}\frac{H^2}{h\mathbb{H}}\Big)\mathscr{A}_h(v_h,v_h), \tag{4.44}$$

where we also have employed the coercivity bound of Lemma 4.1 in the last inequality. Inserting estimates (4.40) and (4.44) into (4.39) we obtain the desired result. □

Remark 4.6. *According to our estimates the statement of Theorem 4.2 becomes*

$$K(P_{ad})\lesssim\max_{1\le j\le N_H}\Big(\frac{\overline{\rho}_j}{\underline{\langle\rho\rangle}_j}\Big)\Big(\frac{p^2}{q}\frac{H}{h}-\frac{p^2}{q^2}\frac{H^2}{h\mathbb{H}}\Big)N_{\mathbb{S}}. \tag{4.45}$$

In particular, in the lowest order case, i.e., when $p=q=1$, we have $K_h(P_{ad})\lesssim H^2/h\mathbb{H}$, which is in agreement with the corresponding bound derived in [DK16]. On the other hand if the size of the coarse subdomain and fine meshes are fixed we deduce that $K_p(P_{ad})\lesssim p^2/q$.

4.4 Condition number estimates for non-nested grids

In this Section we provide a bound for the condition number of the P_{ad} introduced in Section 4.1 when the fine $\mathscr{T}_h$ and coarse $\mathscr{T}_H$ grids are non-nested. For the sake of simplicity, here we assume $\rho=1$ on Ω and we consider the massively parallel case, i.e., when $\mathscr{T}_{\mathbb{H}}=\mathscr{T}_h$. Moreover, we also make the following additional assumption on $\mathscr{T}_h$.

A.3 For every polytopic element $\kappa \in \mathcal{T}_h$, there exists a set of m_κ overlapping shape-regular simplices $\mathcal{K}_i$, $i = 1,\dots,m_\kappa$ such that

$$dist(\kappa,\partial\mathcal{K}_i) \lesssim \frac{diam(\mathcal{K}_i)}{p^2}, \quad \text{and} \quad |\mathcal{K}_i| \gtrsim |\kappa|,$$

for all $i = 1,\dots,m_\kappa$.

Thanks to **A.3**, the following inverse inequality holds, cf. [CDGH16, AHH$^+$17].

Lemma 4.5. *Let $v_h \in V_h$ and let $\kappa \in \mathcal{T}_h$. Then it holds*

$$\|\nabla v_h\|^2_{L^2(\kappa)} \lesssim p^4 h_\kappa^{-2} \|v_h\|^2_{L^2(\kappa)} \qquad \forall \kappa \in \mathcal{T}_h.$$

Proof. We refer to [CDGH16, CDGH17] for the proof of this result. □

We first provide a non-nested counterpart of Lemma 4.2 which allows us to prove the validity of Assumption 4.3 also for non-nested spaces V_h and V_H. Also for this case, the key aspect of our analysis is represented by the conforming approximant introduced in Definition 4.1. In particular, we recall the following result.

Theorem 4.4. *Let $\mathcal{G}_h(v_h) = \nabla_h v_h + \mathcal{R}_1([\![v_h]\!])$ be the discrete gradient operator of $v_h \in V_h$ and let $\widetilde{v_h} \in H^1_0(\Omega)$ such that*

$$\int_\Omega \nabla\widetilde{v}_h \cdot \nabla w \, \mathrm{d}\mathbf{x} = \int_\Omega \mathcal{G}_h(v_h) \cdot \nabla w \, \mathrm{d}\mathbf{x} \quad \forall w \in H^1_0(\Omega).$$

Then, the following approximation and stability results hold:

$$\|v_h - \widetilde{v}_h\|_{L^2(\Omega)} \lesssim \frac{h}{p} \|\sigma^{\frac{1}{2}}_{h,1} [\![v_h]\!]\|_{L^2(\mathcal{F}_h)},$$
$$|\widetilde{v}_h|_{H^1(\Omega)} \lesssim \|v_h\|_{h,1},$$

Remark 4.7. *Theorem 4.4 provides global bounds of $v_h \in V_h$ in the L^2-norm. This result is a particular case of Theorem 4.1, where local bounds on each coarse element $\mathcal{D}_j \in \mathcal{T}_H$ are provided. We refer to [AHS16] for the proof of Theorem 4.4.*

On the basis of the previous result, Lemma 4.2 can be generalized to non-nested spaces as follows.

Lemma 4.6. *For any $v_h \in V_h$ there exists a coarse function $v_H \in V_H$ such that*

$$\|v_h - R_0^\top v_H\|_{L^2(\Omega)} \lesssim \frac{H}{q} \|v_h\|_{h,1}, \tag{4.46}$$

$$\|v_h - R_0^\top v_H\|_{h,1} \lesssim \Big(\frac{p^2}{q}\frac{H}{h}\Big) \|v_h\|_{h,1}, \tag{4.47}$$

Proof. Let $v_h \in V_h$ and let $v_H \in V_H$ be defined as $v_H = \Pi_H \widetilde{v}_h$, with $\widetilde{v}_h$ as defined in Theorem 4.4 while Π_H is the hp-interpolation introduced in Lemma 1.6. Then, by employing the triangle inequality we have

$$\begin{aligned}\|v_h - R_0^\top v_H\|_{L^2(\Omega)} \lesssim \|v_h - \widetilde{v}_h\|_{L^2(\Omega)} + \|\widetilde{v}_h - \Pi^h_{L^2}(\widetilde{v}_h)\|_{L^2(\Omega)} \\ + \|\Pi^h_{L^2}(\widetilde{v}_h) - R_0^\top(\Pi_H(\widetilde{v}_h))\|_{L^2(\Omega)}\end{aligned}$$

where $\Pi^h_{L^2} : L^2(\Omega) \to V_h$ is the L^2-projection operator onto V_h. From the definition of $R_0^\top$, we note that $\Pi^h_{L^2}(w_H) = R_0^\top(w_H)$ for all $w_H \in V_H$. Hence, exploiting Lemma 1.6 together with Assumption 1.2, cf. Remark 1.5, gives

$$\begin{aligned}\|v_h - R_0^\top v_H\|_{L^2(\Omega)} &\lesssim \|v_h - \widetilde{v}_h\|_{L^2(\Omega)} + \|\widetilde{v}_h - \Pi^h_{L^2}(\widetilde{v}_h)\|_{L^2(\Omega)} \\ &\quad + \|\Pi^h_{L^2}(\widetilde{v}_h - \Pi_H(\widetilde{v}_h))\|_{L^2(\Omega)} \\ &\leq \|v_h - \widetilde{v}_h\|_{L^2(\Omega)} + \|\widetilde{v}_h - \Pi_h(\widetilde{v}_h)\|_{L^2(\Omega)} \\ &\quad + \|\widetilde{v}_h - \Pi_H(\widetilde{v}_h)\|_{L^2(\Omega)} \\ &\lesssim \|v_h - \widetilde{v}_h\|_{L^2(\Omega)} + \frac{h}{p}\|\widetilde{v}_h\|_{H^1(\Omega)} + \frac{H}{q}\|\widetilde{v}_h\|_{H^1(\Omega)};\end{aligned}$$

here we have also used that $\|\Pi^h_{L^2}(v)\|_{L^2(\Omega)} \leq \|v\|_{L^2(\Omega)}\ \forall v \in L^2(\Omega)$ and $\|v - \Pi^h_{L^2}(v)\|_{L^2(\Omega)} \leq \|v - w\|_{L^2(\Omega)}\ \forall w \in L^2(\Omega)$. Applying the Poincaré inequality to $\widetilde{v}_h \in H^1_0(\Omega)$ and exploiting the bounds of Theorem 4.4, inequality (4.46) immediately follows by observing that $h \leq H$ and $q \leq p$. In order to obtain (4.47) we proceed as follows:

$$\|v_h - R_0^\top v_H\|^2_{h,1} = \|\nabla_h(v_h - R_0^\top v_H)\|^2_{L^2(\mathcal{T}_h)} + \|\sigma^{\frac{1}{2}}_{h,1}[\![v_h - R_0^\top v_H]\!]\|^2_{L^2(\mathcal{F}_h)}. \tag{4.48}$$

The first term on the right hand side of (4.48) can be bounded by employing (4.46) and Lemma 4.5 as follows:

$$\begin{aligned}\|\nabla_h(v_h - R_0^\top v_H)\|^2_{L^2(\mathcal{T}_h)} &= \sum_{\kappa\in\mathcal{T}_h} \|\nabla_h(v_h - R_0^\top v_H)\|^2_{L^2(\kappa)} \\ &\lesssim \frac{p^4}{h^2}\|v_h - R_0^\top v_H\|^2_{L^2(\Omega)} \lesssim \frac{p^4}{q^2}\frac{H^2}{h^2}\|v_h\|^2_{h,1}.\end{aligned} \tag{4.49}$$

The second term on the right hand side of (4.48) can be bounded by employing the definition of $\sigma_{h,1}$, Lemma 4.5 and (4.46) as follows:

$$\begin{aligned}\|\sigma^{\frac{1}{2}}_{h,1}[\![v_h - R_0^\top v_H]\!]\|^2_{L^2(\mathcal{F}_h)} &\lesssim \frac{p^2}{h}\sum_{\kappa\in\mathcal{T}_h} \|v_h - R_0^\top v_H\|^2_{L^2(\partial\kappa)} \\ &\lesssim \frac{p^2}{h}\frac{p^2}{h}\|v_h - R_0^\top v_H\|^2_{L^2(\Omega)} \\ &\lesssim \frac{p^4}{q^2}\frac{H^2}{h^2}\|v_h\|^2_{h,1}.\end{aligned} \tag{4.50}$$

By inserting (4.49) and (4.50) into (4.48) we obtain (4.47). □

Thanks to Lemma 4.6 we can prove the following Theorem which states the validity of Assumption 4.3 for non-nested spaces.

Theorem 4.5. *Assumption 4.3 holds with*

$$C_\sharp^2 \eqsim \Big(\frac{p^4}{q^2}\frac{H^2}{h^2}\Big).$$

Proof. Let $v_h \in V_h$. Proceeding as in the proof of Theorem 4.3, by choosing $v_0 = v_H$ as in Lemma 4.6, v_h can be decomposed as $v_h = \sum_{i=0}^{N_h} R_i^\top v_i$, being $v_i = R_i(v_h - R_0^\top v_0) \in V_i$, $i = 1,\dots,N_h$, such that

$$\Big|\sum_{i=0}^{N_h} \mathscr{A}_i(v_i,v_i)\Big| \lesssim \|v_h - R_0^\top v_0\|_{h,1}^2 + \mathscr{A}_h(v_h,v_h),$$

where we have employed (4.33), (4.35) and (4.38) with the hypothesis $\mathscr{T}_\mathbb{H} = \mathscr{T}_h$. The thesis then immediately follows by employing (4.47) together with the coercivity of $\mathscr{A}_h$. □

Remark 4.8. *Based on Theorem 4.5, for non-nested coarse and fine spaces V_H and V_h, respectively, the condition number of the additive Schwarz operator can be bounded by*

$$K(P_{ad}) \lesssim \Big(\frac{p^4}{q^2}\frac{H^2}{h^2}\Big)(N_\mathbb{S}+1). \tag{4.51}$$

4.5 Numerical results

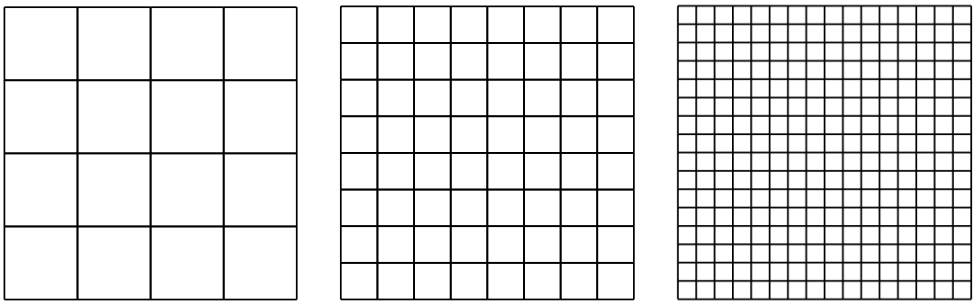


Figure 4.2: Example of nested grids $\mathscr{T}_H$, $\mathscr{T}_\mathbb{H}$ and $\mathscr{T}_h$ with $N_H = 16$, $N_\mathbb{H} = 64$ and $N_h = 256$, respectively.

In order to verify (4.45) we first consider a set of numerical experiments based on employing *nested* grids $\mathscr{T}_h$, $\mathscr{T}_\mathbb{H}$ and $\mathscr{T}_H$ characterized by quadrilateral cartesian elements obtained by successive refinements as shown in Figure 4.2. We solve (4.5) with the Preconditioned Conjugate Gradient method employing B_{ad} as preconditioner (ASPCG) and we report number of iterations needed to reduce the Euclidean norm of the relative residual vector below a tolerance of 10^{-8}. We choose the penalty parameter $C_\sigma = 10$. Then, we estimate the condition number $K(P_{ad})$ by exploiting the extreme eigenvalues estimate based on the PCG iterations, see [Saa03, Sect. 6.7.3],

for example. In Tables 4.1–4.3 we show the h-dependence of the condition number of P_{ad} (and the iteration counts between parentheses) obtained by varying $p \in \{1,2,3\}$ and choosing $q = p$. Each line of the tables is obtained by fixing $N_{\mathbb{H}}$ and $N_H = \frac{N_{\mathbb{H}}}{4}$, such that $H = 2\mathbb{H}$, and by varying N_h such that the size h of the fine grid decreases as $\frac{\mathbb{H}}{2}$, $\frac{\mathbb{H}}{4}$, $\frac{\mathbb{H}}{8}$ and $\frac{\mathbb{H}}{16}$. As expected we observe that the condition number doubles if we halve the size h of the fine grid: indeed, each line of Tables 4.1–4.3 shows that the values of condition number grows linearly by fixing $\mathbb{H}$ and H and by increasing the N_h, i.e., by decreasing h. Similar behavior is observed on each column, where the mesh size h is kept fixed, while $\mathbb{H}$ and H are variable. Moreover, we also observe that the value of the condition number is approximately constant along the main and minor diagonals of each table which is in agreement with our selection of of h, $\mathbb{H}$ and H.

Table 4.1: Condition number (and iteration counts) of ASPCG with $p = q = 1$ on quadrilateral grids.

	$N_h \rightarrow$			
$\downarrow N_H$	256	1024	4096	16384
16	33.44 (46)	71.39 (72)	147.16 (109)	298.46 (159)
64	-	38.46 (56)	80.78 (82)	165.41 (120)
256	-	-	40.07 (62)	83.73 (91)
1024	-	-	-	40.23 (61)

Table 4.2: Condition number (and iteration counts) of ASPCG with $p = q = 2$ on quadrilateral grids.

	$N_h \rightarrow$			
$\downarrow N_H$	256	1024	4096	16384
16	98.40 (81)	198.44 (121)	397.98 (176)	797.03 (254)
64	-	97.04 (91)	196.41 (127)	395.11 (186)
256	-	-	96.50 (90)	195.05 (131)
1024	-	-	-	96.34 (88)

Table 4.3: Condition number (and iteration counts) of ASPCG with $p = q = 3$ on quadrilateral grids.

	$N_h \rightarrow$			
$\downarrow N_H$	256	1024	4096	16384
16	178.82 (103)	358.25 (150)	716.85 (214)	1433.97 (312)
64	-	187.12 (120)	374.92 (168)	750.68 (242)
256	-	-	187.03 (118)	375.09 (169)
1024	-	-	-	186.10 (114)

We now consider a second experiment characterized by the choice $h = \mathbb{H}$. This particular choice is really interesting from the computational point of view because it allows for a massively parallel implementation as the local solvers coincide with the elements of the fine grid $\mathcal{T}_h$, where a very small linear system is solved with a direct method. In this case the bound (**??**) reduces to $K(P_{ad}) = \mathcal{O}\left(\frac{H^2}{h^2}\right)$. The results in this setting are shown in Tables 4.4–4.6 for $p \in \{1,2,3\}$ and $q = p$. Given that each line of the table is obtained by fixing the value of H and by varying $h = \frac{H}{2}, \frac{H}{4}, \ldots,$ the expected quadratic griwth in the condition number is observed. Similar behavior is again observed on each column. With regards the iteration counts shown in parenthesis in Tables 4.1–4.6, we observe the expected rate which is the square root of the one of the condition number, that means a linear growing in this case.

Table 4.4: Condition number (and iteration counts) of the massively parallel ASPCG on nested quadrilateral grids with $p = q = 1$.

	$N_h \rightarrow$				
$\downarrow N_H$	64	256	1024	4096	16384
16	14.78 (26)	43.41 (55)	181.43 (110)	720.46 (220)	2816.89 (440)
64	-	17.67 (38)	43.68 (61)	199.80 (130)	805.01 (265)
256	-	-	18.55 (41)	45.39 (67)	191.46 (131)
1024	-	-	-	18.66 (40)	45.52 (67)
4096	-	-	-	-	18.70 (40)

Table 4.5: Condition number (and iteration counts) of the massively parallel ASPCG on nested quadrilateral grids with $p = q = 2$.

	$N_h \rightarrow$				
$\downarrow N_H$	64	256	1024	4096	16384
16	48.32 (47)	112.65 (94)	403.46 (179)	1516.03 (354)	5869.24 (709)
64	-	47.37 (61)	115.48 (101)	407.03 (188)	1560.31 (370)
256	-	-	47.38 (61)	115.06 (100)	400.94 (187)
1024	-	-	-	47.30 (59)	114.68 (96)
4096	-	-	-	-	47.21 (57)

In order to test the performance of our method in a more general grid setting than the case presented above, we now consider a set of experiments based on employing Voronoi polygonal fine grids $\mathcal{T}_h$: for each grid we then consider a sequence of nested polygonal grids $\mathcal{T}_H$ obtained by successive levels of agglomeration, cf. [AHH$^+$17]. Here, Level 1 represents the fine grid, the grid of Level 2 is obtained by agglomerating elements of Level 1, and so on. The agglomeration of polygonal grids has been performed with *Metis* [KK09]. For each fine grid of size h the agglomeration process has been performed in order to guarantee that the size of the coarser partitions is approximately $H = 2h,\ 4h, \ldots$. Here, we only consider the case $h = \mathbb{H}$. The results

Table 4.6: Condition number (and iteration counts) of the massively parallel ASPCG on nested quadrilateral grids with $p = q = 3$.

	$N_h \rightarrow$				
$\downarrow N_H$	64	256	1024	4096	16384
16	88.81 (63)	189.98 (121)	601.92 (215)	2276.52 (428)	8914.26 (851)
64	-	93.18 (82)	194.21 (121)	627.33 (222)	2384.63 (436)
256	-	-	93.15 (81)	192.75 (120)	627.14 (216)
1024	-	-	-	92.66 (79)	191.71 (116)
4096	-	-	-	-	92.11 (75)

shown in Tables 4.7, 4.8 and 4.9 for $p = q = 1,\ 2$ and 3, respectively, confirm the rate expected from the theory both for the condition number and the iteration count, which is quadratic and linear, respectively. Indeed, by observing from top to bottom each column of Tables 4.7–4.9, where we fix $\mathbb{H}$ and vary $H = 2h, 4h, \ldots$, each numerical value for the condition number is approximately 4 times the one which precede it. Same considerations hold true for the iteration counts, whose value doubles if we double the granularity H. Moreover, the values of the condition number and iteration counts are approximately constant on each line of Tables 4.7–4.9, where a constant ratio between h and H is considered. This behavior is in accord with the theoretical bound (**??**).

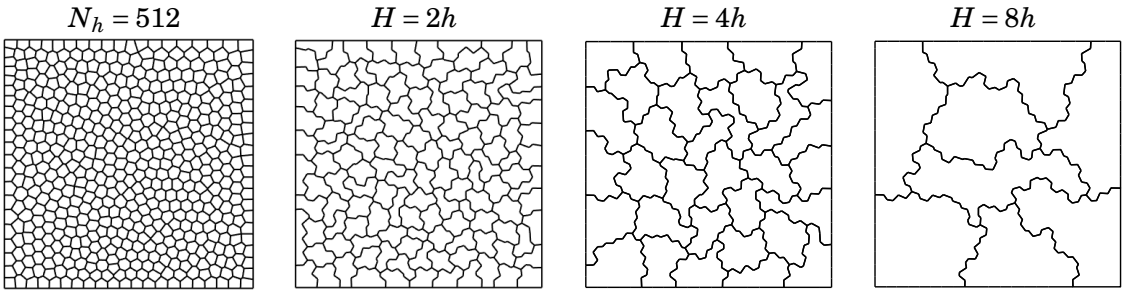


Figure 4.3: Example of a sequence of nested polygonal grids employed for the numerical simulations.

Table 4.7: Condition number (and iteration counts) of the massively parallel ASPCG on nested polygonal grids with $p = q = 1$.

	$N_h = 512$	$N_h = 1024$	$N_h = 2048$	$N_h = 4096$
$H = 2h$	20.70 (45)	21.89 (46)	20.91 (46)	23.08 (48)
$H = 4h$	71.32 (85)	72.31 (86)	73.42 (86)	83.77 (91)
$H = 8h$	265.18 (134)	250.66 (150)	269.70 (163)	261.36 (163)
$H = 16h$	549.35 (184)	636.73 (222)	827.87 (270)	818.09 (289)

Table 4.8: Condition number (and iteration counts) of the massively parallel ASPCG on nested polygonal grids with $p = q = 1$.

	$N_h = 512$	$N_h = 1024$	$N_h = 2048$	$N_h = 4096$
$H = 2h$	54.61 (68)	61.91 (70)	58.22 (68)	60.78 (69)
$H = 4h$	184.63 (127)	178.50 (125)	180.82 (123)	209.67 (128)
$H = 8h$	640.27 (226)	593.27 (234)	726.88 (237)	627.21 (233)
$H = 16h$	1556.61 (345)	1694.94 (365)	2190.17 (432)	1984.41 (437)

Table 4.9: Condition number (and iteration counts) of the massively parallel ASPCG on nested polygonal grids with $p = q = 3$.

	$N_h = 512$	$N_h = 1024$	$N_h = 2048$	$N_h = 4096$
$H = 2h$	88.63 (80)	102.96 (82)	90.30 (79)	104.24 (82)
$H = 4h$	282.14 (148)	291.77 (145)	278.15 (140)	343.19 (148)
$H = 8h$	976.52 (275)	906.59 (270)	1137.55 (276)	949.21 (271)
$H = 16h$	2496.60 (446)	2703.39 (456)	3637.50 (519)	3241.64 (513)

Given the definition of $R_0^\top$ the method presented here allows non-nested coarse spaces $V_H \not\subset V_h$ to be employed. In order to show the validity of (4.45) for the non-nested case, we consider a similar set of experiments presented for quadrilateral grids case with $\mathbb{H} = h$; here, the fine grid $\mathcal{T}_h$ and the coarser one $\mathcal{T}_H$ are both Voronoi polygonal tessellation of size h and $H > h$, respectively, but they are generated independently of each other. In this way $\mathcal{T}_h$ and $\mathcal{T}_H$ are non-nested. The results of Tables 4.10, 4.11 and 4.12 show that the quadratic rate expected from (4.45) for the condition number of P_{ad} is verified also for this case. This behavior is in accord with the theoretical bound (4.51) presented in Section 4.4.

Table 4.10: Condition number (and iteration counts) of the massively parallel ASPCG on non-nested polygonal grids with with $p = q = 1$.

	$N_h \rightarrow$				
$\downarrow N_H$	64	256	1024	4096	16384
16	23.29 (38)	92.13 (77)	387.61 (159)	1624.26 (324)	6370.86 (657)
64	-	25.91 (39)	106.42 (84)	411.02 (167)	1774.19 (342)
256	-	-	26.73 (41)	100.89 (82)	425.97 (169)
1024	-	-	-	31.81 (44)	118.61 (86)
4096	-	-	-	-	30.56 (43)

The dependence of the bound (4.45) with respect to the polynomial degree of approximation is investigated through the following set of experiments based on p-refinement on a fixed couple

Table 4.11: Condition number (and iteration counts) of the massively parallel ASPCG on non-nested polygonal grids with with $p = q = 2$.

	$N_h \rightarrow$				
$\downarrow N_H$	64	256	1024	4096	16384
16	88.07 (66)	264.43 (123)	970.30 (237)	3225.10 (446)	13121.48 (918)
64	-	82.70 (64)	303.61 (124)	1046.59 (238)	3829.80 (466)
256	-	-	83.83 (63)	289.57 (120)	1142.71 (240)
1024	-	-	-	100.16 (65)	313.34 (118)
4096	-	-	-	-	88.89 (60)

Table 4.12: Condition number (and iteration counts) of the massively parallel ASPCG on non-nested polygonal grids with $p = q = 3$.

	$N_h \rightarrow$				
$\downarrow N_H$	64	256	1024	4096	16384
16	148.36 (83)	429.84 (143)	1602.53 (275)	5405.40 (529)	21263.66 (1058)
64	-	142.42 (76)	405.44 (135)	1525.94 (263)	5170.13 (498)
256	-	-	157.47 (80)	452.41 (137)	1469.08 (249)
1024	-	-	-	147.97 (77)	402.98 (124)
4096	-	-	-	-	135.77 (70)

of fine/coarse grids in the setting with $\mathbb{H} = h$. For the nested case, we consider a total of four tests: two of them are characterized by quadrilateral fine grids with $N_h = 256$ and $N_h = 1024$ cells, and two other tests are based on employing the polygonal grids of Figure 4.3 where the fine meshes possess $N_h = 262$ and $N_h = 516$ polygonal elements. For each test the coarse mesh $\mathcal{T}_H$ is obtained by agglomeration of fine elements in order to guarantee $H \eqsim \frac{h}{4}$. The p-dependence of the condition number ($K(P_{ad})$) and the iteration counts ($it(P_{ad})$) is shown in Figures 4.4 (a) and (b), respectively. According to (4.45) the expected bound $K(P_{ad}) \lesssim p$ ($it(P_{ad}) \lesssim \sqrt{p}$) seems to be verified with good accuracy. In order to investigate the p-refinement behavior of $K(P_{ad})$ when $\mathcal{T}_h$ and $\mathcal{T}_H$ are non-nested we repeat the two experiments with the polygonal grids considered for the nested case, but here we choose $\mathcal{T}_H$ to be a Voronoi grid generated independently from $\mathcal{T}_h$. The couples of non-nested grids employed for those experiments are shown in Figure 4.5. In this particular case, Figures 4.4 (a) and (b) show the rates $K(P_{ad}) \lesssim p^2$ and $it(P_{ad}) \lesssim p$, that are in accord with the bound (4.51) given in Section 4.4.

We also investigate the behavior of the condition number of P_{ad} with respect to the diffusion coefficient ρ. Here, we make the assumption that ρ is piecewise constant on Ω. We consider two experiments based on the nested polygonal couples of fine/coarse grids $\mathcal{T}_h/\mathcal{T}_H$, where $\mathcal{T}_h$ is a Voronoi polygonal grid on $\Omega = (0,1)^2$ with 4096 polygonal elements and $\mathcal{T}_H$ is obtained by successive agglomeration of elements of $\mathcal{T}_h$, here $N_H = 16$. Moreover, we choose the polynomial

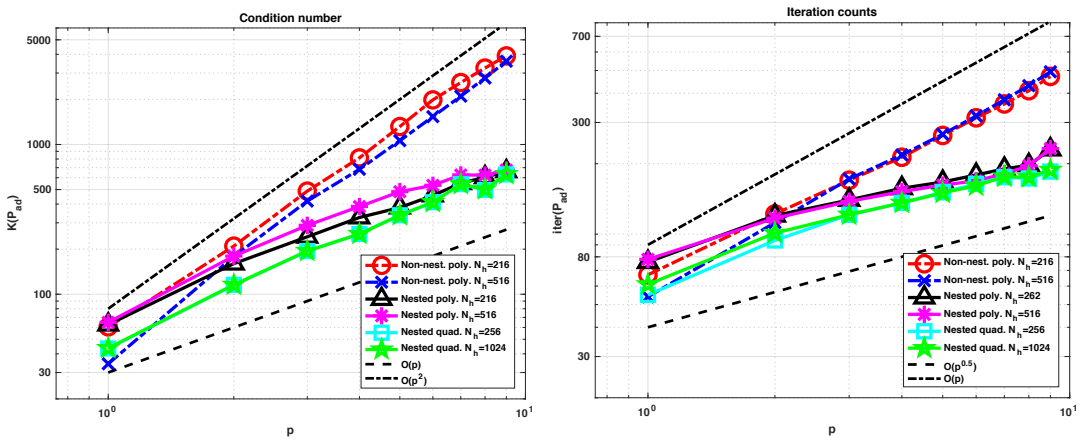


Figure 4.4: Dependence on p of the condition number (left) and the iteration counts (right).

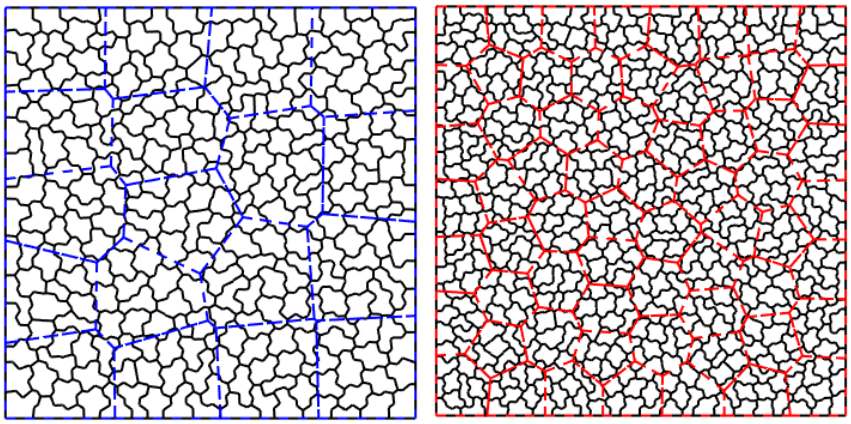


Figure 4.5: Non-nested pairs of fine/coarse grids $\mathcal{T}_h$ (solid) and $\mathcal{T}_H$ (dashed), respectively, employed for investigating p-dependence.

Table 4.13: Condition number as a function of the maximum jump of ρ when $\mathcal{T}_H$ is aligned with the discontinuities of ρ ($p = q = 1$ and $p = q = 2$).

	$\rho_e \rightarrow$						
	1	10	10^2	10^3	10^4	10^5	10^6
$p = 1$	$8.18 \cdot 10^2$	$8.22 \cdot 10^2$	$8.57 \cdot 10^2$	$8.62 \cdot 10^2$	$8.62 \cdot 10^2$	$8.61 \cdot 10^2$	$8.61 \cdot 10^2$
$p = 2$	$1.98 \cdot 10^3$	$2.06 \cdot 10^3$	$2.09 \cdot 10^3$	$2.09 \cdot 10^3$	$2.09 \cdot 10^3$	$2.09 \cdot 10^3$	$2.09 \cdot 10^3$

degree as $p = q = 1$ or $p = q = 2$. In the first experiment we fix $\rho|_{\mathcal{D}_j} = \rho_o = 1$ on the elements $\mathcal{D}_j \in \mathcal{T}_H$ with odd index j and we set $\rho|_{\mathcal{D}_j} = \rho_e \in \{10^0, 10^1, \ldots, 10^6\}$ on the polygonal sub-domains with even index j. The results of Table 4.13 show the independence with respect to the jumps of ρ when those jumps are aligned with the sub-domains of $\mathcal{T}_H$. In the second experiment we proceed in a similar fashion but here we take different values of ρ on odd and even polygonal elements $\kappa \in \mathcal{T}_h$: in this way $\mathcal{T}_H$ is not aligned with the discontinuities of ρ, then the maximum

Table 4.14: Condition number as function of the maximum jump of ρ when $\mathcal{T}_H$ is not aligned with the discontinuities of ρ ($p = q = 1$ and $p = q = 2$).

	$\rho_e \rightarrow$						
	1	10	10^2	10^3	10^4	10^5	10^6
$p = 1$	$8.18 \cdot 10^2$	$9.47 \cdot 10^2$	$2.89 \cdot 10^3$	$2.13 \cdot 10^4$	$2.02 \cdot 10^5$	$2.01 \cdot 10^6$	$2.01 \cdot 10^7$
$p = 2$	$1.98 \cdot 10^3$	$2.39 \cdot 10^3$	$7.75 \cdot 10^3$	$6.34 \cdot 10^4$	$6.20 \cdot 10^5$	$6.18 \cdot 10^6$	$6.18 \cdot 10^7$

jump inside the polygonal sub-domains $\mathcal{D}_j \in \mathcal{T}_H$ is given by $\rho_e - 1$. As expected from the theory, the results presented in Table 4.14 show that the condition number of P_{ad} grows as ρ_e.

We now consider the performance of the ASPCG algorithm on a tetrahedral meshes in three-dimensions. To this end, we set $\Omega = (0,1)^3$; furthermore, the elements of the coarse mesh are general shaped polyhedra obtained by successive agglomeration, cf., the previous example. The results for $p = q = 1$ and $p = q = 3$ are reported in Tables 4.15 and 4.16, respectively. Here, we have also added a line with the condition number of the operator $A_h : V_h \times V_h \rightarrow V_h$ defined as $(A_h u_h, v_h)_{L^2(\Omega)} = \mathcal{A}_h(u_h, v_h)$ for all $u_h, v_h \in V_h$, and, in parenthesis, the iteration counts of the Conjugate Gradient method for solving (1.8) without preconditioning. Analogous behaviour of the condition number and iteration counts to those presented in the previous example are observed. In particular, we observe that the condition number is roughly constant on the diagonals and sub-diagonals of the two tables, while, along each row, i.e., when $\mathcal{T}_H$ is fixed, the expected quadratic growth in $K(P_{ad})$ is observed. Similar considerations are also noted for the iteration counts.

Table 4.15: Condition number (and iteration counts) of the massively parallel ASPCG on nested polyhedral grids with $p = q = 1$.

	$N_h \rightarrow$					
$\downarrow N_H$	384	3072	24576	196608	1572864	12582912
48	107 (85)	411 (156)	1497 (294)	6216 (580)	25791 (1089)	94276 (2012)
384	-	136 (95)	499 (169)	1878 (311)	7407 (584)	28762 (1106)
3072	-	-	146 (96)	480 (165)	1904 (306)	7861 (578)
24576	-	-	-	144 (94)	491 (164)	1973 (306)
196608	-	-	-	-	144 (94)	496 (164)
1572864	-	-	-	-	-	145 (94)

Table 4.16: Condition number (and iteration counts) of the massively parallel ASPCG on nested polyhedral grids with $p = q = 3$.

	$N_h \rightarrow$			
$\downarrow N_H$	384	3072	24576	196608
48	607.41 (174)	2120.20 (309)	6760.65 (515)	26674.37 (924)
384	-	655.20 (179)	2334.88 (314)	7507.31 (536)
3072	-	-	693.12 (182)	2295.10 (316)
24576	-	-	-	697.82 (182)

The preconditioner presented in this chapter can be employed as smoothing operator in the V-cycle multigrid method introduced in Chapter 3, in a similar manner as the additive Schwarz

Table 4.17: Convergence factor (and iteration counts) of the V-cycle solver with the Additive Schwarz smoother as a function of m (airfoil profile test case, $p = 1$).

	set 1			set 2			set 3		
	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.
$m = 3$	0.30 (12)	0.27 (11)	0.28 (11)	0.37 (15)	0.34 (14)	0.35 (14)	0.39 (15)	0.39 (15)	0.39 (15)
$m = 5$	0.09 (6)	0.09 (6)	0.09 (6)	0.14 (8)	0.14 (8)	0.14 (8)	0.14 (8)	0.14 (8)	0.14 (8)
$m = 8$	0.01 (4)	0.01 (4)	0.01 (4)	0.02 (4)	0.02 (4)	0.02 (4)	0.02 (4)	0.02 (4)	0.02 (4)

Table 4.18: Convergence factor (and iteration counts) of the V-cycle solver with the Additive Schwarz smoother as a function of m (airfoil profile test case, $p = 2$).

	set 1			set 2			set 3		
	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.
$m = 3$	0.63 (30)	0.57 (25)	0.58 (26)	0.64 (32)	0.63 (31)	0.62 (29)	0.66 (34)	0.63 (30)	0.62 (30)
$m = 5$	0.30 (12)	0.30 (12)	0.30 (12)	0.33 (13)	0.34 (13)	0.34 (13)	0.36 (14)	0.34 (13)	0.34 (13)
$m = 8$	0.09 (6)	0.10 (6)	0.10 (6)	0.12 (7)	0.12 (7)	0.12 (7)	0.12 (7)	0.12 (7)	0.12 (7)

Table 4.19: Convergence factor (and iteration counts) of the V-cycle solver with the Additive Schwarz smoother as a function of m (airfoil profile test case, $p = 3$).

	set 1			set 2			set 3		
	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.
$m = 3$	0.71 (40)	0.78 (55)	0.72 (43)	0.75 (48)	0.73 (44)	0.75 (49)	0.71 (41)	0.68 (36)	0.69 (38)
$m = 5$	0.35 (14)	0.42 (16)	0.35 (14)	0.44 (17)	0.44 (17)	0.45 (18)	0.42 (16)	0.41 (16)	0.41 (16)
$m = 8$	0.13 (7)	0.13 (7)	0.12 (7)	0.19 (9)	0.19 (9)	0.2 (9)	0.17 (8)	0.17 (8)	0.17 (8)

Table 4.20: Convergence factor (and iteration counts) of the V-cycle solver with the Additive Schwarz smoother as a function of m (airfoil profile test case, $p = 4$).

	set 1			set 2			set 3		
	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.	2 lev.	3 lev.	4 lev.
$m = 3$	0.81 (68)	0.89 (123)	0.82 (70)	0.81 (68)	0.79 (60)	0.87 (104)	0.79 (59)	0.77 (52)	0.80 (61)
$m = 5$	0.54 (23)	0.60 (27)	0.55 (24)	0.56 (24)	0.54 (23)	0.54 (23)	0.50 (20)	0.49 (20)	0.49 (20)
$m = 8$	0.24 (10)	0.27 (11)	0.29 (12)	0.27 (11)	0.27 (11)	0.33 (13)	0.23 (10)	0.23 (10)	0.23 (10)

smoother considered in Section 3.4. Here, we solve problem (4.5) on the domain with circular boundary and airfoil profile shaped hole introduced in Section 3.4.1, by employing the multigrid V-cycle solver based on the same sequence of polygonal partitions considered for the tests in Chapter 3. Tables 4.17–4.20 show the result in terms of convergence factor (and iteration counts) for $p = 1,\dots,4$, respectively. First, we observe that also in this case the convergence factor is uniform with respect to the granularity of the fine mesh and the number of levels employed, while it depends on the polynomial degree of approximation when the number of smoothing steps m is not chosen sufficiently large. Each simulation confirms that the multigrid solver with the smoother operator presented here outperforms the iterative method analyzed in Chapter 3. Moreover, besides the strong irregularity of the domain, the methods seems to converge also for high order polynomial approximation degree, showing a regular behavior which was not observed in Section 3.4.1 for $p > 2$.

4.5.1 Application to multigrid methods

In this section we aim to accelerate the convergence properties of multigrid methods with the additive Schwarz preconditioner by also optimizing the number of computations. The method is defined as follows. Let $L \in \mathbb{N}$, $L > 1$ and let $\mathcal{T}_1, \dots, \mathcal{T}_L$ be a sequence of polytopic grids such that $h_l \geq h_{l+1}$, $l = 1, \dots, L-1$, where $h_l = \max_{\kappa \in \mathcal{T}_l} diam(\kappa)$. Here $\mathcal{T}_L = \mathcal{T}_h$ represents the fine polytopic grid where problem (1.8) is defined. For each $\mathcal{T}_l$ we define the corresponding DG spaces V_l as (1.3).

Remark 4.9. *The sequence of grids $\{\mathcal{T}_l\}_{l=1}^L$ is possibly non-nested and it can be obtained, for example, by successive applications of agglomeration and edge coarsening techniques. L represents the number of levels of the multigrid iteration.*

We define the prolongation operators $I_{l-1}^l : V_{l-1} \to V_l$, $l = 2, \dots, L$, as

$$(I_{l-1}^l w_{l-1}, v_l)_{L^2(\Omega)} = (w_{l-1}, v_l)_{L^2(\Omega)} \quad \forall\, v_l \in V_l,$$

where $(\cdot,\cdot)_{L^2(\Omega)}$ is the $L^2(\Omega)$-inner product. The restriction operator $I_l^{l-1} : V_l \to V_{l-1}$ is defined as the adjoint of I_{l-1}^l with respect to the $L^2(\Omega)$-inner product, i.e.,

$$(I_l^{l-1} v_l, w_{l-1})_{L^2(\Omega)} = (v_l, I_{l-1}^l w_{l-1})_{L^2(\Omega)} \quad \forall w_{l-1} \in V_{l-1},\ v_l \in V_l.$$

Then, we introduce the operators $A_l : V_l \to V_l$ such that

$$\begin{aligned} (A_L w_L, v_L)_{L^2(\Omega)} &= \mathscr{A}_h(w_L, v_L) \quad \forall\, w_L, v_L \in V_L = V_h, \\ (A_l w_l, v_l)_{L^2(\Omega)} &= \mathscr{A}_h(I_l^L w_l, I_l^L v_l) \quad \forall w_l, v_l \in V_l,\ l = 1, \dots, L-1, \end{aligned} \tag{4.52}$$

where $I_l^L = I_{L-1}^L \cdots I_l^{l+1}$, $\forall\, l = 1, \dots, L-1$. By observing that problem (1.8) can be written as

$$\text{find } u_L \in V_L :\ A_L u_L = f_L,$$

where $f_L \in V_L$ such that $(f_L, v_L)_{L^2(\Omega)} = \int_\Omega f v_L \mathrm{d}\mathbf{x}$ $\forall\, v_L \in V_L$, given an initial guess $u_L^0 \in V_L$ and $m \in \mathbb{N}$, the geometric multigrid algorithm for the solution of (1.8) is outlined in Algorithm 6. In particular, $\mathsf{MG}(L, f_L, u^{(k)}, m)$ represents the approximate solution obtained after one iteration of the V-cycle scheme with L levels, which is defined by induction in Algorithm 7. The smoothing steps are given by the algorithm $ASPCG$, i.e., $z = ASPCG(A, z^{(0)}, g, m)$ represents the output of m steps of *Preconditioned Conjugate Gradient* method applied to the linear system of equations $Ax = g$, by using the massively parallel Additive Schwarz preconditioner and starting from the initial guess $z^{(0)}$. For each level, the preconditioner is defined as in Section 4.1 and it is based on the splitting $V_l = \sum_{i=0}^{N_l} V_{l,i}$, $l = 2, \dots, L$, where N_l is the number of elements of $\mathcal{T}_l$ and $V_{l,0} = V_1$, $l = 2, \dots, L$, being V_1 the coarsest discrete space of the multigrid space sequence $\{V_l\}_{l=1}^L$.

Remark 4.10. *We point out that, from the implementation point of view, the method only requires to solve, in parallel, very small local problems, one on each element of the grids $\mathcal{T}_l$, $l > 1$, and the discrete problem posed on the coarsest level V_1.*

Algorithm 6 Multigrid V-cycle algorithm

Initialize $u^{(0)} \in V_J$;
for $k = 0, 1, \dots$ **do**
  $u^{(k+1)} = \mathsf{MG}(L, f_L, u^{(k)}, m)$;
  $u^{(k)} = u^{(k+1)}$;
end for

Algorithm 7 One iteration of Multigrid V-cycle scheme

Pre-smoothing:
if l=1 **then**
  $\mathsf{MG}(1, g, z^{(0)}, m) = A_1^{-1} g$.
else
  Pre-smoothing:
  $z^{(m)} = ASPCG(A_l, z^{(0)}, g, m)$;

  Coarse grid correction:
  $r_{l-1} = I_l^{l-1}(g - A_l z^{(m)})$;
  $e_{l-1} = \mathsf{MG}(l-1, r_{l-1}, 0, m)$;
  $z^{(m+1)} = z^{(m)} + I_{l-1}^{l} e_{l-1}$;

  Post-smoothing:
  $z^{(2m+1)} = ASPCG(A_l, z^{(m+1)}, g, m)$;

  $\mathsf{MG}(l, g, z_0, m) = z^{(2m+1)}$.
end if

Table 4.21: AS coarse operator: inherited; Multigrid operators: inherited.

		Nested				Non-nested			
		L = 2	L = 3	L = 4	L = 5	L = 2	L = 3	L = 4	L = 5
$\mathscr{T}_1$	p=1	12	17	24	35	8	11	14	13
	p=2	21	27	40	62	12	19	22	22
	p=3	25	31	49	86	16	25	27	27
$\mathscr{T}_2$	p=1	12	17	23	39	8	11	14	13
	p=2	20	27	45	69	13	18	20	21
	p=3	24	32	52	100	16	25	26	26
$\mathscr{T}_3$	p=1	12	17	24	39	8	11	13	13
	p=2	22	27	40	70	14	17	19	22
	p=3	25	32	52	89	16	23	27	27

We fix the number of pre- and post-smoothing steps $m = 3$ and we investigate the efficiency of Algorithm 6 to solve problem (1.8) with $d = 2$ and $\Omega = (0,1)^2$. In particular, we measure the number of iterations needed to reduce the norm of the relative residual below a given tolerance of 10^{-8}, by varying the size h of the finest grid, i.e. its number of elements N_h, the polynomial degree p and the number of levels L. Here, we consider three grids $\{\mathscr{T}_n\}_{n=1}^3$ with $N_{h_1} = 1024$, $N_{h_2} = 2048$ and

Table 4.22: AS coarse operator: computed; Multigrid operators: computed.

		Nested				Non-nested			
		L = 2	L = 3	L = 4	L = 5	L = 2	L = 3	L = 4	L = 5
$\mathscr{T}_1$	p=1	16	28	48	-	11	19	28	37
	p=2	28	57	98	180	20	35	56	100
	p=3	35	75	124	229	26	43	80	136
$\mathscr{T}_2$	p=1	16	26	52	-	10	19	27	53
	p=2	28	50	98	-	20	33	53	111
	p=3	35	66	130	-	28	43	71	174
$\mathscr{T}_3$	p=1	18	32	-	-	10	19	25	45
	p=2	30	60	190	335	20	31	45	88
	p=3	38	85	255	-	24	38	62	132

Table 4.23: AS coarse operator: inherited; Multigrid operators: computed.

		Nested				Non-nested			
		L = 2	L = 3	L = 4	L = 5	L = 2	L = 3	L = 4	L = 5
$\mathscr{T}_1$	p=1	11	15	18	19	8	11	13	12
	p=2	21	23	28	33	15	21	24	22
	p=3	26	28	36	42	23	27	29	30
$\mathscr{T}_2$	p=1	11	15	18	19	8	11	13	12
	p=2	20	23	29	31	15	20	23	23
	p=3	24	28	33	38	22	25	28	29
$\mathscr{T}_3$	p=1	12	16	19	20	8	12	13	12
	p=2	20	24	28	34	16	19	22	23
	p=3	25	27	35	41	22	24	28	29

$N_{h_3} = 4096$ number of elements, respectively, we vary $p \in \{1,2,3\}$ and $L = 2,\dots,5$. For each $\mathscr{T}_n$, we define two sequences of grids: one characterized by nested polygonal grids obtained by successive agglomeration, cf. Figure 4.3, and the other one composed by non-nested polygonal Voronoi grids independently generated the one from each other. The results are shown in Table 4.21. The table shows that, in accord with the multigrid analysis presented in [AP18], the iteration counts are independent of h but they depend on p as the number of smoothing steps is kept fixed. Each line of Table 4.21 shows the dependence on the number of levels L employed for the multigrid iteration. According to the analysis presented in [ASV15, Section 5], as the quadratic forms of the multigrid sub-levels are defined by restriction, the results of Table 4.21 related to the nested sequences show that the iteration counts increase by increasing L. However, by observing the results related to the non-nested sequences we deduce that, at least for a sequence of Voronoi grids, the iteration counts are independent of L for non-nested grids.

The choice of the operators $\{A_l\}_{l=1}^{L}$ as (4.52) is not univocal. Indeed, in order to guarantee the independence also with respect to the number of levels L, a possible choice is to define A_l on each sub-level without restriction. For each $l \in \{1,\dots,L-1\}$, let $\mathscr{A}_l(\cdot,\cdot) : V_l \times V_l \to \mathbb{R}$ be the bilinear

form (1.5) defined on the space V_l. Then we define A_l as

$$(A_l w_l, v_l)_{L^2(\Omega)} = \mathscr{A}_l(w_l, v_l) \quad \forall w_l, v_l \in V_l,\ l = 1, \ldots, L-1. \tag{4.53}$$

Remark 4.11. *Note that the implementation of the choice* (4.52) *only requires the computation of the matrix version of* I_l^{l+1}*, while the choice* (4.53) *also requires the direct computation of the matrix version of* A_l*.*

The results obtained by solving (1.8) with the choice (4.53) are shown in Table 4.22 for nested and non-nested grids. The behavior observed In this case is worse than the case described above, because, here, we have dependence with respect to L also when non-nested grids are employed. Moreover, the method might fail to converge when a relative high number of levels are employed. This bad behavior is motivated by the fact that the choice (4.53) is not optimal from the point of view of the additive Schwarz smoother, where restriction is required in order to apply the theory presented in Section 4.3. Then, we also present a final test where the operators of the multigrid iteration are defined as (4.53), while the operators of the additive Schwarz smoother, employed in the algorithm ASPCG, are defined by restriction as (4.52). The results of Table 4.23 show that this choice is the optimal one from the theoretical point of view as it allows to assure that the number of iterations needed to attain the convergence is also independent of the number of levels L, both for nested and for non-nested grids.

CONCLUSIONS AND FURTHER PERSPECTIVES

In this thesis we have introduced and analyzed a class of high-order discontinuous Galerkin discretizations of second-order elliptic partial differential equations on polytopic meshes. We have focused on the implementation aspects of the method and the design of efficient solution techniques for the linear system of equation stemming from the presented method. The main original results can be summarized as follows.

- We have proposed a new approach for the numerical evaluation of the integrals required to assemble the mass and stiffness matrices arising from the DG discretizations of second order elliptic problems, where the underlying mesh is composed by polygonal/polyhedral elements. Exploiting the idea introduced in [CLS15], we have described in details a new efficient method for the fast integration of polynomial *homogeneous functions*. The method is based on successive applications of Stokes' theorem. The latter allows us to exactly compute the underlying numerical integral using only the values of the integrand at the vertices of the polytopic domain, and hence leads to an exact cubature rule whose quadrature points are the vertices of the polytope. We have analyzed in detail the computational complexity of the new algorithm in integrating two- and three-dimensional polynomial functions, and we have shown that our method outperforms classical quadrature rules, such as the sub-tessellation algorithm coupled with Gaussian integration, also for triangular-shaped domains. Exploiting this fast integration algorithm, we have described a technique for efficiently assemble the mass and stiffness matrices arising from the DG discretizations. We have shown that this approach presents a remarkable gain in terms of CPU time with respect to classical quadrature rules, maintaining the same degree of accuracy. On one hand, the number of computations is optimized with respect to the polynomial degree of the integrand, on the other, less memory storage is required as no sub-tessellation and quadrature nodes and weights are required.

- We have analyzed a V-cycle multigrid solver for symmetric hp-version DG methods on polygonal/polyhedral grids. Our method generalizes the W-cycle multigrid method on nested polytopic grids of [AHH$^+$17] to V-cycle algorithms with non-nested meshes. Here, the possibility to employ non-nested polytopic meshes allows us to choose the sequence of grids standing at the basis of the multigrid method based on employing agglomeration procedures together with edge-coarsening; thanks to this possibility we avoid agglomerated

coarser grids characterized by an increasing number of faces so that the overall efficiency of the solvers is improved. The analysis of the proposed solver is based on the general abstract framework presented in [BPX91], where geometric multilevel V-cycle method with non-inherited quadratic forms is analyzed. One iteration of the proposed method consists of an iterative application of the smoothing Richardson operator and a recursive subspace correction step on non-nested coarser spaces. Here, the key point is the *projection operator* which is defined as the L^2-projection between two consecutive (non-nested) partitions. We have proved that our non-nested multigrid method converges uniformly with respect to the number of multigrid levels and the discretization parameters, i.e. the size of the fine grid and the polynomial approximation degree p, provided that the number of smoothing steps is chosen of order $p^{2+\mu}$, $\mu \in \{0,1\}$. We have shown that the proposed algorithm is numerically effective when enough smoothing steps are chosen, and we also have demonstrated that the multigrid iteration converges even though this hypothesis is not strictly satisfied; in the latter case, the convergence factor of the method is uniform with respect to the number of levels and the size of the underlying fine grid, but it increases with p. We have then proposed a further improvement of the method by considering a two-level additive Schwarz preconditioners employed as smoothing operator instead of the Richardson iteration. Also in this case, the solvers on the coarser space are non-inherited. The improvement of this smoother in terms of iteration counts needed to reduce the residual under a given tolerance has been proved through several numerical experiments, also for geometries with curved boundaries, where a non-nested multigrid framework is needed as the coarser grids may not fit the geometry. From the implementation point of view, we point out that the assembly of the prolongation and projection matrices needs the knowledge of the intersections between elements of two consecutive levels. Our computations make use of the tool *PolygonClipper* [Hol], but its extension to the three dimensional case could be expensive. In three dimensions, agglomeration-based procedures which make use of edge-coarsening techniques can also be used to generate the sequence of meshes in the three dimensional case.

- We have designed and analyzed a class of additive Schwarz preconditioner for the linear system of equations arising from DG discretizations of second order elliptic problems with polytopic grids. The preconditioner is based on a partition composed of disjoint polytopic subdomains, where local solvers are applied in parallel, and a non-overlapping partition employed for the coarse space correction. Here the subdomains can be chosen as small as a single element of the fine mesh, leading to a massively parallel implementation of the preconditioner. We have analyzed the preconditioner under the assumption that the coarse and fine grids are nested. In particular, the condition number of the resulting preconditioned system is shown to be of order $\mathcal{O}(\frac{\max\rho}{\min\rho}\frac{H^2}{h^2}\frac{p^2}{q})$, where ρ is the diffusion coefficient, h and H are the sizes of the fine and coarse grids respectively, while p and q are the

polynomial approximation degrees employed on the fine and the coarse spaces, respectively. Several numerical tests have been presented to confirm the theoretical bound. We have also investigated numerically the behavior of the condition number of the preconditioned system when the coarse and the fine grids are non-nested. The numerical results confirm the same dependence with respect to H and h, but the dependence with respect to p seems to be cubic rather then quadratic, i.e. the estimate seems to be $\mathcal{O}(\frac{H^2}{h^2}\frac{p^3}{q})$. The theoretical analysis of the non-nested case is under investigation. Furthermore, if the diffusion coefficient ρ is piecewise constant on each subdomain of the coarse partition, then the numerical results shows that the condition number does not depend on the jump of the coefficients, i.e. the estimate becomes $\mathcal{O}(\frac{H^2}{h^2}\frac{p^2}{q})$.

BPX ABSTRACT CONVERGENCE ANALYSIS

In this section we report the details of the BPX convergence analysis for the V-cycle multigrid algorithms employed in Section 3.2.1. More precisely, we report the proof of Theorem 3.1 presented in [DGTZ07]. For $j = 1, \dots, J$, let V_j be a sequence of finite-dimensional vector spaces and let $\mathscr{A}_j : V_j \times V_j \to \mathbb{R}$ be symmetric positive definite bilinear forms. We solve the following problem: given $f \in L^2(\Omega)$, find $u \in V_j$ satisfying

$$\mathscr{A}_j(u, v) = (f, v)_{L^2(\Omega)} \qquad \forall v \in V_j.$$

Let the operators $A_j : V_j \to V_j$, $I_j^{j-1} : V_j \to V_{j-1}$ and $P_j^{j-1} : V_j \to V_{j-1}$ be defined as in Section 3.2, and let $B_j : V_j \to V_j$ be a generic smoother operator. To analyze the convergence, we set $G_j = \mathrm{Id}_j - B_j^{-1} A_j$ and $G_j^* = \mathrm{Id}_j - (B_j^{-1})^* A_j$, where G_j^* denotes the adjoint of G_j with respect to $\mathscr{A}_j(\cdot, \cdot)$, and set

$$\tilde{G}_j^m = \begin{cases} (G_j^* G_j)^{\frac{m}{2}} & \text{if } m \text{ is even} \\ (G_j^* G_j)^{\frac{m-1}{2}} G_j^* & \text{if } m \text{ is odd} \end{cases}$$

We consider the multigrid V-cycle method shown in Algorithm 4 with $m_1 = m_2 = m$, for which the error propagation operator can be written as

$$\begin{cases} \mathbb{E}_{1,m} v &= 0 \\ \mathbb{E}_{j,m} v &= (\tilde{G}_j^m)^* (\mathrm{Id}_j - I_{j-1}^j P_j^{j-1} + I_{j-1}^j \mathbb{E}_{j-1,m} P_j^{j-1})(\tilde{G}_j^m) v, \; j = 2, \dots, J, \end{cases} \tag{54}$$

We recall the three standard Hypotheses **H.1**, **H.2** and **H.3** given in Section 3.2.1:

H.1 Stability estimate: $\exists\, C_Q > 0$ such that

$$|||(\mathrm{Id}_j - I_{j-1}^j P_j^{j-1}) v_h|||_{1,j} \le C_Q |||v_h|||_{1,j} \qquad \forall v_h \in V_j, \quad j = 2, \dots, J.$$

H.2 Regularity-approximation property: $\exists\, C_A > 0$ such that

$$\left| \mathscr{A}_j((\mathrm{Id}_j - I_{j-1}^j P_j^{j-1}) v_h, v_h) \right| \le C_A \frac{|||v_h|||_{2,j}^2}{\Lambda_j} \qquad \forall v_h \in V_j, \quad j = 2, \dots, J.$$

H.3 Smoothing property: $\exists\, C_S > 0$ such that

$$\frac{\|v_h\|_{L^2(\Omega)}}{\Lambda_j} \le C_S \left(\mathbb{S} v_h, v_h \right)_{L^2(\Omega)} \qquad \forall v_h \in V_j, \quad j = 2, \dots, J,$$

where $\mathbb{S} = \left(\mathrm{Id}_j - G_j^* G_j \right) A_j^{-1}$.

Then, the convergence of Algorithm 4 is stated in the following abstract theorem.

Theorem .6. *Assume that Hypotheses **H.1**, **H.2** and **H.3** hold. Then, for $j = 2, \dots, J$:*

$$\left|\mathscr{A}_j(\mathbb{E}_{j,m}u, u)\right| \leq \delta_j \mathscr{A}_j(u,u) \qquad \forall u \in V_j, \tag{55}$$

where

$$\delta_j = \frac{C_A C_S}{m - C_A C_S}, \tag{56}$$

provided that $m > 2C_A C_S$.

Proof. As shown in [BPX87] and [BPX91], we proceed by induction. For $j = 1$, we have a zero on the left-hand side of (55) because $\mathbb{E}_{1,m}v = 0$ as consequence of its definition given in (54). Then inequality (55) holds for $j = 1$. We now assume that the bound (55) holds for the level $j-1$ and we show that it holds for level j. In view of Hypothesis **H.2**, we have

$$\left|\mathscr{A}_j((\mathrm{Id}_j - I_{j-1}^j P_j^{j-1})\tilde{G}_j^m u, \tilde{G}_j^m u)\right| \leq C_A \frac{\|A_j G_j^m u\|^2_{L^2(\Omega)}}{\Lambda_j}. \tag{57}$$

Next, we define

$$\overline{G}_j = \begin{cases} G_j^* G_j & \text{if } m \text{ is even,} \\ G_j G_j^* & \text{if } m \text{ is odd.} \end{cases}$$

By Hypothesis **H.3** we have

$$\frac{\|A_j \tilde{G}_j^m u\|^2_{L^2(\Omega)}}{\Lambda_j} \leq C_S \mathscr{A}_j((\mathrm{Id}_j - \overline{G}_j)\overline{G}_j^m u, u). \tag{58}$$

Since the spectrum of $\overline{G}_j$ is contained in $[0,1]$, as shown in [BPX91], we have

$$\mathscr{A}_j((\mathrm{Id}_j - \overline{G}_j)\overline{G}_j^m u, u) \leq \frac{1}{m}\sum_{i=0}^{m-1} \mathscr{A}_j((\mathrm{Id}_j - \overline{G}_j)\overline{G}_j^i u, u) = \frac{1}{m}\{\mathscr{A}_j(u,u) - \mathscr{A}_j(\overline{G}_j^m u, u)\}.$$

Note that $\mathscr{A}_j(\overline{G}_j^m u, u) = \mathscr{A}_j(\tilde{G}_j^m u, \tilde{G}_j^m u)$ by (57)-(58). We then get

$$\left|\mathscr{A}_j((\mathrm{Id}_j - I_{j-1}^j P_j^{j-1})\tilde{G}_j^m u, \tilde{G}_j^m u)\right| \leq \frac{C_A C_S}{m}\{\mathscr{A}_j(u,u) - \mathscr{A}_j(\tilde{G}_j^m u, \tilde{G}_j^m u)\}. \tag{59}$$

Next, by setting

$$t = \frac{\mathscr{A}_j(\tilde{G}_j^m u, \tilde{G}_j^m u)}{\mathscr{A}_j(u,u)} \qquad \forall u \neq 0, u \in V_j,$$

and $t = 0$ for $u = 0$, we observe that $t \in [0,1]$. We now rewrite (59) as

$$\left|\mathscr{A}_j((\mathrm{Id}_j - I_{j-1}^j P_j^{j-1})\tilde{G}_j^m u, \tilde{G}_j^m u)\right| \leq \frac{C_A C_S (1-t)}{m}\mathscr{A}_j(u,u). \tag{60}$$

On the other hand, from the Cauchy-Schwarz inequality and Hypothesis **H.1** we have

$$
\begin{aligned}
(61)\qquad & \left|\mathscr{A}_j((\mathrm{Id}_j - I_{j-1}^j P_j^{j-1})\tilde{G}_j^m u, \tilde{G}_j^m u)\right| \\
& \le \{\mathscr{A}_j((\mathrm{Id}_j - I_{j-1}^j P_j^{j-1})\tilde{G}_j^m u, (\mathrm{Id}_j - I_{j-1}^j P_j^{j-1})\tilde{G}_j^m u)\}^{\frac{1}{2}} \{\mathscr{A}_j(\tilde{G}_j^m u, \tilde{G}_j^m u)\}^{\frac{1}{2}} \\
& \le C_Q \mathscr{A}_j(\tilde{G}_j^m u, \tilde{G}_j^m u) = C_Q t \mathscr{A}_j(u,u).
\end{aligned}
$$

Combining (60) and (61), we get

$$
\left|\mathscr{A}_j((\mathrm{Id}_j - I_{j-1}^j P_j^{j-1})\tilde{G}_j^m u, \tilde{G}_j^m u)\right| \le \min\left\{C_Q t, \frac{C_A C_S}{m}(1-t)\right\}\mathscr{A}_j(u,u).
$$

By the relation

$$
\mathscr{A}_j(P_j^{j-1}\tilde{G}_j^m u, P_j^{j-1}\tilde{G}_j^m u) = \mathscr{A}_j(\tilde{G}_j^m u, \tilde{G}_j^m u) - \mathscr{A}_j(\tilde{G}_j^m u, (\mathrm{Id}_j - I_{j-1}^j P_j^{j-1})\tilde{G}_j^m u),
$$

the induction hypothesis and the symmetry of $\mathscr{A}_j(\cdot,\cdot)$, we get

$$
\begin{aligned}
\left|\mathscr{A}_j(\mathbb{E}_{j,m}u, u)\right| &\le \left|\mathscr{A}_j((\mathrm{Id}_j - I_{j-1}^j P_j^{j-1})\tilde{G}_j^m u, \tilde{G}_j^m u)\right| + \left|\mathscr{A}_{j-1}(\mathbb{E}_{j-1,m} P_j^{j-1}\tilde{G}_j^m u, P_j^{j-1}\tilde{G}_j^m u)\right| \\
&\le (1+\delta)\left|\mathscr{A}_j((\mathrm{Id}_j - I_{j-1}^j P_j^{j-1})\tilde{G}_j^m u, \tilde{G}_j^m u)\right| + \delta\mathscr{A}_j(\tilde{G}_j^m u, \tilde{G}_j^m u) \\
&\le (1+\delta)\min\left\{C_Q t, \frac{C_A C_S}{m}(1-t)\right\}\mathscr{A}_j(u,u) + \delta t \mathscr{A}_j(u,u).
\end{aligned}
$$

Now, to show that (55) holds for the level j, we only need to verify

$$
(62)\qquad (1+\delta)\min\left\{C_Q t, \frac{C_A C_S}{m}(1-t)\right\} + \delta t \le \frac{C_A C_S}{m - C_A C_S} \qquad \forall t \in [0,1].
$$

When $t = 0$, the left-hand side of (62) is zero, whereas for $t = 1$, (62) is the induction hypothesis. Next, we consider the case $t \in (0,1)$. By the induction hypothesis (56) on level $j-1$, it is sufficent to show that

$$
(1+\delta)C_Q \min\left\{\frac{t}{1-t}, \frac{C_A C_S}{C_Q m}\right\} \le \delta.
$$

We consider two cases. If

$$
\frac{C_A C_S}{C_Q m + C_A C_S} \le t < 1,
$$

i.e. $\frac{t}{1-t} \ge \frac{C_A C_S}{C_Q m}$, we get

$$
\min\left\{\frac{t}{1-t}, \frac{C_A C_S}{C_Q m}\right\} = \frac{C_A C_S}{C_Q m},
$$

and therefore

$$
(1+\delta)C_Q \min\left\{\frac{t}{1-t}, \frac{C_A C_S}{C_Q m}\right\} = \frac{C_A C_S}{m - C_A C_S}.
$$

Otherwise, if

$$
0 \le t \le \frac{C_A C_S}{C_Q m + C_A C_S},
$$

i.e. $\frac{t}{1-t} \le \frac{C_A C_S}{C_Q m}$, we have

$$
\min\left\{\frac{t}{1-t}, \frac{C_A C_S}{C_Q m}\right\} = \frac{t}{1-t},
$$

and therefore

$$(1+\delta)C_Q \min\left\{\frac{t}{1-t}, \frac{C_A C_S}{C_Q m}\right\} = \frac{mC_Q}{m - C_A C_S}\frac{t}{1-t} \leq \frac{C_A C_S}{m - C_A C_S}.$$

Thus, equation (62) holds for both cases and the proof is complete. □

Remark .12. *In the BPX framework of [BPX91], the assumption* $\mathscr{A}_j(I_{j-1}^j u, I_{j-1}^j u) \leq \mathscr{A}_{j-1}(u,u)$ *is able to guarantee the non-negativity of the operator* $\mathbb{E}_{j,m}$*. As a consequence it is possible to write*

$$\mathscr{A}_j(\mathbb{E}_{j,m}u, u) \leq \delta_j \mathscr{A}_j(u,u) \qquad \forall u \in V_j,$$

instead of (55)*. This implies that the related multigrid error operator has a norm strictly less than* 1 *for any* $m > 0$*. In our case, we cannot guarantee the non-negativity of the operator* $\mathbb{E}_{j,m}$*, but Theorem .6 shows that, under the constrain* $m > 2C_A C_S$*, the error operator* $\mathbb{E}_{j,m}$ *is still a contraction.*

BIBLIOGRAPHY

[AA07] P. F. Antonietti and B. Ayuso de Dios.
Schwarz domain decomposition preconditioners for discontinuous Galerkin approximations of elliptic problems: non-overlapping case.
M2AN Math. Model. Numer. Anal., 41(1):21–54, 2007.

[AA08] P. F. Antonietti and B. Ayuso de Dios.
Multiplicative Schwarz methods for discontinuous Galerkin approximations of elliptic problems.
M2AN Math. Model. Numer. Anal., 42(3):443–469, 2008.

[ABCM02] 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(5):1749 – 1779, 2001/2002.

[ABM09] P. F. Antonietti, F. Brezzi, and L. D. Marini.
Bubble stabilization of discontinuous Galerkin methods.
Comput. Methods Appl. Mech. Engrg., 198(21-26):1651–1659, 2009.

[ABM18] P. F. Antonietti, F. Bonaldi, and I. Mazzieri.
A high-order discontinuous Galerkin approach to the elasto-acoustic problem.
MOX report 18/2018. Submitted, 2018.

[ACC+16] P. F. Antonietti, A. Cangiani, J. Collis, Z. Dong, E. H. Georgoulis, S. Giani, and P. Houston.
Review of discontinuous Galerkin Finite Element Methods for partial differential equations on complicated domains, volume 114 of *Lecture Notes in Computational Science and Engineering*, chapter 8, pages 281 – 310.
Springer, 1st edition, 2016.

[AdVMV14] P. F. Antonietti, L. Beirão da Veiga, D. Mora, and M. Verani.
A stream Virtual Element formulation of the Stokes problem on polygonal meshes.
SIAM J. Numer. Anal., 52(1):386 – 404, 2014.

[AdVSV16] P. F. Antonietti, L. Beirão da Veiga, S. Scacchi, and M. Verani.

A C1 Virtual Element Method for the Cahn-Hilliard equation with polygonal meshes.
SIAM J. Numer. Anal., 54(1):34 – 56, 2016.

[AFRV16] P. F. Antonietti, C. Facciolà, A. Russo, and M. Verani.
Discontinuous Galerkin approximation of flows in fractured porous media.
Technical Report 22/2016, MOX Report, 2016.

[AFS+16] P. F. Antonietti, L. Formaggia, A. Scotti, M. Verani, and N. Verzotti.
Mimetic Finite Difference approximation of flows in fractured porous media.
ESAIM Math. Model. Numer. Anal., 50(3):809 – 832, 2016.

[AGH13] 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.

[AGH14] P. F. Antonietti, S. Giani, and P. Houston.
Domain decomposition preconditioners for discontinuous Galerkin methods for elliptic problems on complicated domains.
J. Sci. Comput., 60(1):203–227, 2014.

[AH11] P. F. Antonietti and P. Houston.
A class of domain decomposition preconditioners for hp-discontinuous Galerkin finite element methods.
J. Sci. Comput., 46(1):124 – 149, 2011.

[AHH+17] P. F. Antonietti, P. Houston, X. Hu, M. Sarti, and M. Verani.
Multigrid algorithms for hp-version interior penaty discontinuous Galerkin methods on polygonal and polyhedral meshes.
Calcolo, 54(4):1169 – 1198, 2017.

[AHP18a] P. F. Antonietti, P. Houston, and G. Pennesi.
Additive Schwarz preconditioners for high-order discontinuous Galerkin methods on non-nested polytopic grids.
In preparation, 2018.

[AHP18b] P. F. Antonietti, P. Houston, and G. Pennesi.
Fast numerical integration on polytopic meshes with applications to discontinuous Galerkin finite element methods.
J. Sci. Comput., 0:1–32, 2018.

[AHS16] P. F. Antonietti, P. Houston, and I. Smears.

A note on optimal spectral bounds for nonoverlapping domain decomposition preconditioners for hp-version discontinuous Galerkin methods.
Internat. J. Numer. Methods Engrg., 13(4):513–524, 2016.

[ALM16] B. Ayuso de Dios, K. Lipnikov, and G. Manzini.
The nonconforming Virtual Element Method.
ESAIM Math. Model. Numer. Anal., 50(3):879 – 904, 2016.

[AM18] P. F. Antonietti and I. Mazzieri.
DG methods for the elastodynamics equations on polygonal/polyhedral grids.
MOX report 06/2018. Submitted, 2018.

[AMV17] P. F. Antonietti, G. Manzini, and M. Verani.
The fully nonconforming Virtual Element Method for biharmonic problems.
Math. Mod. Meth. Appl. Sci., Accepted: 16 October 2017.

[AP18] P. F. Antonietti and G. Pennesi.
V-cycle multigrid algorithms for discontinuous Galerkin methods on non-nested polytopic meshes.
J. Sci. Comput., 0:1–28, 2018.

[Arn82] D. N. Arnold.
An interior penalty finite element method with discontinuous elements.
SIAM J. Numer. Anal., 19(4):742 – 760, 1982.

[AS09] P. F. Antonietti and E. Suli.
Domain decomposition preconditioning for discontinuous Galerkin approximations of convection-diffusion problems.
Lect. Notes Comput. Sci. Eng., 70:259–266, 2009.

[ASV15] P. F. Antonietti, M. Sarti, and M. Verani.
Multigrid algorithms for hp-discontinuous Galerkin discretizations of elliptic problems.
SIAM J. Numer. Anal., 53(1):598 – 618, 2015.

[ASVZ17] P. F. Antonietti, M. Sarti, M. Verani, and L. Zikatanov.
A uniform Additive Schwarz preconditioner for high-order discontinuous Galerkin approximations of elliptic problems.
J. Sci. Comput., 70(2):608–630, 2017.

[Bak77] G. A. Baker.
Finite element methods for elliptic equations using nonconforming elements.
Math. Comp., 31(137):45 – 59, 1977.

[BBC+12] 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.

[BBC+13] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, and A. Russo.
Basic principles of Virtual Element Methods.
Math. Models Methods Appl. Sci., 23(1):199 – 214, 2013.

[BBC14] F. Bassi, L. Botti, and A. Colombo.
Agglomeration-based physical frame dG discretizations: an attempt to be mesh free.
Math. Mod. Meth. Appl. Sci., 24(8):1495–1539, 2014.

[BBCR12] F. Bassi, L. Botti, A. Colombo, and S. Rebay.
Agglomeration based discontinuous Galerkin discretization of the Euler and Navier-Stokes equations.
Comput. & Fluids, 61:77–85, 2012.

[BBMR16a] L. Beirão da Veiga, F. Brezzi, L. D. Marini, and A. Russo.
Mixed Virtual Element Methods for general second order elliptic problems on polygonal meshes.
ESAIM Math. Model. Numer. Anal., 50(3):727 – 747, 2016.

[BBMR16b] L. Beirão da Veiga, F. Brezzi, L. D. Marini, and A. Russo.
Virtual Element Method for general second-order elliptic problems on polygonal meshes.
Math. Models Methods Appl. Sci., 26(4):729 – 750, 2016.

[BCGS11] S. C. Brenner, J. Cui, T. Gudi, and L.-Y. Sung.
Multigrid algorithms for symmetric discontinuous Galerkin methods on graded meshes.
Numer. Math., 119(1):21–47, 2011.

[BCS09] S. C. Brenner, J. Cui, and L.-Y. Sung.
Multigrid methods for the symmetric interior penalty method on graded meshes.
Numer. Linear Algebra Appl., 16(6):481–501, 2009.

[BD81] R. E. Bank and T. Dupont.
An optimal order process for solving finite element equations.
Math. Comp., 36(153):35 – 51, 1981.

[BKP94] J. H. Bramble, D. Y. Kwak, and J. E. Pasciak.

Uniform convergence of multigrid V-cycle iterations for indefinite and nonsymmetric problems.
SIAM J. Numer. Anal., 31(6):1746 – 1763, 1994.

[BLM14] L. Beirão da Veiga, K. Lipnikov, and G. Manzini.
The Mimetic Finite Difference method for elliptic problems, volume 11.
Springer, Cham, 2014.

[BLS05a] F. Brezzi, K. Lipnikov, and M. Shashkov.
Convergence of the Mimetic Finite Difference method for diffusion problems on polyhedral meshes.
SIAM J. Numer. Anal., 43(5):1872 – 1896 (electronic), 2005.

[BLS05b] F. Brezzi, K. Lipnikov, and V. Simoncini.
A family of Mimetic Finite Difference methods on polygonal and polyhedral meshes.
Math. Mod. Meth. Appl. S., 15(10):1533 – 1551, 2005.

[BLS06] F. Brezzi, K. Lipnikov, and M. Shashkov.
Convergence of Mimetic Finite Difference method for diffusion problems on polyhedral meshes with curved faces.
Math. Mod. Meth. Appl. Sci., 16(2):275 – 297, 2006.

[BO07] S. C. Brenner and L. Owens.
A W-cycle algorithm for a weakly over-penalized interior penalty method.
Comput. Methods Appl. Mech. Engrg., 196(37-40):3823–3832, 2007.

[BP92] J. H. Bramble and J. E. Pasciak.
The analysis of smoothers for multigrid algorithms.
Math. Comp., 58(198):467 – 488, 1992.

[BP93] J. H. Bramble and J. E. Pasciak.
New estimates for multilevel algorithms including the V-cycle.
Math. Comp., 60(202):447 – 471, 1993.

[BPX87] J. H. Bramble, J. E. Pasciak, and J. Xu.
New convergence estimates for multigrid algorithms.
Math. Comp., 49(180):311–329, 1987.

[BPX91] J. H. Bramble, J. E. Pasciak, and J. Xu.
The analysis of multigrid algorithms with nonnested space or noninherited quadratic forms.
Math. Comp., 56(193):1–34, 1991.

[Bre99] S. C. Brenner.
Convergence of nonconforming multigrid methods without full elliptic regularity.
Math. Comp., 68(225):25–53, 1999.

[BV90] D. Braess and R. Verfürth.
Multigrid methods for nonconforming finite element methods.
SIAM J. Numer. Anal., 22(4):979 – 986, 1990.

[BZ05] S. C. Brenner and J. Zhao.
Convergence of multigrid algorithms for interior penalty methods.
Appl. Numer. Anal. Comput. Math., 2(1):3–18, 2005.

[CDG08] B. Cockburn, B. Dond, and J. Guzmán.
A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems.
Math. Comp., 77(264):1887–1916, 2008.

[CDG17] A. Cangiani, Z. Dong, and E. H. Georgoulis.
hp-Version space-time discontinuous Galerkin methods for parabolic problems on prismatic meshes.
SIAM J. Sci. Comput., 39(4):A1251 – A1279, 2017.

[CDGH16] A. Cangiani, Z. Dong, E. H. Georgoulis, and P. Houston.
hp-version discontinuous Galerkin methods for advection-diffusion-reaction problems on polytopic meshes.
ESAIM Math. Model. Numer. Anal., 50(3):699 – 725, 2016.

[CDGH17] A. Cangiani, Z. Dong, E. H. Georgoulis, and P. Houston.
hp-Version discontinuous Galerkin methods on polygonal and polyhedral meshes.
SpringerBriefs in Mathematics. Springer International Publishing, 2017.

[CGH14] A. Cangiani, E. H. Georgoulis, and P. Houston.
hp-Version discontinuous Galerkin methods on polygonal and polyhedral meshes.
Math. Models Methods Appl. Sci., 24(10):2009 – 2041, 2014.

[CGL09] B. Cockburn, J. Gopalakrishnan, and R. Lazarov.
Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems.
SIAM J. Numer. Anal., 47(2):1319–1365, 2009.

[CGS10] B. Cockburn, J. Gopalakrishnan, and F.-J. Sayas.
A projection-based error analysis of HDG methods.
Math. Comp., 79(271):1351–1367, 2010.

[CGW09] B. Cockburn, J. Guzmán, and H. Wang.
Superconvergent discontinuous Galerkin methods for second-order elliptic problems.
Math. Comp., 78(265):1–24, 2009.

[CKS00] B. Cockburn, G. E. Karniadakis, and C.-W. Shu.
Discontinuous Galerkin Methods. Theory, computation and applications.
Springer Berlin, 2000.

[CLS15] E. B. Chin, J. B. Lasserre, and N. Sukumar.
Numerical integration of homogeneous functions on convex and nonconvex polygons and polyhedra.
Comput. Mech., 56(6):967 – 981, 2015.

[CLS17] E. B. Chin, J. B. Lasserre, and N. Sukumar.
Modeling crack discontinuities without element-partitioning in the extended finite element method.
Internat. J. Numer. Methods Engrg., 110(11):1021 – 1048, 2017.

[CMS17] A. Cangiani, G. Manzini, and O. J. Sutton.
Conforming and nonconforming Virtual Element Methods for elliptic problems.
IMA J. Numer. Anal., 37(3):1317 – 1354, 2017.

[CXZ98] T. F. Chan, J. Xu, and L. Zikatanov.
An agglomeration multigrid method for unstructured grids.
Contemp. Math., 218:67–81, 1998.

[DE11] D. A. Di Pietro and A. Ern.
Mathematical aspects of discontinuous Galerkin methods.
Springer Science & Business Media, 2011.

[DE15a] D. A. Di Pietro and A. Ern.
A Hybrid High-Order locking-free method for linear elasticity on general meshes.
Comput. Methods Appl. Mech. Engrg., 283(1):1 – 21, 2015.

[DE15b] D. A. Di Pietro and A. Ern.
Hybrid High-Order methods for variable-diffusion problems on general meshes.
C. R. Math. Acad. Sci. Soc. R. Can., 353(1):31 – 34, January 2015.

[DEH16] J. Droniou, R. Eymard, and R. Herbin.
Gradient schemes: generic tools for the numerical analysis of diffusion equations.
ESAIM Math. Model. Numer. Anal., 50(3):749 – 781, 2016.

[DEL14] D. A. Di Pietro, A. Ern, and S. Lemaire.

An arbitrary-order and compact-stencil discretization of diffusion on general meshes based on local reconstruction operators.
Comput. Method Appl. Math., 14(4):461 – 472, October 2014.

[DEL16] D. A. Di Pietro, A. Ern, and S. Lemaire.
A review of hybrid high-order methods: formulations, computational aspects, comparison with other methods.
In G. R. Barrenechea, F. Brezzi, A. Cangiani, and E. H. Georgoulis, editors, *Building bridges: connections and challenges in modern approaches to numerical partial differential equations*, volume 114 of *Lecture Notes in Computational Science and Engineering*. Springer, Cham, 2016.

[DGS07] M. Dryja, J. Galvis, and M. Sarkis.
BDDC methods for discontinuous Galerkin discretization of elliptic problems.
J. Complex., 23:715 – 739, 2007.

[DGTZ07] H. Y. Duan, S. Q. Gao, R. C. E. Tan, and S. Zhang.
A generalized BPX multigrid framework covering nonnested V-cycle methods.
Math. Comp., 76(257):137–152, 2007.

[DK16] M. Dryja and P. Krzyżanowski.
A massively parallel nonoverlapping additive Schwarz method for discontinuous Galerkin discretization of elliptic problems.
Numer. Math., 132(2):347–367, 2016.

[DKS14] M. Dryja, P. Krzyżanowski, and M. Sarkis.
Additive Schwarz method for dG discretization of anisotropic elliptic problems.
Lect. Notes Comput. Sci. Eng., 98:407–415, 2014.

[DLVZ06] V. A. Dobrev, R. D. Lazarov, P. S. Vassilevski, and L. Zikatanov.
Two-level preconditioning of discontinuous Galerkin approximations of second-order elliptic equations.
Numer. Linear Algebra Appl., 13(9):753–770, 2006.

[Dry03] M. Dryja.
On discontinuous Galerkin methods for elliptic problems with discontinuous coefficients.
Comput. Methods Appl. Mat., 3(1):76 – 85, 2003.

[DS10] M. Dryja and M. Sarkis.
Additive average Schwarz methods for discretization of elliptic problems with highly discontinuous coefficients.
Comput. Methods Appl. Math., 10(2):164–176, 2010.

[FB10] T.-P. Fries and T. Belytschko.
The extended/generalized finite element method: an overview of the method and its applications.
Internat. J. Numer. Methods Engrg., 84(3):253–304, 2010.

[FK01] 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.

[FK02] X. Feng and O. A. Karakashian.
Analysis of two-level overlapping additive Schwarz preconditioners for a discontinuous Galerkin method.
In *Domain decomposition methods in science and engineering (Lyon, 2000)*, Theory Eng. Appl. Comput. Methods, pages 237–245. Internat. Center Numer. Methods Eng. (CIMNE), Barcelona, 2002.

[FK05] X. Feng and O. A. Karakashian.
Two-level non-overlapping Schwarz preconditioners for a discontinuous Galerkin approximation of the biharmonic equation.
J. Sci. Comput., 22/23:289–314, 2005.

[GH14] S. Giani and P. Houston.
Domain decomposition preconditioners for discontinuous Galerkin discretizations of compressible fluid flows.
Numer. Math. Theory Methods Appl., 7(2):123 – 148, 2014.

[GK03] J. Gopalakrishnan and G. Kanschat.
A multilevel discontinuous Galerkin method.
Numer. Math., 95(3):527 – 550, 2003.

[GL10] E. H. Georgoulis and O. Lakkis.
A posteriori error bounds for discontinuous Galerkin methods for quasilinear parabolic problems.
In G. Kreiss, P. Lötstedt, A. Målqvist, and M. Neytcheva, editors, *Numerical Mathematics and Advanced Applications 2009*, pages 351–358. Springer Berlin Heidelberg, 2010.

[GP00] J. Gopalakrishnan and J. E. Pasciak.
Multigrid for the Mortar finite element method.
SIAM J. Numer. Anal., 37(3):1029 – 1052, 2000.

[Gri05] D. J. Griffiths.
Introduction to Quantum Mechanics.
Pearson Education, 2005.

[GS05] E. H. Georgoulis and E. Süli.
Optimal error estimates for the hp-version interior penalty discontinuous Galerkin finite element method.
IMA J. Numer. Anal., 25(1):205–220, 2005.

[GW08] A. Gerstenberger and A. W. Wall.
An Extended Finite Element Method/Lagrange multiplier based approach for fluid-structure interaction.
Comput. Methods Appl. Mech. Engrg., 197(19 - 20):1699 – 1714, 2008.

[HNS15] D. J. Holdych, D. R. Noble, and R. B. Secor.
Quadrature rules for triangular and tetrahedral elements with generalized functions.
Internat. J. Numer. Methods Engrg., 73(9):1310 – 1327, 2015.

[Hol] S. Holz.
Polygon clipper.
`https://it.mathworks.com/matlabcentral/fileexchange/8818-polygon-clipper`.

[HS97a] 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(4):15–25, 1997.

[HS97b] W. Hackbusch and S. A. Sauter.
Composite Finite Elements for the approximation of PDEs on domains with complicated micro-structures.
Numer. Math., 75(4):447–472, 1997.

[HSS97] J. Hyman, M. Shashkov, and S. Steinberg.
The numerical solution of diffusion problems in strongly heterogeneous non-isotropic materials.
J. Comput. Phys., 132(1):130 – 148, 1997.

[HW08] J. S. Hesthaven and T. Warburton.
Nodal discontinuous Galerkin methods: algorithms, analysis, and applications.
Springer Berlin, 1st edition, 2008.

[KC17] O. A. Karakashian and C. Collins.
Two-level additive Schwarz methods for discontinuous Galerkin approximations of second-order elliptic problems.
IMA J. Numer. Anal., 37(4):1800 – 1830, 2017.

[KK98] G. Karypis and V. Kumar.
A fast and high quality multilevel scheme for partitioning irregular graphs.
SIAM J. Sci. Comput., 20(1):359 – 392, 1998.

[KK09] G. Karypis and V. Kumar.
Metis: Unstructured graph partitioning and sparse matrix ordering system, version 4.0.
`http://www.cs.umn.edu/~metis`, 2009.

[Las98] J. B. Lasserre.
Integration on a convex polytope.
Proc. Amer. Math. Soc., 126(8):2433 – 2441, 1998.

[LB74] C. L. Ladson and C. W. Jr. Brooks.
Development of a computer program to obtain ordinates for NACA 6- and 6A-series airfoils.
NASA Technical Memorandum X-3069, 1974.

[Lio87] P.-L. Lions.
On the Schwarz alternating method. I.
In R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 1–42. Philadelphia, PA, 1988. SIAM. Paris, France, January 7-9, 1987.

[Lio88] P.-L. Lions.
On the Schwarz alternating method. II.
In T. Chan, R. Glowinski, J. Périaux, and O. Widlund, editors, *Domain Decomposition Methods. Second International Symposium on Domain Decomposition Methods*, pages 47–70. Philadelphia, PA, 1989. SIAM. Los Angeles, California, January 14-16, 1988.

[Lio89] P.-L. Lions.
On the Schwarz alternating method. III.
In T. Chan, R. Glowinski, J. Périaux, and O. Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equa-*

tions, pages 202–223. SIAM, Philadelphia, PA, 1990. Houston, Texas, March 20-22, 1989.

[LLD09] C.-J. Li, P. Lambertu, and C. Dagnino.
Numerical integration over polygons using an eight-node quadrilateral spline finite element.
J. Comput. Appl. Math., 233(2):279 – 292, 2009.

[LM77] J. N. Lyness and G. Monegato.
Quadrature rules for regions having regular hexagonal symmetry.
SIAM J. Numer. Anal., 14(2):283 – 295, 1977.

[LT03] C. Lasser and A. Toselli.
An overlapping domain decomposition preconditioner for a class of discontinuous Galerkin approximations of advection-diffusion problems.
Math. Comp., 72(243):1215–1238, 2003.

[LVY14] K. Lipnikov, D. Vassilev, and I. Yotov.
Discontinuous Galerkin and mimetic finite difference methods for coupled Stokes-Darcy flows on polygonal and polyhedral grids.
Numer. Math., 126(2):321 – 360, 2014.

[MDB99] N. Moës, J. Dolbow, and T. Belytschko.
A finite element method for crack growth without remeshing.
Int. J. Numer. Meth. Eng., 46(1):131 – 150, 1999.

[MRW96] J. Ma, V. Rokhlin, and S. Wandzura.
Generalized Gaussian quadrature of systems of arbitrary functions.
SIAM J. Numer. Anal., 33(3):971 – 996, 1996.

[MS11] S. E. Mousavi and N. Sukumar.
Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons.
Comput. Mech., 47(5):535 – 554, 2011.

[MXS10] S. E. Mousavi, H. Xiao, and N. Sukumar.
Generalized Gaussian quadrature rules on arbitrary polygons.
Internat. J. Numer. Methods Engrg., 82(1):99 – 113, 2010.

[NBM09] S. Natarajan, S. Bordas, and D. R. Mahapatra.
Numerical integration over arbitrary polygonal domains based on Schwarz–Christoffel conformal mapping.
Internat. J. Numer. Methods Engrg., 80(1):103 – 134, 2009.

[PS04] P. O. Persson and G. Strang.
A simple mesh generator in MATLAB.
SIAM Rev., 46(2):329–345, 2004.

[PTVF07] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery.
Numerical Recipes 3rd Edition: The Art of Scientific Computing.
Cambridge University Press, New York, NY, USA, 3 edition, 2007.

[Qia15] H. Qian.
Counting the floating point operations (flops).
`https://it.mathworks.com/matlabcentral/fileexchange/50608-counting-the-floating-point-operations--flops-`, 2015.

[QSS07] A. Quarteroni, R. Sacco, and F. Saleri.
Numerical Mathematics.
Springer, 2007.

[RH73] W. H. Reed and T. R. Hill.
Triangular mesh methods for the neutron transport equation.
Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[Riv08] B. Rivière.
Discontinuous Galerkin methods for solving elliptic and parabolic equations.
SIAM, 2008.

[Saa03] Y. Saad.
Iterative methods for sparse linear systems: second edition, volume 82 of *Other Titles in Applied Mathematics*.
Society for Industrial and Applied Mathematics, 2003.

[SB96] C. P. Simon and L. E. Blume.
Mathematics for Economists.
W. W. Norton and Company: New York, 1996.

[SMB00] N. Sukumar, N. Moës, and T. Belytschko.
Extended Finite Element Method for three-dimensional crack modelling.
Internat. J. Numer. Methods Engrg., 48(11):1549 – 1570, 2000.

[Sme18] I. Smears.
Nonoverlapping domain decomposition preconditionersfor discontinuous Galerkin approximationsof Hamilton–Jacobi–Bellman equations.
J. Sci. Comput., 74:145–174, 2018.

[SS67] A. H. Stroud and D. Secrest.
Gaussiam quadrature formulas.
ZAMM Z. Angew. Math. Mech., 47(2):138 – 139, 1967.

[ST04] N. Sukumar and A. Tabarraei.
Conforming Polygonal Finite Elements.
Internat. J. Numer. Methods Engrg., 61(12):2045–2066, 2004.

[Ste70] E. M. Stein.
Singular Integrals and Differentiability Properties of Functions.
Princeton, University Press, Princeton, N.J., 1970.

[SV07] A. Sommariva and M. Vianello.
Product Gauss cubature over polygons based on Green's integration formula.
BIT, 47(2):441 – 453, 2007.

[SW10] B. Stamm and T. P. Wihler.
hp-optimal discontinuous Galerkin methods for linear elliptic problems.
Math. Comp., 79(272):2117 – 2133, 2010.

[SW13] Y. Sudhakar and W. A. Wall.
Quadrature schemes for arbitrary convex/concave volumes and integration of weak form in enriched partition of unity methods.
Comput. Methods Appl. Mech. Engrg., 258(1):39 – 54, 2013.

[SZ92] L. R. Scott and S. Zhang.
Higher-dimensional nonnested multigrid methods.
Math. Comp., 58(198):457 – 466, 1992.

[Tay96] M. E. Taylor.
Partial Differential Equations: Basic Theory.
Springer-Verlag, New york, 1996.

[TPPM12] C. Talischi, G. H. Paulino, A. Pereira, and I. F. M. Menezes.
Polymesher: a general-purpose mesh generator for polygonal elements written in MATLAB.
Struct. Multidiscip. Optim., 45(3):309–328, 2012.

[TS08] A. Tabarraei and N. Sukumar.
Extended Finite Element Method on polygonal and quadtree meshes.
Comput. Methods Appl. Mech. Eng., 197(5):425–438, 2008.

[TW04] A. Toselli and O. B. Widlund.

Domain decomposition methods – algorithms and theory, volume 34 of *Springer Series in Computational mathematics*.
Springer, 2004.

[VB15] G. Ventura and E. Benvenuti.
Equivalent polynomials for quadrature in Heaviside function enriched elements.
Internat. J. Numer. Methods Engrg., 102(3 - 4):688 – 710, 2015.

[Ven06] G. Ventura.
On the elimination of quadrature subcells for discontinuous functions in the extended finite-element method.
Internat. J. Numer. Methods Engrg., 66(5):761 – 795, 2006.

[WH03] T. Warburton and J. S. Hesthaven.
On the constants in hp-finite element trace inverse inequalities.
Comput. Methods Appl. Mech. Eng., 192(25):2765–2773, 2003.

[Whe78] M. F. Wheeler.
An elliptic collocation finite element method with interior penalties.
SIAM J. Numer. Anal., 15(1):152 – 161, 1978.

[XC01] X. Xu and J. Chen.
Multigrid for the mortar element method for $P1$ nonconforming element.
Numer. Math., 88(2):381 – 389, 2001.

[XG10] H. Xiao and Z. Gimbutas.
A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions.
Comput. Math. Appl., 59(2):663 – 676, January 2010.

[XLC02] X. Xu, L. Li, and W. Chen.
A multigrid method for the Mortar-type Morley element approximation of a plate bending problem.
SIAM J. Numer. Anal., 39(5):1712 – 1731, 2002.

[Xu92] J. Xu.
Iterative methods by space decomposition and subspace correction.
SIAM Rev., 34(4):581–613, 1992.

[YR98] N. Yarvin and V. Rokhlin.
Generalized Gaussian quadratures and singular value decompositions of integral operators.
SIAM J. Sci. Comput., 20(2):669 – 718, 1998.

[YS16] A. M. Yogitha and K. T. Shivaram.
Numerical integration of arbitrary functions over a convex and non convex polygonal domain by eight noded linear quadrilateral finite element method.
Aust. J. Basic Appl. Sci., 10(16):104 – 110, 2016.

[Zha90] S. Zhang.
Optimal-order nonnested multigrid methods for solving finite element equations I: on quasi-uniform meshes.
Math. Comp., 55(191):23 – 36, 1990.

[ZZ97] S. Zhang and Z. Zhang.
Treatments of discontinuity and bubble functions in the multigrid method.
Math. Comp., 66(219):1055 – 1072, 1997.