Università degli Studi di Milano
Facoltà di Scienze e Tecnologie
Dipartimento di matematica ”Federigo Enriques”

Dottorato di Ricerca in Scienze Matematiche,
XXXVI ciclo

Balancing Domain Decomposition by Constraints preconditioners for Virtual Element discretizations of Saddle-point problems

Candidate
Tommaso Bevilacqua

Advisor
Prof. Simone Scacchi

Anno Accademico 2022/2023

Contents

Introduction

The Virtual Element Method (VEM, [10]) is a recent technology for the numerical approximation of Partial Differential Equations (PDEs) which can deal with computational grids of very general polygonal/polyhedral shape. Effective VEM discretizations have been developed for several PDEs; the interested reader should consult the recent special issue [9] and the book [5] for further details. Regarding computational fluid dynamics, divergence-free VEM discretizations of the Stokes, Navier-Stokes and Oseen equations have been proposed in [12–15]. The core idea behind VEM is to use approximated discrete bilinear forms, whose computation requires only the integration of polynomials on the element boundary and interior. The resulting discrete solution is conforming and the accuracy guaranteed by such discrete bilinear forms turns to be sufficient to achieve the correct order of convergence. The advantage of these methods is that they can be applied on a wide choice of general polygonal meshes without the need to integrate complex non-polynomial functions on the elements, keeping an high degree of accuracy. Due to the arbitrary shape of polytopal elements, the linear systems arising from VEM discretizations of PDEs are generally worse conditioned than in case of Finite Element Methods (FEM) therefore, it is fundamental to provide efficient and scalable preconditioners for these methods. Some recent studies have proposed multigrid and domain decomposition preconditioners for scalar elliptic equations in primal form: see [6, 7] for a multigrid preconditioner, [17, 18, 61, 62] for Balancing Domain Decomposition by Constraints (BDDC) and Dual-Primal Finite Element Tearing and Interconnecting (FETI-DP) preconditioners, and [32, 33] for Overlapping Additive Schwarz preconditioners.

A few works have investigated the efficient solution of VEM approximations for saddle point problems. In [40, 41] parallel block algebraic multigrid preconditioners have been proposed for three-dimensional VEM approximations of elliptic, Stokes, and Maxwell equations in mixed form. BDDC preconditioners for three-dimensional scalar elliptic equations in mixed form have been constructed and analyzed in [43]. To our knowledge, the development of effective non-overlapping domain decomposition preconditioners for VEM discretizations of the Stokes equations is still an open problem. Therefore the main goal of our thesis is to work in this direction developing BDDC algorithm for these equations. We recall that BDDC preconditioners [44, 45] belong to the class of non-overlapping domain decomposition methods and they can be regarded as an evolution of Balancing Neumann Neumann preconditioners [49, 81, 91]. These methods perform a decomposition of the problems into independent, parallel local problems. Additionally, to obtain scalability in the number of subdomains, a coarse space has to ensure global transport of information. They have been extensively applied to solve linear systems that arise from finite element discretizations of PDEs (see e.g. [46, 66, 67, 80, 99, 101]) and also extended to various innovative discretizations techniques for PDEs, such as Mortar discretizations [59], discontinuous Galerkin methods [35, 47], isogeometric analysis [57, 98], weak Galerkin methods [96] and, as we said before, virtual element methods. In particular, regarding the Stokes equations, they have been studied in [69, 70]. Moreover, they have been extended to non-symmetric problems [95], such as hybridizable discontinuous Galerkin (HDG) discretization for Oseen equations [97], and FEM discretization of Navier-Stokes equations [52–54]. We also remark that BDDC presents several features in common with the dual-primal finite element tearing and interconnecting (FETI-DP) algorithm. In particular, the BDDC and FETI-DP operators share almost the same eigenvalues [26, 71], thus they exhibit analogous convergence properties.

The novelties of this study are to construct, analyze and numerically validate a BDDC preconditioner for two and three dimensional divergence-free VEM discretizations of the Stokes equations introduced in [14] and [12]. Our algorithm represents an extension to VEM of the BDDC preconditioner proposed in [70] for FEM discretizations of the Stokes equations with discontinuous pressure spaces. From the theoretical point of view, we prove the scalability and quasi-optimality of the method in the case of a homogeneous fluid with piecewise constant viscosity on the subdomains. We validate the theoretical estimates with several parallel numerical tests, and we provide numerical evidence of the robustness of the preconditioner with respect to the degree of approximation and the shape of the polyhedral elements. We also propose different techniques to enrich the primal coarse space, already experimented in the FEM [64] and in the VEM for linear diffusion and compressible elasticity [56, 62]. We confirm the robustness of our approach on various configurations of high viscosity jumps and on a challenging multi-sinker test case with heterogeneous viscosity [87]. Moreover we extended our algorithm to the resolution of non-symmetric saddle point problems, for which we provide the construction and several numerical experiments both in two and three dimensions.

In **Chapter 1** of the thesis we introduce the notations and we give a brief recap on the Sobolev spaces we are going to use throughout our work. We give some geometry assumptions that are needed for the VEM discretization exploiting also some advantages of these type of methods and we introduce the projection operators that are the main ingredient of this type of numerical approximation. We include here also the formulation, in classical and variational form, of the Stokes and Oseen model problems.

In **Chapter 2**, we introduce the divergence-free VEM discretization that we use in our analysis. We recall the construction of the discrete spaces both in two and in three dimensions, also showing how to compute the polynomial projections, that usually are left to the reader in the VEM papers. We also provide a proof for a Stokes optimal VEM interpolant in the three dimensional framework. For sake of completeness we report some convergence tests for the numerical methods that we will use in the following section.

Chapter 3 contains the main novelties of our work. Here, we introduce the BDDC methods applied to the saddle point problem that arises from the VEM discretization of the Stokes equations described in Chapter 2. We give a quick introduction of the Domain Decomposition technique and how to adapt it to the VEM methods. We provide the construction of these BDDC methods and we prove a convergence rate estimate of the preconditioned system, independent of the number of subdomains and polylogarithmic with respect to the ratio H/h, where H denotes the subdomain size and h the mesh size. Such an estimate yields the scalability and quasi-optimality of the resulting algorithm. We confirm the theoretical estimate and show the robustness of the solver with respect to different polygonal and polyhedral meshes and different mesh partitioning techniques with several numerical tests.

Chapter 4 is focused on the enrichment of the coarse space of our methods. We provide here two adaptive approaches in two dimensions, and one in three dimensions, to construct a coarse space that is able to keep the condition number below a fixed tolerance value. All the adaptive coarse spaces are constructed solving eigenvalue problems on edges or faces of the subdomain partition. We also introduce a new heuristical approach to enrich the coarse space that does not require the solution of these eigenvalue problems. We tested the robustness of all these algorithms on meshes that exhibit high viscosity jumps and with different configurations of the coefficients.

Finally, in **Chapter 5** we extend the BDDC algorithm applied to the non-symmetric saddle point problem that arises from the VEM discretization of the Oseen problem. We also provide some numerical simulations including scalability and optimality tests in both two and three dimensions, confirming the good behavior of the preconditioner that is comparable with the performance for the Stokes equations.

All the numerical simulations for the tests in two dimensions have been performed using a MATLAB R2023A© serial code, therefore no CPU time analysis is provided. The parallel code for our three dimensional simulations has been written in C++, exploiting the Vem++ library [39] for the VEM discretization, and the PETSc library [8] for our distributed memory implementation. We refer to [100] for the details related to the BDDC implementation in PETSc. All the numerical tests presented in the following have been performed on the Linux cluster INDACO (www.indaco.unimi.it) of the University of Milan, constituted by 16 nodes, each carrying 2 INTEL XEON E5-2683V4 processors at 2.1 GHz, with 16 cores each.

Chapter 1

Notations and Preliminaries

In this chapter we introduce some preliminary notations, that will be adopted in the rest of the thesis and the model problem that we are going to study. We first recall some standard notation on Sobolev spaces [1] and usual norms that are common in the domain decomposition framework [91]. Then we recall some geometry assumptions that are needed for the VEM discretization [12,14] and some polynomial projection operators [2] that cover a key role in these methods. We conclude introducing the model problem we are going to study.

1.1 A brief recap on Sobolev spaces

Let $\Omega \subseteq \mathbb{R}^d$ with $d = 2, 3$ be the computational domain and let be $\omega \subseteq \mathbb{R}^d$ an open bounded domain, with $\omega = \Omega$ in some cases. We assume that the domain ω is a bounded open Lipschitz set with Lipschitz continuous boundary according to:

Definition 1.1.1. The boundary $\partial\omega$ is Lipschitz continuous if there exists a finite number of open sets U_i , $i = 1, ..., m$, that cover $\partial\omega$ such that, for every i, the intersection $\partial\omega \cap U_i$ is the graph of a Lipschitz continuous function and $\omega \cap U_i$ lies on one side of this graph.

We recall for $1 \leq p \leq \infty$ the spaces of p-summable functions

$$L^p(\omega) = \left\{ u : \omega \to \mathbb{R} \quad | \quad \int_\omega |u|^p < \infty \right\},$$

with the norm, for $1 \leq p < \infty$:

$$\|u\|^p_{L^p(\omega)} = \int_\omega |u|^p \, \mathrm{d}x,$$

and, for $p = \infty$:

$$\|u\|_{L^\infty(\omega)} = \inf\{C \geq 0 : |u(x)| \leq C \ \textit{almost everywhere}\}.$$

A great importance has the particular case $p = 2$, that is the space of square-summable functions on ω

$$L^2(\omega) = \Big\{ u : \omega \to \mathbb{R} \quad | \quad \int_\omega |u|^2 < \infty \Big\}.$$

It is a Hilbert space with the scalar product

$$(u, v)_{L^2(\omega)} = \int_\omega u\, v \,\mathrm{d}x$$

and the induced norm

$$\|u\|^2_{L^2(\omega)} = (u, v)_{L^2(\omega)} = \int_\omega |u|^2 \,\mathrm{d}x.$$

For any integer $s \geq 0$, we recall the Sobolev spaces

$$W^{s,p}(\omega) = \Big\{ u \in L^p(\omega), \quad D^{\boldsymbol{\alpha}} u \in L^p(\omega), \quad |\boldsymbol{\alpha}| \leq s \Big\},$$

where $\boldsymbol{\alpha} := (\alpha_1, \alpha_2, ...\alpha_d) \in \mathbb{N}^d$ is a multi-index, and:

$$D^{\boldsymbol{\alpha}} u := \frac{\partial^{|\boldsymbol{\alpha}|} u}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} ... \partial x_d^{\alpha_d}}, \quad \text{with} \quad |\boldsymbol{\alpha}| = \alpha_1 + \alpha_2 + ... + \alpha_d$$

are the weak derivatives taken in sense of distributions. On this space, we introduce a norm

$$\|u\|_{W^{s,p}(\omega)} = \sum_{|\boldsymbol{\alpha}| \leq s} \|D^{\boldsymbol{\alpha}} u\|_{L^p(\omega)}.$$

As before, we have a particular case for $p = 2$, we denote the Hilbert space $H^s(\omega) = W^{s,2}(\omega)$ with the inner product and the induced norm

$$(u, v)_{H^s(\omega)} = \sum_{|\boldsymbol{\alpha}| \leq s} \int_\omega D^{\boldsymbol{\alpha}} u\, D^{\boldsymbol{\alpha}} v \,\mathrm{d}x, \quad \|u\|^2_{s,\omega} = (u, u)_{H^s(\omega)},$$

and the seminorm

$$|u|^2_{H^s(\omega)} = \sum_{|\boldsymbol{\alpha}| = s} \int_\omega D^{\boldsymbol{\alpha}} u\, D^{\boldsymbol{\alpha}} v \,\mathrm{d}x.$$

Remark 1.1.1. We also recall the differential operators: Δ and ∇ denote the Laplacian and the gradient for scalar functions, $\boldsymbol{\Delta}$, $\boldsymbol{\nabla}$, $\boldsymbol{\varepsilon}$, and div, denote the vector Laplacian, the gradient, the symmetric gradient operator and the divergence operator for vector fields, whereas **div** denotes the vector-valued divergence operator for tensor fields.

We note that, for $s = 0$, we obtain the space $L^2(\omega)$ previously defined, and for $s = 1$ the Sobolev space

$$H^1(\omega) = \left\{ u : \omega \to \mathbb{R} \quad | \quad \int_\omega |\nabla u|^2 + \int_\omega |u|^2 < \infty \right\}$$

with

$$|u|^2_{H^1(\Omega)} = \int_\Omega |\nabla u|^2, \quad \|u\|^2_{H^1(\Omega)} = \|u\|^2_{L^2(\Omega)} + |u|^2_{H^1(\Omega)}.$$

We define $H^s_0(\omega)$ as the closure of $C^\infty_0(\omega)$ in $H^s(\omega)$, where $C^\infty_0(\omega)$ consists of functions in $C^\infty(\omega)$ with compact support in ω.

We need also some trace and norms definitions. Since we assumed that $\partial\omega$ is Lipschitz continuous, we can then define the space $H^s(\partial\omega)$, $s \geq 0$, consisting of functions on $\partial\omega$, s.t.

$$\|u\|^2_{H^s(\partial\omega)} = \|u\|^2_{H^{[s]}(\partial\omega)} + |u|^2_{H^s(\partial\omega)} < \infty,$$

where $[s]$ denotes the greater natural number smaller than s, and with the seminorm defined as

$$|u|^2_{H^s(\partial\omega)} = \sum_{|\boldsymbol{\alpha}|=[s]} \int_{\partial\omega} \int_{\partial\omega} \frac{|D^{\boldsymbol{\alpha}} u(x) - D^{\boldsymbol{\alpha}} u(y)|^2}{|x-y|^{2\sigma+n-1}} \, \mathrm{d}S_x \, \mathrm{d}S_x.$$

We have two lemmas from [51]:

Lemma 1.1.1. *Let ω be a Lipschitz region and let $s > 1/2$. Then the operator $\gamma_0 : C^\infty(\bar{\omega}) \to C^\infty(\partial\omega)$, mapping a function into its restriction on the boundary, can be extended continuously to an operator $\gamma_0 : H^s(\omega) \to H^{s-1/2}(\partial\omega)$.*

Lemma 1.1.2. *With the same assumptions as in Lemma 1.1.1, there exist a continuous lifting operator $\mathcal{R}_0 : H^{s-1/2}(\partial\omega) \to H^s(\omega)$, s.t. $\gamma_0(\mathcal{R}_0 u) = u$, $u \in H^{s-1/2}(\partial\omega)$.*

In the analysis developed in the following chapters, we are interested into proper subsets of $\partial\Omega$. we then recall that these previous definitions and properties can be generalized to a proper subset $\Gamma \subseteq \partial\omega$ with non-vanishing (n-1)-dimensional measure and which is relatively open with respect to $\partial\omega$.

Let Γ be a proper subset of $\partial\omega$. Then $H^s_0(\Gamma)$ (defined as the kernel of γ_0) coincides with $H^s(\Gamma)$ for $s \leq 1/2$. However the extensions by zero of functions in $H^{1/2}_0(\Gamma)$ do not, in general, belong to $H^{1/2}(\partial\omega)$. For this reason we define the space:

$$H^{1/2}_{00}(\Gamma) = \left\{ u \in H^{1/2}(\Gamma) \,|\, \mathcal{E}u \in H^{1/2}(\partial\omega) \right\},$$

where $\mathcal{E}u$ is the extension by zero of u to $\partial\omega$. This space coincides with the interpolation space:

$$[H^1_0(\Gamma, L^2(\Gamma))]_{1/2} = \left\{ u \in L^2(\Gamma) | \quad t^{-1}K(t, u; H^1_0(\Gamma), L^2(\Gamma)) \in L^2(0, \infty) \right\}$$

where

$$K(t, u; H^1_0(\Gamma)^2 = inf_{u_0+u_1=u}\{\|u_0\|^2_{L^2(\Gamma)} + t^2\|u_1\|^2_{H^1_0(\Gamma)}\},$$

and it is equipped with the norm:

$$\|u\|^2_{H^{1/2}_{00}(\Gamma)} = \|u\|^2_{L^2(\Gamma)} + \int_0^\infty t^{-2}K(t, u; H^1_0(\Gamma)^2 \mathrm{d}t.$$

Remark 1.1.2. If $u \in H^{1/2}(\partial\omega)$ vanishes almost everywhere on $\partial\omega \setminus \Gamma$, then one can prove that the two norms $\|u\|_{H^{1/2}(\partial\omega)}$ and $\|u\|_{H^{1/2}_{00}(\Gamma)}$ are equivalent.

Remark 1.1.3. This spaces, norms and seminorms are naturally extendable for vector spaces $[H^s(\omega)]^d$ with $s \geq 0$ and $d = 2, 3$. Also, with an abuse of notation we will write $\| \cdot \|_{s,\omega}$ instead of $\| \cdot \|_{[H^s(\omega)]^d}$, $| \cdot |_{s,\omega}$ instead of $| \cdot |_{[H^s(\omega)]^d}$ and $(\cdot, \cdot)_{s,w}$ instead of $(\cdot, \cdot)_{[H^s(\omega)]^d}$, with $d = 1, 2, 3$, when no confusion can arise. Moreover we will omit the subscript ω when it corresponds to the whole computational domain Ω.

1.2 Geometry definitions

With an abuse of notation, we will denote by K a general polygon or polyhedron depending if we are in a $2D$ or $3D$ context (the framework considered will be always clear throughout the thesis, thus no confusion can arise).

For a general polygon/polyhedron K we denote by V one of the N_V vertices, by e one of the N_e edges and in three dimensions, by f one of the N_f faces of K. In a $2D$ context we simply denote by $\mathbf{n}^e_K$ (respectively, $\mathbf{n}_K$) the unit outward normal vector to e (respectively, to ∂K), while in $3D$ for each polyhedron P, each face f of K and each edge e of f we denote by:

- $\mathbf{n}^f_K$ (respectively, $\mathbf{n}_K$) the unit outward normal vector to f (respectively, to ∂K);
- $\mathbf{n}^f_e$ (respectively, $\mathbf{n}_f$) the unit vector in the plane of f that is normal to the edge e (respectively, to ∂f) and outward with respect to f;
- $\mathbf{n}^f_K$ (respectively, $\mathbf{n}_K$) the unit outward normal vector to f (respectively, to ∂K);
- $\mathbf{t}^e_f$ (respectively, $\mathbf{t}_f$) the unit vector tangent to e (respectively, to ∂f) counterclockwise with respect to $\mathbf{n}^f_K$;
- $\boldsymbol{\tau}^f_1$ and $\boldsymbol{\tau}^f_2$ two orthogonal unit vectors lying on f and such that $\boldsymbol{\tau}^f_1 \wedge \boldsymbol{\tau}^f_1 = \mathbf{n}^f_K$, i.e. $\boldsymbol{\tau}^f_1$ and $\boldsymbol{\tau}^f_2$ constitute the axis of a local coordinate system on f;

- $\boldsymbol{\tau}_e$ a unit vector tangent to the edge e.

Remark 1.2.1. The vectors $\mathbf{t}_f^e$, $\mathbf{t}_f$, $\boldsymbol{\tau}_1^f$ and $\boldsymbol{\tau}_2^f$ depend on K but we omit the subscript for lightening the notation.

1.2.1 Mesh assumptions

Let $\{\mathcal{T}_h\}_h$ be a sequence of decompositions of Ω into general N_K polyhedral elements K with a mesh size:

$$h := \sup_{K \in \mathcal{T}_h} h_K,$$

where h_K is the diameter of K. We denote with n_K, n_E, n_V and n_F respectively the total number of elements, edges, vertices and faces in T_h.

Mesh assumptions in 2D. As in [14], we suppose that, for all h, each element $K \in \mathcal{T}_h$ satisfies the following assumptions:

- (**A1**) K is star-shaped with respect to a ball of radius $\geq \gamma h_k$,
- (**A2**) the distance between any two vertices of K is $\geq c h_K$,

where γ and c are positive constants.

Mesh assumptions in 3D. As in [12] we suppose that, for all h, each element $K \in \mathcal{T}_h$ satisfies the following assumptions:

- (**A1**) K is star-shaped with respect to a ball B_K of radius $\geq \gamma h_K$,
- (**A2**) every face f of K is star-shaped with respect to a disk B_f of radius $\geq \; \gamma \; h_K$,
- (**A3**) every edge e of K satisfies $h_e \geq \gamma h_K$,

where γ is a uniform positive constant.

Remark 1.2.2. These hypotheses both in two and three dimension could be weakened as in [10], for example assuming that every K is a union of a finite (and uniformly bounded) number of star-shaped domains, each satisfying (**A1**).

The mesh assumptions described above are very general and allows to treat mesh discretizations and configurations that can be challenging or not allowed in a FEM context. Firstly, we point out that two or more consecutive edges are allowed to form a straight angle. For example, a square with four extra point on an edge is simply regarded as an octagon (Figure 1.1a). This means that hanging nodes are allowed geometrically in the mesh, but in practice they are not. Adaptive mesh refinements are then easily applicable, since we do not have to worry about this type of configurations. There are several other advantages working with these mesh hypothesis since we can handle fractures or mesh deformations (Figure 1.1b) and moreover we can glue them up together mesh derived from different decomposition (Figure 1.1c). As a representative sample of the increasing list of technologies that make use of polygonal/polyhedral meshes, we refer to these papers and monographs [22, 24, 28, 29, 34, 37, 38, 72, 78, 79, 84–86].

1.3 Polynomial spaces and projections

We denote with $\mathcal{O}$ a generic geometrical entity (element, face, edge) having diameter $h_{\mathcal{O}}$, baricenter $\mathbf{x}_{\mathcal{O}}$ and we introduce for any $\mathcal{O}$ and $n \in \mathbb{N}$ the spaces:

- $\mathbb{P}_n(\mathcal{O})$ the set of polynomials on $\mathcal{O}$ of degree $\leq n$ (with $\mathbb{P}_{-1}(\mathcal{O}) := \{0\}$),
- $\widehat{\mathbb{P}}_{n\setminus m}(\mathcal{O}) := \mathbb{P}_n(\mathcal{O}) \setminus \mathbb{P}_m(\mathcal{O})$ for $n > m$, denotes the space of polynomials in $\mathbb{P}_n(\mathcal{O})$ with monomials of degree strictly greater than m.

A natural basis associated with the space $\mathbb{P}_n(\mathcal{O})$ is the set of scaled monomials:

$$\mathcal{M}_n(\mathcal{O}) := \{m_{\boldsymbol{\alpha}}, \quad \text{with} \quad |\boldsymbol{\alpha}| \leq n\}$$

where, for any multi-index $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_d) \in \mathcal{N}^d$:

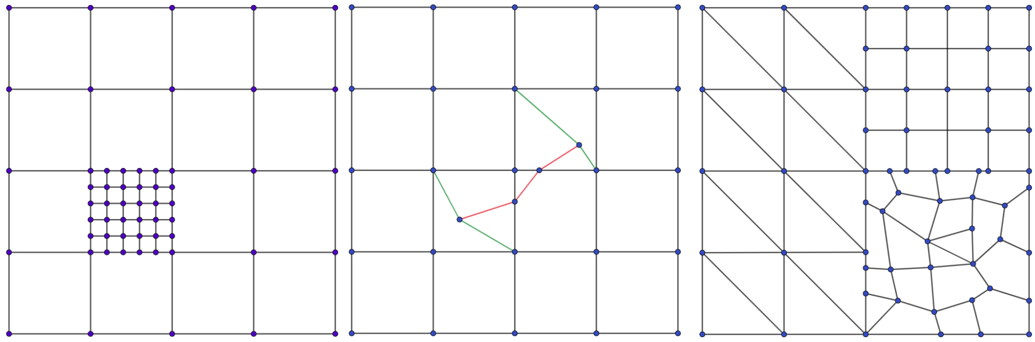
$$m_{\boldsymbol{\alpha}} := \prod_{i=1}^{d} \left(\frac{x_i - x_{\mathcal{O},i}}{h_{\mathcal{O}}} \right) \quad \text{and} \quad |\boldsymbol{\alpha}| := \sum_{i=1}^{d} \alpha_i. \tag{1.1}$$

We denote with $\mathbf{x} = (x_1, ..., x_d)$ the independent variable. We also introduce the following useful polynomial decompositions [41, 42]:

$$\begin{aligned} [\mathbb{P}_n(\mathcal{O})]^2 &= \nabla \mathbb{P}_{n+1}(\mathcal{O}) \oplus \left(\mathbf{x}^{\perp} \mathbb{P}_{n-1}(\mathcal{O})\right) \qquad && \text{if } \dim(\mathcal{O}) = 2, \\ [\mathbb{P}_n(\mathcal{O})]^3 &= \nabla \mathbb{P}_{n+1}(\mathcal{O}) \oplus \left(\mathbf{x} \wedge [\mathbb{P}_{n-1}(\mathcal{O}]^3)\right) \qquad && \text{if } \dim(\mathcal{O}) = 3, \end{aligned} \tag{1.2}$$

where $\mathbf{x}^{\perp} := (x_2, -x_1)$.

Now we define some suitable projection operators that will cover a fundamental role into the VEM context. Their purpose is to project the unknowns virtual functions on to the space of polynomials. The idea is to choose accurate the degrees of freedom (DoFs) for the discrete



(a) "Hanging" nodes. (b) Fracture. (c) Gluing mesh.

Figure 1.1: General mesh decomposition that can be easily handled by the Virtual Element Methods.

space in a way that these operators can be directly computable from them. These projection operators play a central role in the VEM being used to define the discrete bilinear forms and to approximate the errors. For any $n \in \mathbb{N}$ and each geometric entity $\mathcal{O}$ (element or face), we introduce the following polynomial projections:

- the L^2-*projection* $\Pi_n^{0,\mathcal{O}} : L^2(\mathcal{O}) \to \mathbb{P}_n(\mathcal{O})$, defined for any $v \in L^2(\mathcal{O})$ by:

$$\int_{\mathcal{O}} q_n (v - \Pi_n^{0,\mathcal{O}} v) \, \mathrm{d}\mathcal{O} = 0 \qquad \text{for all } q_n \in \mathbb{P}_n(\mathcal{O}), \tag{1.3}$$

  with obvious extension for vector functions $\Pi_n^{0,\mathcal{O}} : [L^2(\mathcal{O})]^d \to [\mathbb{P}_n(\mathcal{O})]^d$ and tensor functions $\Pi_n^{0,\mathcal{O}} : [L^2(\mathcal{O})]^{d\times d} \to [\mathbb{P}_n(\mathcal{O})]^{d\times d}$,

- the H^1-*seminorm projection* $\Pi_n^{\nabla,\mathcal{O}} : H^1(\mathcal{O}) \to \mathbb{P}_n(\mathcal{O})$, defined for any $v \in H^1(\mathcal{O})$ by:

$$\begin{cases} \displaystyle\int_{\mathcal{O}} \nabla q_n \cdot \nabla (v - \Pi_n^{\nabla,\mathcal{O}} v) \, \mathrm{d}\mathcal{O} = 0 & \text{for all } q_n \in {}_n(\mathcal{O}), \\ \displaystyle\int_{\partial\mathcal{O}} (v - \Pi_n^{\nabla,\mathcal{O}} v) \, \mathrm{d}\sigma = 0, \end{cases} \tag{1.4}$$

  with obvious extension for vector functions $\Pi_n^{\nabla,\mathcal{O}} : [H^1(\mathcal{O})]^d \to [\mathbb{P}_n(\mathcal{O})]^d$. We make note that the second condition in (1.4) is needed to fix the constant part.

1.4 Model problems

We now introduce the model problems that we are going to study. We firstly recall the Stokes model that will be the core of our work and it will be used in the Chapters 2,3 and 4. Then we also introduce the Oseen model, a non-symmetric problem that will be analyzed in Chapter 5.

1.4.1 Stokes Equation

Let $\Omega \subseteq \mathbb{R}^d$ with $d = 2, 3$ be a bounded Lipschitz domain, with $\Gamma = \partial\Omega$, and consider the stationary Stokes problem on Ω with homogeneous Dirichlet boundary conditions:

$$\begin{cases} \text{Find } (\mathbf{u}, p) \text{ such that} \\ -\nu \, \mathbf{div} \, \boldsymbol{\varepsilon}(\mathbf{u}) - \nabla p = \mathbf{f} & \text{in } \varOmega \\ \operatorname{div} \mathbf{u} = 0 & \text{in } \varOmega \\ \mathbf{u} = 0 & \text{on } \varGamma, \end{cases} \tag{1.5}$$

where $\mathbf{u}$ and p are respectively the velocity and the pressure fields, $\mathbf{f} \in [H^{-1}(\varOmega)]^d$ represents the external force and $\nu \in L^\infty(\varOmega)$ uniformly positive in $\varOmega$ is the viscosity. We assume that the scalar viscosity field ν is piece-wise constant with respect to the decomposition $\mathcal{T}_h$, i.e., ν is constant on each polyhedron $K \in \mathcal{T}_h$. Let us consider the spaces:

$$\mathbf{V} := [H_0^1(\mathit{\Omega})]^d, \qquad Q := L_0^2(\mathit{\Omega}) = \left\{ q \in L^2(\mathit{\Omega}) \quad s.t. \quad \int_{\mathit{\Omega}} q \, \mathrm{d}\Omega = 0 \right\} \tag{1.6}$$

with the norms previously introduced.

Let the bilinear forms $a(\cdot,\cdot) : \mathbf{V} \times \mathbf{V} \to \mathbb{R}$ and $b : \mathbf{V} \times Q \to \mathbb{R}$ be defined as:

$$a(\mathbf{u}, \mathbf{v}) := \int_\Omega \nu \nabla \mathbf{u} : \nabla \mathbf{v} \, \mathrm{d}\Omega \qquad \text{for all } \mathbf{u}, \mathbf{v} \in \mathbf{V} \tag{1.7}$$

$$b(\mathbf{v}, q) := \int_\Omega \mathrm{div}\, \mathbf{v} q \, \mathrm{d}\Omega \qquad \text{for all } \mathbf{u} \in \mathbf{V}, q \in Q. \tag{1.8}$$

Then a standard variational formulation of problem (1.5) reads:

$$\begin{cases} \text{find } (\mathbf{u}, p) \in \mathbf{V} \times Q \text{ such that} \\ a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = (\mathbf{f}, \mathbf{v})_0 & \text{for all } \mathbf{v} \in \mathbf{V}, \\ b(\mathbf{u}, q) = 0 & \text{for all } q \in Q. \end{cases} \tag{1.9}$$

It is well-known that:

- $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ are continuous, *i.e.*

$$\begin{aligned} |a(\mathbf{u}, \mathbf{v})| &\leq \|a\| \|\mathbf{u}\|_1 \|\mathbf{v}\|_1 && \text{for all } \mathbf{u}, \mathbf{v} \in \mathbf{V}, \\ |b(\mathbf{v}, q)| &\leq \|b\| \|\mathbf{v}\|_1 \|q\|_0 && \text{for all } \mathbf{v} \in \mathbf{V}\text{and } q \in Q, \end{aligned}$$

  where $\|a\|$ and $\|b\|$ are the usual norms of the two bilinear forms;

- $a(\cdot,\cdot)$ is coercive *i.e.*, there exists a positive constant α such that

$$|a(\mathbf{v}, \mathbf{v})| \geq \alpha \|\mathbf{v}\|_1^2 \quad \text{for all } \mathbf{v} \in \mathbf{V};$$

- the bilinear form $b(\cdot,\cdot)$ satisfies the inf-sup condition [23], *i.e.*

$$\exists \beta > 0 \text{ such that} \qquad \sup_{\mathbf{v} \in \mathbf{V}, \mathbf{v} \neq \mathbf{0}} \frac{|b(\mathbf{v}, q)|}{\|\mathbf{v}\|_1} \geq \beta \|q\|_0 \quad \text{for all } q \in Q. \tag{1.10}$$

Therefore, problem (1.9) has a unique solution $(\mathbf{u}, p) \in \mathbf{V} \times Q$ such that

$$||\mathbf{u}||_1 + ||p||_0 \leq C ||\mathbf{f}||_{H^{-1}(\mathit{\Omega})}, \tag{1.11}$$

where the constant C depends only on $\mathit{\Omega}$ and ν; see [23].

Symmetric gradient formulation. In our work, the previous definition of the bilinear form a (1.7) has been used into the two dimensional context. When we analyze the three-dimensional problem we exploit an equivalent definition that involves the symmetric gradient, we define:

$$a(\mathbf{u},\mathbf{v}) := \int_\Omega \nu \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) \, \mathrm{d}\Omega \qquad \text{for all } \mathbf{u},\mathbf{v} \in \mathbf{V}. \tag{1.12}$$

The bilinear forms (1.7) and (1.12) are equivalent as it is shown by the following (see [66] Section 2):

Lemma 1.4.1. *There exists a constant $c > 0$ such that:*

$$c\|\nabla \mathbf{u}\|_0 \leq \|\varepsilon(\mathbf{u})\|_0 \leq \|\nabla \mathbf{u}\|_0 \qquad \forall \mathbf{u} \in [H^1(\Omega)]^3, \mathbf{u} \perp ker(\varepsilon),$$

where $ker(\varepsilon)$ is the space of the rigid body modes of the elasticity problem.

Remark 1.4.1. We recall here the usual base for the space of the rigid body modes. Given a domain $\widehat{\Omega}$ with diameter H we have the three translations:

$$\mathbf{r}_1 := \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{r}_2 := \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{r}_3 := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{1.13}$$

and the three rotations:

$$\mathbf{r}_4 := \frac{1}{H}\begin{bmatrix} x_2 - \widehat{x}_2 \\ -x_1 + \widehat{x}_1 \\ 0 \end{bmatrix}, \quad \mathbf{r}_5 := \frac{1}{H}\begin{bmatrix} -x_3 + \widehat{x}_3 \\ 0 \\ x_1 - \widehat{x}_1 \end{bmatrix}, \quad \mathbf{r}_6 := \frac{1}{H}\begin{bmatrix} 0 \\ x_3 - \widehat{x}_3 \\ -x_2 + \widehat{x}_2 \end{bmatrix}, \tag{1.14}$$

where $\widehat{x} \in \widehat{\Omega}$. The shift of the origin makes this basis for the space of rigid body modes well conditioned, and, thanks to the scaling and shift, the L_2-norms of these six functions scale in the same way with H.

We also recall here a Korn-type inequality that will be useful in Chapter 3.

Lemma 1.4.2. *Let $\Omega \subset \mathbb{R}^3$ be a Lipschitz domain of diameter H and $\Sigma \subset \partial\Omega$ be an open subset with positive surface measure. There exists a positive constant C, independent of H such that:*

$$\inf_{\mathbf{r} \in ker(\varepsilon)} \|\mathbf{u} - \mathbf{r}\|_0^2 \leq CH|\mathbf{u}|_{E(\Sigma)} \qquad \forall \mathbf{u} \in [H^{1/2}(\Sigma)]^3,$$

where $|\mathbf{u}|_{E(\Sigma)} := \inf_{\mathbf{u} \in [H^1(\Omega)]^3, \mathbf{v}_{|\Sigma} = \mathbf{u}} \|\varepsilon(\mathbf{u})\|_0$.

1.4.2 Oseen Equation

Let $\Omega \subseteq \mathbb{R}^d$ with $d = 2, 3$ be a bounded Lipschitz domain, with $\Gamma = \partial\Omega$, and consider the stationary Oseen problem on Ω with homogeneous Dirichlet boundary conditions:

$$\begin{cases} \text{Find } (\mathbf{u}, p) \text{ such that} & \\ -\nu \, \mathbf{div} \, \boldsymbol{\varepsilon}(\mathbf{u}) + (\nabla \mathbf{u}) \, \boldsymbol{\beta} + \sigma \mathbf{u} - \nabla p = \mathbf{f} & \text{in } \mathit{\Omega} \\ \operatorname{div} \mathbf{u} = 0 & \text{in } \mathit{\Omega} \\ \mathbf{u} = 0 & \text{on } \mathit{\Gamma}. \end{cases} \tag{1.15}$$

where again, $\mathbf{u}$ and p are respectively the velocity and the pressure fields, $\mathbf{f} \in [H^{-1}(\mathit{\Omega})]^d$ represents the external force and $\nu, \sigma \in \mathbb{R}^+$ are respectively the viscosity diffusive coefficient and the reaction coefficient. $\boldsymbol{\beta} \in [W^{1,\infty}(\Omega)]^d$ with $\operatorname{div} \boldsymbol{\beta} = 0$ is the transport advective field.
Again, we consider the spaces $\mathbf{V}$ and Q as in (1.6), and the bilinear forms $A(\cdot, \cdot) : \mathbf{V} \times \mathbf{V} \to \mathbb{R}$, $b : \mathbf{V} \times Q \to \mathbb{R}$ and $c(\cdot, \cdot) : \mathbf{V} \times \mathbf{V} \to \mathbb{R}$ be defined as:

$$A(\mathbf{u}, \mathbf{v}) := \nu \int_\Omega \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) \, \mathrm{d}\Omega + \sigma \int_\Omega \mathbf{u} \cdot \mathbf{v} \, \mathrm{d}\Omega \qquad \text{for all } \mathbf{u}, \mathbf{v} \in \mathbf{V} \tag{1.16}$$

$$b(\mathbf{v}, q) := \int_\Omega \operatorname{div} \mathbf{v} \, q \, \mathrm{d}\Omega \qquad \text{for all } \mathbf{u} \in \mathbf{V}, q \in Q. \tag{1.17}$$

$$c(\mathbf{u}, \mathbf{v}) := \int_\Omega [(\nabla \mathbf{u}) \, \boldsymbol{\beta}] \cdot \mathbf{v} \, \mathrm{d}\Omega \qquad \text{for all } \mathbf{u}, \mathbf{v} \in \mathbf{V}. \tag{1.18}$$

Since div $\boldsymbol{\beta} = 0$, one can check that the bilinear form $c(\cdot, \cdot)$ is skew symmetric, that means $c(\mathbf{u}, \mathbf{v}) = -c(\mathbf{v}, \mathbf{u})$ for all $\mathbf{u}, \mathbf{v} \in \mathbf{V}$. Therefore $c(\cdot, \cdot)$ it is equal to its skew-symmetric part, defined as:

$$c^{skew}(\mathbf{u}, \mathbf{v}) = \frac{1}{2}\big(c(\mathbf{u}, \mathbf{v}) - c(\mathbf{v}, \mathbf{u})\big) \qquad \text{for all } \mathbf{u}, \mathbf{v} \in \mathbf{V}, \tag{1.19}$$

in general, at discrete level they lead to different bilinear forms. Then a standard variational formulation of problem (1.5) reads:

$$\begin{cases} \text{find } (\mathbf{u}, p) \in \mathbf{V} \times Q \text{ such that} & \\ A(\mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = (\mathbf{f}, \mathbf{v})_0 & \text{for all } \mathbf{v} \in \mathbf{V}, \\ b(\mathbf{u}, q) = 0 & \text{for all } q \in Q. \end{cases} \tag{1.20}$$

This problem can be written in the equivalent kernel form:

$$\begin{cases} \text{find } (\mathbf{u}, p) \in \mathbf{Z} \text{ such that} & \\ A(\mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \mathbf{v}) = (\mathbf{f}, \mathbf{v})_0 & \text{for all } \mathbf{v} \in \mathbf{Z}, \end{cases} \tag{1.21}$$

where $\mathbf{Z} = \{\mathbf{v} \in \mathbf{V} \text{ s.t. } \operatorname{div} \mathbf{v} = 0\}$.
This problem is well-posed and the unique solution $\mathbf{u} \in \mathbf{Z}$ is such that:

$$\nu \|\varepsilon(\mathbf{u})\|_0 + \sigma \|\mathbf{u}\|_0 \leq c\, \|\mathbf{f}\|_0,$$

where $c > 0$ depends on the Korn inequality constant.

Chapter 2

Divergence-free Virtual Element Methods

In this chapter we will recall the Divergence-free Virtual Element Methods, originally introduced in [12, 14], that we will use in the next chapters to discretize the Stokes problem (1.9). Since a complete book of these methods to which refer to does not exist yet, here we will briefly recall their formulations and we complete the proofs that usually are skipped in the papers. We also give a proof for an optimal VEM interpolant for the three dimensional Stokes case.
The outline of this chapter is the following: we firstly introduce the space of functions for the VEM methods and the discrete bilinear forms (in two and three dimension), then we define the discrete problem and finally we give the convergence theorem.

2.1 Virtual Element spaces and bilinear forms

As usual, the idea is to define two discrete spaces of function to approximate the two continuous ones $\mathbf{V}$ and Q. Like in the FEM context the idea is to define locally some space of functions and then glow them up together to construct the global spaces.

2.1.1 Virtual Elements in 2D

We start giving a look on how the VEM space are defined in two dimension.

Pressure space. To construct the local discrete space for the pressure Q_h we simply define:

$$Q_h^K := \mathbb{P}_{k-1}(K), \tag{2.1}$$

and it is easy to see that:

$$\dim(Q_h^K) = \dim(\mathbb{P}_{k-1}(K)) = \frac{(k+1)k}{2}. \tag{2.2}$$

Velocity space. We define now the discrete space for the velocity functions. To do so we need to introduce other space of polynomials. For $k \in \mathbb{N}$, other than $\mathbb{P}_k(K)$, let us define the spaces:

- $\mathbb{B}_k(K) := \{v \in C^0(\partial K) \text{ s.t. } v_{|e} \in \mathbb{P}_k(e) \quad \forall \text{ edge } e \in \partial K\}$,
- $G_k(K) := \nabla(\mathbb{P}_{k+1}(K)) \subseteq [\mathbb{P}_k(K)]^2$,
- $G_k(K)^\perp \subseteq [\mathbb{P}_k(K)]^2$ the L^2-orthogonal complement to $G_k(K)$.

On each element $K \in T_h$ we define, for $k \geq 2$, the following finite dimensional local virtual element spaces:

$$\begin{aligned}\mathbf{V}_h^K := \Big\{ \mathbf{v} \in [H^1(K)]^2 \quad \text{s.t.} \quad & \mathbf{v}_{|\partial K} \in \mathbb{B}_k(\partial K)]^2, \\ & \begin{cases} -\nu \boldsymbol{\Delta} \mathbf{v} - \nabla s \in G_{k-2}(K)^\perp, \\ \operatorname{div} \mathbf{v} \in \mathbb{P}_{k-1}(K), \end{cases} \quad \text{for some } s \in L^2(K) \Big\}\end{aligned} \tag{2.3}$$

Remark 2.1.1. We note that all the operators and equations above have to be interpreted in the distributional sense. In particular, the definition of $\mathbf{V}_h^K$ above is associated to a Stokes-like variational problem on K. We note that $[\mathbb{P}_k(K)]^2 \subseteq \mathbf{V}_h^K$, this is a crucial property of these space because as we will see it leads to an exactly divergence-free discrete velocity. We mention that one can use different definitions of the velocity local space, like using the natural vector extension of the standard Poisson space used in [10]. In this case we would not able to recover the divergence-free property.

We have (see the proof in [14]):

$$\begin{aligned}\dim(\mathbf{V}_h^K) &= \dim([\mathbb{B}_k(\partial K)]^2) + \dim(G_{k-2}(K)^\perp) + (\dim(\mathbb{P}_{k-1}(K)) - 1) \\ &= 2N_e k + \frac{(k-1)(k-2)}{2} + \frac{(k+1)k}{2} - 1.\end{aligned} \tag{2.4}$$

Degrees of freedom. Once we have defined the local space for the velocity and pressure and we know their dimension, we need to introduce suitable sets of DoFs to be able to handle these functions. Given a function $\mathbf{v} \in \mathbf{V}_h^K$ we take the following linear operators $\mathbf{D_V}$, split into four subsets (Figure 2.1):

- $\mathbf{D_V^1}$: the values of $\mathbf{v}$ at the vertices of the polygon K,
- $\mathbf{D_V^2}$: the values of $\mathbf{v}$ at $k-1$ distinct points of every edge $e \in \partial K$ (for the implementation we will take the $k-1$ internal points of the $(k+1)$-Gauss-Lobatto quadrature rule in e),

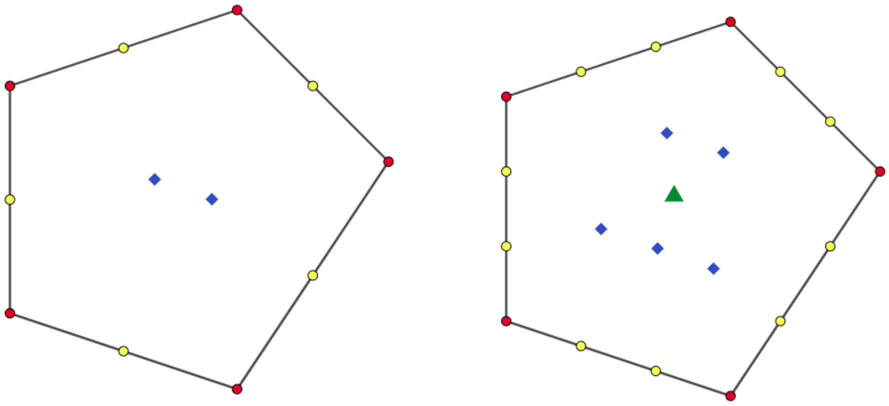


Figure 2.1: DoFs for $k = 2$, $k = 3$ for a velocity function $\mathbf{v} \in \mathbf{V}_h^k$. We denote $\mathbf{D_V^1}$ with the red dots, $\mathbf{D_V^2}$ with the yellow dots, $\mathbf{D_V^3}$ with the green triangles, $\mathbf{D_V^4}$ with the blue square inside the elements.

- $\mathbf{D_V^3}$: the moments of the values of $\mathbf{v}$

$$\int_K \mathbf{v} \cdot \mathbf{g}_{k-2}^{\perp} dK \qquad \text{for all } \mathbf{g}_{k-2}^{\perp} \in \mathbf{G}_{k-2}(K)^{\perp},$$

- $\mathbf{D_V^4}$: the moments up to order $k-1$ and greater than zero of div$\mathbf{v}$ in $K, i.e.$

$$\int_K (\text{div } \mathbf{v}) q_{k-1} dK \qquad \text{for all } q_{k-1} \in \mathbb{P}_{k-1}(K)/\mathbb{R}.$$

Furthermore, for the local pressure, given $q \in Q_h^K$, we consider the linear operators $\mathbf{D_Q}$:

- $\mathbf{D_Q}$: the moments up to order $k-1$ of $q, i.e.$

$$\int_K q \; p_{k-1} dK \qquad \text{for all } p_{k-1} \in \mathbb{P}_{k-1}(K).$$

Since $\mathbf{D_V}$ and $\mathbf{D_Q}$ are unisolvent respectively of $\mathbf{V}_h^K$ and Q_h^K (see [14]), we can define the global virtual element spaces:

$$\mathbf{V}_h := \{\mathbf{v} \in [H^1(\Omega)]^2 \quad \text{s.t.} \quad \mathbf{v}_{|K} \in \mathbf{V}_h^K \quad \text{for all } K \in T_h\} \tag{2.5}$$

and

$$Q_h := \{q \in L^2(\Omega) \quad \text{s.t.} \quad q_{|K} \in Q_h^K \quad \text{for all } K \in T_h\}, \tag{2.6}$$

with obvious associated sets of global DoFs. A computation shows that:

$$\dim(\mathbf{V}_h) = n_K\left(\frac{(k+1)k}{2} - 1 + \frac{(k-1)(k-2)}{2}\right) + 2(n_V + (k-1)n_E) \tag{2.7}$$

and

$$\dim(Q_h) = n_K\,\frac{(k+1)k}{2} - 1. \tag{2.8}$$

We can now define the polynomial projections that we need to discretize the bilinear forms.

Proposition 2.1.1 (Polynomial Projection)**.** *The dofs $\mathbf{D_V}$ allow us to compute exactly (with reference to* (1.3) *and* (1.4)*):*

$$\begin{aligned}
&\Pi_k^{\nabla,K} : \mathbf{V}_h^{K,k} \to [\mathbb{P}_k(K)]^2\\
&\Pi_{k-2}^{0,K} : \mathbf{V}_h^{K,k} \to [\mathbb{P}_{k-2}(K)]^2\\
&\boldsymbol{\Pi}_{k-1}^{0,K} : \boldsymbol{\nabla}\mathbf{V}_h^{K,k} \to [\mathbb{P}_{k-1}(K)]^{2\times 2}.
\end{aligned} \tag{2.9}$$

Proof. We start from the first operator. By definition of H_1 projection, to determine the polynomial $\Pi_k^{\nabla,K}$ for any $\mathbf{v} \in \mathbf{V}_h^{K,k}$ we need to compute:

$$\int_K \boldsymbol{\nabla}\mathbf{v} : \boldsymbol{\nabla}\mathbf{p}_k \qquad \text{for all } \mathbf{p}_k \in [\mathbb{P}_k(K)]^2.$$

Integrating twice by parts and using the decomposition in (1.2), that allows us to write $\boldsymbol{\Delta}\mathbf{p}_k = \nabla p_{k-1} + \mathbf{g}_{k-2}^{\perp}$ with $\boldsymbol{\Delta}\mathbf{p}_k \in [\mathbb{P}_{k-2}(K)]^2, p_{k-1} \in \mathbb{P}_{k-1}(K)$ and $\mathbf{g}_{k-2}^{\perp} \in G_{k-2}(K)^{\perp}$:

$$\begin{aligned}
\int_K \boldsymbol{\nabla}\mathbf{v} : \boldsymbol{\nabla}\mathbf{p}_k &= -\int_K \mathbf{v}\cdot\boldsymbol{\Delta}\mathbf{p}_k + \int_{\partial K} \mathbf{v}\cdot\big(\nabla\mathbf{p}_k\,\mathbf{n}_K\big)\\
&= \int_{\partial K} \mathbf{v}\cdot\big(\nabla\mathbf{p}_k\,\mathbf{n}_K\big) - \int_K \mathbf{v}\cdot\mathbf{g}_{k-2}^{\perp} - \int_K \mathbf{v}\cdot\nabla p_{k-1}\\
&= \int_{\partial K} \mathbf{v}\cdot\big(\nabla\mathbf{p}_k\,\mathbf{n}_K - q_{k-1}\,\mathbf{n}_K\big) - \int_K \mathbf{v}\cdot\mathbf{g}_{k-2}^{\perp} + \int_K p_{k-1}\,\mathrm{div}\,\mathbf{v}.
\end{aligned} \tag{2.10}$$

We see that the first term of the latter line of (2.10) is computable from the $\mathbf{D_V^1}$ and $\mathbf{D_V^2}$, the second by $\mathbf{D_V^3}$ and the last by the $\mathbf{D_V^4}$.

For the second operator, for any $\mathbf{v} \in \mathbf{V}_h^{K,k}$ and for all $\mathbf{p}_{k-2} \in [\mathbb{P}_{k-2}(K)]^2$, using the decomposition $\mathbf{p}_{k-2} = \nabla p_{k-1} + \mathbf{g}_{k-2}^{\perp}$ with $p_{k-1} \in \mathbb{P}_{k-1}(K)$ and $\mathbf{g}_{k-2}^{\perp} \in G_{k-2}(K)^{\perp}$ and integrating by parts, we can compute:

$$\begin{aligned}
\int_K \mathbf{v}\cdot\mathbf{p}_{k-2} &= \int_K \mathbf{v}\cdot\mathbf{g}_{k-2}^{\perp} + \int_K \mathbf{v}\cdot\nabla p_{k-1}\\
&= \int_K \mathbf{v}\cdot\mathbf{g}_{k-2}^{\perp} + \int_{\partial K} \mathbf{v}\cdot\big(q_{k-1}\,\mathbf{n}_K\big) - \int_K p_{k-1}\,\mathrm{div}\,\mathbf{v}.
\end{aligned} \tag{2.11}$$

It is easy to see that the terms of the right hand side of the latter equation can be computed by the $\mathbf{D_V}$.
The third operator is obtained in the same way as the first, for any $\mathbf{v} \in \mathbf{V}_h^{K,k}$ and for all $\underline{\mathbf{p}}_{k-1} \in [\mathbb{P}_{k-1}(K)]^{2\times 2}$, computing:

$$\int_K \boldsymbol{\nabla} \mathbf{v} : \underline{\mathbf{p}}_{k-1}, \tag{2.12}$$

this is done integrating twice by part and using the decomposition $\operatorname{div}\big(\underline{\mathbf{p}}_{k-1}\big) = \nabla p_{k-1} + \mathbf{g}_{k-2}^{\perp}$ with $\operatorname{div}\big(\underline{\mathbf{p}}_{k-1}\big) \in [\mathbb{P}_{k-2}(K)]^2, p_{k-1} \in \mathbb{P}_{k-1}(K)$ and $\mathbf{g}_{k-2}^{\perp} \in G_{k-2}(K)^{\perp}$. □

Remark 2.1.2. For sake of clarity, we just point out that the first and the latter operators in (2.11) are obtained combining the contributes computed in (2.10) and (2.12) with the orthogonality condition to fix the constant part as in (1.4). These projection operators are obtained by testing against the monomial basis functions (1.1) and they are expressed as matrices. We refer to [11] for a complete example on how to compute these operators.

Remark 2.1.3. Although we proved this projection here for a general degree k, in our 2D work we only use the polynomial projection for $k = 2$. We also mention that we proved the computability of the projection operator of the gradient of a virtual function since it is used to analyse the H_1 error and it is needed to discretize the $c(\cdot,\cdot)$ bilinear form in Chapter 5.

Discrete bilinear forms.

With these ingredients we are able to define two discrete versions of the bilinear forms $a(\cdot,\cdot)$ as in (1.7), and $b(\cdot,\cdot)$ as in (1.8). For what concerns $b(\cdot,\cdot)$ we do not need to introduce any approximation of the bilinear form, so we simply set:

$$b(\mathbf{v}, q) := \sum_{K\in T_h} b^K(\mathbf{v}, q) = \sum_{K\in T_h} \int_K \operatorname{div}\mathbf{v}\, q \; \mathrm{d}K \qquad \text{for all } \mathbf{u} \in \mathbf{V}_h, q \in Q_h. \tag{2.13}$$

Since q is a polynomial in each element $K \in T_h$, we notice that (2.13) is computable from the DoFs $\mathbf{D_V1}$, $\mathbf{D_V2}$ and $\mathbf{D_V4}$.
The approximation of the bilinear form $a(\cdot,\cdot)$ on the virtual space $\mathbf{V}_h$, is more involved. In fact, for an arbitrary couple $(\mathbf{w},\mathbf{v}) \in \mathbf{V}_h^K \times \mathbf{V}_h^K$, the quantity $a^K(\mathbf{w},\mathbf{w})$ is not computable. Following [14], we now define a computable discrete local bilinear form

$$a_h^K(\cdot,\cdot) : \mathbf{V}_h^K \times \mathbf{V}_h^K \to \mathbb{R} \tag{2.14}$$

approximating the continuous form $a^K(\cdot,\cdot)$, and satisfying the usual properties requested in the VEM framework [10]:

- $\mathbf{k-consistency}$: for all $\mathbf{q}_k \in [\mathbb{P}_k(K)]^2$ and $\mathbf{v} \in \mathbf{V}_h^K$

$$a_h^K(\mathbf{q}_k, \mathbf{v}_h) = a_h^K(\mathbf{q}_k, \mathbf{v}_h); \tag{2.15}$$

- **stability**: there exist two positive constants α_* and α^*, independent of h and K, such that, for all $\mathbf{v} \in \mathbf{V}_h^K$, it holds

$$\alpha_* a^K(\mathbf{v}_h, \mathbf{v}_h) \leq a_h^K(\mathbf{v}_h, \mathbf{v}_h) \leq \alpha^* a^K(\mathbf{v}_h, \mathbf{v}_h). \tag{2.16}$$

In the same way as in (2.10), we are able to show that $a^K(\mathbf{q}_k, \mathbf{v}_h)$ is computable in terms of the DoFs $\mathbf{D_V}$. At this point it is sufficient to take $a_h^K(\mathbf{w}_k, \mathbf{v}_h) = a^K(\Pi_k^{\nabla,K}\mathbf{w}_k, \Pi_k^{\nabla,K}\mathbf{v}_h)$ to ensure the property (2.15), but in general (2.16) would not be verified. So, we introduce a symmetric positive definite bilinear form (stabilizing-term) $S^K : \mathbf{V}_h^K \times \mathbf{V}_h^K \to \mathbb{R}$, that satisfies

$$c_* a^K(\mathbf{v}_h, \mathbf{v}_h) \leq S^K(\mathbf{v}_h, \mathbf{v}_h) \leq c^* a^K(\mathbf{v}_h, \mathbf{v}_h) \quad \text{for all } \mathbf{v}_h \in \mathbf{V}_h \text{ such that } \Pi_k^{\nabla,K}\mathbf{v}_h = \mathbf{0}, \tag{2.17}$$

where c_* and c^* are two positive constants independent of h and K.
Setting

$$a_h^K(\mathbf{w}_k, \mathbf{v}_h) = a^K(\Pi_k^{\nabla,K}\mathbf{w}_k, \Pi_k^{\nabla,K}\mathbf{v}_h) + S^K((I - \Pi_k^{\nabla,K})\mathbf{w}_h, (I - \Pi_k^{\nabla,K})\mathbf{v}_h) \tag{2.18}$$

for all $\mathbf{w}_h, \mathbf{v}_h \in \mathbf{V}_h^K$, we can prove the:

Lemma 2.1.1. *The bilinear form* (2.18) *satisfies the k-consistency property* (2.15) *and the stability property* (2.16).

Proof. Property (2.15) follows immediately from the definition of the projection operators: for $\mathbf{q} \in [\mathbb{P}_k(K)]^2$ we have $S^K((I-\Pi_k^{\nabla,K})\mathbf{q}, (I-\Pi_k^{\nabla,K})\mathbf{v}) = 0$ for all $\mathbf{v} \in \mathbf{V}_h^K$. Hence, for all $\mathbf{v} \in \mathbf{V}_h^K$ it holds

$$a_h^K(\mathbf{q}, \mathbf{v}) = a^K(\Pi_k^{\nabla,K}\mathbf{q}, \Pi_k^{\nabla,K}\mathbf{v}) = a^K(\mathbf{q}, \mathbf{v}). \tag{2.19}$$

Property (2.16) follows from (2.17): for all $\mathbf{v} \in \mathbf{V}_h^K$

$$\begin{aligned} a_h^K(\mathbf{v}, \mathbf{v}) &\leq a^K(\Pi_k^{\nabla,K}\mathbf{v}, \Pi_k^{\nabla,K}\mathbf{v}) + c^* a^K((I - \Pi_k^{\nabla,K})\mathbf{v}, (I - \Pi_k^{\nabla,K})\mathbf{v}) \\ &\leq \max\{1, c^*\}(a^K(\Pi_k^{\nabla,K}\mathbf{v}, \Pi_k^{\nabla,K}\mathbf{v}) + a^K(\mathbf{v} - \Pi_k^{\nabla,K}\mathbf{v}, \mathbf{v} - \Pi_k^{\nabla,K})\mathbf{v}) \\ &= \alpha^* a^K(\mathbf{v}, \mathbf{v}). \end{aligned} \tag{2.20}$$

Similarly, for all $\mathbf{v} \in \mathbf{V}_h^K$,

$$\begin{aligned} a_h^K(\mathbf{v}, \mathbf{v}) \geq \; &\min\{1, c_*\}(a^K(\Pi_k^{\nabla,K}\mathbf{v}, \Pi_k^{\nabla,K}\mathbf{v}) + a^K(\mathbf{v} - \Pi_k^{\nabla,K}\mathbf{v}, \mathbf{v} - \Pi_k^{\nabla,K})\mathbf{v}) \\ &= \alpha_* a^K(\mathbf{v}, \mathbf{v}). \end{aligned} \tag{2.21}$$

□

Remark 2.1.4. The choice of the bilinear form S^K in general depend on the problem and on the DoFs. Condition (3.27) states that the stabilizing term $S^K(\mathbf{v}, \mathbf{v})$ scales as $a^K(\mathbf{v}, \mathbf{v})$ on the kernel of $\Pi_k^{\nabla,K}$. In our numerical tests we denote with $\tilde{\mathbf{v}}_h, \tilde{\mathbf{w}}_h \in \mathbb{R}^{N_K}$ the vectors containing the values of N_K local DoFs associated to $\mathbf{v}_h, \mathbf{w}_h \in \mathbf{V}_h^K$; and we set

$$S^K(\mathbf{v}_h, \mathbf{w}_h) = \alpha^K \tilde{\mathbf{v}}_h^T \tilde{\mathbf{w}}_h, \tag{2.22}$$

with α^K a suitable positive constant independent of the element size.

Finally we define the global approximated bilinear form $a_h(\cdot,\cdot) : \mathbf{V}_h^K \times \mathbf{V}_h^K \to \mathbb{R}$ by simply summing the local contributions:

$$a_h(\mathbf{w}_h, \mathbf{v}_h) = \sum_{K \in T_h} a_h^K(\mathbf{w}_h, \mathbf{v}_h) \quad \text{for all } \mathbf{w}_h, \mathbf{v}_h \in \mathbf{V}_h. \tag{2.23}$$

Load term approximation Now we present a computable approximation of the right-hand side $(\mathbf{f}, \mathbf{v})$ in (1.9). Let $K \in T_h$, we define the approximate load term $\mathbf{f}_h$ as

$$\mathbf{f}_h := \Pi_{k-2}^{0,K} \mathbf{f}_h \quad \text{for all } K \in T_h, \tag{2.24}$$

and consider:

$$(\mathbf{f}_h, \mathbf{v}_h) = \sum_{K \in T_h} \int_K \mathbf{f}_h \cdot \mathbf{v}_h dK = \sum_{K \in T_h} \int_K \Pi_{k-2}^{0,K} \mathbf{f} \cdot \mathbf{v}_h dK = \sum_{K \in T_h} \int_K \mathbf{f} \cdot \Pi_{k-2}^{0,K} \mathbf{v}_h dK. \tag{2.25}$$

We observe that (2.25) can be exactly computed for all $\mathbf{v}_h \in \mathbf{V}_h$. In fact, $\Pi_{k-2}^{0,K} \mathbf{v}_h$ is computable in terms of the DoFs $\mathbf{D_V}$: for all $\mathbf{q}_h \in [\mathbb{P}_{k-2}(K)]^2$ we have

$$\int_K \Pi_{k-2}^{0,K} \mathbf{v}_h \cdot \mathbf{q}_{k-2} dK = \int_K \mathbf{v}_h \cdot \mathbf{q}_{k-2} dK = \int_K \mathbf{v}_h \cdot \nabla q_{k-1} dK + \int_K \mathbf{v}_h \cdot \mathbf{g}_{k-2}^{\perp} dK \tag{2.26}$$

for suitable $q_{k-1} \in \mathbb{P}_{k-1}(K)$ and $\mathbf{g}_{k-2}^{\perp} \in G_{k-2}(K)^{\perp}$. As a consequence, we get

$$\int_K \Pi_{k-2}^{0,K} \mathbf{v}_h \cdot \mathbf{q}_{k-2} dK = -\int_K \operatorname{div} \mathbf{v}_h \cdot q_{k-1} dK + \int_{\partial K} q_{k-1} \mathbf{v}_h \cdot \mathbf{n} dK + \int_K \mathbf{v}_h \cdot \mathbf{g}_{k-2}^{\perp} dK, \tag{2.27}$$

and the right-hand side is directly computable from $\mathbf{D_V}$.
We have the following lemma [10]

Lemma 2.1.2. *Let $\mathbf{f}_h$ be defined as in (2.24), and let us assume $\mathbf{f} \in H^{k-1}(\Omega)$. Then for all $\mathbf{v}_h \in \mathbf{V}_h$, it holds*

$$|(\mathbf{f}_h - \mathbf{f}, \mathbf{v}_h)| \leq Ch^k |\mathbf{f}|_{k-1} \|\mathbf{v}_h\|_1. \tag{2.28}$$

2.1.2 Virtual Elements in 3D

Pressure space. We start by constructing the discrete space Q_h. This is a natural extension of the two-dimensional space [14] and, following [12], we define:

$$Q_h^K := \mathbb{P}_{k-1}(K), \tag{2.29}$$

therefore the corresponding dofs are chosen defining for each $q \in Q_h^K$ the following linear operator:

- $\mathbf{D_Q}$: the moments up to order $k-1$ of q:

$$\int_K q p_{k-1} \, \mathrm{d}K \qquad \text{for any } p_{k-1} \in \mathbb{P}_{k-1}(K).$$

The global space is given by:

$$Q_h := \{q \in L^2(\Omega) \quad \text{s.t.} \quad q_{|K} \in Q_h^K \quad \text{for all } K \in \mathcal{T}_h\}. \tag{2.30}$$

Velocity space. The space $\mathbf{V}_h$, as defined in [12], is the three-dimensional extension of the two-dimensional velocity space introduced in [14], where the extensive use of the enhancement technique [2] is needed to achieve the computability of suitable polynomial projection operators. We start by considering each face f of a polyhedral element K, then we define:

$$\begin{aligned} \widehat{\mathbb{B}}_k(f) := \{v \in & H^1(f) \text{ s.t. (i) } v_{|f} \in C^0(\partial f), v_{|e} \in \mathbb{P}_k(e) \text{ for all } e \in \partial f, \\ & \text{(ii) } \Delta_f v \in \mathbb{P}_{k+1}(f), \\ & \text{(iii) } (v - \Pi_k^{\nabla,f} v, \widehat{p}_{k+1})_f = 0 \text{ for all } \widehat{p}_{k+1} \in \widehat{\mathbb{P}}_{k+1\setminus k-2}(f)\} \end{aligned} \tag{2.31}$$

and the boundary space:

$$\widehat{\mathbb{B}}_k(\partial K) := \{v \in C^0(\partial K) \text{ s.t. } v_{|f} \in \widehat{\mathbb{B}}_k(f) \text{ for any } f \in \partial K\}. \tag{2.32}$$

Then on the polyhedron K we first define the virtual element space:

$$\begin{aligned} \widetilde{\mathbf{V}}_h^K := \{\mathbf{v} \in & [H^1(K)]^3 \text{ s.t. (i) } \mathbf{v}_{|\partial K} \in [\widehat{\mathbb{B}}_k(\partial K)]^3, \\ & \text{(ii) } \Delta \mathbf{v} + \nabla s \in \mathbf{x} \wedge [\mathbb{P}_{k-1}(K)]^3 \text{ for some } s \in L_0^2(K), \\ & \text{(iii) } \operatorname{div} \mathbf{v} \in \mathbb{P}_{k-1}(K)\}, \end{aligned} \tag{2.33}$$

being $\mathbf{x} = (x_1, x_2, x_3)$ the independent variables and $\wedge$ the cross product. The velocity space is then defined as:

$$\begin{aligned} \mathbf{V}_h^K := \{\mathbf{v} \in \widetilde{\mathbf{V}}_h^K \text{ s.t. } & (\mathbf{v} - \Pi_k^{\nabla,K} \mathbf{v}, \mathbf{x} \wedge \widehat{\mathbf{p}}_{k-1})_K = 0 \\ & \forall \widehat{\mathbf{p}}_{k-1} \in [\widehat{\mathbb{P}}_{k-1\setminus k-3}(K)]^3\}. \end{aligned} \tag{2.34}$$

Remark 2.1.5. The "super-enhanced" constraints (iii) in (2.31) and in (2.34) are necessary to achieve the computability of the polynomial projection operators $\Pi^{0,f}_{k+1}$ and $\Pi^{0,K}_k$ (see Prop. 2.1.2).

Remark 2.1.6. Note that the approximation property is again guaranteed by the fact that the spaces $\mathbf{V}_h^K$ and Q_h^K contain $[\mathbb{P}_k(K)]^3$ and $\mathbb{P}_{k-1}(K)$, respectively.

Degrees of freedom Given $\mathbf{v} \in \mathbf{V}_h^K$, the dofs of the local velocity space $\mathbf{V}_h^K$ are defined by means of the following set of linear operators:

- $\mathbf{D^1_V}$: the values of $\mathbf{v}$ at the vertices of K;
- $\mathbf{D^2_V}$: the values of $\mathbf{v}$ at $k-1$ distinct points of every edge e of K;
- $\mathbf{D^3_V}$: the face moments of $\mathbf{v}$ (split into normal and tangential components):
$$\int_f (\mathbf{v} \cdot \mathbf{n}_K^f) p_{k-2} \,\mathrm{d}f, \quad \int_f \mathbf{v}_\tau \cdot \mathbf{p}_{k-2} \,\mathrm{d}f, \tag{2.35}$$
for all $p_{k-2} \in \mathbb{P}_{k-2}(f)$ and $\mathbf{p}_{k-2} \in [\mathbb{P}_{k-2}(f)]^2$, where $\mathbf{n}_K^f$ is the normal vector associated to the face f and $\mathbf{v}_\tau$ is the 2D vector field defined on ∂K, s.t. on each face $f \in \partial K$:
$$\mathbf{v}_\tau := \mathbf{v} - (\mathbf{v} \cdot \mathbf{n}_K^f)\mathbf{n}_K^f;$$
- $\mathbf{D^4_V}$: the volume moments of $\mathbf{v}$:
$$\int_K \mathbf{v} \cdot (\mathbf{x} \wedge \mathbf{p}_{k-3}) \,\mathrm{d}K \qquad \text{for all } \mathbf{p}_{k-3} \in [\mathbb{P}_{k-3}(K)]^3; \tag{2.36}$$
- $\mathbf{D^5_V}$: the volume moments of $\operatorname{div}\mathbf{v}$:
$$\int_K \operatorname{div}\mathbf{v}\, \widehat{p}_{k-1} \,\mathrm{d}K \qquad \text{for all } \widehat{p}_{k-1} \in \widehat{\mathbb{P}}_{k-1\setminus 0}(K). \tag{2.37}$$

The global space $\mathbf{V}_h$ is obtained by gluing the local spaces:
$$\mathbf{V}_h := \{\mathbf{v} \in [H^1(\Omega)]^3 \text{ s.t. } \mathbf{v}_{|K} \in \mathbf{V}_h^K \text{ for all } K \in \mathcal{T}_h\}. \tag{2.38}$$

Recalling Proposition 5.1 in [12], we now show how to compute some useful projection operators as we have done before in the two dimensional case. We want to do so, since this aspect is not usually covered in the papers and it is left to the reader. Moreover the first two projection operators in Prop. 2.1.2 show the importance of having defined a super-enhanced VEM space for the velocity in (2.31) and (2.34).

Proposition 2.1.2 (Polynomial projections)**.** *The DoFs $\mathbf{D_V}$ allow us to compute exactly the face projections*

$$\Pi_k^{\nabla,f} : [\widehat{\mathbb{B}}_k(f)]^3 \to [\mathbb{P}_k(f)]^3, \quad \Pi_{k+1}^{0,f} : [\widehat{\mathbb{B}}_k(f)]^3 \to [\mathbb{P}_{k+1}(f)]^3$$

for any $f \in \partial K$ of an element K, and the element projections:

$$\begin{aligned} &\Pi_k^{\nabla,K} : \mathbf{V}_h(K) \to [\mathbb{P}_k(K)]^3, \\ &\Pi_{k-1}^{0,K} : \nabla(\mathbf{V}_h(K)) \to [\mathbb{P}_{k-1}(K)]^{3\times 3}, \\ &\boldsymbol{\Pi}_k^{0,K} : \mathbf{V}_h(K) \to [\mathbb{P}_k(K)]^3 \end{aligned}$$

in the sense that, given any $\mathbf{v}_h \in \mathbf{V}_h(K)$, we are able to compute the polynomials $\Pi_k^{\nabla,f}\mathbf{v}_h$, $\Pi_{k+1}^{0,f}\mathbf{v}_h$, $\Pi^{\nabla,K}\mathbf{v}_h$, $\Pi_k^{0,K}(\nabla\mathbf{v}_h)$ and $\boldsymbol{\Pi}_k^{0,K}\mathbf{v}_h$ using only, as unique information, the DoFs values $\mathbf{D_V}$ of $\mathbf{v}_h$.

Proof. The projection operator $\Pi_k^{\nabla,f}$ is determined once we have computed the contributes:

$$\int_f \nabla\mathbf{v}_h : \nabla\mathbf{p}_k \qquad \forall\mathbf{p}_k \in [\mathbb{P}_k(f)]^3. \tag{2.39}$$

By simply integrating by part we obtain:

$$\int_f \nabla\mathbf{v}_h : \nabla\mathbf{p}_k = -\int_f \Delta\mathbf{p}_k \cdot \mathbf{v}_h + \int_{\partial f} (\nabla\mathbf{p}_k\, \mathbf{n}_f) \cdot \mathbf{v}_h. \tag{2.40}$$

Since $\Delta\mathbf{p}_k \in [\mathbb{P}_{k-2}(f)]^3$, the first term is computable from $\mathbf{D_V^3}$ (after having properly splitted into normal and tangential component) and the latter from $\mathbf{D_V^1}$ and $\mathbf{D_V^2}$.
For what concerns the operator $\Pi_{k+1}^{0,f}$ we need to compute:

$$\int_f \mathbf{v}_h \cdot \mathbf{p}_{k+1} \qquad \forall\mathbf{p}_{k+1} \in [\mathbb{P}_{k+1}(f)]^3. \tag{2.41}$$

We can note that, for any $\mathbf{p} \in [\mathbb{P}_{k-2}(f)]^3$ we have exactly the $\mathbf{D_V^3}$ as before. When $\mathbf{p} \in [\mathbb{P}_{k+1\setminus k-2}(f)]^3$ instead we do not have any information from the dofs. The help is given by the operator $\Pi_k^{\nabla,f}$ that now we know explicitly, and that, due to the definition of our spaces $\widehat{\mathbb{B}}_k(f)$ (2.31), allows us to compute (2.41).
The operator $\Pi_k^{\nabla,K}$ is determined by computing:

$$\int_K \nabla\mathbf{v}_h : \nabla\mathbf{p}_k \qquad \forall\mathbf{p}_k \in \mathbb{P}_k(K). \tag{2.42}$$

Firstly integrating by parts, using the second polynomial decomposition in (1.2) that allows us to write $\Delta\mathbf{p}_k \in [\mathbb{P}_{k-2}(K)]^3$ as $\Delta\mathbf{p}_k = \mathbf{x} \wedge \mathbf{p}_{k-3} + \nabla\widehat{p}_{k-1}$ for some $\widehat{p}_{k-1} \in \mathbb{P}_{k-1\setminus 0}(K)$ and

$\mathbf{p}_{k-3} \in [\mathbb{P}_{k-3}(K)]^3$, we obtain:

$$\begin{aligned}\int_K \nabla \mathbf{v}_h : \nabla \mathbf{p}_k = & - \int_K \Delta \mathbf{p}_k \cdot \mathbf{v}_h + \int_{\partial K} (\nabla \mathbf{p}_k \, \mathbf{n}_K) \cdot \mathbf{v}_h = \\ & - \int_K \nabla \widehat{p}_{k-1} \cdot \mathbf{v}_h - \int_K (\mathbf{x} \wedge \mathbf{p}_{k-3}) \cdot \mathbf{v}_h + \int_{\partial K} (\nabla \mathbf{p}_k \, \mathbf{n}_K) \cdot \mathbf{v}_h\end{aligned} \tag{2.43}$$

The second term is directly computable from $\mathbf{D^3_V}$ and the third term, since $\nabla \mathbf{p}_k \cdot \mathbf{n} \in [\mathbb{P}_{k-1}(\partial K)]^3$, is computable in the same way as in (2.41) from $\mathbf{D^4_V}$. The first term is obtained integrating by parts:

$$- \int_K \nabla \widehat{p}_{k-1} \cdot \mathbf{v}_h = \int_K \operatorname{div} \mathbf{v}_h \, \widehat{p}_{k.1} + \int_{\partial K} (\widehat{p}_{k-1} \, \mathbf{n}_K) \cdot \mathbf{v}_h, \tag{2.44}$$

where the first came directly from $\mathbf{D^5_V}$ and the second as the last term of the previous equation. We only give a hint for the operator $\mathbf{\Pi}^{0,K}_{k-1}$, since the procedure is really similar to the previous one. In this case we integrate by parts, $\forall \underline{\mathbf{p}}_{k-1} \in [\mathbb{P}_{k-1}(K)]^{3\times 3}$:

$$\int_K \nabla \mathbf{v}_h : \underline{\mathbf{p}}_{k-1} = - \int_K \mathbf{v}_h \cdot \operatorname{div} \underline{\mathbf{p}}_{k-1} + \int_{\partial K} (\underline{\mathbf{p}}_{k-1} \, \mathbf{n}_K) \cdot \mathbf{v}_h, \tag{2.45}$$

then we use the split $\operatorname{div} \underline{\mathbf{p}}_{k-1} = \mathbf{x} \wedge \mathbf{p}_{k-3} + \nabla \widehat{p}_{k-1}$ for some $\mathbf{p}_{k-3} \in [\mathbb{P}_{k-3}(K)]^3$ and $\widehat{p}_{k-1} \in \mathbb{P}_{k-1}(K)$ and we proceed as before.
The proof for the latter operator could be found in [12]. □

Discrete bilinear forms and load term approximation. We can now discuss the discretization of the bilinear forms defined in (1.9). First, we decompose into local contribution the bilinear forms $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$ and the load term $\mathbf{f}$:

$$a(\mathbf{u}, \mathbf{v}) := \sum_{K \in \mathcal{T}_h} a^K(\mathbf{u}, \mathbf{v}), \quad b(\mathbf{v}, p) := \sum_{K \in \mathcal{T}_h} b^K(\mathbf{v}, p), \quad (\mathbf{f}, \mathbf{v}) := \sum_{K \in \mathcal{T}_h} (\mathbf{f}, \mathbf{v})_K, \tag{2.46}$$

for all $\mathbf{u}, \mathbf{v} \in [H^1(\Omega)]^3$.

As previously, we note that we do not need any approximation for the divergence bilinear form since we can compute exactly $b(\mathbf{v}_h, q_h)$ for all $\mathbf{v}_h \in \mathbf{V}_h$ and $q_h \in Q_h$ directly form the $\mathbf{D^1_V}, \mathbf{D^2_V}$ and $\mathbf{D^5_V}$.

Instead, the bilinear form $a(\cdot, \cdot)$ is not directly computable from the dofs when both entries are "virtual". Following [12], we define the approximation:

$$a_h^K(\mathbf{u}, \mathbf{v}) := \int_K (\Pi^{0,K}_{k-1} \varepsilon(\mathbf{u})) : (\Pi^{0,K}_{k-1} \varepsilon(\mathbf{v})) \, \mathrm{d}K + S^K((I - \Pi^{\nabla,K}_k)\mathbf{u}, (I - \Pi^{\nabla,K}_k)\mathbf{v}), \tag{2.47}$$

for all $\mathbf{u}, \mathbf{v} \in \mathbf{V}_h^K$, where:

$$\Pi_{k-1}^{0,K}\varepsilon(\mathbf{u}) = \frac{\Pi_{k-1}^{0,K}\nabla\mathbf{u} + (\Pi_{k-1}^{0,K}\nabla\mathbf{u}))^T}{2}.$$

The approximate bilinear form (2.47) is obtained as the sum of the first contribute, the consistency part and the second one, the stabilization part, where $S^P : \mathbf{V}_h^K \times \mathbf{V}_h^K \to \mathbb{R}$ is a suitable symmetric bilinear form that has to scale like the H^1-seminorm.

Remark 2.1.7. For the numerical experiments in Section 3.4.2 and 4.4, we use, instead of $\Pi_k^{\nabla,K}$, a different projection operator $\Pi_k^{\mathcal{D},K}$ since it is nothing but a Euclidean projection with respect to the DoF vectors and it is easier to implement. This dof-based stabilization has been introduced in Section 6 of [12], and we remand there for further details.

The load term is approximated by taking:

$$(\mathbf{f}_h, \mathbf{v})_K := \int_K \Pi_k^{0,K} f \cdot \mathbf{v} \,\mathrm{d}K. \tag{2.48}$$

Finally, the global forms are obtained by simply gluing elements' contributions:

$$a_h(\mathbf{u}, \mathbf{v}) := \sum_{K\in\mathcal{T}_h} a_h^K(\mathbf{u}, \mathbf{v}), \quad (\mathbf{f}_h, \mathbf{v}) := \sum_{K\in\mathcal{T}_h} (\mathbf{f}_h, \mathbf{v})_K, \tag{2.49}$$

for all $\mathbf{u}, \mathbf{v} \in \mathbf{V}_h$.

2.2 Discrete problem and theoretical results

Using the discrete spaces (2.5) and (2.6) in 2D, and (2.38) and (2.30) in 3D and the respectively discrete linear and bilinear forms previously introduced, the discrete Stokes problem reads as follows:

$$\begin{cases} \text{find } (\mathbf{u}_h, p_h) \in \mathbf{V}_{h,0} \times Q_{h,0} \text{ such that} \\ a_h(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = (\mathbf{f}_h, \mathbf{v}_h) & \text{for all } \mathbf{v}_h \in \mathbf{V}_{h,0}, \\ b(\mathbf{u}_h, q_h) = 0 & \text{for all } q_h \in Q_{h,0}, \end{cases} \tag{2.50}$$

where $\mathbf{V}_{h,0} := \mathbf{V}_h \cap [H_0^1(\Omega)]^d$ and $Q_{h,0} := Q_h \cap L_0^2(\Omega)$.
The convergence theorem is given by the fact that we have an optimal interpolation Stokes-like operator (Lemma 2.2.1) and that the pair $(\mathbf{V}_h, Q_h)$ is inf-sup stable with $\beta_h > 0$ (Prop. 2.2.1). The proof for the bidimensional case can be found in [14], while for the three dimensional case we need to combine the arguments in [14], [15] and [27]. To give an idea on how to combine these techniques we prove here the Stokes optimal interpolant.

Lemma 2.2.1. *Let* $\mathbf{v} \in [H^{1+s}(K)]^3$, $0 \leq s \leq 1$, *there exists* $\mathbf{v}_I \in \mathbf{V}_h(K)$ *s.t.:*

$$\|\mathbf{v} - \mathbf{v}_I\|_{0,K} + |\mathbf{v} - \mathbf{v}_I|_{1,K} \leq h^{1+s}|\mathbf{v}|_{1+s,\widetilde{K}}.$$

Proof. The proof of this lemma is divided into three steps.
Step 1. Interpolant on faces.
Let K be an element of the VEM tassellation and f a face with $f \in \partial K$. We consider $\widetilde{\mathcal{T}_h}$ a sub-triangulation of $\mathcal{T}_h$ and let $\mathbf{v}_c$ be the Clement interpolant of $\mathbf{v}$ relative to the sub-triangulation. We have that: $\|\mathbf{v} - \mathbf{v}_c\|_0 + |\mathbf{v} - \mathbf{v}_c|_1 \leq h|\mathbf{v}|_{1,f}$. Now we interpolate $\mathbf{v}_c$ on the larger face space:

$$\begin{aligned}\mathbb{B}_k(f) := \{v \in &H^1(f) \text{ s.t. (i) } v_{|e} \in C^0(\partial f), v_{|e} \in \mathbb{P}_k(e) \text{ for all } e \in \partial f,\\ &\text{(ii) } \Delta_f v \in \mathbb{P}_{k+1}(f)\}\end{aligned} \tag{2.51}$$

and we define $\mathbf{w}_I \in [\mathbb{B}_k(f)]^3$ as the solution of:

$$\begin{cases} -\Delta \mathbf{w}_I = -\Delta \Pi^0_{k+1}\mathbf{v}_c & \text{on } f \\ \mathbf{w}_I = \mathbf{w}_c & \text{on } \partial f. \end{cases} \tag{2.52}$$

We see that $\Delta\Pi^0_{k+1}\mathbf{v}_c \in [\mathbb{P}_{k-2}(f)]^3 \subset [\mathbb{P}_{k+1}(f)]^3$, $\mathbf{v}_c$ is continuous by definition on Ω and $\mathbf{v}_c|_e \in [\mathbb{P}_k(e]^3)$ $\forall e \in \partial f$, so we conclude $\mathbf{w}_I \in [\mathbb{B}_k(f)]^3$. Subtracting $\Pi^0_{k+1}\mathbf{v}_c$ in the second equation:

$$\begin{cases} -\Delta \mathbf{w}_I = -\Delta \Pi^0_{k+1}\mathbf{v}_c & \text{on } f \\ \mathbf{w}_I - \Pi^0_{k+1}\mathbf{v}_c = \mathbf{w}_c - \Pi^0_{k+1}\mathbf{v}_c & \text{on } \partial f, \end{cases} \tag{2.53}$$

we have:

$$\begin{aligned}|\mathbf{w}_I - \Pi^0_{k+1}\mathbf{v}_c|_{1,f} \leq \inf\{|\mathbf{z}|_{1,f}, \mathbf{z} \in [H^1(f)]^3 : \mathbf{z} = \mathbf{v}_c - \Pi^0_{k+1}\mathbf{v}_c \text{ on } \partial f\}\\ \leq |\mathbf{v}_c - \Pi^0_{k+1}\mathbf{v}_c|_{1,f}.\end{aligned} \tag{2.54}$$

Now using the triangular inequality:

$$|\mathbf{v}_c - \mathbf{w}_I|_{1,f} \leq |\mathbf{v}_c - \Pi^0_{k+1}\mathbf{v}_c|_{1,f} + |\Pi^0_{k+1}\mathbf{v}_c - \mathbf{w}_I|_{1,f} \leq 2|\mathbf{v}_c - \Pi^0_{k+1}\mathbf{v}_c|_{1,f}. \tag{2.55}$$

Sofar we have $\mathbf{w}_I \in [\mathbb{B}_k(f)]^3$, but we are looking for $\mathbf{v}_I \in [\widehat{\mathbb{B}}_k(f)]^3$. To construct $\mathbf{v}_I$ we set $\mathbf{v}_I = \mathbf{w}_I$ on ∂f and:

$$\begin{aligned}\int_f \mathbf{v}_I \cdot \mathbf{p} = \int_f \mathbf{w}_I \cdot \mathbf{p} \qquad &\forall \mathbf{p} \in [\mathbb{P}_{k-2}]^3\\ \int_f \mathbf{v}_I \cdot \mathbf{p} = \int_f \Pi^{\nabla,f}_k \mathbf{w}_I \cdot \mathbf{p} \qquad &\forall \mathbf{p} \in [\mathbb{P}_{k+1\backslash k-1}]^3,\end{aligned} \tag{2.56}$$

Remembering that $\mathbf{v}_I = \mathbf{w}_I$ on ∂f and integrating by parts:

$$|\mathbf{w}_I - \mathbf{v}_I|^2_{1,f} = \int_f |\nabla(\mathbf{w}_I - \mathbf{v}_I)|^2 = -\int_f \Delta(\mathbf{w}_I - \mathbf{v}_I)(\mathbf{w}_I - \mathbf{v}_I). \tag{2.57}$$

By definition $\mathbf{w}_I - \mathbf{v}_I \in [\mathbb{B}_k(f)]^3$, so $\Delta(\mathbf{w}_I - \mathbf{v}_I) \in \mathbb{P}_{k+1}(f) \implies \Delta(\mathbf{w}_I - \mathbf{v}_I) = \mathbf{p} + \Pi^{0,f}_{k-2}\Delta(\mathbf{w}_I - \mathbf{v}_I)$ for some $\mathbf{p} \in [\mathbb{P}_{k+1\backslash k-2}(f)]^3$. In this way we can write $\mathbf{p} = -\Delta(\mathbf{w}_I - \mathbf{v}_I) + \Pi^{0,f}_{k-2}\Delta(\mathbf{w}_I - \mathbf{v}_I) = -(I - \Pi^{0,f}_{k-2})\Delta(\mathbf{w}_I - \mathbf{v}_I)$.

Now using (2.57), the equivalence for the moments up to degree $k-2$ in (2.56) and an inverse estimate in [18]:

$$\begin{aligned}|\mathbf{w}_I - \mathbf{v}_I|^2_{1,f} = -\int_f \mathbf{p}\cdot(\mathbf{w}_I - \mathbf{v}_I) - \int_f (\mathbf{w}_I - \mathbf{v}_I)\cdot \Pi^{0,f}_{k-2}\Delta(\mathbf{w}_I - \mathbf{v}_I) =\\ -\int_f \mathbf{p}\cdot(\mathbf{w}_I - \mathbf{v}_I) = -\int_f \mathbf{p}\cdot(\mathbf{w}_I - \Pi^{\nabla,f}_k \mathbf{w}_I) \leq \|\mathbf{p}\|_{0,f}\cdot\|\mathbf{w}_I - \Pi^{\nabla,f}_k\mathbf{w}_I\|_{0,f} =\\ \|I - \Pi^{0,f}_{k-2}\|\|\Delta(\mathbf{w}_I - \mathbf{v}_I)\|_{0,f}\|\mathbf{w}_I - \Pi^{\nabla,f}_k\mathbf{w}_I\|_{0,f}\\ \leq c\cdot h^{-1}|\mathbf{w}_I - \mathbf{v}_I|_{1,f}\cdot\|\mathbf{w}_I - \Pi^{\nabla,f}_k\mathbf{w}_I\|_{0,f}\end{aligned} \tag{2.58}$$

Now using a triangular inequality, the fact that $\Pi^{0,f}_k\mathbf{w}_I = \Pi^{\nabla,f}_k(\Pi^{0,f}_k\mathbf{w}_I)$ and a Poincaré estimate:

$$\begin{aligned}|\mathbf{w}_I - \mathbf{v}_I|_{1,f} =\leq c\cdot h^{-1}\|\mathbf{w}_I - \Pi^{\nabla,f}_k\mathbf{w}_I\|_{0,f} \leq c\cdot h^{-1}(\|\mathbf{w}_I - \Pi^{0,f}_k\mathbf{w}_I\|_{0,f}\\ +\|\Pi^{0,f}_k\mathbf{w}_I - \Pi^{\nabla,f}_k\mathbf{w}_I\|_{0,f}) \leq c(1+c_\Delta)\cdot h^{-1}\|\mathbf{w}_I - \Pi^{0,f}_k\mathbf{w}_I\|_{0,f} \leq\\ c_2|\mathbf{w}_I - \Pi^{0,f}_k\mathbf{w}_I|_{1,f}.\end{aligned} \tag{2.59}$$

Using a triangular inequality, by the stability of the projection $\Pi^{0,f}_k$, (2.55) and (2.54):

$$\begin{aligned}|\mathbf{w}_I - \mathbf{v}_I|_{1,f} \leq c_2(|\mathbf{w}_I - \Pi^{0,f}_k\mathbf{v}_c|_{1,f} + |\Pi^{0,f}_k\mathbf{v}_c - \Pi^{0,f}_k\mathbf{w}_I|_{1,f})\\ \leq c_2(1+c_0)|\mathbf{v}_c - \Pi^{0,f}_k\mathbf{v}_c|_{1,f}.\end{aligned} \tag{2.60}$$

We can also estimate:

$$|\mathbf{v}_c - \mathbf{v}_I|_{1,f} \leq |\mathbf{v}_c - \mathbf{w}_I|_{1,f} + |\mathbf{w}_I - \mathbf{v}_I|_{1,f} \leq c|\mathbf{v}_c - \Pi^{0,f}_k\mathbf{v}_c|_{1,f} \tag{2.61}$$

and

$$\begin{aligned}|\mathbf{v}_c - \Pi^{0,f}_k\mathbf{v}_c|_{1,f} \leq |\mathbf{v}_c - \mathbf{v}|_{1,f} + |\mathbf{v} - \Pi^{0,f}_k\mathbf{v}|_{1,f} + |\Pi^{0,f}_k\mathbf{v} - \Pi^{0,f}_k\mathbf{v}_c|_{1,f}\\ \leq (1+c_0)|\mathbf{v}_c - \mathbf{v}|_{1,f} + |\mathbf{v}|_{1,f} \leq c_3|\mathbf{v}|_{1,f}.\end{aligned} \tag{2.62}$$

Combining the estimates obtained and by triangular inequality we have:

$$|\mathbf{v} - \mathbf{v}_I|_{1,f} \leq |\mathbf{v} - \mathbf{v}_c|_{1,f} + |\mathbf{v}_c - \mathbf{v}_I|_{1,f} \leq c|\mathbf{v}|_{1,f} \tag{2.63}$$

Remembering that $\mathbf{v}_I - \mathbf{v}_c|_{\partial f} = 0$ and using a Poincaré-Friedrichs type inequality we can obtain an L^2 estimate and gain a power h.

Step 2. Interpolant on enriched element space.

$\forall f \in \partial K$ we consider $\mathbf{w}_I^f$ and $\mathbf{v}_I^f$ and let be $\mathbf{w}_I^{\partial K}$ and $\mathbf{v}_I^{\partial K}$ their gluing (continuous by construction). Again we consider the 3D Clement interpolant of $\mathbf{v}$ relative to the sub-tassellation $\widehat{\mathcal{T}}$ made by tetrahedron of $\mathcal{T}$ and let be $\mathbf{v}_\pi := \Pi_k^{0,K}\mathbf{v}_c$. Now $\Delta\mathbf{v}_\pi \in [\mathbb{P}_k(K)]^3 \implies \Delta\mathbf{v}_\pi = \nabla q_\pi + \widehat{\mathbf{g}}$ with $q_\pi \in \mathbb{P}_{k+1}(K)$ and $\widehat{\mathbf{g}} \in \mathbf{x} \wedge [\mathbb{P}_{k-1}(K)]^3$.

We now define $\mathbf{w}_I \in \widetilde{\mathbf{V}_h}$ as the solution of:

$$\begin{cases} -\Delta\mathbf{w}_I - \nabla s = \widehat{\mathbf{g}} & \text{in } K \\ \text{div } \mathbf{w}_I = \Pi_{k-1}^{0,K}(\text{div } \mathbf{v}_c) & \text{in } K \\ \mathbf{w}_I = \mathbf{v}_I^{\partial K} & \text{on } \partial K, \end{cases} \tag{2.64}$$

and $\widetilde{\mathbf{v}}$ the solution of the auxiliary problem:

$$\begin{cases} -\Delta\widetilde{\mathbf{v}} - \nabla\widetilde{s} = \widehat{\mathbf{g}} & \text{in } K \\ \text{div } \widetilde{\mathbf{v}} = \text{div } \mathbf{v}_c & \text{in } K \\ \widetilde{\mathbf{v}} = \mathbf{v}_I^{\partial K} & \text{on } \partial K, \end{cases} \tag{2.65}$$

Adding and subtracting $\Delta\mathbf{v}_\pi$ at the first equation of (2.65) and $\mathbf{v}_\pi$ at the other two:

$$\begin{cases} -\Delta(\mathbf{v}_\pi - \widetilde{\mathbf{v}}) - \nabla(-q_\pi - \widetilde{s}) = 0 & \text{in } K \\ \text{div } (\mathbf{v}_\pi - \widetilde{\mathbf{v}}) = \text{div } (\mathbf{v}_\pi - \mathbf{v}_c) & \text{in } K \\ \widetilde{\mathbf{v}} - \mathbf{v}_\pi = \mathbf{v}_I^{\partial K} - \mathbf{v}_\pi = (\mathbf{v}_I^{\partial K} - \mathbf{v}_c)|_{\partial K} + (\mathbf{v}_c - \Pi_k^{0,K}\mathbf{v}_c)|_{\partial K} & \text{on } \partial K, \end{cases} \tag{2.66}$$

we have $\forall f \in \partial K$ $(\mathbf{v}_I^{\partial K} - \mathbf{v}_c)|_{\partial f} = \mathbf{0}$. Let be P_f a regular pyramid $P_f \subset K$, by the trace theorem on $f \implies \exists\psi_f \in [H^1(P_f)]^3$, $\psi_f|_{\partial P_f \setminus f} = \mathbf{0}$ and $c > 0$ s.t. $|\psi_f|_{1,P_f} \leq c_f\|\mathbf{v}_I^f - \mathbf{v}_c\|_{1/2,f}$. Let be $\psi := \sum_{f \in \partial P} \psi_f + \mathbf{v}_c - \Pi_k^{0,E}\mathbf{v}_c$. Now:

$$\begin{aligned} |\mathbf{v}_\pi - \widetilde{\mathbf{v}}|_{1,K} \leq \inf\{|\mathbf{z}|_{1,K}, \mathbf{z} \in [H^1(K)]^3 : \text{div}\mathbf{z} = \text{div}\,(\mathbf{v}_\pi - \mathbf{v}_c) \text{ in } K, \\ \mathbf{z} = \mathbf{v}_I^{\partial K} - \mathbf{v}_\pi \text{ on } \partial K\} \leq |\psi|_{1,K} \leq \sum_{f \in \partial K} |\psi_f|_{1,P_f} + |\mathbf{v}_c - \Pi_k^{0,K}\mathbf{v}_c|_{1,K} \leq \\ c_f \sum_{f \in \partial K} \|\mathbf{v}_I^f - \mathbf{v}_c\|_{1/2,f} + |\mathbf{v}_c - \Pi_k^{0,K}\mathbf{v}_c|_{1,K} \end{aligned} \tag{2.67}$$

We estimate now the first term of the last inequality using the Gagliardo Niremberg estimate, the standard interpolation, the fact $\forall a,b\in\mathbb{R}\ ab\le h^{-1}a^2+hb^2$ and a Poincare inequality:

$$\begin{aligned}\|\mathbf{v}_I^f-\mathbf{v}_c\|_{1/2,f}\le\|\mathbf{v}_I^f-\mathbf{v}_c\|_{0,f}^2+c_{GN}\|\mathbf{v}_I^f-\mathbf{v}_c\|_{0,f}|\mathbf{v}_I^f-\mathbf{v}_c|_{1,f}\le\\ \|\mathbf{v}_I^f-\mathbf{v}_c\|_{0,f}^2+c_{GN}(h^{-1}\|\mathbf{v}_I^f-\mathbf{v}_c\|_{0,f}+h|\mathbf{v}_I^f-\mathbf{v}_c|_{1,f})\le\\ ch(1+h)|\mathbf{v}_I^f-\mathbf{v}_c|_{1,f}\le ch|\mathbf{v}_c-\Pi_k^{0,P}\mathbf{v}_c|_{1,f}^2\\ \le ch|\mathbf{v}|_{1,f}^2\le c|\mathbf{v}|_{1,K}^2,\end{aligned}\tag{2.68}$$

and we conclude $|\mathbf{v}_\pi-\widetilde{\mathbf{v}}|_{1,K}\le c|\mathbf{v}|_{1,K}$. Subtracting the system (2.65) at (2.64), we have:

$$\begin{cases}-\Delta(\widetilde{\mathbf{v}}-\mathbf{w}_I)-\nabla(\widetilde{s}-s)=0 & \text{in } K\\ \operatorname{div}(\widetilde{\mathbf{v}}-\mathbf{w}_I)=\operatorname{div}\mathbf{v}_c-\Pi_{k-1}^{0,K}\operatorname{div}\mathbf{v}_c & \text{in } K\\ \widetilde{\mathbf{v}}-\mathbf{v}_I=\mathbf{0} & \text{on } \partial K,\end{cases}\tag{2.69}$$

By the standard theory of saddle point problem [23]:

$$\begin{aligned}|\widetilde{\mathbf{v}}-\mathbf{w}_I|_{1,K}\le\frac{c(\alpha)}{\beta}\|\operatorname{div}\mathbf{v}_c-\Pi_{k-1}^{0,K}\operatorname{div}\mathbf{v}_c\|_{0,K}\le\frac{c(\alpha)}{\beta}(\|(I-\Pi_{k-1}^{0,K})\operatorname{div}\mathbf{v}\|_{0,K}\\ +\|(I-\Pi_{k-1}^{0,K})(\operatorname{div}\mathbf{v}-\operatorname{div}\mathbf{v}_c)\|_{0,K})\le\frac{c(\alpha)}{\beta}|\mathbf{v}|_{1,\widetilde{K}},\end{aligned}\tag{2.70}$$

then

$$|\mathbf{v}-\mathbf{w}_I|_{1,K}\le|\mathbf{v}-\mathbf{v}_\pi|_{1,K}+|\mathbf{v}_\pi-\widetilde{\mathbf{v}}|_{1,K}+|\widetilde{\mathbf{v}}-\mathbf{w}_I|_{1,K}\le|\mathbf{v}|_{1,K}.\tag{2.71}$$

It left to estimate $|\mathbf{v}_I-\mathbf{w}_I|_{1,K}$ with the norm of $|\mathbf{v}|_{1,K}$, and then we can conclude.
Step 3. Interpolant on the VEM space.
Let be $\mathbf{v}_I\in\mathbf{V}_h^K$ interpolant in sense of the Dofs, we have:

- $\mathbf{v}_I=\mathbf{w}_I$ on ∂K (it means that they have the same $\mathbf{D_v1},\mathbf{D_v2}$ and $\mathbf{D_v3}$);
- $\mathbf{D_v5}$ are equals ($\mathbf{v}_I-\mathbf{w}_I=\mathbf{0}$ on $\partial K\implies\int_K\operatorname{div}(\mathbf{v}_I=\mathbf{w}_I)=0$);
- $\forall\mathbf{g}\in[\mathbb{P}_{k-3}(K)]^3\qquad\int_K\mathbf{v}_I(\mathbf{x}\wedge\mathbf{g})=\int_K\mathbf{w}_I(\mathbf{x}\wedge\mathbf{g})$;
  $\forall\mathbf{g}\in[\mathbb{P}_{k-1\backslash k-3}(K)]^3\qquad\int_K\mathbf{v}_I(\mathbf{x}\wedge\mathbf{g})=\int_K\Pi_k^(\nabla,K)\mathbf{w}_I(\mathbf{x}\wedge\mathbf{g})\implies\int_K(\mathbf{v}_I-\mathbf{w}_I)(\mathbf{x}\wedge\mathbf{g})=\int_K\Pi(\Pi_k^{(\nabla,K)}\mathbf{w}_I-\mathbf{w}_I)(\mathbf{x}\wedge\mathbf{g})\qquad\forall\mathbf{g}\in[\mathbb{P}_{k-1}(K)]^3$ where Π is the L^2 projection on the space $[\mathbb{P}_{k-1\backslash k-3}(K)]^3$.

Defining $\mathbf{d}_I:=\mathbf{v}_I-\mathbf{w}_I$, we have $\Delta\mathbf{d}_I+\nabla\widetilde{s}=\widehat{\mathbf{g}}$ for some $\widetilde{s}\in L_0^2(K),\widehat{\mathbf{g}}\in[\mathbb{P}_{k-1}(K)]^3$. We consider the problem:

$$\begin{cases}-\Delta\mathbf{d}_I-\nabla\widetilde{s}=\widehat{\mathbf{g}} & \text{in } K\\ \operatorname{div}\mathbf{d}_I=0 & \text{in } K\\ \mathbf{d}_I=\mathbf{0} & \text{on } \partial K\\ \int_K\mathbf{d}_I(\mathbf{x}\wedge\mathbf{g})=\int_K\Pi(\Pi_k^{(\nabla,K)}\mathbf{w}_I-\mathbf{w}_I)(\mathbf{x}\wedge\mathbf{g}) & \forall\mathbf{g}\in[\mathbb{P}_{k-1}(K)]^3,\end{cases}\tag{2.72}$$

after providing an inf-sup condition (same technique in [15] Sec. 4), this problem is well-posed and we have the stability estimate:

$$h|\mathbf{d}_I|_{1,K} + \|\widetilde{s}\|_{0,K} + \|\mathbf{g}\|_{0,K} \leq \|\Pi(\Pi_k^{(\nabla,K)}\mathbf{w}_I - \mathbf{w}_I)\|_{0,K}, \tag{2.73}$$

and then by the stability of Π and by the triangular inequality:

$$\begin{aligned} |\mathbf{d}_I|_{1,K} \leq h^{-1}\|\Pi(\Pi_k^{(\nabla,K)}\mathbf{w}_I - \mathbf{w}_I)\|_{0,K} &\leq |\Pi_k^{(\nabla,K)}\mathbf{w}_I - \mathbf{w}_I|_{1,K} \\ &\leq 2|\mathbf{w}_I - \mathbf{v}|_{1,K} + |\Pi_k^{(\nabla,K)}\mathbf{v} - \mathbf{v}|_{1,K} \leq c|\mathbf{v}|_{1,K}. \end{aligned} \tag{2.74}$$

We conclude by triangular inequality combining (2.71) and (2.74) The $|\cdot|_{0,K}$ estimate can be recovered again using a Poincare-Friedrichs type inequality. □

Proposition 2.2.1. *Given the discrete spaces $\mathbf{V}_h$ and Q_h defined in (2.5) and (2.6) in 2D and (2.38) and (2.30) in 3D, there exists a positive $\widetilde{\beta}$, independent of h, such that:*

$$\sup_{\mathbf{v}_h \in \mathbf{V}_h \, \mathbf{v}_h \neq \mathbf{0}} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_1} \geq \widetilde{\beta}\|q_h\|_0 \qquad \textit{for all} q_h \in Q_h. \tag{2.75}$$

We have the following existence and convergence theorem (for the proof see [14], for the 2D and [12] for the 3D).

Theorem 2.2.1. *Under the Assumptions* $(\mathbf{A1}) - (\mathbf{A3})$, *let* $(\mathbf{u}, p) \in [H_0^1(\Omega)] \times L_0^2(\Omega)$ *be the solution of the problem (1.5) and* $(\mathbf{u}_h, p_h) \in \mathbf{V}_{h,0} \times Q_{h,0}$ *be the unique solution of the problem (2.50). Assuming moreover* $\mathbf{u}, \mathbf{f} \in [H^{s+1}(\Omega)]^d$ *and* $p \in H^s(\Omega)$, $0 < s \leq k$, *then:*

$$\begin{aligned} |\mathbf{u} - \mathbf{u}_h|_1 &\lesssim h^s \mathcal{F}(\mathbf{u}, \nu) + h^{s+2}\mathcal{H}(\mathbf{f}, \nu), \\ \|p - p_h\|_0 &\lesssim h^s|p|_s + h^s\mathcal{K}(\mathbf{u}, \nu) + h^{s+2}|\mathbf{f}|_{s+1}, \end{aligned} \tag{2.76}$$

for suitable functions $\mathcal{F}, \mathcal{H}, \mathcal{K}$ independent of h.

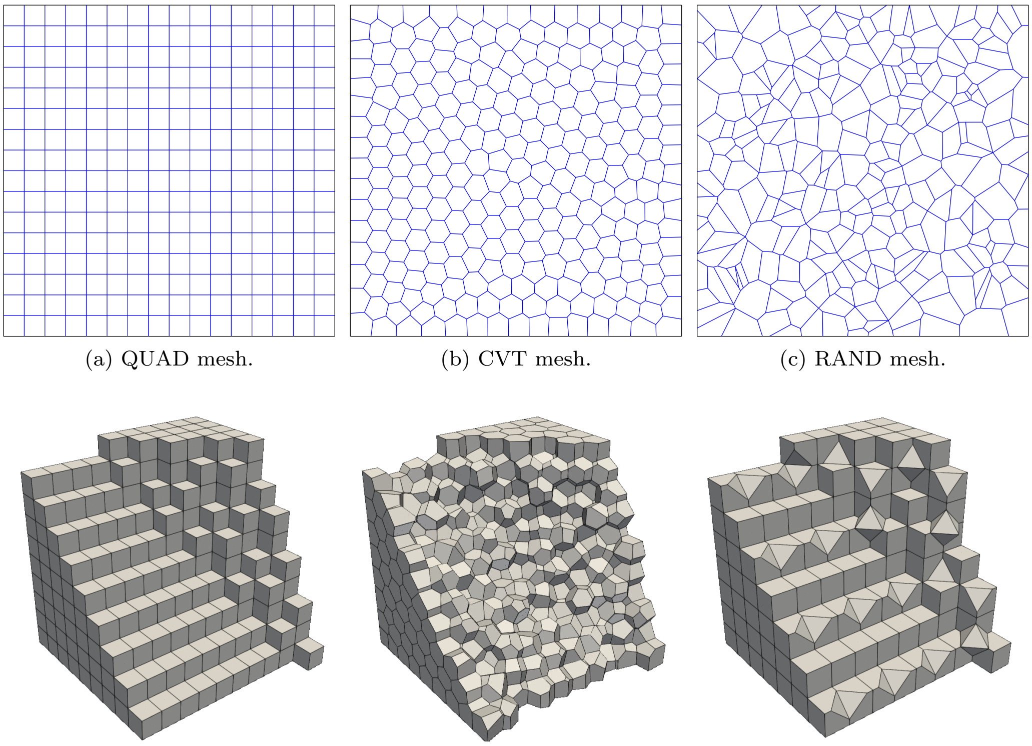
Remark 2.2.1. Since the error of the velocity in (2.76) does not depend on the pressure, one can design a reduced scheme with a smaller number of dofs as for the two-dimensional case (Section 5.3 in [12]).

We write here the discrete variational problem (2.50) in matrix form:

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix}, \tag{2.77}$$

where the matrices A and B are associated with the discrete bilinear forms $a_h(\cdot,\cdot)$ and $b_h(\cdot,\cdot)$.

Remark 2.2.2. In the next two chapters, we omit the underscore h since we will always refer to the finite-dimensional space. We also write $\widehat{\mathbf{V}} \times Q$ instead of $\widehat{\mathbf{V}}_{h,0} \times Q_{h,0}$, only for the sake of simplifying the notation.



(a) QUAD mesh.

(b) CVT mesh.

(c) RAND mesh.

(d) CUBE mesh

(e) CVT mesh

(f) OCTA mesh

Figure 2.2: Example of VEM mesh discretizations in two and three dimensions.

2.3 Numerical Error and convergence test

We will now shortly analyze the convergence error of the VEM methods we use in our work, for more details we remand to [12, 14].

Remark 2.3.1. As we will see in the next chapter, the scope of our work is focused on the iterative resolution of a saddle point problem (3.9) that arise from a reduce scheme using a Schur complement technique. We will study the parallel resolution of that problem with an iterative method without analyzing the error since it is out of the scope. For sake of completeness, we briefly recall here how to compute the error in the VEMs and we just show some results of convergence.

The VEM solution $\mathbf{u}_h$ is not explicitly known point-wise inside the elements. Therefore, even when the analytical solution $\mathbf{u}$ is available the method error is not computable. We then compute the method error comparing $\mathbf{u}$ with a suitable polynomial projection of $\mathbf{u}_h$. To do so, we exploit the polynomial projection $\mathbf{\Pi}_k^{0,K}$ previously introduced in Prop. 2.1.1 and 2.1.2 and we compute

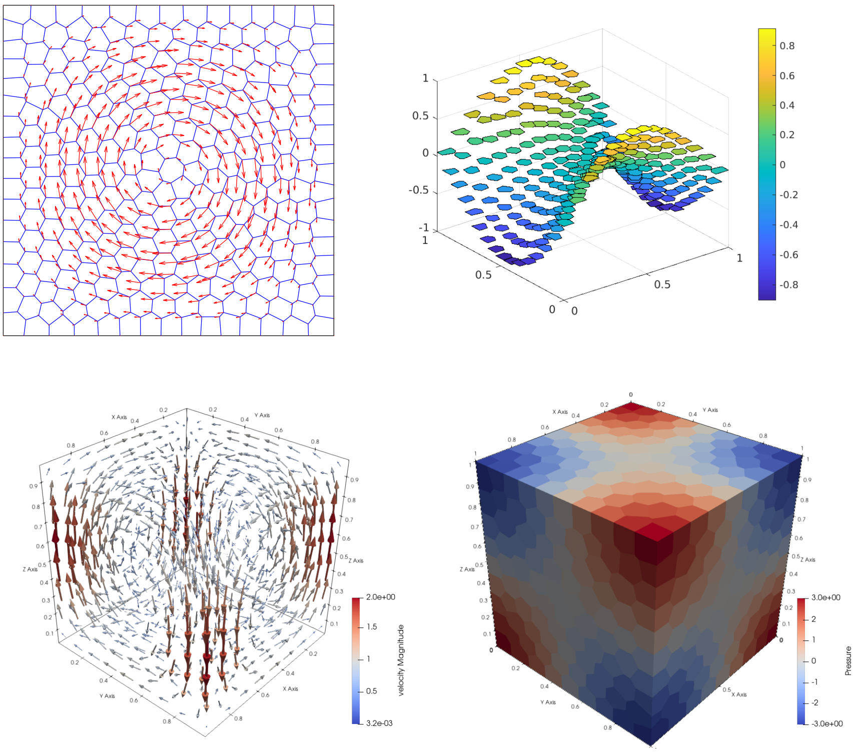


Figure 2.3: Velocity field (left) and pressure (right) discrete solutions for a CVT mesh discretization in two (up) and three (down) dimensions.

quantities:

$$\delta(\mathbf{u}) := \left(\sum_{K \in \mathcal{T}_h} \| \nabla \mathbf{u} - \mathbf{\Pi}_k^{0,K} \nabla \mathbf{u}_h \|_{0,K}^2 \right)^{1/2} \quad \text{and} \quad \delta(p) = \|p - p_h\|_0. \tag{2.78}$$

Remark 2.3.2. All the numerical test in two dimension are performed in MATLAB R2023A©, while for the three dimensional extension we use the Vem++ library [39] a new C++ library specifically written to handle Virtual Element Methods.

2D. In two dimension we solve the Stokes equations on the unit square domain $\Omega = [0,1] \times [0,1]$, applying homogeneous Dirichlet boundary conditions on the whole $\partial\Omega$ and we use two different type of mesh discretization as in Figure 2.2, with viscosity coefficient $\nu = 1$ for all the domain Ω.
We choose the load term $\mathbf{f}$ in such a way that the analytical solution is (Figure 2.3):

$$\mathbf{u}(x,y) = \begin{pmatrix} -\sin(\pi x)\sin(\pi x)\sin(2\pi y) \\ \sin(\pi y)\sin(\pi y)\sin(2\pi x) \end{pmatrix}, \quad p(x,y) = \sin(\pi x) - \sin(\pi y).$$

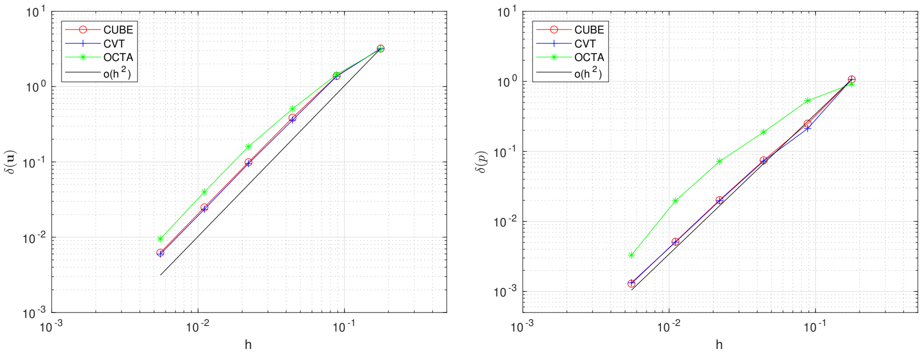


Figure 2.4: Velocity (left) and pressure (right) convergence plots for QUAD, CVT and RAND mesh in 2D for VEM discretization of degree $k = 2$.

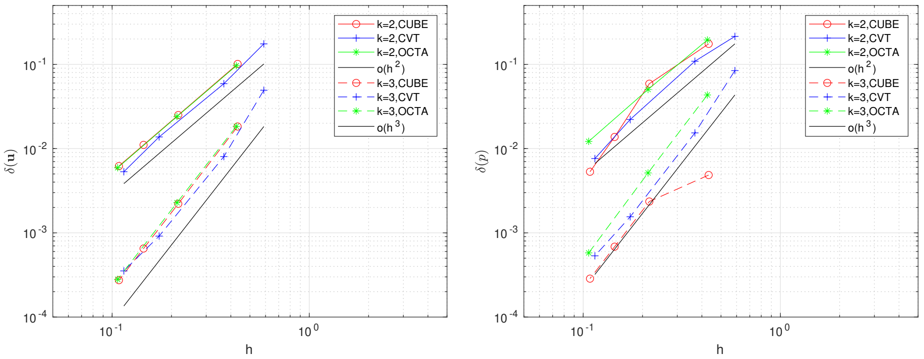


Figure 2.5: Velocity (left) and pressure (right) convergence plots for CUBE, CVT and OCTA mesh in 3D for VEM discretization of degree $k = 2, 3$.

We reported in Figure 2.4 the plot of the two errors for the 2d VEM discretization of degree $k = 2$. As we can see, the convergence rate is of the order 2 for both velocity and pressure for all the different type of mesh considered.

3D. In three dimension we solve the equations on the unit cube domain $\Omega = [0, 1] \times [0, 1]$, applying homogeneous Dirichlet boundary conditions on the whole $\partial\Omega$. In this case we use three different type of mesh discretizations as in in Figure 2.2.
We choose the load term $\mathbf{f}$ in such a way that the analytical solution is (Figure 2.3):

$$\mathbf{u}(x, y, z) = \begin{pmatrix} \sin(\pi x)\cos(\pi y)\cos(\pi z) \\ \cos(\pi x)\sin(\pi y)\cos(\pi z) \\ -2\cos(\pi x)\cos(\pi y)\sin(\pi z) \end{pmatrix}, \; p(x, y, z) = -\pi\cos(\pi x)\cos(\pi y)\cos(\pi z).$$

In Figure 2.5 we reported the plot of the two errors for the 3d VEM discretization of degree $k = 2, 3$. The convergence rate is of the order 2 for both velocity and pressure for the discretization of degree $k = 2$ and order 3 for the case $k = 3$, for all the mesh considered.

Chapter 3

BDDC Algorithm for Virtual Element methods

In this chapter we introduce the BDDC methods applied to the saddle point problem that arises from the VEM discretization of the Stokes equations described in Chapter 2. We start giving a brief introduction of the Domain Decomposition procedure and how it is adapted to the VEM context. We then recall how these BDDC methods are constructed and assembled and we analyze their convergence rate. We provide several numerical results to validate the theoretical estimates. For sake of simplicity the formulation of the method is given in three dimensions, making some observations related to the two dimensional case showing the differences when it is relevant.

3.1 Domain decomposition for VEM spaces

We decompose $\mathcal{T}_h$ into N non-overlapping subdomains Ω_i with characteristic size H_i:

$$\bar{\mathcal{T}}_h = \bigcup_{i=1}^{N} \bar{\Omega}_i, \qquad \Gamma = \bigcup_{i \neq j} \partial\Omega_i \cap \partial\Omega_j, \tag{3.1}$$

where each Ω_i is the union of different polyhedral of the tassellation $\mathcal{T}_h$ and Γ is the interface (skeleton) among the subdomains. Each local interface is defined as $\Gamma_i = \partial\Omega_i \cap \Gamma$ and it is constituted by:

- *macro faces* $\mathcal{F}$, two dimensional open sets shared by two subdomains;
- *macro edges* $\mathcal{E}$, monodimensional open sets shared by more than two subdomains;
- *vertices* $\mathcal{V}$, the end points of the *macro edges.*

We let $\mathcal{E}_H$, and $\mathcal{F}_H$ denote, respectively, the set of macro edges $\mathcal{E}$ and of macro faces $\mathcal{F}$ of the subdomain decomposition interior to Ω, and $\mathcal{F}_H^i$ and $\mathcal{E}_H^i$ denote the set of, respectively, macro

faces and macro edges of the subdomain Ω^i. We identify with $\mathcal{E}_k$ one of the $\mathcal{N}_\mathcal{E}$ total macro edges of the interface Γ, with $k = 1, ..., \mathcal{N}_\mathcal{E}$ and with $\mathcal{F}_l$ one of the $\mathcal{N}_\mathcal{F}$ total macro faces of the interface Γ, with $l = 1, ..., \mathcal{N}_\mathcal{F}$. Sometimes we will use the notation $\mathcal{E}_k(\mathcal{F}_l)$ to underline that the edge $\mathcal{E}_k$ belongs to the face $\mathcal{F}_l$. We denote with $\mathcal{V}_i$ one of the $\mathcal{N}_\mathcal{V}$ vertices.
We note that, in two dimension we keep the same notation using only edges and vertices.

We assume that the decomposition (3.1) is shape-regular in the sense of [18]:
There exist constant $\gamma^\star > 0$ and $N^\star > 0$ such that the subdomain decomposition satisfies the following properties:

- it is geometrically conforming, that is, for all $i = 1, .., N$, if a vertex, edge, or face of Ω_i is contained in $\partial\Omega_i \cap \partial\Omega_j$ with $j = 1, ..., N$ and $j \neq i$, it is also, respectively, a vertex, edge, or face of Ω_j;
- the subdomains Ω_i are shape regular of diameter H_i with constants $\gamma_{\Omega_i} > \gamma^\star$ and $N_{\Omega_i} < N_\star$;
- for all i , there exists a scalar $\nu_i > 0$ such that $\nu|_{\Omega_i} \simeq \nu_i$;
- the decomposition is quasi-uniform: there exists an H such that for all i we have $H_i \simeq H$.

We introduce now a splitting for the velocity components' DoFs. To do so, we identify each DoF with a natural geometrical position, e.g. $\mathbf{D}_V^1$, $\mathbf{D}_V^2$ are clearly located in the coordinates of their evaluation. The moments $\mathbf{D}_V^3$ are placed ideally in the center of the faces of the elements, while the moments $\mathbf{D}_V^4$ and $\mathbf{D}_V^5$ are placed into the baricenter of the elements. We then denote as:

- *boundary* DoFs, the DoFs that live on the interface Γ. The DoFs that belong to this set are the subset of $\mathbf{D}_V^1$, $\mathbf{D}_V^2$ and $\mathbf{D}_V^3$ that have a geometrical position placed on Γ;

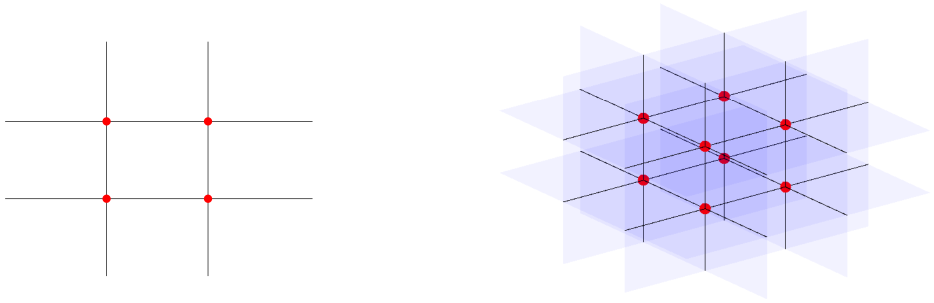


Figure 3.1: 2D and 3D interface of the subdomains (excluding $\partial\Omega$): in red the points that indicate the vertices of the subdomains, in black the macro edges and in light blue the macro faces.

- *interior* DoFs, the DoFs that live inside of a subdomain (that do not intersect Γ). All the $\mathbf{D}_V^4$ and $\mathbf{D}_V^5$ belongs to this set, plus the $\mathbf{D}_V^1$, $\mathbf{D}_V^2$ and $\mathbf{D}_V^3$ that do not live on the interface.

Remark 3.1.1. With the same concept, in two dimension, we classify the DoFs of the velocity components ($\mathbf{D_V3}$ and $\mathbf{D_V4}$) as *interior* DoFs, since they are placed in the baricenter of each element. Instead, we split the $\mathbf{D_V1}$ and $\mathbf{D_V2}$ DoFs into *boundary* and *interior*, since they are pointwise evaluation and some of them live on the interface Γ.

Following the notations introduced in [70] and [21], we decompose the discrete velocity and pressure space $\widehat{\mathbf{V}}$ and Q into:

$$\widehat{\mathbf{V}} = \mathbf{V}_I \bigoplus \widehat{\mathbf{V}}_\Gamma, \quad Q = Q_I \bigoplus Q_0, \tag{3.2}$$

with $Q_0 := \prod_{i=1}^N \{q \in \Omega_i | q \ \textit{is constant in} \ \Omega_i\}$.
$\widehat{\mathbf{V}}_\Gamma$ is the continuous space of the traces on Γ of functions in $\widehat{\mathbf{V}}$ (Figure 3.2.1 (a)). $\mathbf{V}_I$ and Q_I are direct sums of subdomain interior velocity spaces $\mathbf{V}_I^{(i)}$, and subdomain interior pressure spaces $Q_I^{(i)}$, respectively, i.e.,

$$\mathbf{V}_I = \bigoplus_{i=1}^N \mathbf{V}_I^{(i)}, \quad Q_I = \bigoplus_{i=1}^N Q_I^{(i)}. \tag{3.3}$$

The elements of $\mathbf{V}_I^{(i)}$ have support in the subdomain Ω_i and vanish on its interface Γ_i, while the elements of $Q_I^{(i)}$ are restrictions of elements in Q to Ω_i which satisfy $\int_{\Omega_i} q_I^{(i)} = 0$. We also define the space of interface velocity variables of the subdomain Ω_i by $\mathbf{V}_\Gamma^{(i)}$, and the associated product space by $\mathbf{V}_\Gamma = \prod_{i=1}^N \mathbf{V}_\Gamma^{(i)}$. Generally functions in $\mathbf{V}_\Gamma$ are discontinuous across the interface (see Figure 3.2.1 (c)).
Here we define $R_\Gamma^{(i)} : \widehat{\mathbf{V}}_\Gamma \to \mathbf{V}_\Gamma^{(i)}$, the operator which maps functions in the continuous interface velocity space $\widehat{\mathbf{V}}_\Gamma$ to their subdomain components in the space $\mathbf{V}_\Gamma^{(i)}$. We denote the direct sum of the $R_\Gamma^{(i)}$ with R_Γ.
With the decomposition of the solution space given in (3.2), the global saddle-point problem (2.77) can be written as: find $(\mathbf{u}_I, p_I, \mathbf{u}_\Gamma, p_0) \in (\mathbf{V}_I, Q_I, \widehat{\mathbf{V}}_\Gamma, Q_0)$, such that:

$$\begin{bmatrix} A_{II} & B_{II}^T & \widehat{A}_{\Gamma I}^T & 0 \\ B_{II} & 0 & \widehat{B}_{I\Gamma} & 0 \\ \widehat{A}_{\Gamma I} & \widehat{B}_{I\Gamma}^T & \widehat{A}_{\Gamma\Gamma} & \widehat{B}_{0\Gamma}^T \\ 0 & 0 & \widehat{B}_{0\Gamma}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_I \\ p_I \\ \mathbf{u}_\Gamma \\ p_0 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_I \\ 0 \\ \mathbf{f}_\Gamma \\ 0 \end{bmatrix}. \tag{3.4}$$

The blocks related to the continuous interface velocity are assembled by summing the corresponding subdomain submatrices, e.g., $\widehat{A}_{\Gamma\Gamma} = \sum_{i=1}^N {R_\Gamma^{(i)}}^T \widehat{A}_{\Gamma\Gamma}^{(i)} R_\Gamma^{(i)}$ and $\widehat{B}_{0\Gamma} = \sum_{i=1}^N \widehat{B}_{0\Gamma}^{(i)} R_\Gamma^{(i)}$. Correspondingly, the right-hand side vector $\mathbf{f}_I$ consists of subdomain vectors $\mathbf{f}_I^{(i)}$, and $\mathbf{f}_\Gamma$ is assembled from the subdomain components $\mathbf{f}_\Gamma^{(i)}$; we denote the spaces of the right-hand side vectors $\mathbf{f}_I$ and $\mathbf{f}_\Gamma$ by $\mathbf{F}_I$ and $\mathbf{F}_\Gamma$ respectively.

Remark 3.1.2. The lower left block of (3.4) is zero because the bilinear form $b(\mathbf{u}_I, q_0)$, by the divergence theorem, vanishes for any $\mathbf{v}_I \in \mathbf{V}_I$ and $q_0 \in Q_0$.

3.1.1 Schur Complement and System Reduction

By employing a symmetric permutation, the leading two by two blocks in the coefficient matrix can be rewritten as a block diagonal matrix with blocks corresponding to independent subdomain problems:

$$\begin{bmatrix} A_{II}^{(1)} & {B_{II}^{(1)}}^T & & {A_{\Gamma I}^{(1)}}^T & 0 \\ B_{II}^{(1)} & 0 & & B_{I\Gamma}^{(1)} & 0 \\ & & \ddots & \vdots & \\ A_{\Gamma I}^{(1)} & {B_{I\Gamma}^{(1)}}^T & \dots & \widehat{A}_{\Gamma\Gamma} & \widehat{B}_{0\Gamma}^T \\ 0 & 0 & & \widehat{B}_{0\Gamma} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_I^{(1)} \\ p_I^{(1)} \\ \vdots \\ \mathbf{u}_\Gamma \\ p_0 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_I^{(1)} \\ 0 \\ \vdots \\ \mathbf{f}_\Gamma \\ 0 \end{bmatrix}. \tag{3.5}$$

We proceed eliminating, by static condensation, the independent subdomain variables $(\mathbf{u}_I^{(i)}, p_I^{(i)})$ in the system (3.5). To do so, we solve independent Dirichlet problems:

$$\begin{bmatrix} A_{II}^{(i)} & {B_{II}^{(i)}}^T \\ B_{II}^{(i)} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_I^{(i)} \\ p_I^{(i)} \end{bmatrix} + \begin{bmatrix} {A_{\Gamma I}^{(i)}}^T & 0 \\ B_{I\Gamma}^{(i)} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_\Gamma \\ p_0 \end{bmatrix} = \begin{bmatrix} \mathbf{F}_I^{(i)} \\ 0 \end{bmatrix}, \tag{3.6}$$

thus

$$\begin{bmatrix} \mathbf{u}_I^{(i)} \\ p_I^{(i)} \end{bmatrix} = \begin{bmatrix} A_{II}^{(i)} & {B_{II}^{(i)}}^T \\ B_{II}^{(i)} & 0 \end{bmatrix}^{-1} \left(\begin{bmatrix} \mathbf{F}_I^{(i)} \\ 0 \end{bmatrix} - \begin{bmatrix} {A_{\Gamma I}^{(i)}}^T & 0 \\ B_{I\Gamma}^{(i)} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_\Gamma \\ p_0 \end{bmatrix} \right). \tag{3.7}$$

Then, substituting the solutions of (3.7) in

$$\sum_{i=1}^{N} \begin{bmatrix} A_{\Gamma I}^{(i)} & {B_{I\Gamma}^{(i)}}^T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_I^{(i)} \\ p_I^{(i)} \end{bmatrix} + \begin{bmatrix} \widehat{A}_{\Gamma\Gamma} & \widehat{B}_{0\Gamma}^T \\ \widehat{B}_{0\Gamma} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_\Gamma \\ p_0 \end{bmatrix} = \begin{bmatrix} \mathbf{F}_\Gamma \\ 0 \end{bmatrix} \tag{3.8}$$

we obtain the global interface saddle point problem:

$$\widehat{S}\,\widehat{u} = \begin{bmatrix} \widehat{S}_\Gamma & \widehat{B}_{0\Gamma}^T \\ \widehat{B}_{0\Gamma} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_\Gamma \\ p_0 \end{bmatrix} = \begin{bmatrix} \mathbf{g}_\Gamma \\ 0 \end{bmatrix} = \widehat{\mathbf{g}}, \tag{3.9}$$

where the right-hand side $\widehat{\mathbf{g}} \in \mathbf{F}_\Gamma \times F_0$ is given by

$$\widehat{\mathbf{g}} = \sum_{i=1}^{N} {R_\Gamma^{(i)}}^T \left\{ \begin{bmatrix} \mathbf{f}_\Gamma^{(i)} \\ 0 \end{bmatrix} - \begin{bmatrix} A_{\Gamma I}^{(i)} & {B_{I\Gamma}^{(i)}}^T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} A_{II}^{(i)} & {B_{II}^{(i)}}^T \\ B_{II}^{(i)} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f}_I^{(i)} \\ 0 \end{bmatrix} \right\}, \tag{3.10}$$

$\widehat{S}_\Gamma$ is assembled from the subdomain Stokes Schur complements $S_\Gamma^{(i)}$, which are defined by: given $\mathbf{w}_\Gamma^{(i)} \in \mathbf{V}_\Gamma^{(i)}$, determine $S_\Gamma^{(i)}\mathbf{w}_\Gamma^{(i)} \in \mathbf{F}_\Gamma^{(i)}$ such that

$$\begin{bmatrix} A_{II}^{(i)} & B_{II}^{(i)^T} & A_{\Gamma I}^{(i)^T} \\ B_{II}^{(i)} & 0 & B_{I\Gamma}^{(i)} \\ A_{\Gamma I}^{(i)} & B_{I\Gamma}^{(i)^T} & A_{\Gamma\Gamma}^{(i)} \end{bmatrix} \begin{bmatrix} \mathbf{w}_I^{(i)} \\ q_I^{(i)} \\ \mathbf{w}_\Gamma^{(i)} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \\ S_\Gamma^{(i)}\mathbf{w}_\Gamma^{(i)} \end{bmatrix}, \tag{3.11}$$

These Schur complements are symmetric and positive definite. By the coercivity of $a(\cdot,\cdot)$, we know that the matrices

$$\begin{bmatrix} A_{II}^{(i)} & A_{\Gamma I}^{(i)^T} \\ A_{\Gamma I}^{(i)} & A_{\Gamma\Gamma}^{(i)} \end{bmatrix}$$

are symmetric and positive definite and so the left two by two upper block of the left-hand-side of (3.11) has the same number of negative eigenvalues of the all matrix. The left-hand-side matrices of (3.11) are congruent to:

$$\begin{bmatrix} A_{II}^{(i)} & B_{II}^{(i)^T} & 0 \\ B_{II}^{(i)} & 0 & 0 \\ 0 & 0 & S_\Gamma^{(i)} \end{bmatrix}$$

and so, the positive definiteness of (3.11) follows by the Sylvester's law of inertia.
We define by S_Γ the direct sum of the $S_\Gamma^{(i)}$, then $\widehat{S}_\Gamma$ is given by

$$\widehat{S}_\Gamma = R_\Gamma^T S_\Gamma R_\Gamma = \sum_{i=1}^{N} R_\Gamma^{(i)^T} S_\Gamma^{(i)} R_\Gamma^{(i)}, \tag{3.12}$$

and then we set

$$R = \begin{bmatrix} R_\Gamma & 0 \\ 0 & I \end{bmatrix}, \quad R^{(i)} = \begin{bmatrix} R_\Gamma^{(i)} & 0 \\ 0 & I \end{bmatrix}. \tag{3.13}$$

We see from (3.11) that the action of $S_\Gamma^{(i)}$ on a vector can be evaluated by solving a Dirichlet problem on the subdomain Ω_i as in (3.6). These problems defined on the subdomains are independent and since only the action of $\widehat{S}_\Gamma$ on a vector is required we do not need to assemble all the matrices.
We see that $\widehat{S}$ is symmetric and indefinite, so one can use the generalized minimal residual method to solve the problem (3.9), with usually a positive definite block preconditioner. The BDDC preconditioner, that we will introduce in the next Section, makes the operator of the preconditioned problem (3.9) symmetric and positive definite, so the preconditioned conjugate gradient (CG) method can be used to accelerate the solution. However, as we will see in the following, this property is achieved only under particular conditions and only on proper defined subspaces of $\widehat{\mathbf{V}} \times Q_0$.

Remark 3.1.3. We just make note that once we have solved the problem (3.9), we can use direct solvers on the independent local problems (3.7) on each subdomain and then assemble the global solution. These systems are, indeed, significantly smaller and they can be solved in parallel. For this reason the convergence rate theory and our numerical simulation are only focused on the global saddle point problem (3.9).

3.2 Construction of the BDDC preconditioner

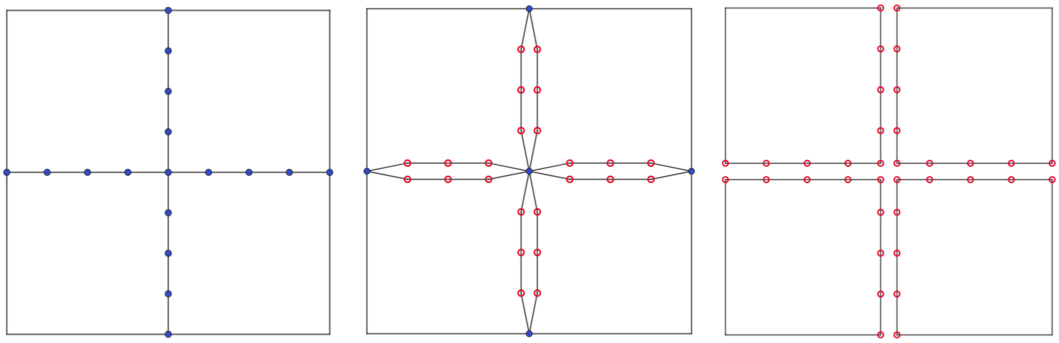
Following the standard BDDC framework for the FEM in [70] we mainly need two ingredients to handle this type of algorithms.
Before entering into the definition of the function space used to construct the BDDC preconditioner, we briefly justify the choice of our notation. The subscript Γ indicates DoFs living on the interface, Π and Δ are instead used to distinguish DoFs of Γ that belong to the primal and dual spaces, respectively, defined here below. Two other subscripts are used: C indicates an operator referred to the coarse space and D is instead used to highlight that an operator has been rescaled by suitable scaling functions, defined later. The hat $\widehat{\cdot}$ refers to a continuous space, the $\widetilde{\cdot}$ means that the space is continuous on primal interface DoFs and discontinuous on the dual ones and finally no hat is used for the product of local spaces, which is discontinuous at all interface DoFs.

3.2.1 The partially assembled space

The first ingredient is a partially assembled interface velocity space, namely $\widetilde{\mathbf{V}}_\Gamma$ (see Figure 3.2.1 (b)):

$$\widetilde{\mathbf{V}}_\Gamma = \widehat{\mathbf{V}}_\Pi \bigoplus \mathbf{V}_\Delta = \widehat{\mathbf{V}}_\Pi \bigoplus \Big(\prod_{i=1}^{N} \mathbf{V}_\Delta^{(i)}\Big). \tag{3.14}$$

Here, $\widehat{\mathbf{V}}_\Pi$ is the continuous coarse-level primal velocity space typically spanned by subdomain vertex nodal basis functions and/or interface edge or face basis with constant values, or with values of weight functions. We will always assume that the basis has been changed so that each primal basis function corresponds to an explicit degree of freedom (see A). In other words, we will have explicit primal unknowns corresponding to the primal continuity constraints on edges or faces. The primal DoFs are shared by neighboring subdomains. The complementary space $\mathbf{V}_\Delta$ is the direct sum of the subdomain dual interface velocity spaces $\mathbf{V}_\Delta^{(i)}$, which correspond to the remaining interface velocity DoFs and are spanned by basis functions which vanish at the primal DoFs. Thus, an element in the space $\widetilde{\mathbf{V}}_\Gamma$ has a continuous primal velocity and typically a discontinuous dual velocity component. We now introduce several restriction, extension, and scaling operators between a variety of spaces. As in [70], $R_\Gamma^{(i)}$ is the operator which maps a function in the space $\widehat{\mathbf{V}}_\Gamma$ to its component in $\mathbf{V}_\Gamma^{(i)}$. We define $R_\Delta^{(i)}$ as the operator which maps the space $\widehat{\mathbf{V}}_\Gamma$ to its dual component in the space $\mathbf{V}_\Delta^{(i)}$. $R_{\Gamma\Pi}$ is the restriction operator from the space $\widehat{\mathbf{V}}_\Gamma$ to its subspace $\widehat{\mathbf{V}}_\Pi$; $R_\Pi^{(i)}$ is the operator which maps $\widehat{\mathbf{V}}_\Pi$ into its Γ_i-component. $\widetilde{R}_\Gamma$ is



(a) $\widehat{\mathbf{V}}_\Gamma$ (b) $\widetilde{\mathbf{V}}_\Gamma := \widehat{\mathbf{V}}_\Pi \bigoplus \mathbf{V}_\Delta$ (c) $\mathbf{V}_\Gamma$

Figure 3.2: Representation of the three velocity interface spaces.

the direct sum of $R_{\Gamma\Pi}$ and the $R_\Delta^{(i)}$, and it is a map from $\widehat{\mathbf{V}}_\Gamma$ into $\widetilde{\mathbf{V}}_\Gamma$. Finally, $\overline{R}_\Gamma$ is the operator that maps the space $\widetilde{\mathbf{V}}_\Gamma$ into the product space $\mathbf{V}_\Gamma$ associated with the set of subdomains.

The relationships among the previous spaces and operators are summarized in the following diagram:

$$\begin{tikzcd}
\mathbf{V}_\Gamma & \widetilde{\mathbf{V}}_\Gamma \arrow[l, "\overline{R}_\Gamma"'] & \\
\mathbf{V}_\Gamma^{(i)} & \widehat{\mathbf{V}}_\Gamma \arrow[l, "R_\Gamma^{(i)}"'] \arrow[u, "\widetilde{R}_\Gamma"'] \arrow[r, "R_\Delta^{(i)}"] \arrow[d, "R_{\Pi\Gamma}"] & \mathbf{V}_\Delta^{(i)} \\
 & \widehat{\mathbf{V}}_\Pi \arrow[r, "R_\Pi^{(i)}"] & \widehat{\mathbf{V}}_\Pi^{(i)}
\end{tikzcd}$$

Remark 3.2.1.

3.2.2 Scaling and average operators

The second main tool in this algorithm are the scaling operators. They are obtained by multiplying the previous ones with suitable matrix $D^{(i)}$ defined on each subdomain. The matrices $D^{(i)}$ can be chosen in different ways and can be either diagonal or not, but they always must provide a partition of unity, i.e.:

$$\widetilde{R}_{D,\Gamma}^T \widetilde{R}_\Gamma = \widetilde{R}_\Gamma^T \widetilde{R}_{D,\Gamma} = I, \tag{3.15}$$

where $\widetilde{R}_{D,\Gamma}$ is a map from $\widehat{\mathbf{V}}_\Gamma$ to $\widetilde{\mathbf{V}}_\Gamma$ and it is obtained as the direct sum of $R_{\Gamma,\Pi}$ and the $R_{D,\Delta}^{(i)} := D^{(i)} R_\Delta^{(i)}$. In this thesis we use diagonal matrices $D^{(i)}$ that define two different scaling namely *multiplicity-scaling* and *ν-scaling*, and a non-diagonal one namely *deluxe-scaling.*

Diagonal scalings. In order to construct these matrices, we introduce a positive scaling factor $\delta_i^\dagger(x)$ for the nodes on the interface Γ_i of each subdomain Ω_i. For the *multiplicity* scaling, we simply define the $\delta_i^\dagger(x)$ as the pseudoinverse counting functions, so:

$$\delta_i^\dagger(x) := 1/\mathcal{N}_x, \quad x \in \Gamma_i \tag{3.16}$$

where $\mathcal{N}_x = card(I_x)$ is the number of the subdomains which have x on their boundaries and I_x is the set of indices of these subdomains. This mean that a node that lives on a macro edge in two dimension or a macro face in three it is shared by two subdomains, while a node that lives on a macro edge in three dimension it is shared by more than three subdomains. The scaled restriction operators $R_{D,\Delta}^{(i)}$ are obtained by simply multiplying each non-zero element of $R_\Delta^{(i)}$, only one for row, by the corresponding scaling factor $\delta_i^\dagger(x)$. The ν-scaling it is obtained by using $\widehat{\delta}_i^\dagger(x)$, a weighted version of the scaling factor $\delta_i^\dagger(x)$. For some $\gamma \in [1/2, \infty]$, we define:

$$\widehat{\delta}_i^\dagger(x) := \frac{\nu_i^\gamma(x)}{\sum_{j\in\mathcal{N}_x} \nu_j^\gamma(x)}, \quad x \in \Gamma_i, \tag{3.17}$$

the scaled restriction operators are obtained as before.

Remark 3.2.2. Since when $\nu_i(x) = 1$ for all the subdomains the two scalings are equivalent, in two dimension we prove the theoretical result by applying the ν-scaling. In three dimensions, for sake of simplicity, we limit to prove the results for the multiplicity scaling.

Deluxe scaling. We take a subdomain Ω_k and for a geometrical entity $\mathcal{O}$, that can be an edge or a face shared by two or more subdomains, we consider the principal minors of the subdomain matrices $S_T^{(k)}$:

$$S_{\mathcal{O}\mathcal{O}}^{(k)} := R_{\mathcal{O}}^{(k)} S_T^{(k)} {R_{\mathcal{O}}^{(k)}}^T, \tag{3.18}$$

where $R_{\mathcal{O}}^{(k)}$ maps $\mathbf{V}_\Gamma^{(k)}$ to the DoFs located on O. Then we split the matrices as follows

$$S_{\mathcal{O}\mathcal{O}}^{(k)} = \begin{bmatrix} S_{\mathcal{O}'\mathcal{O}'}^{(k)} & S_{\mathcal{O}'\mathcal{O}_\Delta}^{(k)} \\ {S_{\mathcal{O}'\mathcal{O}_\Delta}^{(k)}}^T & S_{\mathcal{O}_\Delta\mathcal{O}_\Delta}^{(k)} \end{bmatrix}, \tag{3.19}$$

where $\mathcal{O}_\Delta$ is the dual set of the DoFs associated to the entity $\mathcal{O}$ and $\mathcal{O}' := \Gamma_i \setminus \mathcal{O}_\Delta$. We introduce the Schur complements:

$$\widetilde{S}_{\mathcal{O}_\Delta\mathcal{O}_\Delta}^{(k)} = S_{\mathcal{O}_\Delta\mathcal{O}_\Delta}^{(k)} - {S_{\mathcal{O}'\mathcal{O}_\Delta}^{(k)}}^T {S_{\mathcal{O}'\mathcal{O}'}^{(k)}}^{-1} S_{\mathcal{O}'\mathcal{O}_\Delta}^{(k)}. \tag{3.20}$$

Identifying with $\mathcal{L}$ the subset of indices that share the entity $\mathcal{O}$ we define the following scaling matrices by:

$$D_{\mathcal{O}}^{(k)} := \left(\sum_{i\in\mathcal{L}} \widetilde{S}_{\mathcal{O}_\Delta\mathcal{O}_\Delta}^{(k)}\right)^{-1} \widetilde{S}_{\mathcal{O}_\Delta\mathcal{O}_\Delta}^{(k)} \tag{3.21}$$

and the subdomain (deluxe) scaling matrices by:

$$D^{(k)} := \sum_{\mathcal{O}\subset\Gamma_i} R_{\mathcal{O}}^{(k),T} D_{\mathcal{O}}^{(k)} R_{\mathcal{O}}^{(k)} \tag{3.22}$$

Remark 3.2.3. It is easy to see that all the scaling matrices defined previously are a partition of unity, in the sense:

$$\sum_{i\in\mathcal{L}} D_{\mathcal{O}}^{(i)} = I, \tag{3.23}$$

where I is the identity matrix.

Average operator. In order to analyze the convergence rate of the BDDC method, we define an average operator $E_D = \widetilde{R}\widetilde{R}_D^T$, which maps $\widetilde{\mathbf{V}}_\Gamma \times Q_0$, with generally discontinuous interface velocities, to elements with continuous interface velocities in the same space. For any $\mathbf{v} = (\mathbf{v}_\Gamma, q_0) \in \widetilde{\mathbf{V}}_{\Gamma,B} \times Q_0$,

$$E_D = \begin{bmatrix} \mathbf{v}_\Gamma \\ q_0 \end{bmatrix} = \begin{bmatrix} \widetilde{R}_\Gamma & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \widetilde{R}_{D,\Gamma} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{v}_\Gamma \\ q_0 \end{bmatrix} = \begin{bmatrix} E_{D,\Gamma}\mathbf{v}_\Gamma \\ q_0 \end{bmatrix} \tag{3.24}$$

where $E_D = \widetilde{R}\widetilde{R}_{D,\Gamma}^T$, provides the average of the interface velocities across the interface Γ. Recalling that we can split $\mathbf{v} = \mathbf{v}_\Pi \oplus \mathbf{v}_\Delta$, we have $E_D\mathbf{v} = \mathbf{v}_\Pi \oplus E_{D,\Delta}\mathbf{v}_\Delta$, where $E_{D,\Delta}\mathbf{v}_\Delta$ is the dual part of the averaged vector. Sometimes, especially in the FETI-DP framework, it is convenient work on $P_D = I - E_D$, the complementary projector operator of E_D.

3.2.3 Schur complement and BDDC preconditioner

The interface velocity Schur complement $\widetilde{S}_\Gamma$ is defined on the partially assembled interface velocity space $\widetilde{\mathbf{V}}_\Gamma$ by: given $\mathbf{v}_\Gamma \in \widetilde{\mathbf{V}}_\Gamma$, $\widetilde{S}_\Gamma\mathbf{v}_\Gamma \in \widetilde{\mathbf{F}}_\Gamma$ satisfies:

$$\begin{bmatrix} A_{II}^{(i)} & {B_{II}^{(i)}}^T & {A_{\Delta I}^{(i)}}^T & & {\widetilde{A}_{\Pi I}^{(i)}}^T \\ B_{II}^{(i)} & 0 & B_{I\Delta}^{(i)} & & \widetilde{B}_{I\Pi}^{(i)} \\ A_{\Delta I}^{(i)} & {B_{I\Delta}^{(i)}}^T & A_{\Delta\Delta}^{(i)} & & {\widetilde{A}_{\Pi\Delta}^{(i)}}^T \\ & & & \ddots & \vdots \\ \widetilde{A}_{\Pi I}^{(i)} & {\widetilde{B}_{I\Pi}^{(i)}}^T & \widetilde{A}_{\Pi\Delta}^{(i)} & \dots & \widetilde{A}_{\Pi\Pi} \end{bmatrix} \begin{bmatrix} \mathbf{v}_I^{(i)} \\ p_I^{(i)} \\ \mathbf{v}_\Delta^{(i)} \\ \vdots \\ \mathbf{v}_\Pi^{(i)} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \\ (\widetilde{S}_\Gamma\mathbf{v}_\Gamma)_\Delta^{(i)} \\ \vdots \\ (\widetilde{S}_\Gamma\mathbf{v}_\Gamma)_\Pi^{(i)} \end{bmatrix} \tag{3.25}$$

Here $\widetilde{A}_{\Pi\Pi} = \sum_{i=1}^N {R_\Pi^{(i)}}^T A_{\Pi\Pi}^{(i)} R_\Pi^{(i)}$, $\widetilde{A}_{\Pi I}^{(i)} = {R_\Pi^{(i)}}^T A_{\Pi I}^{(i)}$, $\widetilde{A}_{\Pi\Delta}^{(i)} = {R_\Pi^{(i)}}^T A_{\Pi\Delta}^{(i)}$ and $\widetilde{B}_{I\Pi}^{(i)} = B_{I\Pi}^{(i)} R_\Pi^{(i)}$.

$\widetilde{S}_\Gamma$ can than be obtained from the Schur complements $S_\Gamma^{(i)}$ by assembling only the primal interface velocity part, i.e. as

$$\widetilde{S}_\Gamma = {\overline{R}_\Gamma}^T S_\Gamma \overline{R}_\Gamma. \tag{3.26}$$

As we saw before (3.11) the global interface Schur operator $\widehat{S}_\Gamma$ is obtained by fully assembling

the $S_\Gamma^{(i)}$ across the subdomain interface, therefore it can be also obtained from $\widetilde{S}_\Gamma$ by further assembling the dual interface velocity part, $\widehat{S}_\Gamma = \widetilde{R}_\Gamma^T \widetilde{S}_\Gamma \widetilde{R}_\Gamma$. So we need to define an operator $\widetilde{B}_{0\Gamma}$, which maps the partially assembled interface velocity space $\widetilde{V}_\Gamma$ into F_0, the space of right hand sides corresponding to Q_0, and it is obtained from $\widetilde{B}_{0\Gamma}$ by assembling the dual interface velocity part on the subdomain interfaces, i.e. $\widehat{B}_\Gamma = \widetilde{B}_\Gamma \widetilde{R}_\Gamma$.
Introducing

$$\widetilde{R} = \begin{bmatrix} \widetilde{R}_\Gamma & 0 \\ 0 & I \end{bmatrix}, \quad \widetilde{S} = \begin{bmatrix} \widetilde{S}_\Gamma & \widetilde{B}_{0\Gamma}^T \\ \widetilde{B}_{0\Gamma} & 0 \end{bmatrix}, \tag{3.27}$$

we can write $\widehat{S}$, the operator of the global interface problem (3.9), as

$$\widehat{S} = \begin{bmatrix} \widehat{S}_\Gamma & \widehat{B}_{0\Gamma}^T \\ \widehat{B}_{0\Gamma} & 0 \end{bmatrix} = \begin{bmatrix} \widetilde{R}_\Gamma^T \widetilde{S}_\Gamma \widetilde{R}_\Gamma & \widetilde{R}_\Gamma^T \widetilde{B}_{0\Gamma}^T \\ \widetilde{B}_{0\Gamma} \widetilde{R}_\Gamma & 0 \end{bmatrix} = \widetilde{R}^T \widetilde{S} \widetilde{R}. \tag{3.28}$$

The preconditioner for solving the global saddle-point problem (3.9) is

$$M^{-1} = \widetilde{R}_D^T \widetilde{S}^{-1} \widetilde{R}_D, \tag{3.29}$$

where we have defined

$$\widetilde{R}_D := \begin{bmatrix} \widetilde{R}_{D,\Gamma} & 0 \\ 0 & I \end{bmatrix}, \tag{3.30}$$

and so we have the BDDC preconditioned problem: find $(\mathbf{u}_\Gamma, p_0) \in \widehat{\mathbf{V}}_\Gamma \times Q_0$, such that

$$\widetilde{R}_D^T \widetilde{S}^{-1} \widetilde{R}_D \widehat{S} \begin{bmatrix} \mathbf{u}_\Gamma \\ p_0 \end{bmatrix} = \widetilde{R}_D^T \widetilde{S}^{-1} \widetilde{R}_D \begin{bmatrix} \mathbf{g}_\Gamma \\ 0 \end{bmatrix}. \tag{3.31}$$

What we need in our implementation is to determine the action $\widetilde{S}^{-1}\mathbf{q}$ for any given $\mathbf{q} = (\mathbf{q}_\Gamma, q_0) \in \widetilde{\mathbf{F}}_\Gamma \times F_0$, so we have to solve the linear system

$$\begin{bmatrix} \widetilde{S}_\Gamma & \widetilde{B}_{0\Gamma}^T \\ \widetilde{B}_{0\Gamma} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_\Gamma \\ p_0 \end{bmatrix} = \begin{bmatrix} \mathbf{q}_\Gamma \\ q_0 \end{bmatrix}. \tag{3.32}$$

Given the definition of $\widetilde{S}_\Gamma$ in (5.3), we have that solving (5.10) is equivalent to solve

$$\left[\begin{array}{cccccc} A_{II}^{(i)} & B_{II}^{(i)^T} & A_{\Delta I}^{(i)^T} & & \widetilde{A}_{\Pi I}^{(i)^T} & 0 \\ B_{II}^{(i)} & 0 & B_{I\Delta}^{(i)} & & \widetilde{B}_{I\Pi}^{(i)} & 0 \\ A_{\Delta I}^{(i)} & B_{I\Delta}^{(i)^T} & A_{\Delta\Delta}^{(i)} & & \widetilde{A}_{\Pi\Delta}^{(i)^T} & B_{0\Delta}^{(i)^T} \\ & & & \ddots & \vdots & \\ \widetilde{A}_{\Pi I}^{(i)} & \widetilde{B}_{I\Pi}^{(i)^T} & \widetilde{A}_{\Pi\Delta}^{(i)} & \dots & \widetilde{A}_{\Pi\Pi} & \widetilde{B}_{0\Pi}^T \\ 0 & 0 & B_{0\Delta}^{(i)} & & \widetilde{B}_{0\Pi} & 0 \end{array}\right] \left[\begin{array}{c} \mathbf{u}_I^{(i)} \\ p_I^{(i)} \\ \mathbf{u}_\Delta^{(i)} \\ \vdots \\ \mathbf{u}_\Pi \\ p_0 \end{array}\right] = \left[\begin{array}{c} \mathbf{0} \\ 0 \\ \mathbf{q}_\Delta^{(i)} \\ \vdots \\ \mathbf{q}_\Pi \\ q_0 \end{array}\right] \tag{3.33}$$

where $\widetilde{B}_{0\Pi} = \sum_{i=1}^N B_{0\Pi}^{(i)} R_\Pi^{(i)}$. Now using a block factorization we obtain

$$\widetilde{S}^{-1} = \sum_{i=1}^N \left[\begin{array}{ccc} 0 & 0 & R_{\Delta,i}^T \end{array}\right] \left[\begin{array}{ccc} A_{II}^{(i)} & B_{II}^{(i)^T} & A_{\Delta I}^{(i)^T} \\ B_{II}^{(i)} & 0 & B_{I\Delta}^{(i)} \\ A_{\Delta I}^{(i)} & B_{I\Delta}^{(i)^T} & A_{\Delta\Delta}^{(i)} \end{array}\right]^{-1} \left[\begin{array}{c} 0 \\ 0 \\ R_{\Delta,i} \end{array}\right] + \Phi S_{CC}^{-1} \Phi^T, \tag{3.34}$$

where $R_{\Delta,i}$ maps $\widetilde{\mathbf{F}}_\Gamma \times F_0$ into $\mathbf{F}_\Delta^{(i)}$, the set of right hand sides corresponding to $\mathbf{V}_\Delta^{(i)}$. The matrix S_{CC}, relatively to the primal constraints, has to be completely assembled in this way

$$\begin{aligned} S_{CC} = \sum_{i=1}^N R_C^{(i)^T} \Bigg\{ & \left[\begin{array}{cc} A_{\Pi\Pi}^{(i)} & B_{0\Pi}^{(i)^T} \\ B_{0\Pi}^{(i)} & 0 \end{array}\right] - \left[\begin{array}{ccc} A_{\Pi I}^{(i)} & B_{I\Pi}^{(i)^T} & A_{\Pi\Delta}^{(i)} \\ 0 & 0 & B_{0\Delta}^{(i)} \end{array}\right] \\ & \left[\begin{array}{ccc} A_{II}^{(i)} & B_{II}^{(i)^T} & A_{\Delta I}^{(i)^T} \\ B_{II}^{(i)} & 0 & B_{I\Delta}^{(i)} \\ A_{\Delta I}^{(i)} & B_{I\Delta}^{(i)^T} & A_{\Delta\Delta}^{(i)} \end{array}\right]^{-1} \left[\begin{array}{cc} A_{\Pi I}^{(i)^T} & 0 \\ B_{I\Pi}^{(i)} & 0 \\ A_{\Pi\Delta}^{(i)^T} & B_{0\Delta}^{(i)^T} \end{array}\right] \Bigg\} R_C^{(i)}, \end{aligned} \tag{3.35}$$

where we have defined

$$R_C^{(i)} := \left[\begin{array}{cc} R_\Pi^{(i)} & 0 \\ 0 & I \end{array}\right], \tag{3.36}$$

the maps from $\widehat{\mathbf{V}}_\Pi \times Q_0$ to $\mathbf{V}_\Pi^{(i)} \times Q_0$. Finally we define the matrix

$$\Phi = R_{\Pi 0}^T - \sum_{i=1}^N \left[\begin{array}{ccc} 0 & 0 & R_{\Delta,i}^T \end{array}\right] \left[\begin{array}{ccc} A_{II}^{(i)} & B_{II}^{(i)^T} & A_{\Delta I}^{(i)^T} \\ B_{II}^{(i)} & 0 & B_{I\Delta}^{(i)} \\ A_{\Delta I}^{(i)} & B_{I\Delta}^{(i)^T} & A_{\Delta\Delta}^{(i)} \end{array}\right]^{-1} \left[\begin{array}{cc} A_{\Pi I}^{(i)^T} & 0 \\ B_{I\Pi}^{(i)} & 0 \\ A_{\Pi\Delta}^{(i)^T} & B_{0\Delta}^{(i)^T} \end{array}\right] R_C^{(i)}, \tag{3.37}$$

where $R_{\Pi 0}$ is the map between the space $\widetilde{\mathbf{F}}_\Gamma \times F_0$ and $\widehat{\mathbf{F}}_\Pi \times F_0$.

Remark 3.2.4. As we said before the S_{CC} matrix has to be explicitly assembled. Although it is typically dense, in general the dimension of this matrix it is quite small and the inversion is computed by direct methods. However, in some contexts, when enriching the coarse space using adaptive technique or simply when the coarse space is too large it could be convenient exploit a multilevel formulation of the BDDC preconditioner. The idea consists in considering a subdomain at a finer level as an element of a coarser mesh and applying again the BDDC technology. We do not enter into the details of this and we remand to [92, 93] where it has been first presented and to [75] for the a multilevel extension. Several works have been done in this direction, including saddle point problem and adaptive techniques [54, 76, 83, 88, 89, 94, 101].

3.3 Theoretical analysis of the BDDC preconditioner

We now present an estimate for the eigenvalues of the preconditioned operator $M^{-1}\widehat{S}$, following the theory developed in [70] and adapting it to our VEM formulation. We recall all the definitions and the lemmas that we need, providing a proof only for the mainly ones that present some differences. All the proof of the lemmas that are omitted here can be recovered in [70].
We organized this section as follows. Firstly, we introduce the subspaces and we gave the Assumptions under which the preconditioned saddle point problem is symmetric and positive definite, since the argument is independent on the dimension we also give the convergence rate result. Then, separately for two and three dimensions, we gave the recipes to guarantee that these assumptions are fulfilled.

3.3.1 Benign spaces and convergence rate estimate

In the following, we denote by $a^{(i)}$, $a_h^{(i)}$ and $b^{(i)}$ the restrictions to subdomain Ω_i of the bilinear forms a, a_h and b, respectively. We introduce the $|.|_{S_\Gamma^{(i)}}$ and $|.|_{S_\Gamma}$ seminorms defined by

$$|\mathbf{v}_\Gamma^{(i)}|^2_{S_\Gamma^{(i)}} = \mathbf{v}_\Gamma^{(i)\,T} S_\Gamma^{(i)} \mathbf{v}_\Gamma^{(i)}, \quad |\mathbf{v}_\Gamma|^2_{S_\Gamma} = \mathbf{v}_\Gamma{}^T S_\Gamma \mathbf{v}_\Gamma = \sum_{i=1}^{N} |\mathbf{v}_\Gamma^{(i)}|^2_{S_\Gamma^{(i)}}, \tag{3.38}$$

and we give a norm equivalence result:

Lemma 3.3.1. *There exist positive constant c_1 and c_2, independent of H, h and the shape of the subdomains, such that*

$$c_1\tilde{\beta}^2|\mathbf{v}_\Gamma|^2_{S_\Gamma} \leq |\mathbf{v}_\Gamma|^2_{1/2,\Gamma} \leq c_2|\mathbf{v}_\Gamma|^2_{S_\Gamma} \quad \forall \mathbf{v}_\Gamma \in \mathbf{V}_\Gamma,$$

where $\tilde{\beta}$ is the stability constant inf-sup defined in (2.2.1).

Proof. This proof follows substantially the result presented in Bramble and Pasciak ([25] Theorem 4.1) where a proof for FEM is provided. Given $\mathbf{v}_\Gamma \in \mathbf{V}_\Gamma$, we define the operators $T : \mathbf{V}_\Gamma \to \widehat{\mathbf{V}}$

and $S : \mathbf{V}_\Gamma \rightarrow Q$ that satisfy $\forall i = 1, ..., N$:

$$\begin{aligned}
&(1) \quad S(\mathbf{v}_\Gamma)|_{\Omega_i} \in Q_I^{(i)}, \\
&(2) \quad T(\mathbf{v}_\Gamma)|_\Gamma = \mathbf{v}_\Gamma, \\
&(3) \quad a_h^{(i)}(T(\mathbf{v}_\Gamma), \mathbf{v}) + b^{(i)}(\mathbf{v}, S(\mathbf{v}_\Gamma) = \mathbf{0} \quad \text{for all} \quad \mathbf{v} \in \mathbf{V}_I^{(i)}, \\
&(4) \quad b^{(i)}(T(\mathbf{v}_\Gamma), q) = 0 \quad \text{for all} \quad q \in Q_I^{(i)}.
\end{aligned} \tag{3.39}$$

The above condition uniquely defines S and T. Now given $\mathbf{v}_\Gamma \in \mathbf{V}_\Gamma$, let $\mathbf{v}_\Gamma^{\mathcal{H}} \in \widehat{\mathbf{V}}$ be the discrete harmonic extension of $\mathbf{v}_\Gamma$, i.e. the unique function in $\widehat{\mathbf{V}}$ which equals $\mathbf{v}_\Gamma$ on Γ and satisfies $\forall i = 1, ..., N$:

$$a^{(i)}(\mathbf{v}_\Gamma^{\mathcal{H}}, \mathbf{v}) = 0 \quad \text{for all} \quad \mathbf{v} \in \mathbf{V}_I^{(i)}. \tag{3.40}$$

By the stability of the discrete harmonic extension and the stability of the discrete bilinear form a_h [14], we have on each subdomain:

$$a_h^{(i)}(\mathbf{v}_\Gamma^{\mathcal{H}}, \mathbf{v}_\Gamma^{\mathcal{H}}) \leq c_3 a^{(i)}(\mathbf{v}_\Gamma^{\mathcal{H}}, \mathbf{v}_\Gamma^{\mathcal{H}}) \leq c_3 |\mathbf{v}_\Gamma^{(i)}|^2_{1/2, \Gamma_i}, \tag{3.41}$$

where c_3 is a positive constant independent of h, H and the number of subdomains N. Now, by definition of S and T, and since $b^{(i)}(T(\mathbf{v}_\Gamma), S(\mathbf{v}_\Gamma)) = 0$, we have:

$$a_h^{(i)}(T(\mathbf{v}_\Gamma), T(\mathbf{v}_\Gamma)) = a_h^{(i)}(T(\mathbf{v}_\Gamma), \mathbf{v}_\Gamma^{\mathcal{H}}) + b^{(i)}(\mathbf{v}_\Gamma^{\mathcal{H}}, S(\mathbf{v}_\Gamma)). \tag{3.42}$$

Applying (2.2.1) on the subdomains, we have, for some $c > 0$:

$$\begin{aligned}
||S(\mathbf{v}_\Gamma)||^2_{Q_i} &\leq \tilde{\beta}^{-2} \sup_{\mathbf{w} \in \mathbf{V}_I^{(i)}} \frac{b^{(i)}(\mathbf{w}, S(\mathbf{v}_\Gamma))^2}{\|\mathbf{w}\|_1^2} \leq c\tilde{\beta}^{-2} \sup_{\mathbf{w} \in \mathbf{V}_I^{(i)}} \frac{b^{(i)}(\mathbf{w}, S(\mathbf{v}_\Gamma))^2}{a_h^{(i)}(\mathbf{w}, \mathbf{w})} \\
&= c\tilde{\beta}^{-2} \sup_{\mathbf{w} \in \mathbf{V}_I^{(i)}} \frac{a_h^{(i)}(T(\mathbf{v}_\Gamma), \mathbf{w})^2}{a_h^{(i)}(\mathbf{w}, \mathbf{w})} \leq c\tilde{\beta}^{-2} a_h^{(i)}(T(\mathbf{v}_\Gamma), T(\mathbf{v}_\Gamma)),
\end{aligned} \tag{3.43}$$

Applying Cauchy-Schwarz to the first term in (3.42) and using (3.43), we have:

$$\begin{aligned}
|\mathbf{v}_\Gamma^{(i)}|^2_{S_\Gamma^{(i)}} = a_h^{(i)}(T(\mathbf{v}_\Gamma), T(\mathbf{v}_\Gamma)) &\leq a_h^{(i)}(T(\mathbf{v}_\Gamma), T(\mathbf{v}_\Gamma))^{1/2} a_h^{(i)}(\mathbf{v}_\Gamma^{\mathcal{H}}, \mathbf{v}_\Gamma^{\mathcal{H}})^{1/2} \\
+ c|\mathbf{v}_\Gamma^{\mathcal{H}}|_{H^1(\Omega_i)} ||S(\mathbf{v}_\Gamma)||_{Q_i} &\leq a_h(T(\mathbf{v}_\Gamma), T(\mathbf{v}_\Gamma))^{1/2} a_h(\mathbf{v}_\Gamma^{\mathcal{H}}, \mathbf{v}_\Gamma^{\mathcal{H}})^{1/2} + \\
&c\tilde{\beta}^{-1} a_h(\mathbf{v}_\Gamma^{\mathcal{H}}, \mathbf{v}_\Gamma^{\mathcal{H}})^{1/2} a_h^{(i)}(T(\mathbf{v}_\Gamma), T(\mathbf{v}_\Gamma))^{1/2}
\end{aligned}$$

and then:

$$c\tilde{\beta}^2 |\mathbf{v}_\Gamma^{(i)}|^2_{S_\Gamma^{(i)}} = c\tilde{\beta}^2 a_h^{(i)}(T(\mathbf{v}_\Gamma), T(\mathbf{v}_\Gamma)) \leq a_h^{(i)}(\mathbf{v}_\Gamma^{\mathcal{H}}, \mathbf{v}_\Gamma^{\mathcal{H}}). \tag{3.44}$$

Finally, from (3.41) and summing on the subdomains

$$c\tilde{\beta}^2|\mathbf{v}_\Gamma|^2_{S_\Gamma} = c\tilde{\beta}^2 a_h(T(\mathbf{v}_\Gamma), T(\mathbf{v}_\Gamma)) \le a_h(\mathbf{v}_\Gamma^{\mathcal{H}}, \mathbf{v}_\Gamma^{\mathcal{H}}) \le c_3|\mathbf{v}_\Gamma|^2_{1/2,\Gamma},$$

which yields the first inequality of the thesis with $c_1 = c/c_3$.

For the second inequality we have, by definition of the discrete harmonic extension and again the stability of the discrete bilinear form a_h:

$$|\mathbf{v}_\Gamma|^2_{1/2,\Gamma_i} \le a^{(i)}(\mathbf{v}_\Gamma^{\mathcal{H}}, \mathbf{v}_\Gamma^{\mathcal{H}}) \le a^{(i)}(T(\mathbf{v}_\Gamma), T(\mathbf{v}_\Gamma)) \le c_2 a_h^{(i)}(T(\mathbf{v}_\Gamma), T(\mathbf{v}_\Gamma)) \tag{3.45}$$

and then:

$$|\mathbf{v}_\Gamma|^2_{1/2,\Gamma} \le c_2|\mathbf{v}_\Gamma|^2_{S_\Gamma}. \tag{3.46}$$

□

The operators $\widehat{S}_\Gamma$ and $\widetilde{S}_\Gamma$, given in (3.12) and (3.26), are both symmetric and positive definite, because of the Dirichlet boundary conditions on $\partial\Omega$ and provided that sufficiently many primal constraints are chosen. We can then define the $\widehat{S}_\Gamma$ and $\widetilde{S}_\Gamma$ norms on the spaces $\widehat{V}_\Gamma$ and $\widetilde{V}_\Gamma$ by

$$\begin{aligned}
\|\mathbf{v}_\Gamma\|^2_{\widehat{S}_\Gamma} &= \mathbf{v}_\Gamma^T R_\Gamma^T S_\Gamma R_\Gamma \mathbf{v}_\Gamma = |R_\Gamma \mathbf{v}_\Gamma|^2_{S_\Gamma} \quad \forall \mathbf{v}_\Gamma \in \widehat{\mathbf{V}}_\Gamma,\\
\|\mathbf{v}_\Gamma\|^2_{\widetilde{S}_\Gamma} &= \mathbf{v}_\Gamma^T \overline{R}_\Gamma^T S_\Gamma \overline{R}_\Gamma \mathbf{v}_\Gamma = |\overline{R}_\Gamma \mathbf{v}_\Gamma|^2_{S_\Gamma} \quad \forall \mathbf{v}_\Gamma \in \widetilde{\mathbf{V}}_\Gamma.
\end{aligned}$$

We then define the two spaces, whose utility is that, restricted to such spaces, the interface problem operators $\widehat{S}$ of (3.9) and $\widetilde{S}$ of (3.32) are positive semi-definite. As in [70], we give the following:

Definition 3.3.1. Given the discrete spaces $\widehat{\mathbf{V}}_\Gamma$ and $\widetilde{\mathbf{V}}_\Gamma$, we define the two subspaces

$$\begin{aligned}
\widehat{\mathbf{V}}_{\Gamma,B} &= \{\mathbf{v}_\Gamma \in \widehat{\mathbf{V}}_\Gamma | \widehat{B}_{0\Gamma}\mathbf{v}_\Gamma = 0\},\\
\widetilde{\mathbf{V}}_{\Gamma,B} &= \{\mathbf{v}_\Gamma \in \widetilde{\mathbf{V}}_\Gamma | \widetilde{B}_{0\Gamma}\mathbf{v}_\Gamma = 0\}.
\end{aligned}$$

We call $\widehat{\mathbf{V}}_{\Gamma,B} \times Q_0$ and $\widetilde{\mathbf{V}}_{\Gamma,B} \times Q_0$ the *benign subspaces* of $\widehat{\mathbf{V}}_\Gamma \times Q_0$ and $\widetilde{\mathbf{V}}_\Gamma \times Q_0$.

Lemma 3.3.2. *The interface operator $\widehat{S}$ of (3.9), restricted to the subspace $\widehat{\mathbf{V}}_{\Gamma,B} \times Q_0$ is positive semi-definite. The same is true for $\widetilde{S}$ of (3.26) restricted to $\widetilde{\mathbf{V}}_{\Gamma,B} \times Q_0$.*

We define the $\widehat{S}$ and $\widetilde{S}$ seminorms on the benign subspaces

$$\begin{aligned}
|\mathbf{v}|^2_{\widehat{S}} &= \mathbf{v}^T \widehat{S}\mathbf{v} = \|\mathbf{v}_\Gamma\|_{\widehat{S}_\Gamma} \quad \forall \mathbf{v} = (\mathbf{v}_\Gamma, q_0) \in \widehat{\mathbf{V}}_{\Gamma,B} \times Q_0,\\
|\mathbf{v}|^2_{\widetilde{S}} &= \mathbf{v}^T \widetilde{S}\mathbf{v} = \|\mathbf{v}_\Gamma\|_{\widetilde{S}_\Gamma} \quad \forall \mathbf{v} = (\mathbf{v}_\Gamma, q_0) \in \widetilde{\mathbf{V}}_{\Gamma,B} \times Q_0.
\end{aligned}$$

We make note that for we did not explicitly explain yet how to choose the primal and dual DoFs. This has to be done, as in the FEM case (see [70]), in such a way that the following two Assumptions (or conditions) are satisfied.

Assumption 1. For any $\mathbf{v}_\Delta \in \mathbf{V}_\Delta$, $\int_{\partial\Omega_i} \mathbf{v}_\Delta^{(i)} \cdot \mathbf{n} = 0$ and $\int_{\partial\Omega_i} (E_{D,\Delta}\mathbf{v}_\Delta)^{(i)} \cdot \mathbf{n} = 0$, where $\mathbf{n}$ is the outward normal of $\partial\Omega_i$. We can equivalently write $B_{0\Delta}^{(i)}\mathbf{v}_\Delta^{(i)} = 0$ and $B_{0\Delta}^{(i)}(E_{D,\Delta}\mathbf{v}_\Delta)^{(i)} = 0$.

Assumption 2. There exists a positive constant C, which is independent of H, h and the number of subdomains, but that can depend on the degree of the VEM discretization k, such that

$$|\bar{R}_\Gamma(E_{D,\Gamma}\mathbf{v}_\Gamma)|_{1/2,\Gamma} \le C\left(1+\log\left(\frac{H}{h}\right)\right)|\bar{R}_\Gamma\mathbf{v}_\Gamma|_{1/2,\Gamma}, \quad \forall \mathbf{v}_\Gamma \in \mathbf{V}_\Gamma.$$

The first assumption, also commonly known as *no-net-flux* condition, is needed to ensure that the iterates of the preconditioned conjugate gradient method stay in the benign space during the resolution of the saddle point problem(Lemma 3.3.3). The second one instead ensure the stability of the average operator and it is necessary to guarantee the convergence of the method. These assumptions will be both proven later by properly constructing the BDDC primal space in 2D and 3D in Sec 3.3.3 and 3.3.4, respectively.
With these two assumptions, we have the following results:

Lemma 3.3.3. *Let Assumption 1 hold. Then* $\widetilde{R}_D^T\mathbf{v} \in \widehat{\mathbf{V}}_{\Gamma,B} \times Q_0$*, for any* $\mathbf{v} \in \widetilde{\mathbf{V}}_{\Gamma,B} \times Q_0$.

Lemma 3.3.4. *Let Assumptions 1 and 2 hold. Then there exists a positive constant C, which is independent of H, h and the number of subdomains, such that*

$$|E_D\mathbf{v}|_{\widetilde{S}} \le C\frac{1}{\tilde{\beta}}\left(1+\log\left(\frac{H}{h}\right)\right)|\mathbf{v}|_{\widetilde{S}}, \quad \forall \mathbf{v} = (\mathbf{v}_\Gamma, q_0) \in \widetilde{\mathbf{V}}_{\Gamma,B} \times Q_0,$$

where $\tilde{\beta}$ is the inf-sup stability constant of (2.2.1).

Proof. Given any $\mathbf{v} = (\mathbf{v}_\Gamma, q_0) \in \widetilde{\mathbf{V}}_{\Gamma,B} \times Q_0$, we know, from Lemma 3.3.3, that $\widetilde{R}_D^T\mathbf{v} \in \widehat{\mathbf{V}}_{\Gamma,B} \times Q_0$. Therefore, $E_D\mathbf{v} = \widetilde{R}\widetilde{R}_D^T\mathbf{v} \in \widetilde{\mathbf{V}}_{\Gamma,B} \times Q_0$. We have from the definition of the $\widetilde{S}$-seminorm, that

$$|E_D\mathbf{v}|_{\widetilde{S}}^2 = \|E_{D,\Gamma}\mathbf{v}_\Gamma\|_{\widetilde{S}_\Gamma}^2 = |\bar{R}_\Gamma(E_{D,\Gamma}\mathbf{v}_\Gamma)|_{S_\Gamma}^2 \le C\frac{1}{\tilde{\beta}^2}|\bar{R}_\Gamma(E_{D,\Gamma}\mathbf{v}_\Gamma)|_{1/2,\Gamma}^2, \tag{3.47}$$

where the last inequality follows from Lemma 3.3.1. We have, from Assumption 2 and Lemma 3.3.1

$$\begin{aligned} |\bar{R}_\Gamma(E_{D,\Gamma}\mathbf{v}_\Gamma)|_{1/2,\Gamma}^2 &\le C\left(1+\log\left(\frac{H}{h}\right)\right)^2|\bar{R}_\Gamma\mathbf{v}_\Gamma|_{1/2,\Gamma}^2 \\ \le C\left(1+\log\left(\frac{H}{h}\right)\right)^2|\bar{R}_\Gamma\mathbf{v}_\Gamma|_{S_\Gamma}^2 &\le C\left(1+\log\left(\frac{H}{h}\right)\right)^2\|\mathbf{v}_\Gamma\|_{\widetilde{S}_\Gamma}^2. \end{aligned} \tag{3.48}$$

Consequently we have

$$|E_D\mathbf{v}|_{\widetilde{S}}^2 \le C\frac{1}{\tilde{\beta}^2}\left(1+\log\left(\frac{H}{h}\right)\right)^2\|\mathbf{v}_\Gamma\|_{\widetilde{S}_\Gamma}^2 = C\frac{1}{\tilde{\beta}^2}\left(1+\log\left(\frac{H}{h}\right)\right)^2|\mathbf{v}|_{\widetilde{S}}^2. \tag{3.49}$$

□

We have the following lemma (proof in [70]):

Lemma 3.3.5. *Any vector of the form* $\mathbf{u} = (\mathbf{0}, p_0) \in \widehat{\mathbf{V}}_{\Gamma,B} \times Q_0$ *is an eigenvector of the preconditioner operator* $M^{-1}\widehat{S}$ *with eigenvalue equal to 1.*

Theorem 3.3.1. *Let Assumptions 1 and 2 hold. The preconditioned operator* $M^{-1}\widehat{S}$ *is then symmetric, positive definite with respect to the bilinear form* $\langle \cdot, \cdot \rangle_{\widehat{S}}$ *on the benign space* $\widehat{\mathbf{V}}_{\Gamma,B} \times Q_0$. *Its minimum eigenvalue is 1 and its maximum eigenvalue is bounded by*

$$C\frac{1}{\tilde{\beta}^2}\left(1 + \log\left(\frac{H}{h}\right)\right)^2. \tag{3.50}$$

Here, C *is a constant which is independent of* H, h *and the number of subdomains, and* $\tilde{\beta}$ *is the inf-sup stability constant defined in* (2.2.1).

Proof. We know from Lemma 3.3.5, that any vector of the form $\mathbf{u} = (\mathbf{0}, p_0) \in \widehat{\mathbf{V}}_{\Gamma,B} \times Q_0$ is an eigenvector of the preconditioned operator $M^{-1}\widehat{S}$ with an eigenvalue equal to 1. It is sufficient to find lower and upper bounds of the quotient $\left\langle M^{-1}\widehat{S}\mathbf{u}, \mathbf{u} \right\rangle_{\widehat{S}} / \langle \mathbf{u}, \mathbf{u} \rangle_{\widehat{S}}$, for any $\mathbf{u} = (\mathbf{u}_\Gamma, p_0) \in \widehat{\mathbf{V}}_{\Gamma,B} \times Q_0$, where $\mathbf{u}_\Gamma$ is non zero and therefore $\langle \mathbf{u}, \mathbf{u} \rangle_{\widehat{S}} > 0$.
Lower bound: Given $\mathbf{u} \in \widehat{\mathbf{V}}_{\Gamma,B} \times Q_0$, let

$$\mathbf{v} = \widetilde{S}^{-1}\widetilde{R}_D\widetilde{S}\mathbf{u} \in \widetilde{\mathbf{V}}_{\Gamma,B} \times Q_0. \tag{3.51}$$

We have from the fact that $\widetilde{R}^T\widetilde{R}_D = \widetilde{R}_D^T\widetilde{R} = I$,

$$\langle \mathbf{u}, \mathbf{u} \rangle_{\widehat{S}} = \mathbf{u}^T\widehat{S}\widetilde{R}_D^T\widetilde{R}\mathbf{u} = \mathbf{u}^T\widehat{S}\widetilde{R}_D^T\widetilde{S}^{-1}\widetilde{S}\widetilde{R}\mathbf{u} = \langle \mathbf{v}, \widetilde{R}\mathbf{u} \rangle_{\widetilde{S}}. \tag{3.52}$$

From the Cauchy-Schwartz inequality and the fact that $\widehat{S} = \widetilde{R}^T\widetilde{S}\widetilde{R}$, we find that

$$\langle \mathbf{v}, \widetilde{R}\mathbf{u} \rangle_{\widetilde{S}} \leq \langle \mathbf{v}, \mathbf{v} \rangle_{\widetilde{S}}^{1/2} \langle \widetilde{R}\mathbf{u}, \widetilde{R}\mathbf{u} \rangle_{\widetilde{S}}^{1/2} = \langle \mathbf{v}, \mathbf{v} \rangle_{\widetilde{S}}^{1/2} \langle \mathbf{u}, \mathbf{u} \rangle_{\widetilde{S}}^{1/2}. \tag{3.53}$$

Therefore from (3.52) and (3.53),

$$\langle \mathbf{u}, \mathbf{u} \rangle_{\widetilde{S}} \leq \langle \mathbf{v}, \mathbf{v} \rangle_{\widetilde{S}}. \tag{3.54}$$

Since,

$$\langle \mathbf{v}, \mathbf{v} \rangle_{\widetilde{S}} = \mathbf{u}^T\widehat{S}\widetilde{R}_D^T\widetilde{S}^{-1}\widetilde{S}\widetilde{S}^{-1}\widetilde{R}_D\widehat{S}\mathbf{u} = \left\langle \mathbf{u}, \widetilde{R}_D^T\widetilde{S}^{-1}\widetilde{R}_D\widehat{S}\mathbf{u} \right\rangle_{\widehat{S}} = \left\langle \mathbf{u}, M^{-1}\widehat{S}\mathbf{u} \right\rangle_{\widehat{S}}, \tag{3.55}$$

we obtain, from equations (3.54) and (3.55), that $\langle \mathbf{u}, \mathbf{u} \rangle_{\widetilde{S}} \leq \left\langle \mathbf{u}, M^{-1}\widehat{S}\mathbf{u} \right\rangle_{\widehat{S}}$, which gives a lower bound of 1 for the eigenvalues. Then from Lemma 3.3.5, we know that 1 is the minimum eigenvalue of the preconditioned operator.
Upper bound: Given $\mathbf{u} \in \widehat{\mathbf{V}}_{\Gamma,B} \times Q_0$, take $\mathbf{v} \in \widetilde{\mathbf{V}}_{\Gamma,B} \times Q_0$ as in (3.51). We have, $\widetilde{R}_D^T\mathbf{v} = M^{-1}\widehat{S}\mathbf{u}$.

Since $\widehat{S} = \widetilde{R}^T \widetilde{S} \widetilde{R}$ and by using Lemma 3.3.4, we have

$$\begin{aligned}\langle M^{-1}\widehat{S}\mathbf{u}, M^{-1}\widehat{S}\mathbf{u}\rangle_{\widehat{S}} &= \langle \widetilde{R}_D^T\mathbf{v}, \widetilde{R}_D^T\mathbf{v}\rangle_{\widehat{S}} = \langle \widetilde{R}\widetilde{R}_D^T\mathbf{v}, \widetilde{R}\widetilde{R}_D^T\mathbf{v}\rangle_{\widetilde{S}} \\ &= |E_D\mathbf{v}|^2_{\widetilde{S}} \leq C^2 \frac{1}{\widetilde{\beta}^2}\left(1+\log\left(\frac{H}{h}\right)\right)^2 |\mathbf{v}|^2_{\widetilde{S}}.\end{aligned} \tag{3.56}$$

Therefore from equation (3.55), we have

$$\langle M^{-1}\widehat{S}\mathbf{u}, M^{-1}\widehat{S}\mathbf{u}\rangle_{\widehat{S}} \leq C^2 \frac{1}{\widetilde{\beta}^2}\left(1+\log\left(\frac{H}{h}\right)\right)^2 \langle \mathbf{u}, M^{-1}\widehat{S}\mathbf{u}\rangle_{\widehat{S}}. \tag{3.57}$$

Using the Cauchy-Schwarz inequality and equation (3.57), we have

$$\begin{aligned}\langle \mathbf{u}, M^{-1}\widehat{S}\mathbf{u}\rangle_{\widehat{S}} &\leq \langle M^{-1}\widehat{S}\mathbf{u}, M^{-1}\widehat{S}\mathbf{u}\rangle_{\widehat{S}}^{1/2} \langle \mathbf{u},\mathbf{u}\rangle_{\widehat{S}}^{1/2} \\ &\leq C\frac{1}{\widetilde{\beta}}\left(1+\log\left(\frac{H}{h}\right)\right)\langle \mathbf{u},\mathbf{u}\rangle_{\widehat{S}}^{1/2}\langle \mathbf{u}, M^{-1}\widehat{S}\mathbf{u}\rangle_{\widehat{S}}^{1/2}.\end{aligned} \tag{3.58}$$

This gives

$$\langle \mathbf{u}, M^{-1}\widehat{S}\mathbf{u}\rangle_{\widehat{S}} \leq C\frac{1}{\widetilde{\beta}^2}\left(1+\log\left(\frac{H}{h}\right)\right)^2 \langle \mathbf{u},\mathbf{u}\rangle_{\widehat{S}}, \tag{3.59}$$

and the upper bound of the theorem. □

3.3.2 Auxiliary lemmas

We recall here some edge and face estimates that we will need to prove the two assumptions. For each macro edge $\mathcal{E} = \Gamma_i \cap \Gamma_j \in \mathcal{E}_h$, we define $\mathbf{V}_\Gamma^{\mathcal{E}} := \mathbf{V}_\Gamma^{(i)}|_{\mathcal{E}} = \mathbf{V}_\Gamma^{(j)}|_{\mathcal{E}}$, that is the restriction of the VEM interface velocity space on the macro edge $\mathcal{E}$. We make note that this restricted space coincides with the one dimension finite element space of order k on the grid induced on $\mathcal{E}$ by the tassellation $\mathcal{T}_h$ which is quasi uniform of mesh size h, in view of the mesh assumptions in 1.2.1. In particular it satisfies standard and inverse inequalities: for $d = 2, 3$ and for some $c > 0$ for all r, s with $0 < r \leq k+1, 0 \leq s < min\{3/2, r\}, \mathbf{v} \in [H^r(\mathcal{E})]^d$:

$$\inf_{\mathbf{w}_h \in \mathbf{V}_\Gamma^{\mathcal{E}}} \|\mathbf{v} - \mathbf{w}_h\|_{[H^s(\mathcal{E})]^d} \leq c\left(\frac{h}{H}\right)^{r-s} |\mathbf{v}|_{[H^r(\mathcal{E})]^d}, \tag{3.60}$$

and for all s, r with $0 \leq s \leq r < 3/2$, $\mathbf{v}_h \in \mathbf{V}_\Gamma^{\mathcal{E}}$:

$$\|\mathbf{v}_h\|_{[H^r(\mathcal{E})]^d} \leq c\left(\frac{h}{H}\right)^{s-r} \|\mathbf{v}_h\|_{[H^s(\mathcal{E})]^d}. \tag{3.61}$$

Now, for $\mathbf{v}_\Gamma \in \mathbf{V}_\Gamma$ we consider the splitting:

$$\mathbf{v}_\Gamma = \mathbf{v}_\Pi + \mathbf{v}_\Delta, \tag{3.62}$$

and for $\mathbf{v}_\Gamma^\mathcal{E} \in \mathbf{V}_\Gamma^\mathcal{E}$:

$$\mathbf{v}_\Gamma^\mathcal{E} = \mathbf{v}_\Pi^\mathcal{E} + \mathbf{v}_\Delta^\mathcal{E}, \tag{3.63}$$

where $\mathbf{v}_\Pi^\mathcal{E}$ and $\mathbf{v}_\Delta^\mathcal{E}$ are the restriction to the edge E of the primal and dual DoFs, with the obvious definition of the spaces $\mathbf{V}_\Pi^\mathcal{E}$ and $\mathbf{V}_\Delta^\mathcal{E}$. To simplify the notation, from now, we will write $A \lesssim B$ to say that the quantity A is bounded from above by cB, with a constant c that is independent of H,h and the number of subdomains. We also denote with T, P and D respectively the set of indices of the DoFs living on Γ and the set of indices of primal DoFs and dual DoFs, the subscript $\mathcal{E}$ denotes the restriction to that edge. The following lemma gives us an estimate for an interface velocity function in two dimension.

Lemma 3.3.6. *For all $\mathbf{v}_\Gamma^\mathcal{E} \in \mathbf{V}_\Gamma^\mathcal{E}$, $\mathcal{E} \in \mathcal{E}_H$, we have:*

$$\|\mathbf{v}_\Delta^\mathcal{E}\|_{[H_{00}^{1/2}(\mathcal{E})]^2} \lesssim (1 + \log(H/h))|\mathbf{v}_\Gamma^\mathcal{E}|_{[H^{1/2}(\mathcal{E})]^2} + \sqrt{(1 + \log(H/h))}\|\mathbf{v}_\Gamma^\mathcal{E}\|_{[L^\infty(\mathcal{E})]^2}.$$

Proof. We show the proof for the VEM method of order $k = 2$.
We let $\pi_\Gamma : [L^2(\mathcal{E})]^2 \to \mathbf{V}_\Gamma^\mathcal{E}$ and $\pi_\Gamma^0 : [L^2(\mathcal{E})]^2 \to \mathbf{V}_\Delta^E$ denote the L^2-projection onto respectively $\mathbf{V}_\Gamma^\mathcal{E}$ and $\mathbf{V}_\Delta^\mathcal{E}$. Moreover, let $i_\Gamma^0 : \mathbf{V}_\Gamma^\mathcal{E} \to \mathbf{V}_\Delta^\mathcal{E}$ the restriction operator that maps the restriction to the DoFs edge into the duals one. We have that for $\mathbf{v}_\Gamma^\mathcal{E} \in \mathbf{V}_\Gamma^\mathcal{E}$:

$$\|i_\Gamma^0 \mathbf{v}_\Gamma^\mathcal{E}\|_{[L^2(\mathcal{E})]^2} \lesssim h \sum_{j \in D_\mathcal{E}} |\mathbf{D}_V^j(\mathbf{v}_\Gamma^\mathcal{E})|^2 \lesssim h \sum_{j \in T_\mathcal{E}} |\mathbf{D}_V^j(\mathbf{v}_\Gamma^\mathcal{E})|^2 \lesssim \|\mathbf{v}_\Gamma^E\|_{[L^2(\mathcal{E})]^2}.$$

We consider now $\pi_\Gamma^1 = i_\Gamma^0 \circ \pi_\Gamma : [L^2(\mathcal{E})]^2 \to \mathbf{V}_\Delta^\mathcal{E}$; we see that it is L^2-bounded: for all $\mathbf{v} \in [L^2(\mathcal{E})]^2$

$$\|\pi_\Gamma^1 \mathbf{v}\|_{[L^2(\mathcal{E})]^2} \lesssim \|\pi_\Gamma \mathbf{v}\|_{[L^2(\mathcal{E})]^2} \lesssim \|\mathbf{v}\|_{[L^2(\mathcal{E})]^2}.$$

On the other hand, observing that $i_\Gamma^0 \circ \pi_\Gamma^0 = \pi_\Gamma^0$, we have that, for $\mathbf{v} \in [H_0^1(\mathcal{E})]^2$

$$\begin{aligned}
\|\mathbf{v} - \pi_\Gamma^1 \mathbf{v}\|_{[L^2(\mathcal{E})]^2} &\lesssim \|\mathbf{v} - \pi_\Gamma^0 \mathbf{v} + i_\Gamma^0 \pi_\Gamma^0 \mathbf{v} - i_\Gamma^0 \pi_\Gamma \mathbf{v}\|_{[L^2(\mathcal{E})]^2} \\
&\lesssim \|\mathbf{v} - \pi_\Gamma^0 \mathbf{v}\|_{[L^2(\mathcal{E})]^2} + \|\pi_\Gamma^0 \mathbf{v} - \pi_\Gamma \mathbf{v}\|_{[L^2(\mathcal{E})]^2} \\
&= \|\mathbf{v} - \pi_\Gamma^0 \mathbf{v}\|_{[L^2(\mathcal{E})]^2} + \|\pi_\Gamma^0 \mathbf{v} - \mathbf{v} + \mathbf{v} - \pi_\Gamma \mathbf{v}\|_{[L^2(\mathcal{E})]^2} \\
&\lesssim \|\mathbf{v} - \pi_\Gamma^0 \mathbf{v}\|_{[L^2(\mathcal{E})]^2} + \|\mathbf{v} - \pi_\Gamma \mathbf{v}\|_{[L^2(\mathcal{E})]^2} \lesssim \frac{h}{H}|\mathbf{v}|_{[H^1(\mathcal{E})]^2}.
\end{aligned}$$

By a standard argument, we can prove that π_Γ^1 is H_0^1-bounded. In fact, letting $\Pi_\Gamma^1 : [H_0^1(\mathcal{E})]^2 \to \mathbf{V}_\Delta^\mathcal{E}$ denote the $[H_0^1(\mathcal{E})]^2$ projection onto $\mathbf{V}_\Delta^\mathcal{E}$, defined as

$$\Pi_\Gamma^1(\mathbf{v}) = arg \min_{\mathbf{w} \in \mathbf{V}_\Delta^\mathcal{E}} \left(\frac{1}{2}|\mathbf{w}|_{1,\mathcal{E}}^2 - < \mathbf{v}, \mathbf{w} > \right),$$

we have:

$$\begin{aligned}
|\pi_\Gamma^1 \mathbf{v}|_{[H^1(\mathcal{E})]^2} &\lesssim |\mathbf{v}|_{[H^1(\mathcal{E})]^2} + \left(\frac{h}{H}\right)^{-1} \|\pi_\Gamma^1 \mathbf{v} - \Pi_\Gamma^1 \mathbf{v}\|_{[L^2(\mathcal{E})]^2} \\
&\lesssim |\mathbf{v}|_{[H^1(\mathcal{E})]^2} + \left(\frac{h}{H}\right)^{-1} \|\pi_\Gamma^1 \mathbf{v} - \mathbf{v}\|_{[L^2(\mathcal{E})]^2} + \left(\frac{h}{H}\right)^{-1} \|\mathbf{v} - \Pi_\Gamma^1 \mathbf{v}\|_{[L^2(\mathcal{E})]^2} \\
&\lesssim |\mathbf{v}|_{[H^1(\mathcal{E})]^2} + \left(\frac{h}{H}\right)^{-1} \left(\frac{h}{H}\right) |\mathbf{v}|_{[H^1(\mathcal{E})]^2} + \left(\frac{h}{H}\right)^{-1} \left(\frac{h}{H}\right) |\mathbf{v}|_{[H^1(\mathcal{E})]^2} \\
&\lesssim |\mathbf{v}|_{[H^1(\mathcal{E})]^2}.
\end{aligned}$$

By space interpolation we then deduce that π_Γ^1 is uniformly bounded for all $s \in [0,1], s \neq 1/2$, that is for all $\mathbf{v} \in [H_0^s(\mathcal{E})]^2$ we have:

$$\|\pi_\Gamma^1 \mathbf{w}\|_{[H_0^s(\mathcal{E})]^2} \lesssim \|\mathbf{w}\|_{[H_0^s(\mathcal{E})]^2}$$

with a constant independent of s, whereas for $\mathbf{v} \in [H_{00}^{1/2}(\mathcal{E})]^2$ we have:

$$\|\pi_\Gamma^1 \mathbf{v}\|_{[H_{00}^{1/2}(\mathcal{E})]^2} \lesssim \|\mathbf{v}\|_{[H_{00}^{1/2}(\mathcal{E})]^2}.$$

We now recall that for $0 < s < 1/2$, the space $[H^s(\mathcal{E})]^2$ is embedded in $[H_0^s(\mathcal{E})]^2$ and we have [16]:

$$\|\mathbf{v}\|_{[H_0^s(\mathcal{E})]^2} \lesssim \frac{1}{1-2s} |\mathbf{v}|_{[H^s(\mathcal{E})]^2} + \frac{1}{\sqrt{1-2s}} \|\mathbf{v}\|_{[L^\infty(\mathcal{E})]^2}.$$

Then, for $\varepsilon \in]0, 1/2[$ and $\mathbf{v}_\Gamma^\mathcal{E} \in \mathbf{V}_\Gamma^\mathcal{E}$ we have:

$$\begin{aligned}
\|\pi_\Gamma^1 \mathbf{w}_\Gamma\|_{[H_{00}^{1/2}(\mathcal{E})]^2} \lesssim \left(\frac{h}{H}\right)^{-\varepsilon} \|\pi_\Gamma^1 \mathbf{w}_\Gamma\|_{[H_0^{1/2-\varepsilon}(\mathcal{E})]^2} &\lesssim \left(\frac{h}{H}\right)^{-\varepsilon} \|\mathbf{w}_\Gamma\|_{[H_0^{1/2-\varepsilon}(\mathcal{E})]^2} \\
&\lesssim \frac{1}{\varepsilon}\left(\frac{h}{H}\right)^{-\varepsilon} |\mathbf{v}_\Gamma|_{[H_0^{1/2-\varepsilon}(\mathcal{E})]^2} + \frac{1}{\sqrt{\varepsilon}}\left(\frac{h}{H}\right)^{-\varepsilon} \|\mathbf{v}_\Gamma\|_{[L^\infty(\mathcal{E})]^2} \\
&\lesssim (1 + \log(H/h)) |\mathbf{v}_\Gamma|_{[H^{1/2}(\mathcal{E})]^2} + \sqrt{1 + \log(H/h)} \|\mathbf{v}_\Gamma\|_{[L^\infty(\mathcal{E})]^2},
\end{aligned}$$

where the last inequality is obtained by choosing $\varepsilon = 1/|1 + \log(H/h)|$. Observing that for $\mathbf{v}_\Gamma^\mathcal{E} \in \mathbf{V}_\Gamma^\mathcal{E}$ we have:

$$\mathbf{v}_\Delta^\mathcal{E} = i_\Gamma^0 \mathbf{v}_\Gamma^\mathcal{E} = \pi_\Gamma^1 \mathbf{v}_\Gamma^\mathcal{E},$$

so we get the thesis. □

To take care of the estimates in three dimension we need bounds for edge and face terms. First of all we need a Riesz Basis Property, that gives us the equivalence between the $L^2(f)$ norm of a function in $\widehat{\mathbb{B}}_k(f)$ and the euclidean norm of the vector of its DoFs:

Lemma 3.3.7. *Let f be a face of an element K. For all $\mathbf{v}_h \in [\widehat{\mathbb{B}}_k(f)]^3$ we have:*

$$\int_f |\mathbf{v}_h|^2 \simeq h^2 \sum_{i \in \mathcal{X}} |\mathbf{D}^i_{\mathbf{v}}(\mathbf{v}_h)|^2, \tag{3.64}$$

where $\mathcal{X}$ is the union of the set of DoFs $\mathbf{D}^1_{\mathbf{v}}, \mathbf{D}^2_{\mathbf{v}}$ and $\mathbf{D}^3_{\mathbf{v}}$.

We do not provide the proof of this Lemma because it is the natural vector extension of the one in [18] and [36], with a slightly different choice for the DoFs of the face velocity space. Moreover, it simply involves the Π^0_{k+1} projection instead of Π^0_k.
Finally we need to recall two lemmas that we need for estimating the contributions of the interface velocity functions defined on the edges and faces of the subdomains. The first one guarantees us an estimate for the edges and it is nothing else than the vectorial extension of the one in [18]:

Lemma 3.3.8. *Let be Ω_i a subdomain and let be F a face of Ω_i. Then for $\mathbf{v}_\Gamma \in \mathbf{V}_{\Gamma|F}$ we have*

$$\|\mathbf{v}_\Gamma\|_{[L^2(\partial F)]^3} \lesssim \sqrt{1 + \log(H/h)} \|\mathbf{v}_\Gamma\|_{[H^{1/2}(F)]^3}, \tag{3.65}$$

where the subscript $|\mathcal{F}$ means that we are restricting the interface space $\mathbf{V}_\Gamma$ on the face $\mathcal{F}$.

The second is a *face lemma*:

Lemma 3.3.9. *Let $\mathbf{v}_\Gamma \in \mathbf{V}^{(i)}_\Gamma$. Then, for all faces $\mathcal{F}_i$ of Ω_i it holds that $\theta_{\mathcal{F}_i} \mathbf{v}_{\Gamma|\mathcal{F}_i} \in [H^{1/2}_{00}(\mathcal{F}_i)]^3$ and*

$$\|\theta_{\mathcal{F}_i} \mathbf{v}_\Gamma\|^2_{[H^{1/2}_{00}(\mathcal{F}_i)]^3} \lesssim (1 + \log(H/h))^2 \|\mathbf{v}_\Gamma\|^2_{[H^{1/2}(\mathcal{F}_i)]^3}, \tag{3.66}$$

where $\theta_{\mathcal{F}_i}$ is defined at the beginning of Section 3.3.4.

The proof of this lemma is quite similar to the Lemma 3.3.6 one, adapted for the faces. It basically follows the same procedure of Lemma 5.2 in [18]. The only difference is that we need to substitute the the Scott-Zhang interpolation with the Stokes VEM interpolant defined in Lemma 2.2.1 and the Riesz basis property of Lemma 3.3.7 adapted in our framework.

3.3.3 The coarse space in two dimensions

To satisfy the assumptions 1 and 2 in the previous section we have to choose properly the primal constraints for the interface velocity space. The natural choice it is to make all the vertices of the subdomains as primal DoFs for both the components of the velocity. However, if this is sufficient to guarantee the stability of the average operator as we will see in Lemma 3.3.10, it is not enough to satisfy the no-net-flux. In particular, some extra edge constraints are necessary. For each interface edge Γ_{ij}, which is shared by a pair of subdomains Ω_i and Ω_j, we make

$$\int_{\Gamma_{ij}} \mathbf{v}^{(i)}_\Gamma \cdot \mathbf{n}_{ij} = \int_{\Gamma_{ij}} \mathbf{v}^{(j)}_\Gamma \cdot \mathbf{n}_{ij} \tag{3.67}$$

for a fixed selection of the normal $\mathbf{n}_{ij}$ of Γ_{ij}. Proceeding the discussion with this first choice, after changing the variables, the dual interface velocity component will vanish at the subdomain vertices and its normal component will have a weighted zero average over each Γ_{ij}, i.e. $\int_{\Gamma_i} \mathbf{v_\Gamma}^{(i)} \cdot \mathbf{n}_{ij} = \int_{\Gamma_j} \mathbf{v_\Gamma}^{(j)} \cdot \mathbf{n}_{ij} = 0$. By the definition of the average operator $E_{D,\Delta}$, we have that the average interface velocity is:

$$E_{D,\Delta}\mathbf{v}_\Delta = \frac{\nu_i^\gamma}{\nu_i^\gamma + \nu_j^\gamma}\mathbf{v}_\Delta^{(i)} + \frac{\nu_j^\gamma}{\nu_i^\gamma + \nu_j^\gamma}\mathbf{v}_\Delta^{(j)}$$

on each edge, and hence

$$\int_{\Gamma_{ij}} (E_{D,\Delta}\mathbf{v}_\Delta)^{(i)} \cdot \mathbf{n}_{ij} = 0. \tag{3.68}$$

The stability of the average operator is shown in the following:

Lemma 3.3.10. *For all $\mathbf{v}_\Gamma \in \widetilde{\mathbf{V}}_\Gamma$ we have:*

$$|E_{D,\Gamma}\mathbf{v}_\Gamma|^2_{1/2,\Gamma} \lesssim (1 + \log(H/h))^2 |\mathbf{v}_\Gamma|^2_{1/2,\Gamma} \tag{3.69}$$

Proof. Let consider $\mathbf{v}_\Gamma \in \widetilde{\mathbf{V}}_\Gamma$ and we define $\mathbf{w}_\Gamma := E_D\mathbf{v}_\Gamma$. We have:

$$|\mathbf{w}_\Gamma|_{1/2,\Gamma} = |\mathbf{w}_\Gamma - \mathbf{v}_\Gamma + \mathbf{v}_\Gamma|_{1/2,\Gamma} \leq |\mathbf{w}_\Gamma - \mathbf{v}_\Gamma|_{1/2,\Gamma} + |\mathbf{v}_\Gamma|_{1/2,\Gamma}. \tag{3.70}$$

Since all the vertices of the subdomains are primal, we can rewrite:

$$|\mathbf{w}_\Gamma - \mathbf{v}_\Gamma|_{1/2,\Gamma} = \sum_{i=1}^{N} |\mathbf{w}_\Gamma^{(i)} - \mathbf{v}_\Gamma^{(i)}|_{1/2,\Gamma_i}. \tag{3.71}$$

Using the splitting in (3.62) for both $\mathbf{v}_\Gamma$ and $\mathbf{w}_\Gamma$ and observing that $\mathbf{v}_\Pi = \mathbf{w}_\Pi$, we have that:

$$\sum_{i=1}^{N} |\mathbf{w}_\Gamma - \mathbf{v}_\Gamma|^2_{1/2,\Gamma_i} \lesssim \sum_{i=1}^{N} |\mathbf{w}_\Delta - \mathbf{v}_\Delta^{(i)}|^2_{1/2,\Gamma_i}.$$

We observe that, for a given macro edge $\mathcal{E}_{ij} = \partial\Omega_i \cap \partial\Omega_j$, we have

$$\mathbf{w}_\Delta|_{\mathcal{E}_{ij}} = \frac{\nu_i^\gamma}{\nu_i^\gamma + \nu_j^\gamma}\mathbf{v}_\Delta^{(i)}|_{\mathcal{E}_{ij}} + \frac{\nu_j^\gamma}{\nu_i^\gamma + \nu_j^\gamma}\mathbf{v}_\Delta^{(j)}|_{\mathcal{E}_{ij}}$$

whence

$$(\mathbf{w}_\Delta - \mathbf{v}_\Delta^{(i)})|_{\mathcal{E}_{ij}} = \frac{\nu_j^\gamma}{\nu_i^\gamma + \nu_j^\gamma}\big(\mathbf{v}_\Delta^{(j)} - \mathbf{v}_\Delta^{(i)}\big)|_{\mathcal{E}_{ij}}.$$

Observing that:

$$\nu_i\big(\frac{\nu_j^\gamma}{\nu_i^\gamma+\nu_j^\gamma}\big)^2 \leq \min\{\nu_i,\nu_j\} \leq \nu_j, \tag{3.72}$$

we have:

$$\begin{aligned}
\sum_{i=1,..,N} \nu_i\,|\mathbf{v}_\Delta^{(i)} - \mathbf{w}_\Delta|^2_{1/2,\Gamma_i} &\lesssim \sum_{i=1,..,N}\sum_{\mathcal{E}_{ij}\in\mathcal{E}_H^i} \nu_i\,|\mathbf{v}_\Delta^{(i)}|_{\mathcal{E}_{ij}} - \mathbf{w}_\Delta|_{\mathcal{E}_{ij}}|^2_{[H^{1/2}_{00}(\mathcal{E}_{ij})]^2}\\
&\lesssim \sum_{i=1,..,N}\sum_{\mathcal{E}_{ij}=\partial\Omega_i\cap\partial\Omega_j\in\mathcal{E}_H^i} \nu_i\left(\frac{\nu_j^\gamma}{\nu_i^\gamma+\nu_j^\gamma}\right)^2 |\mathbf{v}_\Delta^{(i)}|_{\mathcal{E}_{ij}} - \mathbf{v}_\Delta^{(j)}|_{\mathcal{E}_{ij}}|^2_{[H^{1/2}_{00}(\mathcal{E}_{ij})]^2}\\
&= \sum_{\mathcal{E}_{ij}=\partial\Omega_i\cap\partial\Omega_j\in\mathcal{E}_H} \frac{\nu_i\,\nu_j^{2\gamma}+\nu_j\,\nu_i^{2\gamma}}{(\nu_i^\gamma+\nu_j^\gamma)^2} |\mathbf{v}_\Delta^{(i)}|_{\mathcal{E}_{ij}} - \mathbf{v}_\Delta^{(j)}|_{\mathcal{E}_{ij}}|^2_{[H^{1/2}_{00}(\mathcal{E}_{ij})]^2}\\
&\lesssim \sum_{\mathcal{E}_{ij}=\partial\Omega_i\cap\partial\Omega_j\in\mathcal{E}_H} \min\{\nu_i,\nu_j\}|\mathbf{v}_\Delta^{(i)}|_{\mathcal{E}_{ij}} - \mathbf{v}_\Delta^{(j)}|_{\mathcal{E}_{ij}}|^2_{[H^{1/2}_{00}(\mathcal{E}_{ij})]^2}.
\end{aligned} \tag{3.73}$$

We recall that we have the following bound for any $\mathbf{v}_{\Gamma_{|\mathcal{E}_{ij}}} \in \mathbf{V}_\Gamma^{\mathcal{E}_{ij}}$ with $\mathbf{v}_{\Gamma_{|\mathcal{E}_{ij}}}(x) = 0$ for some $x \in \Gamma_{|\mathcal{E}_{ij}}$:

$$\|\mathbf{v}_{\Gamma_{|\mathcal{E}_{ij}}}\|_{[L^\infty(\mathcal{E}_{ij})]^2} \lesssim \sqrt{1+\log(H/h)}\,|\mathbf{v}_{\Gamma_{|\mathcal{E}_{ij}}}|_{[H^{1/2}(\mathcal{E}_{ij})]^2}. \tag{3.74}$$

This bound was proven in [16] where it is provided a proof that only relies on the properties of $\mathbf{V}_\Gamma^{\mathcal{E}_{ij}}$. Using Lemma 3.3.6 and (3.74), because $\mathbf{v}_\Gamma^{(i)} - \mathbf{v}_\Gamma^{(j)}$ vanishes at the extremes of Γ_{ij}, we have:

$$\begin{aligned}
|\mathbf{v}_\Delta^{(i)}|_{\mathcal{E}_{ij}} - \mathbf{v}_\Delta^{(j)}|_{\mathcal{E}_{ij}}|^2_{[H^{1/2}_{00}(\mathcal{E}_{ij})]^2} &\lesssim (1+\log(H/h))^2\big|\mathbf{v}_\Gamma^{(i)}|_{\mathcal{E}_{ij}} - \mathbf{v}_\Gamma^{(j)}|_{\mathcal{E}_{ij}}\big|_{[H^{1/2}(\mathcal{E}_{ij})]^2}\\
&\quad+(1+\log(H/h))^2\|\mathbf{v}_\Gamma^{(i)}|_{\mathcal{E}_{ij}} - \mathbf{v}_\Gamma^{(j)}|_{\mathcal{E}_{ij}}\|_{[L^\infty(\mathcal{E}_{ij})]^2}\\
&\lesssim (1+\log(H/h))^2\big|\mathbf{v}_\Gamma^{(i)}|_{\mathcal{E}_{ij}} - \mathbf{v}_\Gamma^{(j)}|_{\mathcal{E}_{ij}}\big|_{[H^{1/2}(\mathcal{E}_{ij})]^2}\\
&\lesssim (1+\log(H/h))^2\big(\big|\mathbf{v}_\Gamma^{(i)}|_{\mathcal{E}_{ij}}\big|_{[H^{1/2}(\mathcal{E}_{ij})]^2} + \big|\mathbf{v}_\Gamma^{(j)}|_{\mathcal{E}_{ij}}\big|_{[H^{1/2}(\mathcal{E}_{ij})]^2}\big),
\end{aligned}$$

whence we obtain

$$\sum_{i=1,..,N} \nu_i\,|\mathbf{w}_\Delta - \mathbf{v}_\Delta|^2_{1/2,\Gamma_i} \lesssim (1+\log(H/h))^2 \sum_{i=1,..,N} \nu_i\,|\mathbf{v}_\Gamma^{(i)}|^2_{1/2,\Gamma_i}$$

We finally have:

$$|\mathbf{v}_\Gamma - \mathbf{w}_\Gamma|^2_{1/2,\Gamma} \lesssim (1+\log(H/h))^2|\mathbf{v}_\Gamma|^2_{1/2,\Gamma},$$

and by applying the inverse triangular inequality we obtain the thesis. □

Remark 3.3.1. We make note that if we use the standard multiplicity scaling we can not more exploit the relation (3.72). In this way we are not able to do the last estimate of (3.73) and we can only obtain:

$$\sum_{i=1,..,N} \nu_i \, |\mathbf{v}_\Delta^{(i)} - \mathbf{w}_\Delta|^2_{1/2,\Gamma_i} \lesssim \sum_{\mathcal{E}_{ij}=\partial\Omega_i\cap\partial\Omega_j\in\mathcal{E}_H} (\nu_i+\nu_j)|\mathbf{v}_\Delta^{(i)}|_{\mathcal{E}_{ij}} - \mathbf{v}_\Delta^{(j)}|_{\mathcal{E}_{ij}}|^2_{[H^{1/2}_{00}(\mathcal{E}_{ij})]^2}. \tag{3.75}$$

The final estimate then concludes with a dependence on the viscosity constant ν, since the high viscosity coefficient penalize the jump. Numerical results in 3.4.1.2 will show this aspect.

3.3.4 The coarse space in three dimensions

In this section we provide a recipe to construct the coarse space in three dimensions. Following [70], we recall that in three dimensions, the interface Γ of a subdomain Ω_i is constituted by faces $\mathcal{F}_l$ shared by two subdomains, edges $\mathcal{E}_k$ that are shared by more than two subdomains (the notation $\mathcal{E}_k(\mathcal{F}_l)$ is to underline that the edge $\mathcal{E}_k$ belongs to the boundary of face $\mathcal{F}_l$) and vertices V_j that are the endpoints of the edges. Now let be G any one of these geometrical entities and let be $\mathbf{v} \in \mathbf{V}$ a generic virtual function, we define the cut-off linear functional θ_G that maps a virtual function $\mathbf{v}$ into another virtual function $\theta_G(\mathbf{v})$, that is equal to $\mathbf{v}$ on all the DoFs that belongs to G and 0 elsewhere, for simplicity we generally write $\theta_G\mathbf{v}$ instead of $\theta_G(\mathbf{v})$. We use the subscript $\mathcal{F}_l$ to refer the DoFs that live only in the interior of the face excluding the boundary face DoFs. When a multi-subscript is present, like $\mathcal{F}_{ij}$, it means that the face is shared between the subdomains Ω_i and Ω_j.

To satisfy Assumption 1, we first make all vertices primal, and then we require that, for any $\mathbf{v}_\Delta$, the two quantities:

$$\int_{\mathcal{F}_{ij}} \mathbf{v}_\Delta^{(i)} \cdot \mathbf{n}_{ij} = \int_{\mathcal{F}_{ij}} (\theta_{\mathcal{F}_{ij}} \mathbf{v}_\Delta^{(i)}) \cdot \mathbf{n}_{ij} + \sum_{\mathcal{E}_k\subset\mathcal{F}_{ij}} \int_{\mathcal{F}_{ij}} (\theta_{\mathcal{E}_k(\mathcal{F}_{ij})} \mathbf{v}_\Delta^{(i)}) \cdot \mathbf{n}_{ij} \tag{3.76}$$

and

$$\begin{aligned} \int_{\mathcal{F}_{ij}} (E_D\mathbf{v})_\Delta^{(i)} \cdot \mathbf{n}_{ij} &= \frac{1}{2} \int_{\mathcal{F}_{ij}} \theta_{\mathcal{F}_{ij}} (\mathbf{v}_\Delta^{(i)} + \mathbf{v}_\Delta^{(j)}) \cdot \mathbf{n}_{ij} \\ &+ \sum_{\mathcal{E}_k\subset\mathcal{F}_{ij}} \sum_{m\in\mathcal{N}_{\mathcal{E}_k}} \frac{1}{card(\mathcal{N}_{\mathcal{E}_k})} \int_{\mathcal{F}_{ij}} (\theta_{\mathcal{E}_k(\mathcal{F}_{ij})} \mathbf{v}_\Delta^{(m)}) \cdot \mathbf{n}_{ij} \end{aligned} \tag{3.77}$$

vanish, where $\mathcal{N}_{\mathcal{E}_k}$ is the set of all the subdomains that share the edge $\mathcal{E}_k$ and $\mathbf{n}_{ij}$ is the unit outward normal vector to the face $\mathcal{F}_{ij}$. To do so, we need that all the integrals of the right-hand side of (3.76) and (3.77) will vanish. This can be achieved by enforcing a primal constraint for each face $\mathcal{F}_{ij}$:

$$\int_{\mathcal{F}_{ij}} (\theta_{\mathcal{F}_{ij}} \mathbf{v}_\Gamma^{(i)}) \cdot \mathbf{n}_{ij} = \int_{\mathcal{F}_{ij}} (\theta_{\mathcal{F}_{ij}} \mathbf{v}_\Gamma^{(j)}) \cdot \mathbf{n}_{ij} \tag{3.78}$$

and a set of primal constraints requiring that for each edge $\mathcal{E}_k$, on each face $\mathcal{F}_{ij}$, the following quantity is the same for all $m \in \mathcal{N}_{\mathcal{E}_k}$:

$$\int_{\mathcal{F}_{ij}} (\theta_{\mathcal{E}_k(\mathcal{F}_{ij})} \mathbf{v}_\Gamma^{(m)}) \cdot \mathbf{n}_{ij} \tag{3.79}$$

To ensure constraint (3.78), we need one primal variable per face, while, to ensure constraint (3.79), we need as many primal variables as the number of faces which share the edge $\mathcal{E}_k$. We remark that in our VEM context, the quantities in (3.78) and (3.79) are directly computable from the DoFs introduced in Chapter 2. For particular subdomain partitions, such as those with cubic subdomains and hexahedral elements, it might happen that some of the primal basis functions are linearly dependent; this situation is harmless in practice since we can perform a singular value decomposition of the basis DoFs and obtain non-singular coarse operators.

To satisfy Assumption 2, we have to ensure that we have the right type of constraints that can control the rigid body modes (at least six constraints: the three translations and the three rotations). Given the fact that the coefficients of the Stokes problem are all the same for each subdomain and that the vertices of the subdomains have been selected as primal constraints, we can prove that the second Assumption is satisfied if also all the faces of the interface Γ are *fully primal* in the sense of the following definition (see [66, def. 5.3]):

Definition 3.3.2. A face $\mathcal{F}_{ij}$ is called fully primal if, in the space of primal constraints over $\mathcal{F}_{ij}$, there exists a set f_m, $m = 1, ..., 6$, of linear functionals on $\mathbf{V}_\Gamma^{(i)}$ with the following properties:

- $|f_m(\mathbf{v}_\Gamma^{(i)})|^2 \leq CH^{-1}(1 + \log(H/h))\|\mathbf{v}_\Gamma^{(i)}\|_{H^{1/2}(\mathcal{F}_{ij})}$;
- $f_m(\mathbf{r}_l) = \delta_{ml} \quad \forall m, l = 1, ..., 6 \quad \mathbf{r}_l \in ker(\varepsilon)$,

with $C > 0$ and $\mathbf{v}_\Gamma^{(i)} \in \mathbf{V}_\Gamma^{(i)}$.

We recall that, to satisfy Assumption 1, we have chosen as primal constraints some averages of the normal component of the velocity over the edges. In Section 7 of [70] (see also further details in [66]), it is shown that this choice of primal DoFs is sufficient to guarantee a set of functionals that makes the face fully primal. It is essential to underline that in some particular cases, like triangular or rectangular faces, some of these constraints can be linearly dependent and it is necessary to introduce some extra edge average in the tangential direction. This condition can be verified numerically because the selection of a set of linearly independent set of constraints can be computed using a QR factorization and selecting six functionals that are robustly independent. For sake of simplicity, our proof will be for the standard multiplicity scaling. It is easy to extend it to a general ν-scaling using the same techniques as in the previous section and to a general non diagonal scaling following [101]. We are ready to state the lemma related to the stability of the average operator:

Lemma 3.3.11. *For all $\mathbf{v}_\Gamma \in \widetilde{\mathbf{V}}_\Gamma$ holds that:*

$$|E_D \mathbf{v}_\Gamma|^2_{\widetilde{S}} \lesssim C \frac{1}{\beta^2} \left(1 + \log\left(\frac{H}{h}\right)\right)^2 |\mathbf{v}_\Gamma|^2_{\widetilde{S}}, \tag{3.80}$$

where C is a positive constant that is independent on h, H and β_h, but it can depend on the degree k of the virtual element discretization.

Proof. Let consider $\mathbf{v}_\Gamma \in \widetilde{\mathbf{V}}_\Gamma$ and we define $\mathbf{w}_\Gamma := E_D \mathbf{v}_\Gamma$. We have:

$$|\mathbf{w}_\Gamma|_{\widetilde{S}} = |\mathbf{w}_\Gamma - \mathbf{v}_\Gamma + \mathbf{v}_\Gamma|_{\widetilde{S}} \leq |\mathbf{w}_\Gamma - \mathbf{v}_\Gamma|_{\widetilde{S}} + |\mathbf{v}_\Gamma|_{\widetilde{S}}. \tag{3.81}$$

Since all the vertices of the subdomains are primal, we can rewrite:

$$|\mathbf{w}_\Gamma - \mathbf{v}_\Gamma|_{\widetilde{S}_\Gamma} = \sum_{i=1}^{N} |\mathbf{w}^{(i)}_\Gamma - \mathbf{v}^{(i)}_\Gamma|_{S^{(i)}_\Gamma} \tag{3.82}$$

and for each subdomain Ω_i we can also use the split:

$$\mathbf{w}^{(i)}_\Gamma - \mathbf{v}^{(i)}_\Gamma = \sum_{\mathcal{F}_{ij} \subset \partial\Omega_i} \theta_{\mathcal{F}_{ij}} (\mathbf{w}^{(i)}_\Gamma - \mathbf{v}^{(i)}_\Gamma) + \sum_{\mathcal{E}_k \subset \partial\Omega_i} \theta_{\mathcal{E}_k} (\mathbf{w}^{(i)}_\Gamma - \mathbf{v}^{(i)}_\Gamma) \tag{3.83}$$

Recalling that a face $\mathcal{F}_{ij}$ is shared by two subdomains i, j and using the explicit definition of E_D:

$$\mathbf{w}^{(i)}_\Gamma - \mathbf{v}^{(i)}_\Gamma = \frac{1}{2}(\mathbf{v}^{(i)}_\Gamma + \mathbf{v}^{(j)}_\Gamma) - \mathbf{v}^{(i)}_\Gamma = \mathbf{v}^{(j)}_\Gamma - \mathbf{v}^{(i)}_\Gamma.$$

Starting from the face contributions, we write:

$$\mathbf{v}^{(j)}_\Gamma - \mathbf{v}^{(i)}_\Gamma = \left(\mathbf{v}^{(j)}_\Gamma - \sum_{m=1}^{6} f^{\mathcal{F}_{ij}}_m (\mathbf{v}^{(j)}_\Gamma)\mathbf{r}_m\right) - \left(\mathbf{v}^{(i)}_\Gamma - \sum_{m=1}^{6} f^{\mathcal{F}_{ij}}_m (\mathbf{v}^{(i)}_\Gamma)\mathbf{r}_m\right) \tag{3.84}$$

where $\{\mathbf{r}_m\}$ for $m = 1, .., 6$ is the basis for the rigid body modes and the $f^{\mathcal{F}_{ij}}_m(\cdot)$ are the functionals that are equal for the faces i and j since the faces are fully primal. For an arbitrary rigid body mode, $\mathbf{r}^{(i)} \in \mathbf{V}^{(i)}_\Gamma$ we write:

$$\mathbf{v}^{(i)}_\Gamma - \sum_{m=1}^{6} f^{\mathcal{F}_{ij}}_m (\mathbf{v}^{(i)}_\Gamma)\mathbf{r}_m = (\mathbf{v}^{(i)}_\Gamma - \mathbf{r}^{(i)}) - \sum_{m=1}^{6} f^{\mathcal{F}_{ij}}_m (\mathbf{v}^{(i)}_\Gamma - \mathbf{r}^{(i)})\mathbf{r}_m \tag{3.85}$$

We can estimate the first term of the right-hand side using lemma 3.3.9:

$$\|\theta_{\mathcal{F}_{ij}} (\mathbf{v}^{(i)}_\Gamma - \mathbf{r}^{(i)})\|_{[H^{1/2}_{00}(\mathcal{F}_{ij})]^3} \lesssim (1 + \log(H/h)) \|\mathbf{v}^{(i)}_\Gamma - \mathbf{r}^{(i)}\|_{[H^{1/2}(\mathcal{F}_{ij})]^3}. \tag{3.86}$$

Then we consider the second term of (3.85), and we estimate it using two additional contributions, by lemma 8 in [66]:

$$\|\theta_{\mathcal{F}_{ij}}\mathbf{r}_m^{(i)}\|_{[H_{00}^{1/2}(\mathcal{F}_{ij})]^3} \lesssim H(1+\log(H/h)) \tag{3.87}$$

and by the definition of fully primal face (3.3.2) and (3.3.8) we have:

$$|f_m^{\mathcal{F}_{ij}}(\mathbf{v}_\Gamma^{(i)}-\mathbf{r}^{(i)})| \lesssim \frac{1}{H}(1+\log(H/h))\|\mathbf{v}_\Gamma^{(i)}-\mathbf{r}^{(i)}\|_{[H^{1/2}(\mathcal{F}_{ij})]^3}. \tag{3.88}$$

Then combining the previous two estimates we have:

$$\|\theta_{\mathcal{F}_{ij}}\sum_{m=1}^{6} f_m^{\mathcal{F}_{ij}}(\mathbf{v}_\Gamma^{(i)}-\mathbf{r}^{(i)})\mathbf{r}_m\|_{[H_{00}^{1/2}(\mathcal{F}_{ij})]^3} \lesssim (1+\log(H/h))\|\mathbf{v}_\Gamma^{(i)}-\mathbf{r}^{(i)}\|_{[H^{1/2}(\mathcal{F}_{ij})]^3}. \tag{3.89}$$

By triangular inequality and since $\mathbf{r}^{(i)}$ is an arbitrary rigid body mode, we can take the minimum of over all the modes and by Lemma 1.4.2:

$$\begin{aligned}\|\theta_{\mathcal{F}_{ij}}(\mathbf{v}_\Gamma^{(i)}-\sum_{m=1}^{6} f_m^{\mathcal{F}_{ij}}(\mathbf{v}_\Gamma^{(i)})\mathbf{r}_m)\|_{[H_{00}^{1/2}(\mathcal{F}_{ij})]^3} \lesssim \|\theta_{\mathcal{F}_{ij}}(\mathbf{v}_\Gamma^{(i)}-\mathbf{r}^{(i)})\|_{[H_{00}^{1/2}(\mathcal{F}_{ij})]^3}+\\ \|\theta_{\mathcal{F}_{ij}}(\mathbf{v}_\Gamma^{(i)}-\sum_{m=1}^{6} f_m^{\mathcal{F}_{ij}}(\mathbf{v}_\Gamma^{(i)}-\mathbf{r}^{(i)})\mathbf{r}_m)\|_{[H_{00}^{1/2}(\mathcal{F}_{ij})]^3} \lesssim\\ (1+\log(H/h))\|\mathbf{v}_\Gamma^{(i)}-\mathbf{r}^{(i)}\|_{[H^{1/2}(\mathcal{F}_{ij})]^3} \lesssim (1+\log(H/h))|\mathbf{v}_\Gamma^{(i)}|_{[H^{1/2}(\mathcal{F}_{ij})]^3}\end{aligned} \tag{3.90}$$

We can repeat in the same way for the jth term and obtain:

$$\begin{aligned}\|\theta_{\mathcal{F}_{ij}}(\mathbf{v}_\Gamma^{(i)}-\mathbf{v}_\Gamma^{(j)})\|_{[H_{00}^{1/2}(\mathcal{F}_{ij})]^3} \lesssim (1+\log(H/h))|\mathbf{v}_\Gamma^{(i)}|_{[H^{1/2}(\mathcal{F}_{ij})]^3}\\ +(1+\log(H/h))|\mathbf{v}_\Gamma^{(j)}|_{[H^{1/2}(\mathcal{F}_{ij})]^3}\end{aligned} \tag{3.91}$$

Regarding the edge terms, we need to estimate contributions that depend on the number of subdomains that share the edge. We propose the estimate for one of these contributions since the others are treated similarly. We consider an edge $\mathcal{E}_k \subset \partial\mathcal{F}_{ij}$, by the fact that all the faces are fully primal, we can reduce these terms to face estimates, we write:

$$\begin{aligned}\|\mathbf{v}_\Gamma^{(i)}-\mathbf{v}_\Gamma^{(j)}\|^2_{[L^2(\mathcal{E}_k)]^3} \lesssim \|\mathbf{v}_\Gamma^{(i)}-\sum_{m=1}^{6} f_m^{\mathcal{F}_{ij}}(\mathbf{v}_\Gamma^{(i)})\mathbf{r}_m\|^2_{[L^2(\mathcal{E}_k)]^3}\\ +\|\mathbf{v}_\Gamma^{(j)}-\sum_{m=1}^{6} f_m^{\mathcal{F}_{ij}}(\mathbf{v}_\Gamma^{(j)})\mathbf{r}_m\|^2_{[L^2(\mathcal{E}_k)]^3}\end{aligned} \tag{3.92}$$

We proceed again considering an arbitrary rigid body mode $\mathbf{r}^{(i)} \in \mathbf{V}_\Gamma^{(i)}$. Using the triangular inequality and (3.3.8):

$$\begin{aligned} \|\mathbf{v}_\Gamma^{(i)} - \sum_{m=1}^{6} f_m^{\mathcal{F}_{ij}}(\mathbf{v}_\Gamma^{(i)})\mathbf{r}_m\|^2_{[L^2(\mathcal{E}_k)]^3} \lesssim \|\mathbf{v}_\Gamma^{(i)} - \mathbf{r}^{(i)}\|_{[L^2(\mathcal{E}_k)]^3} + \\ \|\sum_{m=1}^{6} f_m^{\mathcal{F}_{ij}}(\mathbf{v}_\Gamma^{(i)} - \mathbf{r}^{(i)})\mathbf{r}_m\|^2_{[L^2(\mathcal{E}_k)]^3} \lesssim (1+\log(H/h))\|\mathbf{v}_\Gamma^{(i)}\|^2_{[H^{1/2}(\mathcal{F}_{ij})]^3} + \\ \sum_{m=1}^{6} |f_m^{\mathcal{F}_{ij}}(\mathbf{v}_\Gamma^{(i)} - \mathbf{r}^{(i)})|^2\|\mathbf{r}_m\|^2_{[L^2(\mathcal{E}_k)]^3}. \end{aligned} \tag{3.93}$$

It can be proved that $\|\mathbf{r}_m\|_{[L^2(\mathcal{E}_k)]^3} \lesssim H$ ([67]), and now using (3.88) and minimizing again on all over the rigid body modes:

$$\|\mathbf{v}_\Gamma^{(i)} - \sum_{m=1}^{6} f_m^{\mathcal{F}_{ij}}(\mathbf{v}_\Gamma^{(i)})\mathbf{r}_m\|^2_{[L^2(\mathcal{E}_k)]^3} \lesssim (1+\log(H/h))|\mathbf{v}_\Gamma^{(j)}|^2_{[H^{1/2}(\mathcal{F}_{ij})]^3}. \tag{3.94}$$

We have an analogous result for the jth term and obtain:

$$\begin{aligned} \|\mathbf{v}_\Gamma^{(i)} - \mathbf{v}_\Gamma^{(j)}\|^2_{[L^2(\mathcal{E}_k)]^3} \lesssim (1+\log(H/h))|\mathbf{v}_\Gamma^{(i)}|^2_{[H^{1/2}(\mathcal{F}_{ij})]^3} \\ +(1+\log(H/h))|\mathbf{v}_\Gamma^{(j)}|^2_{[H^{1/2}(\mathcal{F}_{ij})]^3}. \end{aligned} \tag{3.95}$$

We conclude by Lemma 3.3.1, combining (3.91) and (3.95) by summing over the subdomains. □

Assumption 2 is then obtained by combining Lemma 3.3.11 and 3.3.1.

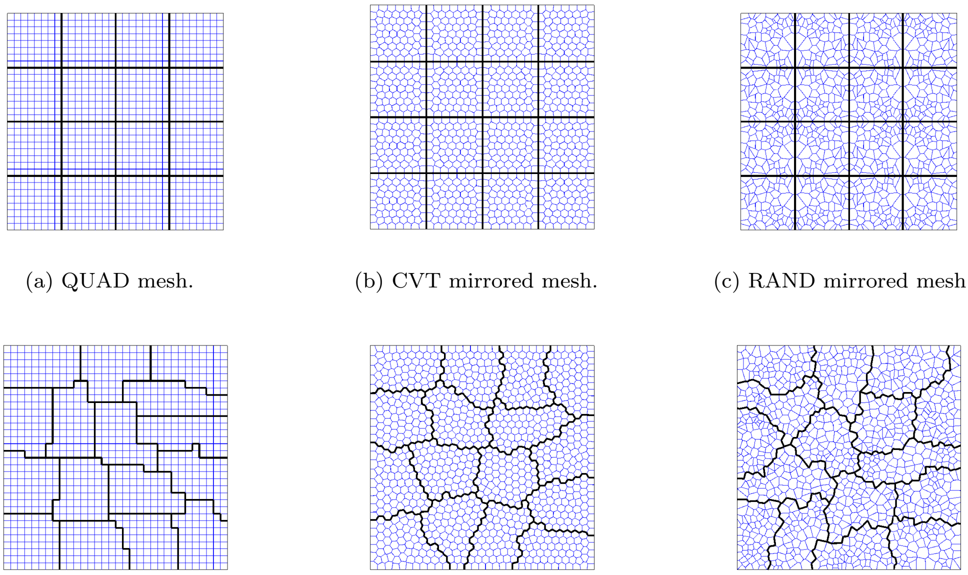
3.4 Numerical Results

In this section, we provide some numerical tests to study the behavior of the BDDC preconditioner with respect to the mesh size h, the number of subdomains N, the shape of the polygonal mesh elements, the subdomain partition technique used and coefficient jumps.

3.4.1 Numerical Results in 2D

We solve the Stokes equations on the unit square domain $\Omega = [0,1]\times[0,1]$, applying homogeneous Dirichlet boundary conditions on the whole $\partial\Omega$. In the following tables, we report the number of iterations to solve the global interface saddle-point problem (3.9) with the non-preconditioned GMRES method or the PCG method, accelerated by BDDC. Where possible, we estimate the extreme eigenvalues using the Lanczos trick [68]. Both in case of PCG and GMRES, we set the tolerance for the relative residual error to 10^{-6}.

Our tests have been executed on different types of polygonal meshes (QUAD, CVT and RAND as in Figure 2.2 using two different partition techniques (Figure 3.3). We identify with SQUARE the partition generated by subdividing the domain Ω into square subdomains and using a fixed



(a) QUAD mesh.

(b) CVT mirrored mesh.

(c) RAND mirrored mesh.

(d) QUAD mesh METIS

(e) CVT mesh METIS.

(f) RAND mesh METIS.

Figure 3.3: Examples of the different type of meshes we consider in our numerical experiments partitioned in 4×4 subdomains. The mesh in (a), (b) and (c) are partitioned by SQUARE, (d), (e) and (f) are done by using METIS.

local mesh discretization that is mirrored between the subdomains (Figure 3.3 (a), (b) and (c)). The second one is the METIS decomposition [58], that for a fixed mesh produce a general decomposition into subdomains by graph partitioning, in this case we do not have straight macro edges (Figure 3.3 (d), (e) and (f)). We use the VEM discretization with degree $k = 2$, that means having polynomials of degree 2 on the boundary of each element for the velocity and piecewise constant functions for the pressure. We underline the fact that we would have obtained the same behavior, both in terms of number of iterations and spectral condition number, also in the case of imposing using the reduced scheme, because the interface problem and the preconditioner are exactly the same due to the decomposition technique used in (3.2) and (3.3).

In the tables we use the following notation: SQUARE (os SQ) and METIS (or MET) = type of the mesh partitioning, nSub = number of subdomain, H/h = ratio between the diameter of subdomain and the element, it = iteration count (GMRES or PCG), λ_{min} = minimum eigenvalue of the preconditioned saddle point problem, λ_{max} = maximum eigenvalue of the preconditioned saddle point problem. When the BDDC preconditioner is used, we identify with BDDC (V) the coarse space generated by using as primal constraints the vertices of the subdomains and with BDDC (V+N) the coarse space generated by using as primal constraints the vertices of the subdomains plus the average of the normal component of the velocity.

SQUARE		GMRES	BDDC (V)			BDDC (V + N)		
	nSub	it	λ_{min}	λ_{max}	it	λ_{min}	λ_{max}	it
QUAD	4×4	57	N.A.	N.A	13	1.00	5.75	10
	8×8	203	N.A.	N.A	26	1.00	7.81	15
	12×12	309	N.A.	N.A	26	1.02	8.31	18
	16×16	436	N.A.	N.A	26	1.02	8.47	19
	20×20	551	N.A.	N.A	26	1.02	8.57	19
CVT	4×4	118	N.A.	N.A	18	1.00	5.67	13
	8×8	284	N.A.	N.A	26	1.00	7.74	17
	12×12	385	N.A.	N.A	26	1.02	8.23	19
	16×16	507	N.A.	N.A	26	1.02	8.42	19
	20×20	625	N.A.	N.A	27	1.01	8.52	19
RAND	4×4	110	N.A.	N.A	18	1.00	5.43	13
	8×8	298	N.A.	N.A	26	1.01	7.49	17
	12×12	401	N.A.	N.A	27	1.02	7.95	19
	16×16	533	N.A.	N.A	28	1.02	8.19	19
	20×20	663	N.A.	N.A	31	1.01	8.38	19

Table 3.1: Weak scaling with subdomain ratio $H/h = 8$ and SQUARE partitioning.

3.4.1.1 Homogeneous fluid

We perform two common scaling test in the domain decomposition and an optimality test. Here, we consider an homogeneous fluid with $\nu_i = 1$ for all the subdomains Ω_i using the same test problem as in 2.3.

Weak Scaling. We fix the ratio $H/h = 8$ for the local problems and we increase the number of subdomains. Table 3.1 reports the results for the different meshes partitioned with the SQUARE technique. As expected, for the non-preconditioned GMRES, we observe that the iteration counts grow when the number of subdomains increases. The BDDC(V) solver appears to be scalable, since the iterations remain bounded when the number of subdomains increase, with a worse behavior for the RAND meshes. In this case we do not fulfill the no-net-flux assumption, so we are not able to have an estimate of the eigenvalue since the preconditioned problem is not symmetric and positive definite. Using the BDDC(V+N) solver, both the assumptions are satisfied, therefore the system is symmetric and positive definite and we are able to give an estimate of the eigenvalues. The results confirm the theoretical estimates, since both the condition number and the number of iterations are independent of number of subdomains. Moreover we can see that the number of iterations and the spectral estimates are independent on the type of the mesh used for the local problems (Figure 3.5). Table 3.2 reports the results again for a weak scaling test using METIS as subdomain partition. The behavior of the GMRES is the same as before, but we can see that also for the BDDC(V) solver the number of iterations increases when increasing the number of subdomains. This also happens for the BDDC(V+N) with, in general, a milder effect. This fact is due to the irregularity of the boundary of the subdomains that the METIS partitioning technique creates. Even though the lower bound for the eigenvalues is respected, the upper one slightly increases, since the different shape of the partition reflects into different sparsity pattern of the matrices. The high spectral number for the QUAD and RAND mesh with

	METIS	GMRES	BDDC (V)			BDDC (V + N)		
	nSub	it	λ_{min}	λ_{max}	it	λ_{min}	λ_{max}	it
QUAD	4×4	217	N.A.	N.A	30	1.00	11.70	20
	8×8	483	N.A.	N.A	42	1.00	12.14	22
	12×12	767	N.A.	N.A	52	1.00	24.92	26
	16×16	1 008	N.A.	N.A	55	1.00	22.01	27
	20×20	1 299	N.A.	N.A	78	1.00	233.35	40
CVT	4×4	236	N.A.	N.A	31	1.00	10.16	18
	8×8	592	N.A.	N.A	38	1.00	10.17	20
	12×12	952	N.A.	N.A	41	1.00	17.12	21
	16×16	1 273	N.A.	N.A	44	1.00	16.41	22
	20×20	1 449	N.A.	N.A	46	1.00	17.34	23
RAND	4×4	252	N.A.	N.A	35	1.00	21.77	20
	8×8	675	N.A.	N.A	47	1.00	30.31	28
	12×12	1 042	N.A.	N.A	56	1.00	36.20	32
	16×16	1 518	N.A.	N.A	63	1.00	45.19	34
	20×20	1 914	N.A.	N.A	63	1.00	71.26	41

Table 3.2: Weak scaling with subdomain ratio $H/h = 8$ and METIS partitioning.

	SQUARE	GMRES	BDDC (V)			BDDC (V + N)		
	H/h	it	λ_{min}	λ_{max}	it	λ_{min}	λ_{max}	it
QUAD	8	57	N.A.	N.A.	13	1.00	5.75	10
	16	79	N.A.	N.A.	15	1.00	7.20	11
	24	94	N.A.	N.A.	17	1.00	8.09	11
	32	107	N.A.	N.A.	17	1.00	8.76	12
CVT	8	118	N.A.	N.A.	18	1.00	5.67	13
	16	153	N.A.	N.A.	19	1.00	7.19	14
	24	157	N.A.	N.A.	20	1.00	8.07	15
	32	187	N.A.	N.A.	21	1.00	8.76	15
RAND	8	110	N.A.	N.A.	18	1.00	5.43	13
	16	169	N.A.	N.A.	20	1.00	7.13	14
	24	176	N.A.	N.A.	22	1.00	8.00	15
	32	194	N.A.	N.A.	23	1.00	8.45	15

Table 3.3: Optimality test with respect to mesh ratio H/h. $nSub = 4 \times 4$ and SQUARE partition.

$nSub = 20 \times 20$ is due to the fact that in this case METIS produces non convex subdomains. Increasing the ratio H/h will clearly reduce the effect caused by the irregular boundaries.

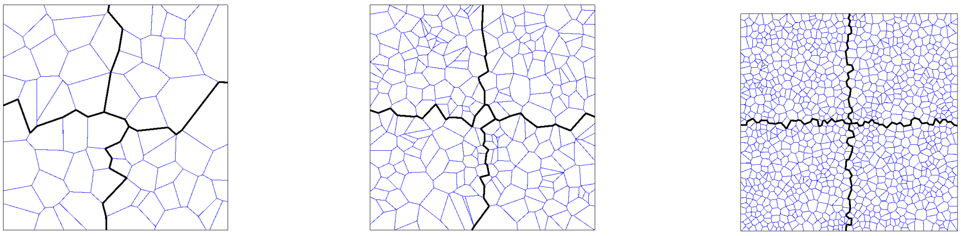
Optimality test. We now perform an optimality test with respect to the mesh size: we fix the number of subdomains at 16 and we increase the local ratio H/h. The results for the SQUARE partitioning are reported in Table 3.3. The GMRES, of course is not optimal, while the BDDC (V) seems to have a quasi-optimal behavior since the iteration count exhibits a logarithmic growth. Also the BDDC (V+N) shows a quasi-optimal behavior since both the iteration count and the spectral estimates respect again, in line with the theory, a logarithmic growth (Figure 3.6). In Table 3.4 we reported the results for the METIS partitioning. As we expect, the iteration count and, when possible, the spectral estimates are greater than in Table 3.3, due to the irregularity of the subdomains. One can note that the iteration count and the spectral estimates do not grow when increasing the coefficient ratio H/h. However, this does

	METIS	GMRES	BDDC (V)			BDDC (V + N)		
	H/h	it	λ_{min}	λ_{max}	it	λ_{min}	λ_{max}	it
QUAD	8	217	N.A.	N.A.	30	1.00	11.70	20
	16	293	N.A.	N.A.	29	1.00	10.79	18
	24	300	N.A.	N.A.	27	1.00	12.12	16
	32	358	N.A.	N.A.	30	1.00	10.53	19
CVT	8	236	N.A.	N.A.	31	1.00	10.16	18
	16	304	N.A.	N.A.	32	1.00	8.67	18
	24	379	N.A.	N.A.	32	1.00	17.69	19
	32	376	N.A.	N.A.	31	1.00	8.53	18
RAND	8	252	N.A.	N.A.	34	1.00	12.44	21
	16	344	N.A.	N.A.	34	1.00	13.93	21
	24	405	N.A.	N.A.	34	1.00	9.68	18
	32	420	N.A.	N.A.	34	1.00	12.04	21

Table 3.4: Optimality test with respect to mesh ratio H/h. $nSub = 4 \times 4$ and METIS partition.

not mean that the solver is optimal, but it is consequence of the evolution of the boundary when increasing the mesh refinement. As shown in Figure 3.4, we can see that when we increase the ratio H/h, the boundary of the subdomains appear to be more regular. When increasing the mesh refinement, the quasi-optimality would increase the iteration count and the condition number but the effect of the irregularity of the boundary is reduced. This explains why the do not see any growth in this test.

Strong scaling. We conclude this group with a strong scaling test. We fixed the global size problem with 16000 elements for all the three different meshes and we solve the global saddle point problem increasing the number of the subdomains. In this case we choose the most natural partitioning technique using SQUARE partition for the QUAD mesh and the METIS for the CVT and RAND ones. Looking at the estimate of Theorem 3.3.1 we expect that the condition number will slightly decrease when increasing the number of the subdomains, since H decrease while h is fixed. We can observe this fact for the QUAD mesh in Table 3.5 where, for the BDDC (V+N) solver, the iteration count remain fixed and the largest eigenvalue slightly decrease. Similar result



(a) H/h = 4

(b) H/h = 8

(c) H/h = 16

Figure 3.4: The different shape of the boundaries for $2x2$ subdomains when increasing the ratio H/h.

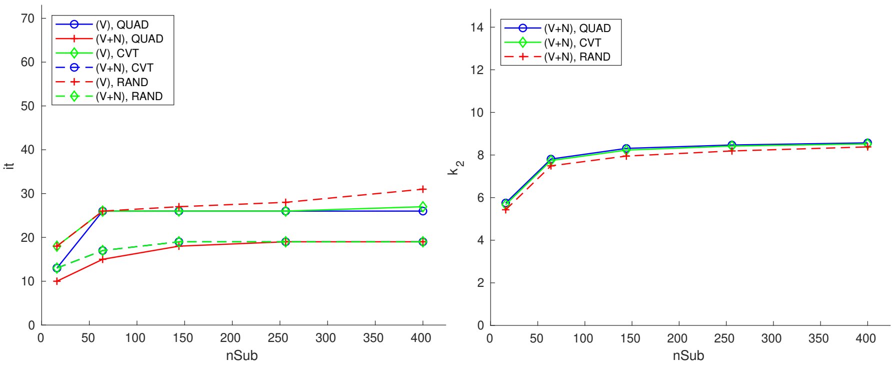


Figure 3.5: Weak scaling. It (left) and k_2 (right) with the BDDC preconditioners for different coarse space and meshes discretizations. Part = SQUARE, $H/h = 8$.

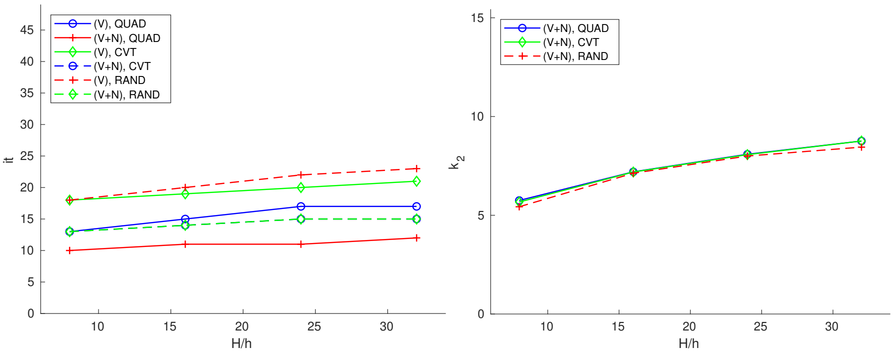


Figure 3.6: Optimality test. It (left) and k_2 (right) with the BDDC preconditioners for different coarse space and meshes discretizations. Part = SQUARE, $nSub = 4 \times 4$.

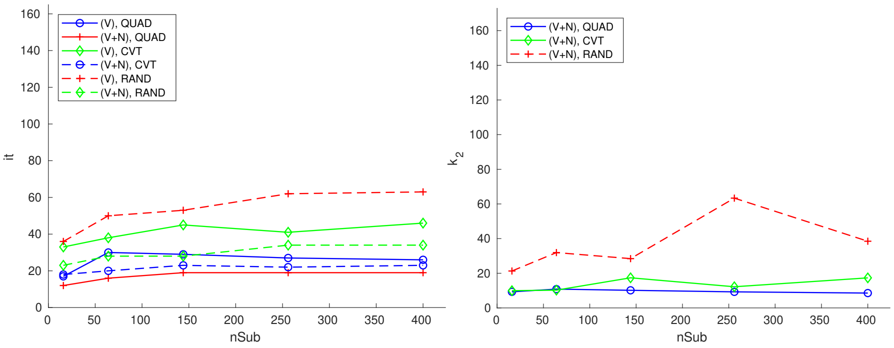


Figure 3.7: Strong scaling. It (left) and k_2 (right) with the BDDC preconditioners for different coarse space and meshes discretizations. Part = SQUARE, $H/h = 8$.

can be observed also for the other two type of meshes where the increasing irregularity of the boundaries hides this effect (Figure 3.7).

		GMRES	BDDC (V)			BDDC (V + N)		
	nSub	it	λ_{min}	λ_{max}	it	λ_{min}	λ_{max}	it
QUAD	4×4	117	N.A.	N.A.	17	1.00	9.28	12
	8×8	318	N.A.	N.A.	30	1.00	10.85	16
	12×12	421	N.A.	N.A.	29	1.00	10.16	19
	16×16	479	N.A.	N.A.	27	1.00	9.27	19
	20×20	551	N.A.	N.A.	26	1.00	8.57	19
CVT	4×4	432	N.A.	N.A.	33	1.00	9.94	18
	8×8	815	N.A.	N.A.	38	1.00	10.26	20
	12×12	1070	N.A.	N.A.	45	1.00	17.38	23
	16×16	1336	N.A.	N.A.	41	1.00	12.21	22
	20×20	1449	N.A.	N.A.	46	1.00	17.34	23
RAND	4×4	512	N.A.	N.A.	36	1.00	21.30	23
	8×8	984	N.A.	N.A.	50	1.00	31.93	28
	12×12	1347	N.A.	N.A.	53	1.00	28.44	28
	16×16	1605	N.A.	N.A.	62	1.00	63.35	34
	20×20	1914	N.A.	N.A.	63	1.00	38.48	34

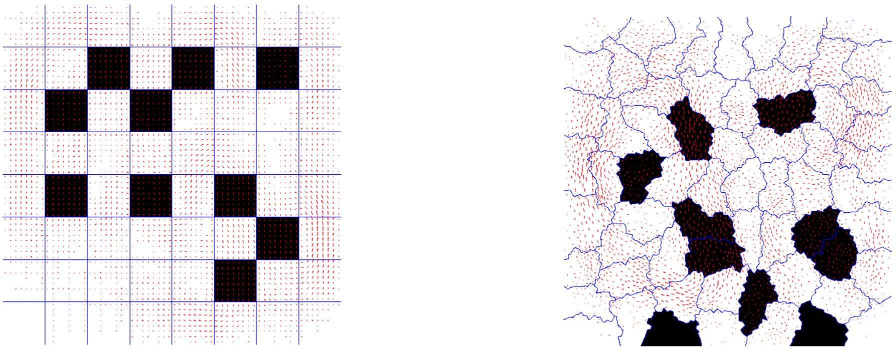
Table 3.5: Strong scaling. $nEl = 25600$, QUAD (SQUARE), CVT and RAND (METIS).

3.4.1.2 Coefficient jumps

In order to verify the robustness of the method with respect to strong variations of the viscosity ν, we now consider problems with ν discontinuous across the interfaces Γ. For a random set of subdomain $\mathcal{L}$ we choose the coefficient $\nu = 1e3$, while for the others we choose $\nu = 1e-3$, applying a jump of 6 orders of magnitude. We then solve the lid-driven cavity benchmark problem, using as load term $\mathbf{f}$ a force that should represent the gravity that take down the high viscosity materials:

$$\mathbf{f} = \begin{cases} \beta & \text{if} \quad x \in \Omega_i, \text{with} \quad i \in \mathcal{L} \\ 0 & \text{otherwise} \end{cases}$$

with $\beta = 10$. In Table 3.6, we report the iteration counts and the spectral estimates for the BDDC (V+N) solver varying the local mesh, the partitioning and the scaling technique used. We keep fixed the number of the subdomains at 64. We can observe that the multiplicity-scaling, even if respect the lower bound meaning that the problem is symmetric and positive definite, it is not able to control neither the iteration nor the upper bound. The ν-scaling exhibits a good behavior, while the deluxe scaling has the better one. In Table 3.7 we reported the same test for 256 subomains. We see that again the multiplicity-scaling is not able to handle the jumps, while the ν and the deluxe scaling are able to keep low the number of the iteration and the spectral estimates. Further consideration and a more exhaustive study, varying the viscosity coefficients will be done in the next Chapter.



(a) SQUARE partitioning

(b) METIS partitioning

Figure 3.8: Examples of jump configuration. In black the subdomains with $\nu = 1e3$, in withe the subdomains with $\nu = 1e-3$. The red arrows show the solution for velocity flow of the lid-driven-cavity problem.

BDDC (V+N)		mul-scaling			ρ-scaling			del-scaling		
Part	Mesh	it	λ_{min}	λ_{max}	it	λ_{min}	λ_{max}	it	λ_{min}	λ_{max}
SQ	QUAD	1.00	NaN	742	1.00	6.22	17	1.00	5.85	16
	CVT	1.00	NaN	732	1.00	6.17	17	1.00	5.87	16
	RAND	1.00	NaN	713	1.00	6.19	17	1.00	6.00	16
MET	QUAD	1.00	NaN	1003	1.00	15.91	23	1.00	4.34	14
	CVT	1.00	NaN	1002	1.00	14.32	21	1.00	5.10	15
	RAND	1.00	NaN	2082	1.00	29.82	29	1.00	5.35	26

Table 3.6: Coefficient jumps. Comparison of the three different type of scaling for the BDDC (V+N) coarse space with 8×8 subdomains.

BDDC (V+N)		mul-scaling			ρ-scaling			del-scaling		
Part	Mesh	it	λ_{min}	λ_{max}	it	λ_{min}	λ_{max}	it	λ_{min}	λ_{max}
SQ	QUAD	1.00	NaN	569	1.00	9.10	19	1.00	6.84	18
	CVT	1.00	NaN	566	1.00	9.20	20	1.00	6.79	18
	RAND	1.00	NaN	568	1.00	9.48	20	1.00	6.74	18
MET	QUAD	1.00	NaN	1810	1.00	17.04	25	1.00	11.72	19
	CVT	1.00	NaN	1501	1.00	15.93	23	1.00	9.45	20
	RAND	1.00	NaN	3577	1.00	33.13	33	1.00	39.22	28

Table 3.7: Coefficient jumps. Comparison of the three different type of scaling for the BDDC (V+N) coarse space with 16×16 subdomains.

3.4.2 Numerical Results in 3D

In this Section, we report the numerical results to validate our theoretical estimates of the BDDC algorithm for solving the 3D Stokes model problem (1.9). We solve a problem on the unit cube $[0,1]^3$ with a known solution as in 2.3, imposing Neumann boundary conditions on two faces of the cube and homogeneous Dirichlet boundary conditions on the other ones. Again we use the

BDDC method as a preconditioner for system the (3.9), which is solved by the CG method with a stopping criterion of a 10^{-8} reduction of the l^2-norm of the relative residual. Here, we consider three types of meshes: hexahedral (Cube), octahedral (Octa), and Voronoi (CVT), see Figure 3.3. Since we need the deluxe scaling to construct our adaptive coarse space in three dimension in the next Chapter, we do the scaling and optimality test directly using this technology. In the tables we use the following notation: procs = number of CPUs, nEl = number of VEM elements, k = degree of VEM approximation, nDoFs = number of DoFs, N_Π = number of primal constraints, it = iteration count (PCG), k_2 = conditioning number, T_{ass}= time to assemble the stiffness matrix and the right-hand side, T_{sol} = time to solve the interface saddle point problem and S^{id} = ideal speed up, S_p = parallel speed up.

Strong Scaling. We first study the strong scalability of our solvers. We keep fixed the global number of the DoFs and the degree of the VEM approximation k, while we increase the number of processors from 4 to 256. We consider CUBE mesh with 408 243 DoFs, a CVT mesh with 311 155 DoFs and an OCTA mesh with 549 939 DoFs for the degree $k = 2$. While a CUBE mesh with 589 344 DoFs, a CVT mesh with 311 155 DoFs and an OCTA mesh with 398 260 DoFs for the degree $k = 3$. We recall that denoting by p the number of processors, the parallel speedup is defined as:

$$S_p := \frac{\text{CPU time with 4 processors}}{\text{CPU time with p processors}}.$$

In Table 3.8, we report the results related to the three polyhedral meshes with $k = 2, 3$. In Figure 3.9, we plot the number of iterations and the parallel speedup for the case $k = 3$. We observe that the CPU time T_{ass}, needed to assemble the stiffness matrix and the right-hand-side is scalable, with a speedup very close to the ideal ones. The BDDC method results scalable since the number of CG iterations remains bounded and the solution time decreases as the number of the processors increases. We note that, as usual in a strong scalability study, the parallel speedup does not increase when the number of processors is large with respect to the local size of the problems. This is because communication time overcomes the time for computation.

Optimality with respect to the mesh size. We now perform an optimality test with respect to the mesh size: we keep fixed the number of processors at 32 and we increase the number of DoFs, maintaining the degree of the VEM discretization $k = 2$. The results are reported in Table 3.9. We observe that the solver has a quasi-optimal behavior irrespective of the type of polyhedral mesh considered since both the iteration count and the condition number exhibit a logarithmic growth as predicted by Theorem 3.3.1.

Optimality with respect to the polynomial degree. In this test, we study the robustness of our preconditioners when increasing the polynomial degree of the VEM discretization. The tests are performed keeping fixed the number of processors again at 32 and the mesh size. The results reported in Table 3.10 show that the BDDC solver exhibits a slight increase of the condition number and iterations count when the degree k increases.

			$k = 2$					$k = 3$				
	procs	S^{id}	T_{ass}	S_p	it	T_{sol}	S_p	T_{ass}	S_p	it	T_{sol}	S_p
CUBE	4		1 299		21	132		3346		26	318	
	8	2	871	1.5	31	62	2.1	1821	1.8	41	193	2.4
	16	4	418	3.1	33	35	3.8	745	4.4	49	92	3.5
	32	8	202	6.4	55	16	8.3	467	7.2	72	37	8.6
	64	16	105	12.4	58	6	22.0	245	13.7	86	29	11.0
	128	32	53	24.5	61	5	26.4	140	23.9	98	13	24.5
	256	64	27	48.1	62	5	26.4	83	40.3	99	11	28.9
CVT	4		2 456		38	711		2 812		41	289	
	8	2	1 423	1.7	41	248	2.9	1 450	1.9	43	103	2.8
	16	4	816	3.0	42	174	4.1	820	3.4	44	70	4.1
	32	8	400	6.1	46	66	10.8	460	6.1	45	20	14.0
	64	16	216	11.4	33	21	33.9	264	10.6	37	6	44.2
	128	32	126	19.5	32	9	79.0	179	15.7	39	5	50.7
	256	64	72	34.1	32	8	93.2	175	16.0	37	6	44.2
OCTA	4		3 225		32	59		3 889		30	102	
	8	2	1 637	2.0	32	59	2.6	1 922	2.0	39	44	2.3
	16	4	838	3.9	41	36	4.2	982	4.0	52	34	3.0
	32	8	434	7.4	57	16	9.5	497	7.8	77	15	6.7
	64	16	218	14.8	64	7	21.7	261	14.9	90	7	14.2
	128	32	116	27.8	76	5	30.4	133	29.3	92	7	14.7
	256	64	56	57.6	70	7	21.7	66	59.1	92	7	15.0

Table 3.8: Strong Scaling.nEl for CUBE = 13 824 , CVT = 4 000 and OCTA = 15 552 meshes for $k = 2$. nEl for CUBE = 8 000 , CVT = 1 000 and OCTA = 4 608 meshes for $k = 3$.

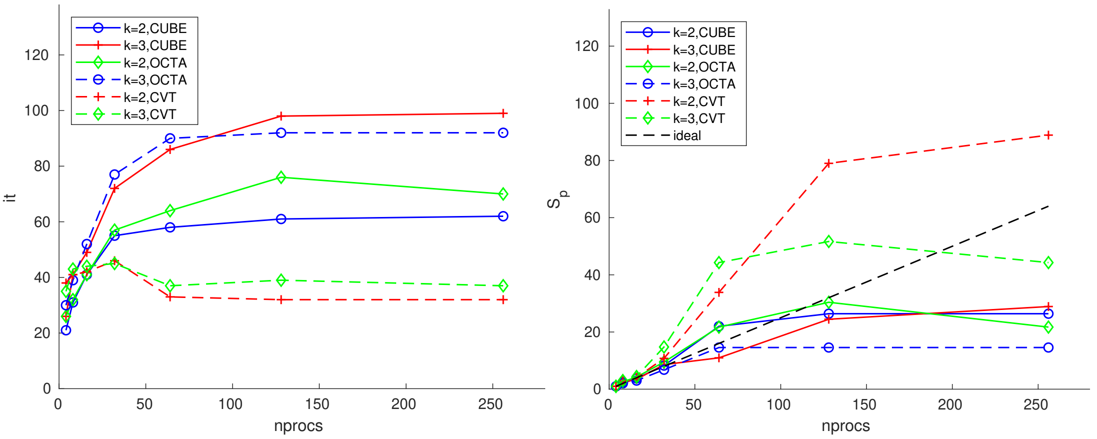


Figure 3.9: Test 1: strong scaling. Iteration of the CG (left) and parallel speedup (right) with the BDDC preconditioners for $k = 2, 3$ and different meshes discretizations.

Table 3.9: Optimality test with respect to the mesh size. $k = 2$ and procs = 32.

nEl	nDoFs	N_Π	it	k_2
4 096	124 195	125	44	44.67
8 000	239 763	152	50	54.53
13 824	408 243	125	54	63.98
21 952	643 387	152	58	72.37
32 768	954 947	125	62	80.63

(a) CUBE meshes.

nEl	nDoFs	N_Π	it	k_2
125	8 945	633	22	7.7
1 000	76 051	801	32	16.09
2 000	154 067	829	40	28.35
4 000	311 155	836	46	37.42
8 000	626 455	833	56	53.10

(b) CVT meshes.

nEl	nDoFs	N_Π	it	k_2
576	22 035	125	36	26.15
4 608	166 179	125	40	37.19
9 000	320 763	157	57	47.31
15 552	549 939	125	57	70.78
30 375	1 065 693	474	57	46.62

(c) OCTA meshes.

Table 3.10: Optimality Test Increasing the polynomial degree k. procs = 32 and nEl for CUBE = 512, CVT = 125 and OCTA = 576.

k	nDoFs	N_Π	it	k_2
2	16 787	125	34	23.86
3	40 667	125	49	46.34
4	76 387	125	64	80.67

(a) CUBE meshes.

k	nDoFs	N_Π	it	k_2
2	8 945	633	22	7.7
3	19 256	633	31	14.89
4	33 487	633	46	28.86

(b) CVT meshes.

k	nDoFs	N_Π	it	k_2
2	22 035	125	36	26.15
3	52 251	125	55	53.66
4	96 675	125	75	91.81

(c) OCTA meshes.

Chapter 4

Enriching the coarse space

In this chapter we study the behavior of the BDDC algorithm applied to the Stokes equations with heterogeneous viscosity. In the previous Chapter, we saw that our preconditioner is robust with respect to coefficient jumps that are constant on the subdomains. Here, we study its performance when the jumps occurs between the elements. Moreover, we show numerical tests for a practical application, where the viscosity function is represented by a continuous function that exhibits sharp gradients to represent high viscosity materials present in the fluid. We introduce two adaptive techniques to enrich the coarse space in two dimensions and one for the three dimensional case. We also present an heuristic and economic approach, currently under study, to enrich the coarse space without solving auxiliary eigenvalue problems. The study related to the two dimensional framework, is currently under development in collaboration with Prof. Axel Klawonn and Dr. Martin Lanser from the University of Cologne [20]. For what concern the three dimensional extension we work together with Prof. Franco Dassi from University of Milano-Bicocca, and Dr. Stefano Zampini from King Abdullah University of Science and Technology (KAUST) [19].

4.1 Adaptive coarse spaces in two dimensions

We start presenting two adaptive techniques to enrich the coarse space in two dimensions. The idea is to solve generalized eigenvalue problems defined on each subdomain macro edge $\mathcal{E}$ and then construct an enriched primal space such that the condition number of the preconditioned system will be bounded from above by a selected $TOL \in [1, \infty)$ times a constant C independent of h, H and N:

$$\kappa_2(M^{-1}S) \leq C\,\mathrm{TOL}.$$

Both approaches are based on a localization of the E_D, or P_D estimate (see eq. (3.24) and following, for theie definitions), since the two operators have the same $\widetilde{S}_\Gamma$-norm [73]. The estimate before is then obtained by applying Theorem 3.3.1, where the Assumption 2 is substituted by the adaptive operator estimate.

Remark 4.1.1. It is important to emphasize that when using a transformation of basis approach (see Appendix A), the adaptive techniques to enrich the coarse space have to be applied after having imposed the no-net-flux condition. In fact we can not ensure a priori that this constraint will be detect by the solution of the eigenvalue problems.
The algorithm in our 2D implementation is the following:

- apply the transformation of basis to impose the no-net-flux condition;
- solve the eigenvalue problems and compute the constraints in the transformed space;
- transform back the coefficient into the original space of function;
- construct the orthonormal transformation imposing the no-net-flux condition and the additional constraints using a QR decomposition.

Since these transformations modify the sparsity pattern of the matrices, when the size of the problem is too large, for example in our three dimensions numerical test, an approach that avoid the explicit change of the basis is needed [101].

4.1.1 First coarse space

The first approach, designed specifically with the deluxe scaling, was originally introduced in [82] and already successfully used for VEM discretization in [19, 43]. A variant, that allows to use any kind of scaling, has been presented in [63] for FEM discretizations.

4.1.1.1 Generalized eigenvalue problem

The proof of the adaptive coarse space that we will present here has been given, for BDDC and finite element discretizations, in Section 4 of [96] and in [64]. Scrolling the second proof, we see that we can reproduce the same proof also for virtual element discretization paying attention to use the VEM edge estimates. We limit here to recall the construction of this coarse space. Let $\mathcal{E}_{ij}$ be the open macro edge shared by two subdomains Ω_i and Ω_j, and we settle in a deluxe scaling context. Thus, for $k = i, j$, defining the matrices $S^{(k)}_{\mathcal{E}_{ij}\mathcal{E}_{ij}}$ and $\widetilde{S}^{(k)}_{\mathcal{E}_{ij}\mathcal{E}_{ij}}$ as in (3.18) and (3.19), the deluxe scaling matrices are given by

$$D^{(k)}_{\mathcal{E}_{ij}} := \left(\widetilde{S}^{(i)}_{\mathcal{E}_{ij}\mathcal{E}_{ij}} + \widetilde{S}^{(j)}_{\mathcal{E}_{ij}\mathcal{E}_{ij}}\right)^{-1} \widetilde{S}^{(k)}_{\mathcal{E}_{ij}\mathcal{E}_{ij}} \quad \text{for} k = i, j. \tag{4.1}$$

We also define the product $A : B = (A^{-1} + B^{-1})^{-1}$, and we solve the eigenvalue problem

$$S^{(i)}_{\mathcal{E}_{ij}} : S^{(j)}_{\mathcal{E}_{ij}} \mathbf{x}_m = \mu_m \widetilde{S}^{(i)}_{\mathcal{E}_{ij}} : \widetilde{S}^{(j)}_{\mathcal{E}_{ij}} \mathbf{x}_m. \tag{4.2}$$

We select the $\mathbf{x}_m$, $m = 1,..,k$ for which $\mu_m <$ TOL and we compute the coefficient c_{ij} for the constraints as

$$c_{ij} := \widetilde{S}^{(i)}_{\mathcal{E}_{ij}} : \widetilde{S}^{(j)}_{\mathcal{E}_{ij}}\, \mathbf{x}_m.$$

We give an other approach for this coarse space for generic scaling matrices $D^{(k)}_{\mathcal{E}_{ij}}$ with $k = i, j$, that satisfy $D^{(i)}_{\mathcal{E}_{ij}} + D^{(j)}_{\mathcal{E}_{ij}} = I$. We solve the generalized eigenvalue problem:

$$S^{(i)}_{\mathcal{E}_{ij}} : S^{(j)}_{\mathcal{E}_{ij}}\, \mathbf{x}_m = \mu_m \big(D^{(j)\,T}_{\mathcal{E}_{ij}} S^{(i)}_{\mathcal{E}_{ij},0} D^{(j)}_{\mathcal{E}_{ij}} + D^{(j)\,T}_{\mathcal{E}_{ij}} S^{(j)}_{\mathcal{E}_{ij},0} D^{(j)}_{\mathcal{E}_{ij}} \big)\, \mathbf{x}_m, \tag{4.3}$$

and we select the $\mathbf{x}_m$, $m = 1,..,k$ for which $\mu_m <$ TOL and we compute the coefficient c_{ij} for the constraints as

$$c_{ij} := \big(D^{(j)\,T}_{\mathcal{E}_{ij}} S^{(i)}_{\mathcal{E}_{ij},0} D^{(j)}_{\mathcal{E}_{ij}} + D^{(j)\,T}_{\mathcal{E}_{ij}} S^{(j)}_{\mathcal{E}_{ij},0} D^{(j)}_{\mathcal{E}_{ij}} \big)\, \mathbf{x}_m.$$

Remark 4.1.2. It is easy to see that using $D^{(k)}_{\mathcal{E}_{ij}}$ defined as in (4.1), the two eigenvalue problems (4.2) and (4.3) are equivalent.

Theorem 4.1.1. *Let the assumptions of (3.3.1) hold. In addition, let the primal space contain all the eigenvectors of the generalized eigenvalue problems (4.2), whose corresponding eigenvalues are lower than 1/TOL for each $\mathcal{E} \in \Gamma$. Then, the preconditioned BDDC operator for (2.77) satisfies*

$$\kappa_2(M^{-1}S) \le C\; TOL, \qquad \forall\; TOL \in [1, \infty), \tag{4.4}$$

where C is independent of $N, h,$ and H, but it depends on the number of the macro edges $N_{\mathcal{E}}$.

4.1.2 Second coarse space

The second coarse space has been successfully used in FETI-DP and BDDC in [65, 74, 77, 89], and firstly theoretically proved in [64]. It has been also recently extended to VEM discretization of diffusion and linear elasticity problems in [62].

4.1.2.1 Notation

As in the FETI-DP framework, we define the jump matrix $B = [B^{(1)} \dots B^{(N)}]$ that connects the dual DoFs on the interface such that $B\mathbf{u}_\Gamma = 0$ if $\mathbf{u}_\Gamma$ is continuous. For $\mathcal{E}_{ij}$ edge shared by Ω_i and Ω_j, we define the operator $B_{\mathcal{E}_{ij}} = [B^{(i)}_{\mathcal{E}_{ij}}\, B^{(j)}_{\mathcal{E}_{ij}}]$ as a submatrix of $[B^{(i)}\, B^{(j)}]$, that consists of a matrix with a one 1 and a one -1 for each rows. We also define their scaled version $B_{D,\mathcal{E}_{ij}} = [B^{(i)}_{D,\mathcal{E}_{ij}} B^{(j)}_{D,\mathcal{E}_{ij}}]$ obtained taking the same rows from $[B^{(i)} B^{(j)}] = [D^{(i)\,T} B^{(i)}\; D^{(j)\,T} B^{(j)}]$.
Let $S_{ij} := \begin{bmatrix} S^{(i)}_\Gamma & \\ & S^{(j)}_\Gamma \end{bmatrix}$ and the restricted version of the P_D operator $P_{D_{ij}} = B^T_{D,\mathcal{E}_{ij}} B_{\mathcal{E}_{ij}}$.
Defining by $\widetilde{\mathbf{V}}_{ij}$ the space of functions belonging to $\mathbf{V}^{(i)}_\Gamma \times \mathbf{V}^{(j)}_\Gamma$ and that are continuous in the primal nodes, we define by R^T_{ij} the restriction operator from $\mathbf{V}^{(i)}_\Gamma \times \mathbf{V}^{(j)}_\Gamma$ to $\widetilde{\mathbf{V}}_{ij}$. We introduce

Π_{ij}, the l_2-orthogonal projection from $\mathbf{V}_\Gamma^{(i)} \times \mathbf{V}_\Gamma^{(j)}$ to $\widetilde{\mathbf{V}}_{ij}$

$$\Pi_{ij} := R_{ij}(R_{ij}^T R_{ij})^{-1} R_{ij}^T$$

and $\bar{\Pi}_{ij}$, another orthogonal projection from $\mathbf{V}_\Gamma^{(i)} \times \mathbf{V}_\Gamma^{(j)}$ to Range $(\Pi_{ij} S_{ij} \Pi_{ij} + \sigma(I - \Pi_{ij}))$, where σ is a suitable positive constant, usually taken as the maximum of the entries of the diagonal S_{ij}

$$\bar{\Pi}_{ij} := I - c\, c^T$$

with $c := \frac{\widehat{c}}{\|\widehat{c}\|}$, $c = (1, ..., 1)^T$ (see [64] Section 5 for further details on these operators).

4.1.2.2 Generalized eigenvalue problem

We then solve the eigenvalue problem

$$\bar{\Pi}_{ij}\Pi_{ij} P_{D_{ij}}^T S_{ij} P_{D_{ij}} \Pi_{ij} \bar{\Pi}_{ij}\, \mathbf{x}_m = \mu_m \big(\bar{\Pi}_{ij}\big(\Pi_{ij} S_{ij} \Pi_{ij} + \sigma(I - \Pi_{ij})\big)\bar{\Pi}_{ij} + \sigma\big(I - \bar{\Pi}_{ij}\big)\big)\, \mathbf{x}_m,$$

and we select the $\mathbf{x}_m$, $m = 1, .., k$ for which $\mu_m \geq$ TOL and we compute the coefficient constraints as c_{ij}^m

$$c_{ij}^m := B_{D_{ij}}^T S_{ij} P_{D_{ij}}\, \mathbf{x}_m.$$

Theorem 4.1.2. *Let the assumptions of* (3.3.1) *hold. In addition, let the primal space contain all the eigenvectors of the generalized eigenvalue problems* (4.2)*, whose corresponding eigenvalues are greater than TOL for each $\varepsilon \in \Gamma$. Then, the preconditioned BDDC operator for* (2.77) *satisfies*

$$\kappa_2(M^{-1}S) \leq C\; TOL, \qquad \forall\; TOL \in [1, \infty), \tag{4.5}$$

where C is independent of $N, h,$ and H, but it depends on the number of macro edges $N_{\mathcal{E}}$.

A complete proof for FETI-DP and finite element discretizations has been given in [64]. Since the interface consists of line segments in both cases, even if the discretization is performed by finite element or virtual element the discrete trace spaces can be constructed in the same way. Moreover, the kernels of the Schur complements are identical regardless if the local discretization is obtained using finite or virtual elements. We do not show the details here, but the proof for the variant using virtual elements turns out to be analogous.

Remark 4.1.3. The eigenvalue problem of this coarse space is larger, since the size of the eigenvalue problem is determined by the number of DoFs of $\mathbf{V}_\Gamma^{(i)} \times \mathbf{V}_\Gamma^{(j)}$, while in the previous algorithm it is determined by the number of DoFs on an edge $\mathcal{E}_{ij}$. However, the advantage of this approach is that it produce sparser matrices. In fact, the left-hand side is not dense because of the structure of the local jump operator and the right-hand side consists of two dense blocks and two zero blocks in the dual part.

4.1.3 Frugal approach

We recall here an heuristic approach to enrich the coarse introduced in [56] and already used for the VEM for stationary diffusion and linear Elasticity in [62]. Since in general the condition number is determined by few large eigenvalue, the idea consists of building constraints for each edge of the interface Γ, without the need to solve an eigenvalue problems. We now try to define frugal constraints for virtual element discretizations for the Stokes equations. When applying the BDDC method to the Stokes problem in two dimensions, we need three constraints for each edge to account for the three (linearized) rigid-body motions, like in the linear elasticity case. Given two subdomains Ω_l, $l = i, j$, with diameter H_l, the kernel of the strain tensor $\boldsymbol{\varepsilon}$ is given by two translations and one rotation:

$$\mathbf{r}_1 := \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{r}_2 := \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{r}_3 := \frac{1}{H_l} \begin{bmatrix} x_2 - \widehat{x}_2 \\ -x_1 + \widehat{x}_1 \end{bmatrix}, \tag{4.6}$$

where $\widehat{\mathbf{x}} \in \Omega_l$ is the center of the rotation. Differently from the approach in [62], we do not rescale the rigid body modes and we simply define the "approximate" eigenvector $v(\mathbf{x})^{(m,l)}_{\mathcal{E}_{ij}}$:

$$v(\mathbf{x})^{(m,l)}_{\mathcal{E}_{ij}} := \begin{cases} r(\mathbf{x})^{(l)}_m, & \mathbf{x} \in \mathcal{E}_{ij}, \\ 0, & \mathbf{x} \in \partial\Omega_l \setminus \mathcal{E}_{ij}, \end{cases} \tag{4.7}$$

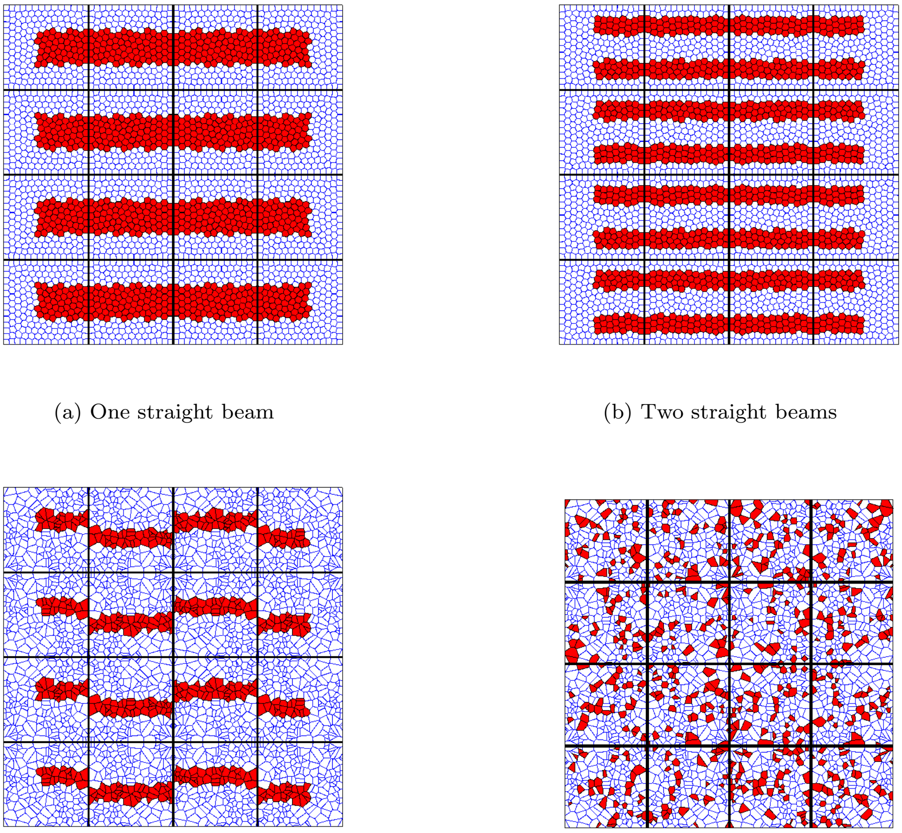
and we define $v(\mathbf{x})^{(m)\,T}_{\mathcal{E}_{ij}} := [v(\mathbf{x})^{(m,i)\,T}_{\mathcal{E}_{ij}}, -v(\mathbf{x})^{(m,j)\,T}_{\mathcal{E}_{ij}}]$.
The three frugal edge constraints are then obtained by:

$$c_{ij} := B^T_{D_{ij}} S_{ij} P_{D_{ij}}\, v(\mathbf{x})^{(m)\,T}_{\mathcal{E}_{ij}}. \tag{4.8}$$

Remark 4.1.4. This approach should work only when the number of the constraints per edge required for the second adaptive coarse space is less than 3 for each edge. We make note that, when using a transformation of basis approach, the rigid body modes have to be transformed too.

4.2 Numerical results in 2D

With our Matlab VEM implementation of degree $k = 2$, we considered a lid-driven cavity benchmark problem, where the heterogeneity is introduced to represent high viscosity material in the fluid. We consider two different type of heterogeneity: the classic one represented as high coefficient jumps between elements (Figure 4.1) and a second type that wants to represent a practical example where the viscosity is a continuous function that exhibits sharp gradients (Figure 4.2). In our experiments we used two different type of meshes (CVT and RAND), and as before, two type of mesh partitioning SQUARE and METIS.



(a) One straight beam

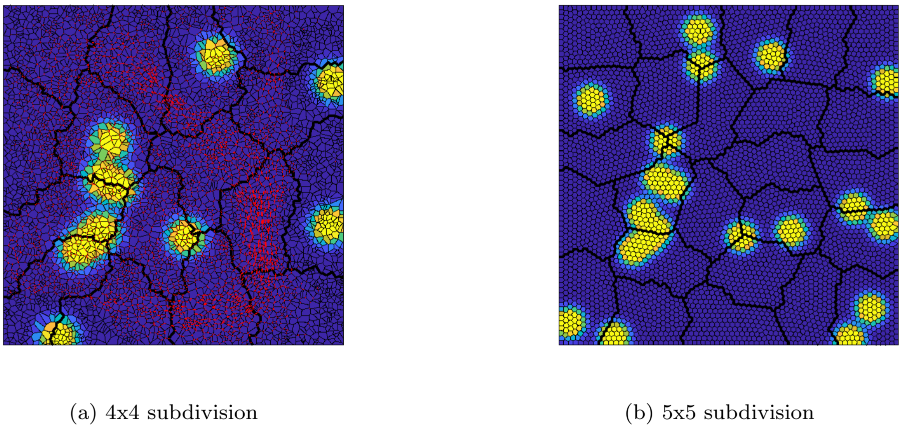
(b) Two straight beams

(c) Beams with offset

(d) Random distribution

Figure 4.1: Different coefficient distributions for a 4×4 domain decomposition with straight macro edges. The red and white elements have coefficient $\nu = 1e3$ and $\nu = 1e-3$ respectively.

We compare four different coarse space: the classic coarse space, the two adaptive coarse space and the frugal one; we also compare the ν and the deluxe scaling. In the tables we use the following notation: nSub = number of subdomains, nEl = number of elements, nSink = number of Sinkers, n_Π = number of primal constraints, it = iteration count (CG), k_2 = condition number, nnf = coarse space spanned by vertex constraints plus no-net-flux for each macro edge, $nnf+fru$ = coarse space spanned by vertex constraints plus no-net-flux and three frugal constraints for each macro edge, $First$ = coarse space spanned by vertex constraints, no-net-flux and first approach of the adaptive technique for each macro edge, $Second$ = coarse space spanned by vertex constraints, no-net-flux and second approach of the adaptive technique for each macro edge. Moreover ”x” in the tables means that the result is not available due to a non convergence of the method.



(a) 4x4 subdivision

(b) 5x5 subdivision

Figure 4.2: Two examples of METIS decomposition of a CVT and a RAND mesh for the Sinker problem. The yellow elements has a coefficient $\nu = 1e3$, while the blue $\nu = 1e-3$. In the left picture is represented the numerical solution for the velocity field.

4.2.1 Jumping Coefficients

We studied different configuration of coefficient jumps looking at straight beams configuration and beams with offset across the interface (Figure 4.1a). For this family of tests we considered $2\times2, 4\times4$ and 8×8 subdomains with a $\nu := 1e3$ for red elements and $\nu = 1e-3$ for the others and we fix the tolerance for both the adaptive coarse space to $TOL = 100$. The mesh is constructed choosing locally for each subdomain a CVT or a RAND mesh and then it is mirrored for each subdomain. In particular the local mesh for each subdomain are made with 200 elements, and the CVT ones are created using 500 Lloyd iterations. We remind that the coarse space always includes the vertices of each subdomain and an extra constraint for the no-net-flux condition for each macro edge, plus the constraints added by the adaptive or frugal technique.

4.2.1.1 Mirrored meshes

One Straight Beam. In Table 4.1 we show the iteration and condition numbers for the two different type of meshes and the two different types of scaling. For 2×2 subdomains the results are singular since all the coarse space work for both the scalings, in fact the adaptive coarse spaces do not need extra constraints. Increasing the number of subdomains, we can see that when using the multiplicity scaling the coarse space spanned just by the no-net-flux condition fails, while the results obtained with both the adaptive coarse spaces agree with the theoretical estimate, keeping the condition number under the fixed tolerance. The frugal coarse space provides good results. The iteration count and the condition numbers are very similar to the adaptive coarse spaces. We can note that sometimes its performance is even better and this is justified by the fact that this coarse space enrich all the edges.

Table 4.1: Coarse space comparison for one straight beam test (Figure 4.1a), increasing the number of subdomains.

		nnf			**nnf + fru**			**First**			**Second**		
CVT		n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**
mul	2x2	6	12	1.60e+1	18	14	16.04	8	12	16.01	6	12	16.01
	4x4	42	246	2.66e+5	114	13	4.66	90	13	5.59	78	13	5.59
	8x8	210	1757	8.06e+5	546	12	4.55	434	17	9.42	378	17	9.42
del	2x2	6	8	4.00	18	9	4.00	8	8	4.00	6	8	4.00
	4x4	42	121	3.30e+4	114	15	10.92	70	15	9.81	59	17	9.81
	8x8	210	760	1.06e+5	546	19	14.93	386	16	10.74	365	18	10.66
		nnf			**nnf + fru**			**First**			**Second**		
RAND		n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**
mul	2x2	6	12	17.12	18	15	17.01	8	13	17.11	6	12	17.11
	4x4	42	212	2.40e+5	114	12	5.90	90	12	5.16	78	12	5.16
	8x8	210	1729	7.24e+5	546	12	5.70	434	16	8.74	380	16	8.42
del	2x2	6	8	4.20	18	10	4.17	8	8	4.20	6	8	4.20
	4x4	42	124	3.23e+4	114	15	14.11	70	16	10.49	58	17	10.48
	8x8	210	778	1.01e+5	546	20	19.46	386	16	11.39	350	18	11.31

We discuss now the number of primal constraints needed for the different coarse space. Making reference to Figure 4.1a, we can easily count the number of constraints needed for all the coarse spaces. In particular, the basic coarse space is spanned by 18 vertices constraints plus 24 edge constraints due to the no-net-flux condition. The frugal coarse space needs to add 3 constraints for each edge, that means 72 constraints. While the first and second adaptive coarse space add respectively 4 and 3 extra edge constraints when a jump occurs, that means 48 and 36 constraints. The fact that the first coarse space need exactly one extra constraint for each edge where a jump occurs has already been observed in [64] for FEM discretizations. Since the extra constraints are needed only on the macro edges with coefficient jumps, in the frugal coarse space, one can decide to add the extra constraints only on specific macro edges where a jump occurs. Eventually, it is also possible detect if the constraints that have to be added belongs to a large eigenvalue, this could be done to limit the dimension of the coarse space.
When using the deluxe scaling, we have in general a good improvement in the performance of all the algorithms.

Two Straight beams. Table 4.2 reports the results for the test for the two straight beams configuration in Figure 4.1b. The behavior of both the adaptive coarse spaces is basically the same as before, the two algorithms are able to keep the condition numbers below the fixed tolerance. As expected, the poorest coarse space fails and also the frugal constraints do not give us good result. In fact, the number of constraints required from the adaptive coarse spaces is greater than the number of primal constraints added in the frugal space.

Beams with offset. Table 4.3 reports the results for the test with the beam with offset configurations in Figure 4.1c. The adaptive coarse spaces perform well, as expected while the basic coarse space fails. The frugal constraints instead, combined with the deluxe scaling gives results in line with the adaptive coarse spaces.

Table 4.2: Coarse space comparison for two straight beam test (Figure 4.1b), increasing the number of subdomains.

		nnf			**nnf + fru**			**First**			**Second**		
CVT		n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**
mul	2x2	6	18	4.96e+1	18	21	49.2e+01	8	18	49.62	6	18	49.62
	4x4	42	520	1.78e+5	114	102	1.16e+4	126	17	14.70	114	17	14.70
	8x8	210	2992	7.07e+5	546	459	3.03e+4	602	20	16.96	547	21	16.96
del	2x2	6	8	12.43e+1	18	12	12.37	8	8	12.43	6	8	12.43
	4x4	42	346	5.24e+4	114	61	6.35e+3	82	22	38.59	71	25	38.64
	8x8	210	1239	7.87e+4	546	386	NaN	506	24	32.94	470	27	32.66

		nnf			**nnf + fru**			**First**			**Second**		
RAND		n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**
mul	2x2	6	19	34.04	18	20	33.89	8	19	34.02	6	19	34.02
	4x4	42	488	1.40e+5	114	101	9.66e+3	126	16	10.67	114	16	10.66
	8x8	210	2781	5.57e+5	546	476	NaN	602	19	15.01	546	19	15.01
del	2x2	6	8	7.97	18	14	8.32	8	8	7.97	6	8	7.97
	4x4	42	316	3.87e+4	114	53	5.23e+3	82	18	23.97	71	22	23.50
	8x8	210	1156	6.13e+4	546	337	NaN	506	28	55.53	466	23	12.83

Table 4.3: Coarse space comparison for straight beams test with offset (Figure 4.1c), increasing the number of subdomains.

		nnf			**nnf + fru**			**First**			**Second**		
CVT		n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**
mul	2x2	6	212	5.82e+5	18	230	NaN	54	14	16.21	54	13	16.21
	4x4	42	1074	NaN	114	684	NaN	334	16	6.31	334	16	6.31
	8x8	210	3178	NaN	546	1300	NaN	1594	18	10.17	1594	18	10.17
del	2x2	6	9	5.09	18	16	12.01	6	9	5.09	6	9	5.09
	4x4	42	189	2.72e+4	114	17	9.57	58	18	20.32	63	21	20.36
	8x8	210	758	8.23e+4	546	18	9.08	330	20	13.62	343	22	13.58

		nnf			**nnf + fru**			**First**			**Second**		
RAND		n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**
mul	2x2	6	154	4.31e+5	18	198	4.00e+5	38	14	13.92	38	14	13.92
	4x4	42	1035	NaN	114	894	NaN	414	16	5.57	414	16	5.58
	8x8	210	2931	7.72e+5	546	1745	NaN	1882	18	9.41	1888	18	8.02
del	2x2	6	10	5.04	18	14	7.74	6	10	5.04	6	10	5.04
	4x4	42	170	2.45e+4	114	19	12.12	58	19	21.46	60	21	21.43
	8x8	210	705	7.27e+4	546	19	10.61	330	22	13.94	349	22	13.83

Table 4.4: Coarse space comparison for random coefficient distribution (Figure 4.1d), increasing the number of subdomains.

		nnf			nnf + fru			First			Second		
CVT		n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**
mul	2x2	9	274	NaN	24	274	6.69e+5	88	22	39.33	84	22	40.12
	4x4	42	1346	NaN	114	565	NaN	403	34	73.87	395	34	73.91
	8x8	273	5838	2.97e+6	546	2601	NaN	2189	52	76.45	2136	47	75.76
del	2x2	9	10	2.85	24	10	2.79	13	10	2.85	9	10	2.85
	4x4	42	34	9.97e+1	114	20	10.82	50	24	65.54	46	24	65.15
	8x8	273	231	3.16e+5	546	30	41.44	274	40	82.15	237	37	84.14
		nnf			**nnf + fru**			**First**			**Second**		
RAND		n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**
mul	2x2	6	198	1.00e+6	18	198	5.22e+5	75	26	34.93	73	26	34.93
	4x4	69	1850	4.89e+6	114	1298	NaN	466	46	95.92	458	46	100.7
	8x8	x	x	x	546	3875	NaN	2186	61	94.65	2156	60	94.71
del	2x2	6	11	6.37	18	17	11.90	8	12	6.37	6	11	6.37
	4x4	69	x	x	114	23	93.94	50	29	71.30	45	28	71.06
	8x8	x	x	x	546	34	93.69	262	42	139.35	244	36	61.08

Table 4.5: Coarse space comparison for the one straight beams testcase with 4 × 4 subdomains (Figure 4.1a), increasing the coefficient jump.

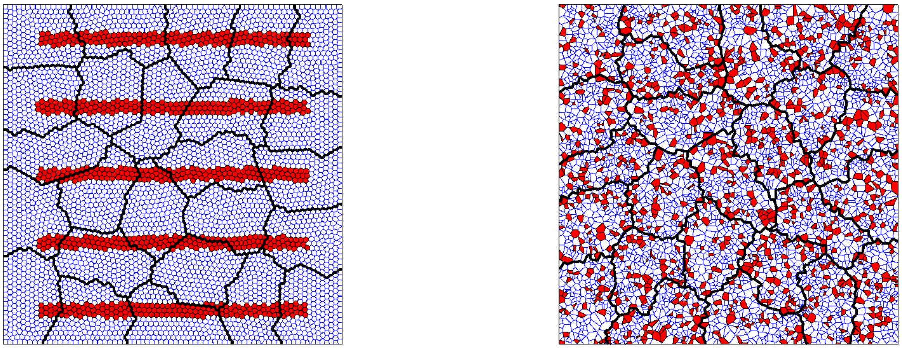
		nnf			nnf + fru			First			Second		
CVT		n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**
mul	1e4	42	82	2.66e+3	114	15	10.10	82	19	44.21	70	22	47.02
	1e5	42	123	2.65e+4	114	15	10.82	90	11	3.92	78	14	5.63
	1e6	42	246	2.66e+5	114	15	10.92	90	11	3.92	78	13	5.63
del	1e4	42	43	2.94e+2	114	12	3.86	66	22	54.34	56	25	55.83
	1e5	42	69	3.26e+3	114	12	4.56	70	16	9.60	60	18	9.70
	1e6	42	121	3.30e+4	114	13	4.66	70	16	9.73	59	18	9.79
		nnf			**nnf + fru**			**First**			**Second**		
RAND		n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**
mul	1e4	42	76	2.41e+3	114	15	13.01	82	19	40.53	70	21	44.53
	1e5	42	123	2.40e+4	114	15	13.97	90	12	4.05	78	14	5.70
	1e6	42	212	2.40e+5	114	15	14.11	90	12	4.05	78	14	5.70
del	1e4	42	43	2.89e+2	114	13	4.75	66	21	53.91	56	23	53.91
	1e5	42	68	3.19e+3	114	13	5.75	70	16	9.30	59	17	9.52
	1e6	42	124	3.23e+4	114	12	5.90	70	16	9.44	61	18	9.64

Random coefficient distribution. Table 4.4 shows the result for a random distribution of the coefficient jumps as in Figure 4.1d. The results are in line with our expectations. The classic coarse space of course fails. Both the adaptive coarse space seem to be robust since they are able to keep the condition number under the fixed tolerance and the deluxe scaling drastically reduce the dimension of the coarse space. The surprising result is given by the frugal coarse space that combined with the deluxe scaling is able to handle this configuration. Of course, this is due to the fact that the number of primal constraints required by the adaptive coarse space is limited, for other more complex configurations this could not happen.

Robustness varying coefficient jump ratio. Finally in Table 4.5 we tested different values for the coefficient jumps using the one beam configuration. We can see that, as expected, increasing the jump, the condition number grow for the classic coarse space. The adaptive coarse spaces and the frugal one are robust, since the iteration count and the condition number do not grow. We can also note that the number of constraints required by the adaptive algorithms stay bounded when the jump increase.

4.2.1.2 METIS partition

We performed also a test with an irregular partition into $3x3$ and $5x5$ subdomains, of a CVT and a RAND mesh with 5000 elements provided by METIS (Figure 4.3). We reported in Table 4.6 the results for a one beam test (4.3a) and a random configuration (4.3b). We only tested the two adaptive coarse spaces. Their behavior follows our expectations and again both two coarse space are robust with respect to the jumps. The deluxe scaling reduce the number of iterations and constraints required.



(a) Five straight beams

(b) Random coefficient distribution

Figure 4.3: Two example of CVT and RAND meshes with 25 subdomains with a METIS decomposition.

Table 4.6: Adaptive coarse space comparison for one straight beam and a random coefficient distribution (Figure 4.3). CVT and RAND mesh with 5000 elements and partitioned by METIS.

CVT		**Beam**						**Random**					
		First			**Second**			**First**			**Second**		
		n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**
mul	3x3	170	31	53.02	152	32	52.92	524	34	43.68	512	36	43.65
	5x5	539	28	53.72	468	31	55.51	1504	51	77.81	1454	51	76.23
del	3x3	55	21	18.31	36	23	38.91	54	18	8.56	32	18	9.73
	5x5	246	19	8.84	163	20	9.55	215	25	18.93	122	24	20.89
RAND		**Beam**						**Random**					
		First			**Second**			**First**			**Second**		
		n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**
mul	3x3	199	26	27.49	173	29	35.32	432	46	78.19	416	46	78.37
	5x5	658	31	44.08	573	34	45.51	1624	53	87.35	1579	53	87.43
del	3x3	63	14	8.34	35	16	8.78	52	20	15.51	32	19	16.08
	5x5	253	20	29.63	163	24	25.37	243	31	20.65	134	32	45.28

4.2.2 Sinkers

We now look at the second type of heterogeneity. As in [19, 87], we test the robustness of our adaptive BDDC algorithm on a multi-sinker test problem with inclusions of equal size placed randomly in the unit square domain so that they can overlap and intersect the boundary. This physically represent drops of an high viscosity material into a general fluid. The viscosity coefficient $\nu(\mathbf{x}) \in [\nu_{min}, \nu_{max}]$, $\mathbf{x} \in \Omega$, $0 < \nu_{min} < \nu_{max} < \infty$, is defined in terms of a C^∞ indicator function $\chi_n(\mathbf{x}) \in [0, 1]$ that accumulates n sinkers via the product of modified Gaussian functions:

$$\begin{aligned} \nu(\mathbf{x}) &:= (\nu_{max} - \nu_{min})(1 - \chi_n(\mathbf{x})) + \nu_{min}, & \mathbf{x} \in \Omega \\ \chi_n(\mathbf{x}) &:= \prod_{i=1}^{n} 1 - \exp\left(-\delta \max\left(0, |\mathbf{c}_i - \mathbf{x}| - \frac{\omega}{2}\right)^2 \right), & \mathbf{x} \in \Omega \end{aligned} \tag{4.9}$$

where $\mathbf{c}_i \in \Omega, i = 1, ..., n$ are the centers of the sinkers, $\delta > 0$ control the exponential decay and $\omega \geq 0$ is the diameter of the sinkers. In this way, the viscosity exhibits sharp gradients. In both of the two family of experiments the dynamic ratio $DR(\nu) := \nu_{max}/\nu_{min}$ can be up to six orders of magnitude. The right hand side is defined as $\mathbf{f}(\mathbf{x}) := (0, \beta(\chi_n(\mathbf{x}-1)))$, with $\beta = 10$, this is due to simulate gravity that takes down the high viscosity material. We consider two different tests. We first set a configuration for the sinkers (Figure 4.2a) where they are placed randomly and we increase the number of the subdomains, Table 4.7. We see that with the multiplicity scaling the behavior of the adaptive coarse space respect again our expectations, while the smaller coarse space and the frugal approach keep an high condition number. We observe that the performance of our algorithm increase when the number of subdomains increase, even with the two non adaptive coarse spaces. This is due to the fact that using the same configuration for the sinkers when the number of subdomains increase some of the inclusions are helped from the vertex constraints. Introducing the deluxe scaling we see that the frugal approach perform very well and seems to be a good alternative. We also have good results even just combining the deluxe

Table 4.7: Test with nSink = 11 and increasing the number of subdomains. Mesh sizes: nEl = 1000 for 2 × 2 subdomains, nEl = 5000 for 4 × 4 and nEl = 12000 for 8 × 8. The CVT meshes are created using 300 Lloyd iterations. The partition is performed by METIS. The coefficient $\delta = 2000$, $w = 0.05$ and $DR = 1e6$.

		nnf			**nnf + fru**			**First**			**Second**		
CVT		n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**
mul	2x2	9	185	1.12e+4	24	135	1.52e+3	49	58	106.62	47	58	107.42
	4x4	69	139	1.22e+4	168	62	4.10e+2	96	49	80.99	84	52	102.99
	8x8	360	185	1.16e+4	840	51	1.21e+2	418	54	101.43	394	55	97.96
del	2x2	9	11	10.03	24	12	6.10	11	11	9.93	9	11	10.03
	4x4	69	17	10.53	168	11	3.67	81	17	10.54	73	17	10.51
	8x8	360	21	25.91	840	12	4.25	386	21	25.95	406	18	9.57

		nnf			**nnf + fru**			**First**			**Second**		
RAND		n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**
mul	2x2	9	224	1.42e+2	24	173	3.28e+3	56	52	99.80	53	51	97.18
	4x4	69	203	1.04e+5	168	89	4.00e+3	104	54	90.37	90	56	97.76
	8x8	354	199	1.72e+4	828	53	247.96	426	51	96.15	400	54	96.32
del	2x2	9	12	13.91	24	11	2.91	13	12	13.89	9	12	13.91
	4x4	69	20	16.79	168	13	5.82	84	19	16.50	71	19	16.73
	8x8	354	21	16.78	828	11	3.33	380	20	16.60	383	19	13.47

Table 4.8: Test with increasing number of randomly placed sinkers. Mesh sizes: nEl = 5000. The CVT mesh is created using 300 Lloyd iterations. The partition is performed by METIS into 5 × 5 subdomains. The coefficient $\delta = 2000$, $w = 0.05$ and $DR = 1e6$.

		nnf			**nnf + fru**			**First**			**Second**		
CVT		n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**
mul	5	117	140	NaN	279	54	396.27	143	46	82.14	134	46	82.21
	10	117	117	6.63e+3	279	52	297.33	148	47	81.07	135	49	101.38
	15	117	179	3.09e+4	279	69	375.24	164	51	80.93	144	53	101.21
	20	117	199	3.07e+4	279	79	606.77	171	54	98.15	150	55	92.60
del	5	117	16	7.88	279	10	2.44	125	15	7.56	123	15	5.87
	10	117	16	8.40	279	10	2.57	131	16	7.72	122	17	8.34
	15	117	17	8.80	279	11	6.23	137	16	8.72	122	17	8.81
	20	117	22	28.78	279	12	6.23	138	20	28.72	122	21	28.03

		nnf			**nnf + fru**			**First**			**Second**		
RAND		n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**	n_Π	**it**	**k2**
mul	5	114	110	2.79e+4	276	52	363.35	132	48	91.52	123	50	91.54
	10	114	120	1.18e+4	276	60	1.30e+3	140	49	90.42	126	49	83.04
	15	114	340	NaN	276	105	1.30e+3	170	61	94.91	152	61	97.77
	20	114	420	8.14e+4	276	122	NaN	191	62	99.80	170	63	97.48
del	5	114	15	5.96	276	12	6.27	123	15	5.96	118	15	5.73
	10	114	16	5.97	276	12	3.38	127	15	5.42	118	16	5.98
	15	114	21	39.63	276	13	4.30	132	20	39.62	118	20	39.56
	20	114	27	413.74	276	13	6.48	137	23	39.14	121	23	39.47

scaling with the coarse space spanned by the no-net-flux condition. In Table 4.8 we instead keep fixed the number of subdomains at 5 × 5 and we increase the number of the inclusions. Again as before the adaptive coarse spaces are robust and also when introducing the deluxe scaling the smaller coarse space and the frugal one perform well.

4.3 Adaptive coarse space in three dimensions

We recall here an adaptive technique to enrich the minimal primal space $\mathbf{V}_\Gamma$ [43, 100], that is nothing else that the three dimensional extension of the first coarse space previously introduced. This is the adaptive coarse space implemented in PETSC in the PCBDDC library, so for more details about the implementation, we refer [100]. The idea is again to solve generalized eigenvalue problems defined on each subdomain face $\mathcal{F}$ and edge $\mathcal{E}$ and then construct an enriched primal space such that the condition number of the preconditioned system will be bounded from above by a selected $\nu_{tol} \in [1, \infty)$ times a constant independent on h, H and N. To construct an adaptive coarse space, we need to settle in a deluxe scaling context [101]. For each face $\mathcal{F}$ shared by two subdomains i, j, we consider the principal minors of the subdomain matrices $S_T^{(k)}$ with $k = i, j$:

$$S_{\mathcal{F}\mathcal{F}}^{(k)} := R_{\mathcal{F}}^{(k)} S_T^{(k)} {R_{\mathcal{F}}^{(k)}}^T, \tag{4.10}$$

where $R_{\mathcal{F}}^{(k)}$ maps $\mathbf{V}_\Gamma^{(k)}$ to the DoFs located on F. Then we split the matrices as follows

$$S_{\mathcal{F}\mathcal{F}}^{(k)} = \left[\begin{array}{cc} S_{\mathcal{F}'\mathcal{F}'}^{(k)} & S_{F'F_\Delta}^{(k)} \\ {S_{\mathcal{F}'\mathcal{F}_\Delta}^{(k)}}^T & S_{\mathcal{F}_\Delta\mathcal{F}_\Delta}^{(k)} \end{array} \right], \qquad k = i, j \tag{4.11}$$

where $\mathcal{F}_\Delta$ is the dual set of the DoFs associated to the face $\mathcal{F}$ and $\mathcal{F}' := \Gamma_i \setminus \mathcal{F}_\Delta$. We introduce the Schur complements:

$$\widetilde{S}_{\mathcal{F}_\Delta\mathcal{F}_\Delta}^{(k)} = S_{\mathcal{F}_\Delta\mathcal{F}_\Delta}^{(k)} - {S_{\mathcal{F}'\mathcal{F}_\Delta}^{(k)}}^T {S_{\mathcal{F}'\mathcal{F}'}^{(k)}}^{-1} S_{\mathcal{F}'\mathcal{F}_\Delta}^{(k)}, \quad k = i, j. \tag{4.12}$$

and then we solve the following eigenvalue problems:

$$\widetilde{S}_{\mathcal{F}_\Delta\mathcal{F}_\Delta}^{(i)} : \widetilde{S}_{\mathcal{F}_\Delta\mathcal{F}_\Delta}^{(j)} \psi = \nu S_{\mathcal{F}_\Delta\mathcal{F}_\Delta}^{(i)} : S_{\mathcal{F}_\Delta\mathcal{F}_\Delta}^{(j)} \psi \tag{4.13}$$

where again $A : B = (A^{-1} + B^{-1})^{-1}$, finally we choose the element of the primal space as $S_{\mathcal{F}_\Delta\mathcal{F}_\Delta}^{(i)} : S_{\mathcal{F}_\Delta\mathcal{F}_\Delta}^{(j)} \Psi$, where Ψ is the matrix formed column-wise by those eigenvectors associated with eigenvalues smaller than a fixed tolerance $1/\nu_{tol}$.

Analogously we repeat the same process for any edge $\mathcal{E}$. Assuming that the edge $\mathcal{E}$ is shared by $\mathcal{N}_\mathcal{E}$ subdomains, we define as in (4.10),(4.11) and (4.12), for $k \in \mathcal{N}_\mathcal{E}$ the matrices $S_{\mathcal{E}\mathcal{E}}^{(k)}, S_{\mathcal{E}\mathcal{E}}^{(k)}$ and

$\widetilde{S}^{(k)}_{\mathcal{E}_\Delta\mathcal{E}_\Delta}$. Then we solve the eigenvalues problem:

$$\Big(\sum_{i\in\mathcal{N}_\mathcal{E}} \widetilde{S}^{(i)\,-1}_{\mathcal{E}_\Delta\mathcal{E}_\Delta}\Big)\psi = \Big(\sum_{i\in\mathcal{N}_\mathcal{E}} S^{(i)\,-1}_{\mathcal{E}_\Delta\mathcal{E}_\Delta}\Big)\psi$$

and we select the elements of the primal space as $(\sum_{i\in\mathcal{N}_\mathcal{E}} \widetilde{S}^{(i)\,-1}_{\mathcal{E}_\Delta\mathcal{E}_\Delta})\Psi$, where Ψ, again is the matrix formed column-wise by those eigenvectors associated with eigenvalues smaller than a fixed tolerance $1/\nu_{tol}$.

We do not provide a proof of the following theorem, and we remand to [43] for further details:

Theorem 4.3.1. *Let the dual space satisfy the no-net-flux condition given in* (1) *and let the average operator preserve subdomain normal fluxes as in* (3.78) *and* (3.79)*. Then, $M^{-1}S$ is symmetric positive definite on the subspace $\widehat{\mathbf{V}}_{\Gamma,B} \times Q_0$; the minimum eigenvalue is 1, and we can algebraically construct a primal space $\mathbf{V}_\Gamma$ such that:*

$$\kappa_2(M^{-1}S) \le C\nu_{tol}, \qquad \forall \nu_{tol} \in [1,\infty), \tag{4.14}$$

where C is independent of $N, h,$ and H .

4.4 Numerical results in 3D

In this Section, we report the numerical results to validate our theoretical estimates of the adaptive BDDC algorithm for solving the Stokes model problem (1.9). We keep the same test problem as in Section 3.4.2 in particular we solve a problem on the unit cube $[0,1]^3$ with a known solution (Figure 2.3a) imposing Neumann boundary conditions on two faces of the cube and homogeneous Dirichlet boundary conditions on the other ones. We always solve the system (3.9), with the BDDC method used as a preconditioner for the CG method with a stopping criterion of a 10^{-8} reduction of the l^2-norm of the relative residual. We keep considering the three types of meshes Cube, Octa and CVT (Figure MESH). In our experiments, we compare two different choices of primal spaces, corresponding to tolerances $\nu_{tol} = 2$ and $\nu_{tol} = \infty$. The first one represents the adaptive coarse space built to keep the condition number under the fixed tolerance $\nu_{tol} = 2$. The latter represents the minimal coarse spaces created as explained in Chapter 3 to satisfy Assumptions 1 and 2. We also compare the BDDC algorithms against our previous block-diagonal preconditioners [40], and the parallel direct solver MUMPS [3, 4]. We conclude by testing the robustness of our adaptive BDDC algorithm on a benchmark problem with variable viscosity. We provide again Strong Scaling and optimality test, also a test varying the tolerance of the adaptive coarse space is performed with the purpose to have an heuristic idea of the optimal value of ν_{tol} to require. We conclude by testing the robustness of our adaptive BDDC algorithm on a benchmark problem with variable viscosity. To facilitate the comparison of the adaptive coarse space against the minimal one, we place some of the tables of the previous chapter next to the adaptive ones.

In the tables we use the following notation: procs = number of CPU cores, nEl = number of VEM elements, k = degree of VEM approximation, nDoFs = number of DoFs, N_Π = number of primal constraints, it = iteration count (GMRES for Block-Schur and Mass, CG for BDDC), k_2 = conditioning number, T_{ass}= time to assemble the stiffness matrix and the right-hand side, T_{ass}= time to assemble the preconditioner, T_{sol} = time to solve the interface saddle point problem and S^{id} = ideal speed up, S_p = parallel speed up.

4.4.1 Test 1: Strong scalability

We first study the strong scalability of our solvers. We keep fixed the global number of DoFs and the degree of the VEM approximation k, while we increase the number of processors from 4 to 256. We consider a CUBE mesh with 408 243 DoFs, a CVT mesh with 311 155 DoFs and an OCTA mesh with 549 939 DoFs. In Table 3.8, we report the results related to the three polyhedral meshes with $k = 2, 3$. In Figure 3.9, we plot the number of iterations and the parallel speedup for the case $k = 2, 3$. We observe that the CPU time T_{ass}, needed to assemble the stiffness matrix and the right-hand-side is scalable, with a speedup very close to the ideal ones. The adaptive BDDC method ($\nu_{tol} = 2$) results algorithmically scalable since the number of CG iterations remains bounded and the solution time decreases as the number of the processors increases. The parallel speedup shows a superlinear rate and improves only up to the point where communication times start to dominate, as usual in the strong scaling tests of domain decomposition methods where local problems are solved using direct factorizations. The differences between CUBE, OCTA and CVT meshes are due to the different sparsity patterns of the local subdomain problems. Also, the minimal coarse space results are scalable for the degree $k = 2$ and 3, with the same behavior as the adaptive BDDC.

4.4.2 Test 2: Optimality test with respect to the mesh size

We now perform an optimality test with respect to the mesh size: we keep fixed the number of processors at 32 and the degree of the VEM discretization $k = 2$, and we increase the number of DoFs by refining the mesh. The results are reported in Table 4.11. We observe that the adaptive solver has an optimal behavior irrespective of the type of polyhedral mesh considered since the number of iterations is independent of the refinement level and the condition number stays below the fixed tolerance. The minimal coarse space shows a quasi-optimal behavior since both the iteration count and the condition number exhibit a logarithmic growth as predicted by Theorem 3.3.1. Similar results also occur for the cases $k = 3$ and 4.

4.4.3 Test 3: Optimality test with respect to the polynomial degree

In this test, we study the robustness of our preconditioners when increasing the polynomial degree of the VEM discretization. The tests are performed by keeping fixed the number of processors (32) and the number of elements. The results reported in Table 3.10 show that the adaptive

					$\nu_{tol} = 2$				$\nu_{tol} = \infty$			
	procs	S^{id}	T_{ass}	S_p	N_Π	it	T_{sol}	S_p	N_Π	it	T_{sol}	S_p
CUBE	4		1 299		447	8	129		4	21	132	
	8	2	871	1.5	1 145	9	46	2.8	21	31	62	2.1
	16	4	418	3.1	2 195	9	29	4.4	41	33	35	3.8
	32	8	202	6.4	4 577	9	12	10.8	125	55	16	8.3
	64	16	105	12.4	7 831	9	5	25.8	311	58	6	22.0
	128	32	53	24.5	13 411	9	4	32.3	643	61	5	26.4
	256	64	27	48.1	22 723	9	5	25.8	1 399	62	5	26.4
CVT	4		2 456		661	9	754		15	38	711	
	8	2	1 423	1.7	1 655	10	334	2.3	130	41	248	2.9
	16	4	816	3.0	3 289	10	157	4.8	337	42	174	4.1
	32	8	400	6.1	6 641	10	61	12.4	905	46	66	10.8
	64	16	216	11.4	10 355	10	21	35.9	2326	33	21	33.9
	128	32	126	19.5	17 566	11	11	68.5	4398	32	9	79.0
	256	64	72	34.1	27 429	11	9	83.8	10 608	32	8	93.2
OCTA	4		3 225		476	8	127		21	32	59	
	8	2	1 637	2.0	1 217	8	50	2.6	21	32	59	2.6
	16	4	838	3.9	2 299	8	30	4.3	51	41	36	4.2
	32	8	434	7.4	4 793	8	12	10.5	125	57	16	9.5
	64	16	218	14.8	8 063	8	5	24.0	311	64	7	21.7
	128	32	116	27.8	14 342	9	6	21.9	643	76	5	30.4
	256	64	56	57.6	26 708	9	5	23.5	1 676	70	7	21.7

					$\nu_{tol} = 2$				$\nu_{tol} = \infty$			
	procs	S^{id}	T_{ass}	S_p	N_Π	it	T_{sol}	S_p	N_Π	it	T_{sol}	S_p
CUBE	4		3346		783	7	270		4	26	318	
	8	2	1821	1.8	1 952	9	111	2.4	30	41	193	2.4
	16	4	745	4.4	3 743	9	72	3.7	69	49	92	3.5
	32	8	467	7.2	7 815	9	29	9.2	152	72	37	8.6
	64	16	245	13.7	13 163	8	12	22.5	311	86	29	11.0
	128	32	140	23.9	23 350	9	9	28.7	815	98	13	24.5
	256	64	83	40.3	43 896	10	12	23.3	2 309	99	11	28.9
CVT	4		2 812		799	9	290		15	41	289	
	8	2	1 450	1.9	1 791	10	83	3.5	139	43	103	2.8
	16	4	820	3.4	3 580	10	59	4.9	341	44	70	4.1
	32	8	460	6.1	6 805	11	20	14.6	888	45	20	14.0
	64	16	264	10.6	9 831	11	7	40.8	2 134	37	6	44.2
	128	32	179	15.7	15 889	12	6	47.5	4 152	39	5	50.7
	256	64	175	16.0	22 874	13	8	37.2	7 840	37	6	44.2
OCTA	4		3 889		669	8	96		7	30	102	
	8	2	1 922	2.0	1 675	8	34	2.8	21	39	44	2.3
	16	4	982	4.0	3 147	8	24	4.0	51	52	34	3.0
	32	8	497	7.8	6 511	8	11	8.9	125	77	15	6.7
	64	16	261	14.9	10 715	8	5	18.1	311	90	7	14.2
	128	32	133	29.3	18 215	8	4	22.3	643	92	7	14.7
	256	64	66	59.1	30 263	8	6	15.7	1 399	92	7	15.0

Table 4.9: Strong Scalability with $k = 2, 3$. nEl for CUBE $= 13\,824$, CVT $= 4\,000$ and OCTA $= 15\,552$ meshes for $k = 2$. nEl for CUBE $= 8\,000$, CVT $= 1\,000$ and OCTA $= 4\,608$ meshes for $k = 3$.

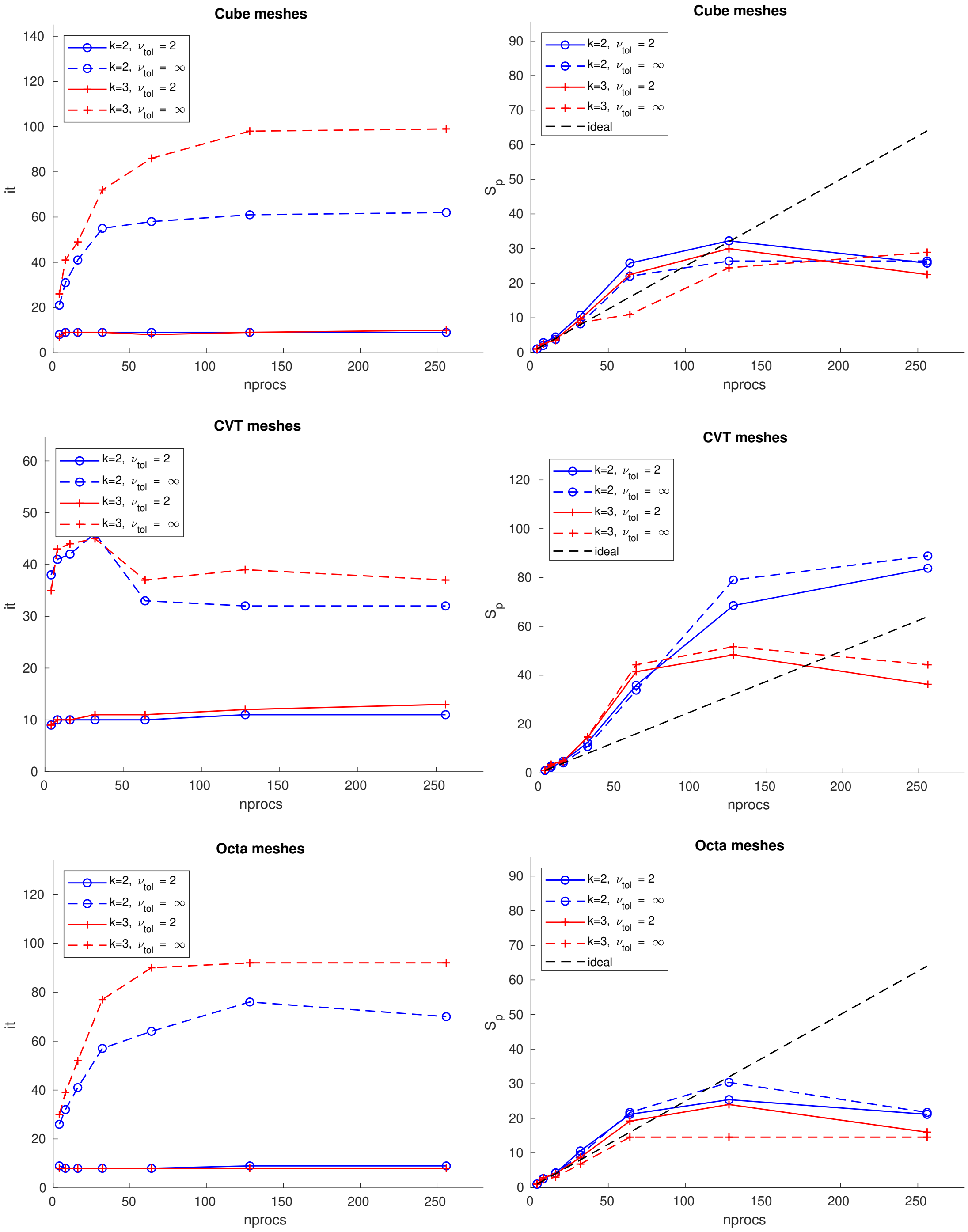


Figure 4.4: Test 1: strong scaling. Iteration of the CG (left) and parallel speedup (right) with the BDDC preconditioners for the two different primal spaces for different meshes and degrees of VEM discretizations.

			$\nu_{tol} = 2$			$\nu_{tol} = \infty$		
	nEl	nDoFs	N_{Π}	it	k_2	N_{Π}	it	k_2
CUBE	4 096	124 195	2 861	9	1.59	125	44	44.67
	8 000	239 763	3 644	9	1.62	152	50	54.53
	13 824	408 243	4 577	9	1.57	125	54	63.98
	21 952	643 387	5 257	9	1.64	152	58	72.37
	32 768	954 947	6 141	9	1.57	125	62	80.63
CVT	125	8 945	1 192	12	2.29	633	22	7.7
	1 000	76 051	3 666	10	2.01	801	32	16.09
	2 000	154 067	5 018	10	1.93	829	40	28.35
	4 000	311 155	6 700	10	1.92	836	46	37.42
	8 000	626 455	8 890	10	1.90	833	56	53.10
OCTA	576	22 035	1 313	9	1.76	125	36	26.15
	4 608	166 179	3 065	8	1.52	125	40	37.19
	9 000	320 763	4 571	9	1.64	157	57	47.31
	15 552	549 939	4 793	8	1.53	125	57	70.78
	30 375	1 065 693	8 513	10	1.85	474	57	46.62

Table 4.10: Optimality test with respect to the mesh size. $k = 2$ and procs $= 32$.

			$\nu_{tol} = 2$			$\nu_{tol} = \infty$		
	k	nDoFs	N_{Π}	it	k_2	N_{Π}	it	k_2
CUBE	2	16 787	1 273	9	1.81	125	34	23.86
	3	40 667	2 683	8	1.44	125	49	46.34
	4	76 387	4 495	8	1.49	125	64	80.67
CVT	2	8 945	1 192	12	2.29	633	22	7.7
	3	19 256	2 444	12	2.26	633	31	14.89
	4	33 487	4 604	14	3.02	633	46	28.86
OCTA	2	22 035	1 313	9	1.76	125	36	26.15
	3	52 251	2 847	8	1.46	125	55	53.66
	4	96 675	4 951	9	1.61	125	75	91.81

Table 4.11: Optimality Test Increasing the polynomial degree k. procs $= 32$ and nEl for CUBE $= 512$, CVT $= 125$ and OCTA $= 576$.

BDDC algorithm is robust with respect to the polynomial degree in all meshes. The BDDC solver with minimal coarse space instead exhibits a slight increase of the condition number and iterations count when the degree k increases.

4.4.4 Test 4: Solvers comparison

In Table 4.12, we compare the performance of the CG method accelerated by the adaptive BDDC preconditioner against the direct solver MUMPS, the GMRES method accelerated by the Block-Schur preconditioner proposed in [41], and preconditioned GMRES by the Block-Mass preconditioner proposed in [50]. The Block-Schur preconditioner is of the form:

$$B = \begin{bmatrix} diag(A)^{-1} & 0 \\ 0 & \widetilde{S}^{-1} \end{bmatrix}$$

	nEl	k	nDoFs	MUMPS T_{sol}	Block-Schur it	 T_{sol}	BDDC it	 T_{sol}
CUBE	32 768	2	954 947	297	705	110	21	26
	13 824	3	1 009 803	416	NC	NC	18	22
	8 000	4	1 119 523	465	NC	NC	13	34
CVT	8 000	2	627 455	913	568	134	16	58
	4 000	3	666 301	971	NC	NC	17	103
	2 000	4	571 696	842	NC	NC	22	93
OCTA	30 375	2	1 065 693	285	893	180	21	60
	15 552	3	1 322 571	355	NC	NC	19	73
	9 000	4	1 436 523	548	NC	NC	34	97

Table 4.12: Solver comparison among different parallel solver with procs = 64.

	ν_{tol}	$k = 2$ N_Π	 it	 k_2	 T_{sol}	 T_{Prec}
CUBE	2	9181	8	1.54	18.1	1509
	5	2683	20	6	16.9	1532
	10	937	30	14.3	19.6	1448
	100	301	61	88.2	22.3	1347
	1000	289	58	88.2	21.6	1338
CVT	2	9380	10	1.95	19.1	2981
	5	3826	18	4.7	20.4	3155
	10	2405	27	10.4	21.8	2883
	100	2250	32	15.2	20.7	2900
	1000	2250	32	15.2	20.6	2634
OCTA	2	12490	10	1.9	66.9	4641
	5	5281	20	6.0	61.8	4734
	10	2170	27	9.6	61.3	4630
	100	1188	51	33.9	68.6	4169
	1000	1188	51	34.0	68.5	4166

	ν_{tol}	$k = 3$ N_Π	 it	 k_2	 T_{sol}	 T_{Prec}
CUBE	2	14068	9	1.7	29.6	5925
	5	5020	18	4.7	22.8	5160
	10	1726	31	12.8	24.3	5903
	100	430	67	163.2	30.6	5160
	1000	370	80	164.6	35.7	5830
CVT	2	12940	14	4.6	32.8	11529
	5	6052	21	9.1	26.2	10358
	10	3497	31	27.8	27.3	11336
	100	2209	43	41.2	26.2	10175
	1000	2209	43	41.2	29	11424
OCTA	2	14869	8	1.5	31.5	13442
	5	5317	19	5.2	25.1	11950
	10	1759	30	12.9	26.4	13475
	100	397	55	44.5	30.5	11825
	1000	289	91	215.6	42.2	13521

Table 4.13: Test 5: Optimality test with respect to the adaptive tolerance with $k = 2, 3$ and $procs = 64$. nEl for CUBE = 32 768 , CVT = 4 000 and OCTA = 30 375 meshes for $k = 2$. nEl for CUBE = 13 824 , CVT = 2 000 and OCTA = 15 552 meshes for $k = 3$.

where $\widetilde{S} = -Bdiag(A)^{-1}B^T$ is the approximate Schur complement of the system (2.77) and the inversion of this matrix is performed by MUMPS, while the latter substitutes the bottom right block with the mass matrix for the pressure. We can see that the adaptive BDDC is significantly faster than the other solvers for all the meshes considered and for the three different degrees of the VEM discretization. We also note that both the Schur complement based preconditioners are not robust for the degree $k = 3$ and 4, since the GMRES method does not converge (NC in the table).

4.4.5 Test 5: Optimality test with respect to the adaptive tolerance

In Table 4.13, we study the behavior of the adaptive coarse space for the degree $k = 2, 3$ when varying the ν_{tol} parameter. We can see that with all the different choice of this parameter the condition number respect the theory for both the degrees. Looking at the time T_{sol} needed to solve the interface problem, we see that there is not so significant difference with the choice of the tolerance, even the fastest time is around the value $\nu_{tol} = 10$. Looking instead at the time T_{Prec}, that correspond to the time needed to set the preconditioner, we see that the optimal time is around $\nu_{tol} = 100$. This time is the highest computational part in this algorithm, but it is reqiured once for all. These contributions have to be computed just at the beginning since they do not change during the iterations. This is a relevant result for real application since we can achieve an acceptable value for the condition number and a fast resolution of the linear system even if we do not introduce too many primal constraints (as show the Figure 4.5). Further considerations on varying the tolerance will be done in the next test.

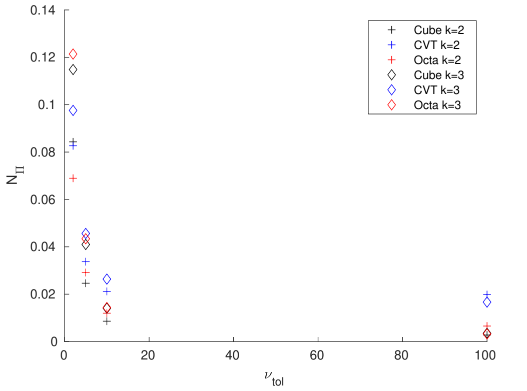


Figure 4.5: Optimality test with respect to the adaptive tolerance with $k = 2, 3$ and $procs = 64$. The plot shows the ratio R_Π between the number of the primal DoFs and the total number of interface DoFs.

	$DR(\nu)$	$1e+0$		$1e+2$		$1e+4$		$1e+6$	
	n	it	k_2	it	k_2	it	k_2	it	k_2
CUBE	1	18	6.2	19	7.7	20	7.6	19	7.7
	5	17	6.6	19	7.7	19	7.6	19	8.4
	10	18	6.6	19	7.6	19	7.6	19	8.5
	20	17	6.6	19	7.4	19	7.7	19	6.1
CVT	1	15	4.7	15	4.8	16	5.2	17	6.1
	5	15	4.7	15	4.8	17	5.8	19	8.1
	10	15	4.7	15	5.1	17	6.1	20	8.1
	20	14	4.7	17	7.7	20	9.7	28	15.2
OCTA	1	17	6.2	19	7.4	19	7.6	20	8.4
	5	17	6.2	18	7.2	19	7.6	19	8.4
	10	17	6.2	18	6.8	20	8.8	19	8.9
	20	17	6.2	18	7.0	19	7.6	18	8.4

Table 4.14: Test 6a. Multi-sinker benchmark problem varying the number of the inclusions. procs = 64 and nEl for CUBE = 13 824, CVT = 4 000 and OCTA = 15 552.

		$k = 2$				
	ν_{tol}	N_Π	it	k_2	T_{sol}	T_{Prec}
CUBE	2	9876	17	4.9	7	256
	5	4158	19	8.5	6	251
	10	2657	30	11.7	6	253
	100	973	77	107.4	9	259
	1000	555	168	460.6	16	265
	10000	469	289	2.20E+03	26	272
	Deluxe	311	2003	1.70E+05	285	533
CVT	2	13054	15	4.0	25	2880
	5	6196	20	8.1	23	2907
	10	4123	34	14.1	24	2881
	100	2547	78	74.6	31	2889
	1000	2405	126	3.10E+02	43	2947
	10000	2377	152	6.40E+02	47	2650
	Deluxe	2326	166	6.60E+02	48	2953
OCTA	2	10180	11	2.2	7	569
	5	4119	19	8.9	6	532
	10	2463	30	11.3	6	567
	100	897	66	59.5	9	574
	1000	548	146	3.30E+02	16	538
	10000	461	252	2.00E+03	26	589
	Deluxe	311	1906	1.75E+05	287	817

Table 4.15: Test 6b. Multi-sinker benchmark problem varying the tolerance of the adaptive coarse space. procs = 64 and nEl for CUBE = 13 824, CVT = 4 000 and OCTA = 15 552.

4.4.6 Test 6: Multi-sinker benchmark problem

To consider a practical application, we conclude by testing the robustness of our adaptive BDDC algorithm on a benchmark problem with heterogeneous viscosity. We perform a multi-sinker test problem as in the two dimensional case with inclusions of equal size placed randomly in the unit cube domain so that they can overlap and intersect the interface among the subdomains. The viscosity coefficient $\nu(\mathbf{x})$ is defined in terms of a C^∞ indicator function $\chi_n(\mathbf{x}) \in [0, 1]$ that accumulates n sinkers via the product of modified Gaussian functions, see [87] for more details about these functions. In this way, the viscosity exhibits sharp gradients, and its dynamic ratio $DR(\nu) := \nu_{max}/\nu_{min}$ in our study can be up to six orders of magnitude.

Figure 4.6 shows two example of the solution of the test used in our simulation. We solved the standard Stokes equations with a load term as in 4.2.2 to simulate the gravity. The pictures exhibit two different flows for the fluid due to the fact that in the left one we imposed Neumann boundary condition on two faces of the cube and homogeneous Dirichlet boundary condition on the rest, while in the right one we took homogeneous Dirichlet boundary conditions on the whole domain. The two different behaviours are clear, since in the left example we see the flow through two faces while in the right one we see the low viscosity fluid being strongly pushed to the top while the inclusions are taken down by the gravity. As previously, in our experiments we imposed Neumann boundary conditions on two faces of the cube and homogeneous Dirichlet on the other ones. We fix the mesh element size and the number of processes at 64, and we study the iteration count and condition number of the adaptive BDDC algorithm with $\nu_{tol} = 5$, varying the dynamic ratio $DR(\nu)$ from 1 to $1e+6$, and the number of sinkers n from 1 to 20. The results reported in the left panel of Table 4.14, obtained on different polyhedral meshes and for a VEM discretization of degree $k = 2$, show the robustness of our adaptive preconditioner since the number of iterations and the condition number are independent of the number of sinkers and the viscosity ratio. We also report in Table 4.15 a test by varying the tolerance ν_{tol} to highlight the need for an adaptive coarse space in terms of computational timings. It is evident the necessity to use an adaptive approach in this situation, since the simple deluxe scaling is not able to handle the heterogeneity. The number of iterations required and the condition number are very high. We observe also that there is not so much gain in terms of timing to solve the linear system imposing a too strict tolerance. It seems to be reasonable to relax this parameter around 100, in a way to not have to deal with a too large primal space.

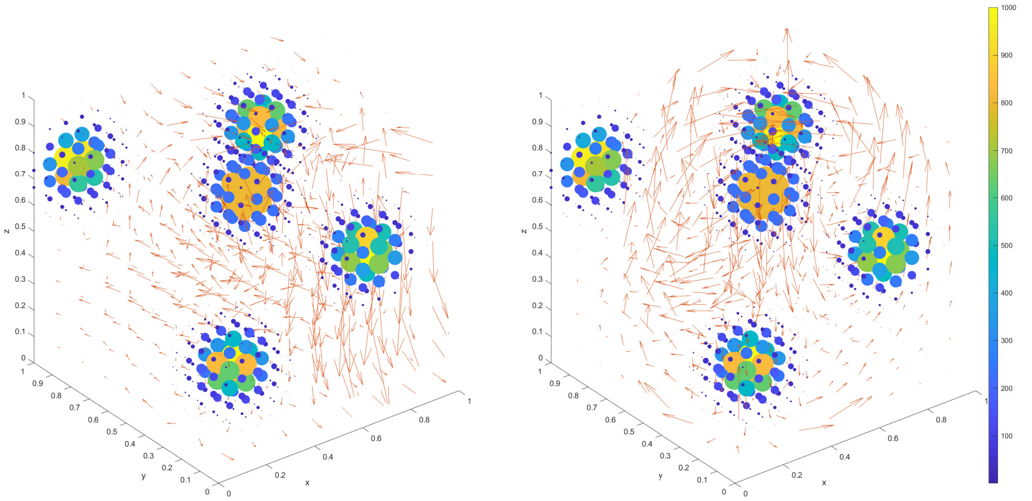


Figure 4.6: Two examples of a viscosity field and fluid flow for a test case with 5 Sinkers. The inclusions are represented by spheres placed in the baricenter of each element with size and colour scaled by the viscosity function. The red arrows represent the velocity field of the fluid scaled by its magnitude. In the left figure we imposed Neumann boundary conditions on the plains $y = 0$ and $y = 1$ and homogeneous Dirichlet boundary condition on the other faces. In the right figure we imposed homogeneous Dirichlet boundary condition on the whole domain.

Chapter 5

Non-Symmetric Problems: The Oseen equation

In this chapter we analyze the BDDC algorithm applied to a non-symmetric saddle point problem that arise from the VEM discretization of the steady Navier-Stokes and Oseen equations. We do not provide here a complete treatment of the theory, but we limit to explain the features and the problems that arise in this context. Since our interest is to test our preconditioner to solve a non-symmetric problem, we limit to apply it to solve the Oseen equation. The same algorithm combined with the Newton method to solve the non-linearity, can be used in the Navier-Stokes context. We also provide some numerical simulations that shows the good behavior of the preconditioner. Due to the lack of the symmetry of the model problem the resolution of the preconditioned linear system is obtained by the GMRES method.

5.1 The VEM discretization

We recall here, very briefly, the VEM technology for this equation. It is well known that, the problem (1.20) leads to instabilities when the convective term $\|\boldsymbol{\beta}\|_{[L^\infty(\Omega)]^d}$ with $d = 2, 3$, is dominant with respect to the diffusive and the reaction term ([30, 31, 48, 55, 90]). In such situations, the spurious oscillations that spoil the numerical solution are prevented by a stabilized form of the problem, that in the VEM context for the Oseen equation was presented in [13]. In our study we do not introduce any stabilization, so to stay on the safe side we assume to be in diffusion-dominated case with $\nu = \sigma = 1$ and the field $\boldsymbol{\beta}$, in the numerical simulations, chosen as the exact solution in order to simulate a Navier-Stokes framework. We do not enter into the details and we remand to [12, 15] for an exhaustive study and analysis of the Navier-Stokes VEM. In the same fashion of Chapter 2 we define the VEM spaces for the velocity and pressure to discretize the Oseen problem (1.20). In particular, we choose exactly the same space as in (2.5),(2.6) in two dimension and (2.38),(2.30) in three dimension.

5.1.1 Bilinear forms and Discrete problem

We do not repeat all the procedure to define the VEM method used, since it is the same of the one in Chapter 2. We limit here to show the local bilinear forms that differ from the previous ones. With this scope, we define $A_h^K : \mathbf{V}^K \times \mathbf{V}^K \to \mathbb{R}$:

$$A_h^K(\mathbf{u}_h, \mathbf{v}_h) := \nu \int_K \mathbf{\Pi}_{k-1}^{0,K}\, \varepsilon(\mathbf{u}_h) : \mathbf{\Pi}_{k-1}^{0,K}\, \varepsilon(\mathbf{v}_h)\, \mathrm{d}K + \sigma \int_K \Pi_k^{0,K}\mathbf{u}_h \cdot \Pi_k^{0,K}\mathbf{v}_h\, \mathrm{d}K, \tag{5.1}$$

$c_h^K : \mathbf{V}^K \times \mathbf{V}^K \to \mathbb{R}$ and its skew symmetric form $c_h^{skew,K} : \mathbf{V}^K \times \mathbf{V}^K \to \mathbb{R}$:

$$\begin{aligned} c_h^K(\mathbf{u}_h, \mathbf{v}_h) &:= \int_K \left[\left(\mathbf{\Pi}_k^{0,K}\, \boldsymbol{\nabla}\mathbf{u}_h\right)\boldsymbol{\beta}\right] \cdot \Pi_k^{0,K}\, \mathbf{v}_h\, \mathrm{d}K, \\ c_h^{skew,K}(\mathbf{u}_h, \mathbf{v}_h) &:= \frac{1}{2}c_h^K(\mathbf{u}_h, \mathbf{v}_h) - \frac{1}{2}c_h^K(\mathbf{v}_h, \mathbf{u}_h), \end{aligned} \tag{5.2}$$

while the bilinear form $b : \mathbf{V}^K \times Q^K \to \mathbb{R}$ is again chosen as the continuous one, since no approximation is needed. The stabilization term S^K is chosen as in Chapter 2, and we introduce the local discrete form $\mathcal{K}_h^K : \mathbf{V}^K \times \mathbf{V}^K \to \mathbb{R}$:

$$\mathcal{K}_h^K(\mathbf{u}_h, \mathbf{v}_h) := A_h^K(\mathbf{u}_h, \mathbf{v}_h) + c_h^{skew,K}(\mathbf{u}_h, \mathbf{v}_h) + S^K(\mathbf{u}_h, \mathbf{v}_h). \tag{5.3}$$

Using the global version of the bilinear form previously introduced, the VEM method for the Oseen equation is then given by:

$$\begin{cases} \text{find } (\mathbf{u}_h, p_h) \in \mathbf{V}_{h,0} \times Q_{h,0} \text{ such that} & \\ \mathcal{K}_h(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = (\mathbf{f}_h, \mathbf{v}_h) & \text{for all } \mathbf{v}_h \in \mathbf{V}_{h,0}, \\ b(\mathbf{u}_h, q_h) = 0 & \text{for all } q_h \in Q_{h,0}, \end{cases} \tag{5.4}$$

Under the assumption we have done before it is shown in [12,15] that this method has the optimal order of convergence. For a complete analysis that takes also into account the instability induced by the convective field in two dimension, can be found in [13].

The discrete variational problem can be written as:

$$\begin{bmatrix} K & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix}, \tag{5.5}$$

where the matrices K and B are associated with the discrete bilinear forms $\mathcal{K}_h(\cdot,\cdot)$ and $b_h(\cdot,\cdot)$, with $K = A + C$, where A is the matrix of the symmetric bilinear form $a_h(\cdot,\cdot)$ and C the one of the non-symmetric one $c_h^{skew}(\cdot,\cdot)$.

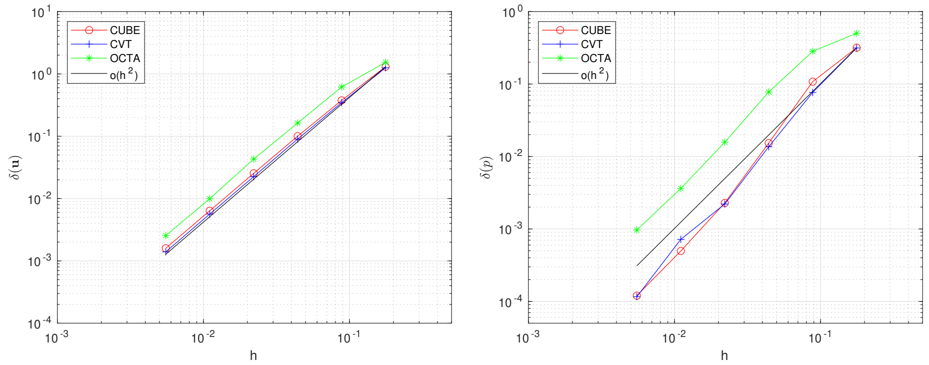


Figure 5.1: Velocity (left) and pressure (right) convergence plot for QUAD, CVT and RAND mesh in 2D for VEM discretization of degree $k = 2$.

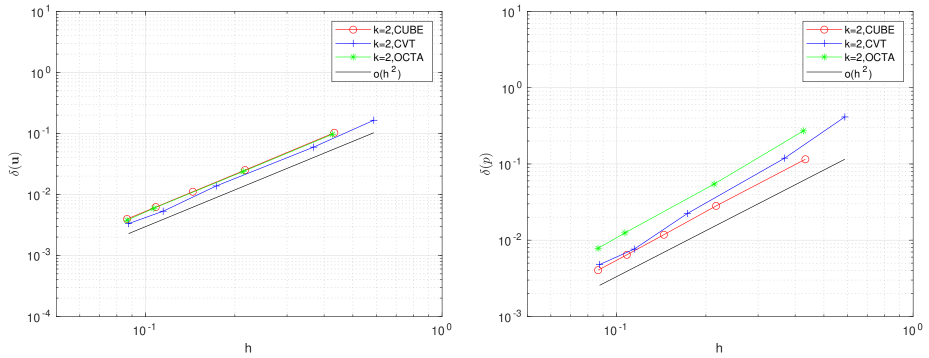


Figure 5.2: Velocity (left) and pressure (right) convergence plot for CUBE, CVT and OCTA mesh in 3D for VEM discretization of degree $k = 2$.

5.1.2 Numerical results

For sake of completeness we performed numerical simulations on the unit square and the unit cube imposing Dirichlet boundary conditions on the whole domain Ω. We analyzed the errors $\delta(\mathbf{u})$ and $\delta(p)$, defined in (2.78), on a test problem with known exact solution $\mathbf{u}$ (the same as in 2.3), $\nu = \sigma = 1$ and the field $\boldsymbol{\beta} = \mathbf{u}$. We reported in Figure 5.1 and 5.2, the convergence plot of the method of order $k = 2$ in two and three dimension. As we can see in the pictures, for this case we recover the optimal error for the convergence without introduce any stabilization.

5.2 Domain decomposition and BDDC construction

In the same fashion of Chapter 3, we introduce the domain decomposition technique. We split $\mathcal{T}_h$ into N non-overlapping subdomains Ω_i with characteristic size H_i, respecting the shape-regularity of the decomposition. We decompose the discrete velocity and pressure space $\widehat{\mathbf{V}}$ and

Q into:

$$\widehat{\mathbf{V}} = \mathbf{V}_I \bigoplus \widehat{\mathbf{V}}_\Gamma, \quad Q = Q_I \bigoplus Q_0, \tag{5.6}$$

with $Q_0 := \prod_{i=1}^N \{q \in \Omega_i | q \ \textit{is constant in} \ \Omega_i\}$.
In this way, we obtain that the global saddle-point problem (2.77) can be written as: find $(\mathbf{u}_I, p_I, \mathbf{u}_\Gamma, p_0) \in (\mathbf{V}_I, Q_I, \widehat{\mathbf{V}}_\Gamma, Q_0)$, such that:

$$\begin{bmatrix} K_{II} & B_{II}^T & \widehat{K}_{I\Gamma} & 0 \\ B_{II} & 0 & \widehat{B}_{I\Gamma} & 0 \\ \widehat{K}_{\Gamma I} & \widehat{B}_{I\Gamma}^T & \widehat{K}_{\Gamma\Gamma} & \widehat{B}_{0\Gamma}^T \\ 0 & 0 & \widehat{B}_{0\Gamma}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_I \\ p_I \\ \mathbf{u}_\Gamma \\ p_0 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_I \\ 0 \\ \mathbf{f}_\Gamma \\ 0 \end{bmatrix}, \tag{5.7}$$

where we make note that now we have lost the symmetry in the blocks related to the bilinear form A. We proceed to eliminate, by static condensation, the independent subdomain variables $(\mathbf{u}_I, p_I)$ solving independent Dirichlet problems:

$$\begin{bmatrix} K_{II} & B_{II}^T \\ B_{II} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_I \\ p_I \end{bmatrix} + \begin{bmatrix} \widehat{K}_{I\Gamma} & 0 \\ \widehat{B}_{I\Gamma} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_\Gamma \\ p_0 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_I \\ 0 \end{bmatrix}, \tag{5.8}$$

and obtain the global interface saddle point problem:

$$\widehat{S}\ \widehat{u} = \begin{bmatrix} \widehat{S}_\Gamma & \widehat{B}_{0\Gamma}^T \\ \widehat{B}_{0\Gamma} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_\Gamma \\ p_0 \end{bmatrix} = \begin{bmatrix} \mathbf{g}_\Gamma \\ 0 \end{bmatrix} = \widehat{\mathbf{g}}, \tag{5.9}$$

where the right-hand side $\widehat{\mathbf{g}} \in \mathbf{F}_\Gamma \times F_0$ is given by

$$\widehat{\mathbf{g}} = \sum_{i=1}^N R_\Gamma^{(i)^T} \left\{ \begin{bmatrix} \mathbf{f}_\Gamma^{(i)} \\ 0 \end{bmatrix} - \begin{bmatrix} K_{\Gamma I}^{(i)} & B_{I\Gamma}^{(i)^T} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} K_{II}^{(i)} & B_{II}^{(i)^T} \\ B_{II}^{(i)} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f}_I^{(i)} \\ 0 \end{bmatrix} \right\}, \tag{5.10}$$

with the same restriction and scaling operators defined in the previous chapters. To construct the preconditioner, we need to reintroduce a partially assembled interface velocity space, namely $\widetilde{\mathbf{V}}_\Gamma$ (see Figure 3.2.1 (b)):

$$\widetilde{\mathbf{V}}_\Gamma = \widehat{\mathbf{V}}_\Pi \bigoplus \mathbf{V}_\Delta = \widehat{\mathbf{V}}_\Pi \bigoplus \Big(\prod_{i=1}^N \mathbf{V}_\Delta^{(i)} \Big). \tag{5.11}$$

Proceeding by block factorization, the inverse of the Schur complement on the partially assembled velocity space, can then be written as:

$$\widetilde{S}^{-1} = \sum_{i=1}^N \begin{bmatrix} 0 & 0 & R_{\Delta,i}^T \end{bmatrix} \begin{bmatrix} K_{II}^{(i)} & B_{II}^{(i)^T} & K_{I\Delta}^{(i)} \\ B_{II}^{(i)} & 0 & B_{I\Delta}^{(i)} \\ K_{\Delta I}^{(i)} & B_{I\Delta}^{(i)^T} & K_{\Delta\Delta}^{(i)} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ R_{\Delta,i} \end{bmatrix} + \Phi S_{CC}^{-1} \Psi^T, \tag{5.12}$$

with the primal basis functions:

$$\begin{aligned}
\Phi &= R_{\Pi 0}^T - \sum_{i=1}^{N} \begin{bmatrix} 0 & 0 & R_{\Delta,i}^T \end{bmatrix} \begin{bmatrix} K_{II}^{(i)} & B_{II}^{(i)^T} & K_{I\Delta}^{(i)} \\ B_{II}^{(i)} & 0 & B_{I\Delta}^{(i)} \\ K_{\Delta I}^{(i)} & B_{I\Delta}^{(i)^T} & K_{\Delta\Delta}^{(i)} \end{bmatrix}^{-1} \begin{bmatrix} K_{I\Pi}^{(i)} & 0 \\ B_{I\Pi}^{(i)} & 0 \\ K_{\Delta\Pi}^{(i)} & B_{0\Delta}^{(i)^T} \end{bmatrix} R_C^{(i)}, \\
\Psi &= R_{\Pi 0}^T - \sum_{i=1}^{N} \begin{bmatrix} 0 & 0 & R_{\Delta,i}^T \end{bmatrix} \begin{bmatrix} K_{II}^{(i)} & B_{II}^{(i)^T} & K_{I\Delta}^{(i)} \\ B_{II}^{(i)} & 0 & B_{I\Delta}^{(i)} \\ K_{\Delta I}^{(i)} & B_{I\Delta}^{(i)^T} & K_{\Delta\Delta}^{(i)} \end{bmatrix}^{-T} \begin{bmatrix} K_{\Pi I}^{(i)^T} & 0 \\ B_{I\Pi}^{(i)} & 0 \\ K_{\Pi\Delta}^{(i)^T} & B_{0\Delta}^{(i)^T} \end{bmatrix} R_C^{(i)},
\end{aligned} \tag{5.13}$$

that are distinct in a non-symmetric framework, with the preconditioner that has the usual form:

$$M^{-1} = \widetilde{R}_D^T \widetilde{S}^{-1} \widetilde{R}_D, \tag{5.14}$$

and the BDDC preconditioned problem: find $(\mathbf{u}_{\boldsymbol{\Gamma}}, p_0) \in \widehat{\mathbf{V}}_\Gamma \times Q_0$, such that

$$\widetilde{R}_D^T \widetilde{S}^{-1} \widetilde{R}_D \widehat{S} \begin{bmatrix} \mathbf{u}_\Gamma \\ p_0 \end{bmatrix} = \widetilde{R}_D^T \widetilde{S}^{-1} \widetilde{R}_D \begin{bmatrix} \mathbf{g}_\Gamma \\ 0 \end{bmatrix}. \tag{5.15}$$

Again the matrix $\widehat{S}$ is indefinite on the space $\widehat{\mathbf{V}} \times Q_0$, but it is positive definite on the benign subsace $\widehat{\mathbf{V}} \cap Ker(\widehat{B}_{0,\Gamma}) \times Q_0$. To ensure that the iterates of the GMRES methods stay in this subspace it is necessary introduce again and satisfy the *no-net-flux* condition. For what concern the theoretical convergence rate of the preconditioner the study is more involved. Due to the asymmetry, the Oseen extension does not have the minimization property and one has to estimate more carefully the non-symmetric part. A detailed study has been done in [97], where in the lower bound estimate of the average operator is no more possible use the minimization property of the Stokes extension and a complex procedure is required. Moreover, in that paper their analysis require a further assumption on the field $\boldsymbol{\beta}$, leading to introduce more constraints in the primal space. Anyway, we decide to not introduce them in our algorithm and, as we will see in the numerical results, we obtain a good behavior of the preconditioner. They will be studied in future works in this direction. Although we do not have a theoretical estimate for the convergence rate of this algorithm on non-symmetric problems, we obtain scalability results that are in line with the same preconditioner applied on symmetric problems.

5.3 Numerical results

In this section, we provide as before some numerical tests to study the behavior of the BDDC preconditioner with respect to the mesh size h, the number of subdomains N the shape of the polygonal mesh elements. All the test reported here are referred to the BDDC algorithm that satisfies both the assumptions to guarantee the stability of the average operator and the iteration that remains into the benign space. Due to the asymmetry of the problem we solve the saddle

		QUAD			CVT			RAND		
	nSub	N_Γ	N_Π	it	N_Γ	N_Π	it	N_Γ	N_Π	it
SQUARE	4x4	3810	42	14	738	42	15	594	42	15
	8x8	8834	210	18	3394	210	18	2786	210	18
	12x12	13794	506	20	7970	506	20	6578	506	19
	16x16	18690	930	20	14466	930	20	11970	930	20
	20x20	23522	1482	19	22882	1482	20	18962	1482	20
METIS	4x4	870	69	20	1470	69	18	1282	66	20
	8x8	3906	360	23	6770	354	21	5698	357	29
	12x12	9342	854	27	15610	872	23	13478	875	33
	16x16	16510	1581	29	28198	1614	25	24190	1611	36
	20x20	23522	1482	34	44338	2574	26	38670	2568	37

Table 5.1: Weak scaling. $k = 2$ with ratio $H/h = 8$.

point problem with the GMRES method using a stopping criteria for the relative residual error to 10^{-8} for both two and three dimension.

5.3.1 Numerical Results in 2D

We solve the Oseen equations on the unit square domain $\Omega = [0,1] \times [0,1]$, applying homogeneous Dirichlet boundary conditions on the whole $\partial\Omega$, with the usual known solution (2.3). In the following tables, we report the number of iterations to solve the global interface saddle-point problem (5.9) with the preconditioned GMRES method, accelerated by BDDC. Also here, our tests have been executed on different types of polygonal meshes (QUAD, CVT and RAND as in Figure 2.2 using the two different partition techniques SQUARE and METIS (Figure 3.3). We use the VEM discretization with degree $k = 2$.

Remark 5.3.1. We remark that our 2D simulations have been provided by MATLAB R2023A© in a serial code, no computational time analysis is provided.

In the tables we use the following notation: SQUARE (or SQ) and METIS (or MET) = type of the mesh partitioning, nSub = number of subdomain, H/h = ratio between the diameter of subdomain and the element, N_Γ = number of interface DoFs, N_Π = number of primal DoFs, it = iteration count (GMRES).

As in Section 3.4.1.1 we perform the two common scaling test in the domain decomposition and an optimality test.

Weak Scaling. In Table 5.1 we reported the results for the different meshes, partitioned with the two techniques. As expected, using with the SQUARE partitioning the iteration counts do not grow when the number of subdomains increases,so the BDDC solver appears to be scalable. A slightly worse behavior happen for the METIS partition, but as we said before this is due to the irregularity of the subdomains.

Optimality test. The results for the optimality test with respect to the mesh size are shown in Table 5.2. We fix the number of subdomains at 16 and we increase the local ratio H/h. With the SQUARE partition, the GMRES have a quasi-optimal behavior since the iteration count

		QUAD			CVT			RAND		
	nSub	N_Γ	N_Π	it	N_Γ	N_Π	it	N_Γ	N_Π	it
SQUARE	8	738	42	14	738	42	15	594	42	15
	16	1506	42	15	1458	42	17	1522	42	17
	24	2274	42	16	2130	42	17	2002	42	17
	32	3042	42	17	2850	42	18	2434	42	19
METIS	8	870	69	20	1470	69	18	1282	66	20
	16	1722	69	19	3030	69	19	2498	69	22
	24	2430	69	16	4506	69	20	3866	69	20
	32	3510	69	19	5986	69	19	4934	69	23

Table 5.2: Optimality test. $k = 2$ with $nSub = 4 \times 4$.

	QUAD			CVT			RAND		
nSub	N_Γ	N_Π	it	N_Γ	N_Π	it	N_Γ	N_Π	it
4x4	3810	42	17	7650	69	18	6726	69	22
8x8	8834	210	20	17026	360	22	15110	351	29
12x12	13794	506	21	26786	866	26	23122	872	30
16x16	18690	930	20	35186	1602	24	30946	1608	35
20x20	23522	1482	19	44338	2574	26	38670	2568	37

Table 5.3: Strong scaling. $k = 2$ and $nEl = 25600$ for QUAD (SQUARE), CVT and RAND (METIS).

exhibits a logarithmic growth. With the METIS partition we see again the same effect as in the weak scaling test for the Stokes equations, where the increasing regularity of the boundaries balances the growth of the iteration count.

Strong scaling. We conclude this group with a strong scaling test with 25600 elements for all the three different meshes. Again the results reflect what we expect keeping the number of iterations bounded when increasing the number of the subdomains.

5.3.2 Numerical Results in 3D

We conclude reporting the numerical results for the BDDC algorithm for solving the 3D Oseen problem. We solve a problem on the unit cube $[0, 1]^3$ with a known solution as in 2.3, imposing full Dirichlet boundary conditions on the whole domain. Again we use the BDDC method as a preconditioner for system the (5.9), which is solved by the GMRES method. The implementation is based on the PETSc library [8], where the PCBDDC [100] library is adapted in our framework after having introduce a little patch to handle the discrete extensions that are no more harmonic. In the tables we use the following notation: procs = number of CPUs, nEl = number of VEM elements, k = degree of VEM approximation, nDoFs = number of DoFs, N_Γ = number of interface DoFs, N_Π = number of primal constraints, it = iteration count (GMRES), k_2 = conditioning number, T_{ass}= time to assemble the stiffness matrix and the right-hand side, T_{sol} = time to solve the interface saddle point problem and S^{id} = ideal speed up, S_p = parallel speed up.

	procs	S^{id}	T_{ass}	S_p	N_Γ	N_Π	it	T_{sol}	S_p
CUBE	8	1	7 253	1.00	37 443	23	32	178.1	1.00
	16	2	3 431	2.11	62 019	53	38	64.9	2.74
	32	4	1 580	4.59	85 827	123	48	28.7	6.21
	64	8	769	9.43	108 895	289	57	15.5	11.49
	128	16	387	18.74	155 013	621	69	9.7	18.36
	256	32	194	37.39	198 177	1 337	61	8.7	20.42
CVT	8	1	11 265	1.00	42 453	126	42	92.7	1.00
	16	2	6 636	1.70	68 691	333	45	38.0	2.44
	32	4	3 394	3.32	92 745	885	45	16.1	5.76
	64	8	1 819	6.19	113 475	2 250	40	7.6	12.20
	128	16	1 051	10.72	154 215	4 859	39	5.1	18.18
	256	32	556	20.26	184 995	10 608	41	9.6	9.66
OCTA	8	1	11 490	1.00	64 011	63	50	177.0	1.00
	16	2	7 459	1.54	105 447	188	53	78.1	2.27
	32	4	3 848	2.99	144 435	472	53	31.6	5.60
	64	8	1 685	6.82	181 143	1 188	52	15.3	11.57
	128	16	857	13.41	238 035	2 101	60	8.7	20.30
	256	32	449	25.59	290 559	3 764	60	8.6	20.58

Table 5.4: Strong Scaling. $k = 2$ and nEl for CUBE $= 32\,768$, CVT $= 4\,000$ and OCTA $= 30\,375$ meshes.

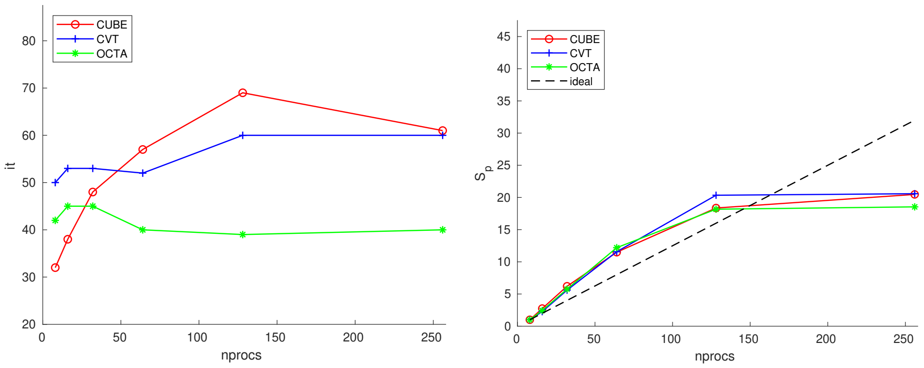


Figure 5.3: Strong scaling. Iteration of the GMRES (left) and parallel speedup (right) with the BDDC preconditioner for different meshes.

Table 5.5: Optimality test with respect to the mesh size. $k = 2$ and procs = 64.

nEl	nDoFs	N_Γ	N_Π	it
512	16 281	6 507	289	26
1 728	53 787	14 931	289	32
8 000	238 763	42 147	289	45
21 952	643 387	83 187	289	54
46 656	1 353 675	138 051	289	60

(a) CUBE meshes.

nEl	nDoFs	N_Γ	N_Π	it
125	8 945	6 987	1 241	22
1 000	76 051	40 005	1 752	34
2 000	154 067	67 353	1 716	38
4 000	311 155	113 475	1 826	44
16 000	1 022 535	301 815	1 854	42

(b) CVT meshes.

nEl	nDoFs	N_Γ	N_Π	it
576	22 035	6 507	289	28
4 608	166 179	26 811	289	43
9 000	320 763	67 491	603	52
15 552	549 939	60 939	289	52
30 375	1 065 693	181 143	1188	52

(c) OCTA meshes.

Strong Scaling. We start looking at the strong scalability of our solver. We fixed the global number of the DoFs and the degree of the VEM approximation at $k = 2$, while we increase the number of processors from 8 to 256. We increase the number of DoFs with respect to the previous tests, we consider CUBE mesh with 954 947 DoFs, a CVT mesh with 627 455 DoFs and an OCTA mesh with 1 065 693 DoFs. In Table 5.4, we report the results related to the three polyhedral meshes with $k = 2$, while in Figure 5.3, we plot the number of iterations and the parallel speedup. We observe that the CPU time T_{ass}, needed to assemble the stiffness matrix and the right-hand-side is scalable, with a speedup very close to the ideal ones, except for the CVT mesh that deteriorates a little. The BDDC method results scalable since the number of GMRES iterations remains bounded and the solution time decreases as the number of the processors increases. Having increased the size of the problem we can see that the effect of the communication time that overcomes the solution time is visible only in the last row of each mesh.

Optimality with respect to the mesh size. We now perform an optimality test with respect to the mesh size: we keep fixed the number of processors at 64 and we increase the number of DoFs, maintaining the degree of the VEM discretization $k = 2$. The results are reported in Table 5.5. As expected, the solver has a quasi-optimal behavior irrespective of the type of polyhedral mesh considered since both the iteration count and the condition number exhibit a logarithmic growth. In the CVT and OCTA meshes this effect is not so evident and the method seems to show an optimal behavior, but again this fact is due to the partitioning that automatically increase the number of the primal constraints.

Optimality with respect to the polynomial degree. We also study the robustness of our preconditioners for the polynomial degree of the VEM discretization $k = 2, 3$ and 4. The tests are performed keeping fixed the number of processors again at 64 and the mesh size. The

Table 5.6: Optimality Test Increasing the polynomial degree k. procs = 64 and nEl for CUBE = 512, CVT = 125 and OCTA = 576.

k	nDoFs	N_Γ	N_Π	it
2	16 787	5 331	125	30
3	40 667	10 683	125	43
4	76 387	17 397	125	56

(a) CUBE meshes.

k	nDoFs	N_Γ	N_Π	it
2	8 945	6 423	702	30
3	19 256	12 495	702	43
4	33 487	19 998	702	59

(b) CVT meshes.

k	nDoFs	N_Γ	N_Π	it
2	22 035	5 331	125	31
3	52 251	10 683	125	46
4	96 675	17 397	125	52

(c) OCTA meshes.

				MUMPS	Block-Schur		BDDC	
	nEl	k	nDoFs	T_{sol}	it	T_{sol}	it	T_{sol}
CUBE	32 768	2	954 947	312	791	123	58	13
	13 824	3	1 009 803	409	NC	NC	82	22
	8 000	4	1 119 523	410	NC	NC	95	34
CVT	8 000	2	627 455	770	522	117	46	19
	4 000	3	666 301	1008	NC	NC	62	39
	2 000	4	571 696	797	NC	NC	74	52
OCTA	30 375	2	1 065 693	287	887	150	50	16
	15 552	3	1 322 571	343	NC	NC	90	34
	4 608	4	741 699	105	NC	NC	47	17

Table 5.7: Solver Comparison among different parallel solver. procs = 64.

results reported in Table 5.6 show that the BDDC solver exhibits, as in the Stokes problem, a mild increase of the iterations count when the degree k increases.

Solver comparison. We conclude with Table 5.7, where we compare the performance of the GMRES method accelerated by the adaptive BDDC preconditioner against the direct solver MUMPS, and the GMRES method accelerated by the Block-Schur preconditioner proposed in [41] and used in the previous Chapter. We do not use in this case the Block-Mass preconditioner proposed in [50], since as we saw before it has a similar behavior of the Block-Schur but with worse performance. Again, we can see that BDDC is significantly faster than the other solvers for all the meshes considered and for the three different degrees of the VEM discretization. We also observe the same phenomena for which the Block-Schur preconditioner is not robust for the degree $k = 3$ and 4, since the GMRES method does not converge (NC in the table).

Conclusions

In this thesis we extended the BDDC algorithms to the class of saddle point problems that arise from divergence-free VEM discretization of the Stokes and Oseen equations. We constructed, analyzed and numerically validated the behavior of this preconditioner in the two and three dimensional framework.

In Chapter 2, we introduced the divergence-free VEM discretization. We recalled the construction of the discrete spaces both in two and in three dimensions, and we showed the technique to compute the polynomial projections. We provided a proof for a Stokes optimal VEM interpolant in the three dimensional framework. We completed the VEM analysis with some convergence tests for the numerical methods that we used in our work.

In Chapter 3 we designed the BDDC methods applied to the saddle point problem that arises from the VEM discretization of the Stokes equations. We described the construction of these algorithms and, in case of a piecewise constant viscosity on the subdomains, we proved a convergence rate estimate of the preconditioned system that is independent of the number of subdomains and polylogarithmic with respect to the ratio H/h, where H denotes the subdomain size and h the mesh size. We confirmed the theoretical estimate showing numerically the scalability and quasi-optimality of the resulting algorithm. In the two dimensional case we proved the robustness of the solver with respect to different polygonal meshes and different mesh partitioning techniques with several numerical tests. We also showed the robustness of the algorithm for the $\nu-$ and deluxe scaling for high coefficient jumps aligned with the subdomains. In the three dimensional framework we numerically proved the scalability and the quasi-optimalty of the preconditioner with several parallel computations including a CPU time analysis.

In Chapter 4 we studied different techniques to enrich the coarse space. In the two dimensional case we constructed two adaptive coarse spaces that we showed to be robust with respect to high jumps in the viscosity coefficient, which can change through the elements. To validate our expectations we provided several numerical tests with different mesh discretizations and different mesh partitioning. We also saw that our new heuristical approach to enrich the coarse space without solving eigenvalue problems, when combined with the deluxe scaling, turned out to be a good competitor for some jump configurations. In the three dimesional framework we also provided an adaptive algorithm and numerically validated its performance with the classical scalability and optimality test also for higher VEM discretizations. We also compared our algorithm with some valuable options of preconditioners seeing that the adaptive BDDC is significantly faster than the other solvers for all the meshes considered and for the different degrees of the VEM discretization. We finally proved the robustness of the adaptive coarse space using a challenging multi-sinker testcase with heterogeneous viscosity.

Finally, in Chapter 5 of the thesis we applied the BDDC algorithm to the non-symmetric saddle point problem that arises from the VEM discretization of the Oseen problem. We provided some numerical simulations including scalability and optimality tests in both two and three dimensions, confirming the expectations for the performance of the preconditioner that is comparable with the symmetric case. We will provide a theoretical analysis in the future.

Future possible extentions of this thesis include the development of BDDC algorithms for VEM discretizations of the time dependent Navier-Stokes equations, Hellinger-Reissner elasticity and coupled Darcy-Stokes problems.

Appendix A

Appendix

A.1 The velocity change of basis

We introduce here the transformation of basis approach used in the two dimensional framework. The idea is to perform a change of basis for the velocity space in such a way that each primal constraint corresponds to an explicit DoF. The strategy used here differs from the classical BDDC transformation of basis [67], since using an adaptive technique we need to enforce multiple constraints for each edge. With this scope we use a generalized transformation of basis approach introduced in [60] that we recall here.

We consider a single macro edge $\mathcal{E}$ of the interface, shared by two subdomains Ω_i and Ω_j. We suppose that during each iteration of the Krylov method, the velocity vector $\mathbf{v}_{\Gamma|\mathcal{E}}$ should fulfills N constraints given by the normalized vectors c_{ij}^l for $l = 1, ..., N$ defined on $\partial\Omega_i \cap \mathcal{E}$(and equal to that on $\partial\Omega_j \cap \mathcal{E}$), i.e., such that

$$c_{ij}^{l^T}\left(\mathbf{v}_{\Gamma|\mathcal{E}}^{(i)} - \mathbf{v}_{\Gamma|\mathcal{E}}^{(j)}\right) = 0 \quad i.e. \quad c_{ij}^{l^T}\mathbf{v}_{\Gamma|\mathcal{E}}^{(i)} = c_{ij}^{l^T}\mathbf{v}_{\Gamma|\mathcal{E}}^{(j)}.$$

We then consider the orthonormalized set of constraint vectors $(c_{ij}^1, ..., c_{ij}^N)$ and we introduce:

$$T_{\mathcal{E},\Pi_{\mathcal{E}}}^{(i)} := [c_{ij}^1, ..., c_{ij}^N].$$

Then we compute a matrix $T_{\mathcal{E},\Delta_{\mathcal{E}}}^{(i)}$, using a modified Grahm-Schmidt algorithm, so that $T_{\mathcal{E}}^{(i)} = [T_{\mathcal{E},\Pi_{\mathcal{E}}}^{(i)}\ T_{\mathcal{E},\Delta_{\mathcal{E}}}^{(i)}]$ is a square matrix and $T_{\mathcal{E}}^{(i)^T}\ T_{\mathcal{E}}^{(i)} = I$. In our specific framework the first coefficient c_{ij}^1 of each edge always corresponds to the no-net-flux condition, while the additional constraints are obtained by computing the coefficients c_{ij}^l with $l = 2, ..., N$ using one of the enrichment techniques for the primal space described in Chapter 4. This transformation is chosen consistently, i.e. for the two subdomains we have

$$T_{\mathcal{E},\Pi_{\mathcal{E}}} := T^{(i)}_{\mathcal{E},\Pi_{\mathcal{E}}} = T^{(j)}_{\mathcal{E},\Pi_{\mathcal{E}}} \quad \text{and} \quad T_{\mathcal{E},\Delta_{\mathcal{E}}} = T^{(i)}_{\mathcal{E},\Delta_{\mathcal{E}}} = T^{(j)}_{\mathcal{E},\Delta_{\mathcal{E}}}.$$

Since for each edge these transformation of basis are independent from the others, the global transformation of basis from the new (non-nodal) basis to the old (nodal) basis has the form

$$T = \begin{bmatrix} I_\Pi & 0 \\ 0 & \text{blockdiag}_{l=1,\dots,N_{\mathcal{E}}}(T_{\mathcal{E}_l}) \end{bmatrix}$$

where I_Π represent the identity matrix for the primal DoFs that are not affected by the transformation (we used a reordering just for sake of simplicity in the notation).

A.2 The pressure change of basis

We briefly recall the transformation of basis for the pressure field needed to introduce the decomposition

$$Q = Q_I \bigoplus Q_0 \quad \text{with} \quad Q_I = \bigoplus_{i=1}^{N} Q_I^{(i)} \tag{A.1}$$

and $Q_0 := \prod_{i=1}^{N}\{q \in \Omega_i | q \textit{ is constant in } \Omega_i\}$, where the elements in $Q_I^{(i)}$ need to satisfy $\int_{\Omega_i} q_I^{(i)} = 0$. The matrix for the change of the base has the classical form as in [67], we recall here the transformation for the subdomain Ω_i, from $Q_I^{(i)} \bigoplus Q_0^{(i)}$ to $Q^{(i)}$. Since the pressure basis functions of degree greater than 1 have null average on the element by definition, we work only on the constant terms that we denote with the subscript 0. We then introduce the matrix transformation for the constant terms

$$T_{p,0} = \begin{bmatrix} 1 & -1 & \dots & -1 \\ 1 & 1 & & \\ \vdots & & \ddots & \\ 1 & & & 1 \end{bmatrix} \tag{A.2}$$

and the transformation on the whole subdomain extending by identity

$$T_p = \begin{bmatrix} T_{p,0} & \\ & I \end{bmatrix}. \tag{A.3}$$

Remark A.2.1. The pressure change of basis is a essential in this preconditioner, as it guarantees the well posedness of the interior problems and of the Schur complements. However, this change of basis modify drastically the sparsity pattern of the matrices. This limits the applicability of the algorithm since the memory required by the subdomain solvers becomes very large when considering real problems. This happens also with the change basis for the velocity space. An approach that avoid these transformations for the pressure and the velocity, has been presented in [101] and used in our three dimensional PETSc version of the algorithm. In our two dimensional framework we explicitly performed both the change of basis, since the problem is smaller.

Bibliography

[1] R. A. Adams and J. J. Fournier, *Sobolev spaces*, Elsevier, 2003.

[2] B. Ahmad, A. Alsaedi, F. Brezzi, L. D. Marini, and A. Russo, *Equivalent Projectors for Virtual Element Methods*, Computers & Mathematics with Applications, 66 (2013), pp. 376–391.

[3] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster, *A fully asynchronous multifrontal solver using distributed dynamic scheduling*, SIAM Journal on Matrix Analysis and Applications, 23 (2001), pp. 15–41.

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

[5] P. F. Antonietti, L. Beirão da Veiga, and G. Manzini, *The Virtual Element Method and its Applications*, Springer International Publishing, 2022.

[6] P. F. Antonietti, S. Berrone, M. Busetto, and M. Verani, *Agglomeration-based geometric multigrid schemes for the Virtual Element Method*, SIAM Journal on Numerical Analysis, 61 (2023), pp. 223–249.

[7] P. F. Antonietti, L. Mascotto, and M. Verani, *A multigrid algorithm for the p-version of the virtual element method*, ESAIM: Math. Model. Numer. Anal., 52 (2018), pp. 337–364.

[8] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. Gropp, et al., *PETSc users manual*, tech. report, Argonne National Laboratory, 2019.

[9] L. Beirão da Veiga, N. Bellomo, F. Brezzi, and L. D. Marini, *Recent results and perspectives of virtual element methods*, Math. Mod. Meth. Appl. Sci., 31 (2021), pp. 2819–3058.

[10] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, and A. Russo, *Basic Principles of Virtual Element Methods*, Math. Mod. Meth. Appl. Sci., 23 (2013), pp. 199–214.

[11] L. Beirão da Veiga, F. Brezzi, L. D. Marini, and A. Russo, *The Hitchhiker's Guide to the Virtual Element Method*, Math. Mod. Meth. Appl. Sci., 24 (2014), pp. 1541–1573.

[12] L. Beirão da Veiga, F. Dassi, and G. Vacca, *The Stokes complex for Virtual Elements in three dimensions*, Mathematical Models and Methods in Applied Sciences, 30 (2020), pp. 477–512.

[13] L. Beirão da Veiga, F. Dassi, and G. Vacca, *Vorticity-stabilized virtual elements for the Oseen equation*, Mathematical Models and Methods in Applied Sciences, 31 (2021), pp. 3009–3052.

[14] L. Beirão da Veiga, C. Lovadina, and G. Vacca, *Divergence Free Virtual Elements for the Stokes Problem on Polygonal Meshes*, ESAIM: Math. Mod. Numer. Anal., 51 (2017), pp. 509–535.

[15] L. Beirão da Veiga, C. Lovadina, and G. Vacca, *Virtual Elements for the Navier–Stokes Problem on Polygonal Meshes*, SIAM J. Numer. Anal., 56 (2018), pp. 1210–1242.

[16] S. Bertoluzza, *Substructuring preconditioners for the three fields domain decomposition method*, Math. Comp., 73 (2004), pp. 659–689.

[17] S. Bertoluzza, M. Pennacchio, and D. Prada, *BDDC and FETI-DP for the virtual element method*, Calcolo, 54 (2017), pp. 1565–1593.

[18] S. Bertoluzza, M. Pennacchio, and D. Prada, *FETI-DP for the Three Dimensional Virtual Element Method*, SIAM J. Numer. Anal., 58 (2020), pp. 1556–1591.

[19] T. Bevilacqua, F. Dassi, S. Zampini, and S. Scacchi, *BDDC preconditioners for virtual element approximations of the three-dimensional Stokes equations*, 2023, https://arxiv.org/abs/2304.09770.

[20] T. Bevilacqua, A. Klawonn, and M. Lanser, *BDDC with adaptively enriched coarse spaces for virtual element discretizations of Stokes' equations*, (2024). In preparation.

[21] T. Bevilacqua and S. Scacchi, *BDDC preconditioners for divergence free virtual element discretizations of the Stokes equations*, Journal of Scientific Computing, 92 (2022), pp. 1–27.

[22] J. Bishop, *A displacement-based finite element formulation for general polyhedra using harmonic shape functions*, Int. J. Numer. Methods Engrg., 97 (2014), pp. 1–31.

[23] D. Boffi, F. Brezzi, and M. Fortin, *Mixed finite element methods and applications*, vol. 44, Springer, 2013.

[24] J. Bonelle and A. Ern, *Analysis of compatible discrete operator schemes for elliptic problems on polyhedral meshes*, ESAIM: M2AN, 48 (2014), pp. 553–581.

[25] J. H. Bramble and J. E. Pasciak, *A domain decomposition technique for Stokes problems*, Appl. Numer. Math., 6 (1990), pp. 251–261.

[26] S. C. Brenner and L.-Y. Sung, *BDDC and FETI-DP without matrices or vectors*, Comput. Meth. Appl. Mech. Eng., 196 (2007), pp. 1429–1435.

[27] S. C. Brenner and L.-Y. Sung, *Virtual element methods on meshes with small edges or faces*, Mathematical Models and Methods in Applied Sciences, 28 (2018), pp. 1291–1336.

[28] F. Brezzi, A. Buffa, and K. Lipnikov, *Mimetic finite differences for elliptic problems*, ESAIM: M2AN, 43 (2009), pp. 277–295.

[29] 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 (2005), pp. 1872–1896.

[30] A. N. Brooks and T. J. Hughes, *Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, Computer methods in applied mechanics and engineering, 32 (1982), pp. 199–259.

[31] E. Burman, M. A. Fernández, and P. Hansbo, *Continuous interior penalty finite element method for Oseen's equations*, SIAM journal on numerical analysis, 44 (2006), pp. 1248–1274.

[32] J. G. Calvo, *On the approximation of a virtual coarse space for domain decomposition methods in two dimensions*, Math. Mod. Meth. Appl. Sci., 28 (2018), pp. 1267–1289.

[33] J. G. Calvo, *An overlapping Schwarz method for virtual element discretizations in two dimensions*, Comput. Math. Appl., 77 (2019), pp. 1163–1177.

[34] A. Cangiani, E. Georgoulis, and P. Houston, *hp-version discontinuous Galerkin methods on polygonal and polyhedral meshes*, Math. Mod. Meth. Appl. Sci., 24 (2014), pp. 2009–2041.

[35] C. Canuto, L. F. Pavarino, and A. B. Pieri, *BDDC preconditioners for continuous and discontinuous Galerkin methods using spectral/hp elements with variable local polynomial degree*, IMA J. Numer. Anal., 34 (2014), pp. 879–903.

[36] L. Chen and J. Huang, *Some error analysis on virtual element methods*, Calcolo, 55 (2018), pp. 1–23.

[37] L. B. da Veiga, K. Lipnikov, and G. Manzini, *Arbitrary-order nodal mimetic discretizations of elliptic problems on polygonal meshes*, SIAM J. Numer. Anal., 49 (2011), pp. 1737–1760.

[38] L. B. da Veiga, K. Lipnikov, and G. Manzini, *The mimetic finite difference method for elliptic problems*, vol. 11 of MS&A. Modeling, Simulation and Applications, Springer, 2014.

[39] F. Dassi, *Vem++, a c++ library to handle and play with the Virtual Element Method*, arXiv preprint arXiv:2310.05748, (2023).

[40] F. Dassi and S. Scacchi, *Parallel block preconditioners for three-dimensional virtual element discretizations of saddle-point problems*, Comput. Meth. Appl. Mech. Eng., 372 (2020), p. 113424.

[41] F. Dassi and S. Scacchi, *Parallel solvers for virtual element discretizations of elliptic equations in mixed form*, Computers & Mathematics with Applications, 79 (2020), pp. 1972–1989.

[42] F. Dassi and G. Vacca, *Bricks for the mixed high-order virtual element method: projectors and differential operators*, Applied Numerical Mathematics, 155 (2020), pp. 140–159.

[43] F. Dassi, S. Zampini, and S. Scacchi, *Robust and scalable adaptive BDDC preconditioners for virtual element discretizations of elliptic partial differential equations in mixed form*, Comput. Meth. Appl. Mech. Eng., 391 (2022), p. 114620.

[44] C. R. Dohrmann, *A preconditioner for substructuring based on constrained energy minimization*, SIAM J. Sci. Comput., 25 (2003), pp. 246–258.

[45] C. R. Dohrmann, *A preconditioner for substructuring based on constrained energy minimization*, SIAM Journal on Scientific Computing, 25 (2003), pp. 246–258.

[46] C. R. Dohrmann and O. B. Widlund, *A BDDC Algorithm with Deluxe Scaling for Three-Dimensional H (curl) Problems*, Comm. Pure Appl. Math., 69 (2016), pp. 745–770.

[47] M. Dryja, J. Galvis, and M. Sarkis, *BDDC methods for discontinuous Galerkin discretization of elliptic problems*, J. Complex., 23 (2007), pp. 715–739.

[48] L. P. Franca and S. L. Frey, *Stabilized finite element methods: II. The incompressible Navier-Stokes equations*, Computer Methods in Applied Mechanics and Engineering, 99 (1992), pp. 209–233.

[49] P. Goldfeld, L. F. Pavarino, and O. B. Widlund, *Balancing Neumann-Neumann preconditioners for mixed approximations of heterogeneous problems in linear elasticity*, Numerische Mathematik, 95 (2003), pp. 283–324.

[50] P. P. Grinevich and M. A. Olshanskii, *An iterative method for the Stokes-type problem with variable viscosity*, SIAM Journal on Scientific Computing, 31 (2009), pp. 3959–3978.

[51] P. Grisvard, *Elliptic problems in nonsmooth domains*, SIAM, 2011.

[52] M. Hanek and J. Šístek, *Application of Multilevel BDDC to the problem of pressure in simulations of incompressible flow*, in Domain Decomposition Methods in Science and Engineering XXVI, Springer, 2023, pp. 299–306.

[53] M. Hanek, J. Šístek, and P. Burda, *An application of the BDDC method to the Navier-Stokes equations in 3-D cavity*, Programs and algorithms of numerical mathematics, (2015), pp. 77–85.

[54] M. Hanek, J. Sistek, and P. Burda, *Multilevel BDDC for Incompressible Navier–Stokes Equations*, SIAM Journal on Scientific Computing, 42 (2020), pp. C359–C383.

[55] P. Hansbo and A. Szepessy, *A velocity-pressure streamline diffusion finite element method for the incompressible Navier-Stokes equations*, Computer Methods in Applied Mechanics and Engineering, 84 (1990), pp. 175–192.

[56] A. Heinlein, A. Klawonn, M. Lanser, and J. Weber, *A frugal FETI-DP and BDDC coarse space for heterogeneous problems*, Universität zu Köln, (2019).

[57] C. Hofer, *Analysis of discontinuous Galerkin dual-primal isogeometric tearing and interconnecting methods*, Math. Mod. Meth. Appl. Sci., 28 (2017), pp. 131–158.

[58] G. Karypis and V. Kumar, *METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices*, FIND THE JOURNAL, (1997).

[59] H. H. Kim, M. Dryja, and O. B. Widlund, *A BDDC Method for Mortar Discretizations Using a Transformation of Basis*, SIAM J. Numer. Anal., 47 (2009), pp. 136–157.

[60] A. Klawonn, M. Kühn, and O. Rheinbach, *Coarse spaces for FETI-DP and BDDC methods for heterogeneous problems: connections of deflation and a generalized transformation-of-basis approach*, Verlag der Österreichischen Akademie der Wissenschaften, (2020).

[61] A. Klawonn, M. Lanser, and A. Wasiak, *Three-level BDDC for Virtual Elements*, tech. report, Universität zu Köln, 2022.

[62] A. Klawonn, M. Lanser, and A. Wasiak, *Adaptive and frugal feti-dp for virtual elements*, Vietnam Journal of Mathematics, 51 (2023), pp. 89–111.

[63] A. Klawonn, P. Radtke, and O. Rheinbach, *FETI-DP with different scalings for adaptive coarse spaces*, PAMM, 14 (2014), pp. 835–836.

[64] A. Klawonn, P. Radtke, and O. Rheinbach, *A comparison of adaptive coarse spaces for iterative substructuring in two dimensions*, Electronic Transactions on Numerical Analysis, 45 (2016), pp. 75–106.

[65] A. Klawonn, P. Radtke, and O. Rheinbach, *A Newton-Krylov-FETI-DP method with an adaptive coarse space applied to elastoplasticity*, in Domain Decomposition Methods in Science and Engineering XXII, Springer, 2016, pp. 293–300.

[66] A. Klawonn and O. B. Widlund, *Dual-primal FETI methods for linear elasticity*, Comm. Pure Appl. Math., 59 (2006), pp. 1523–1572.

[67] A. Klawonn and O. B. Widlund, *Dual-primal FETI methods for linear elasticity*, Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences, 59 (2006), pp. 1523–1572.

[68] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, United States Governm. Press Office Los Angeles, CA, (1950).

[69] J. Li and X. Tu, *A nonoverlapping domain decomposition method for incompressible Stokes equations with continuous pressures*, SIAM J. Numer. Anal., 51 (2013), pp. 1235–1253.

[70] J. Li and O. Widlund, *BDDC algorithms for incompressible Stokes equations*, SIAM J. Numer. Anal., 44 (2006), pp. 2432–2455.

[71] J. Li and O. B. Widlund, *FETI-DP, BDDC, and block Cholesky methods*, Int. J. Numer. Meth. Eng., 66 (2006), pp. 250–271.

[72] K. Lipnikov, G. Manzini, and M. Shashkov, *Mimetic finite difference method*, J. Comput. Phys., 257 (2014), pp. 1163–1227.

[73] J. Mandel, C. R. Dohrmann, and R. Tezaur, *An algebraic theory for primal and dual substructuring methods by constraints*, Applied numerical mathematics, 54 (2005), pp. 167–193.

[74] J. Mandel and B. Sousedík, *Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods*, Computer methods in applied mechanics and engineering, 196 (2007), pp. 1389–1399.

[75] J. Mandel, B. Sousedík, and C. R. Dohrmann, *Multispace and multilevel BDDC*, Computing, 83 (2008), pp. 55–85.

[76] J. Mandel, B. Sousedík, and C. R. Dohrmann, *On multilevel BDDC*, Springer, 2008.

[77] J. Mandel, B. Sousedík, and J. Šístek, *Adaptive BDDC in three dimensions*, Mathematics and Computers in Simulation, 82 (2012), pp. 1812–1831.

[78] S. Mousavi and N. Sukumar, *Numerical integration of polynomials and discontinuous functions on irregular convex polygons and polyhedrons*, Comput. Mech., 47 (2011), pp. 535–554.

[79] S. Natarajan, S. Bordas, and D. Mahapatra, *Numerical integration over arbitrary polygonal domains based on Schwarz–Christoffel conformal mapping*, Int. J. Numer. Meth. Eng., 80 (2009), pp. 103–134.

[80] D.-S. Oh, O. B. Widlund, S. Zampini, and C. R. Dohrmann, *BDDC Algorithms with deluxe scaling and adaptive selection of primal constraints for Raviart–Thomas vector fields*, Math. Comput., 87 (2017), pp. 659–692.

[81] L. F. Pavarino and O. B. Widlund, *Balancing Neumann-Neumann methods for incompressible Stokes equations*, Comm. Pure Appl. Math., 55 (2002), pp. 302–335.

[82] C. Pechstein and C. Dohrmann, *Modern domain decomposition solvers—BDDC, deluxe scaling, and an algebraic approach, slides to a talk at NuMa Seminar, JKU Linz, December 10, 2013.*

[83] J. Peng, S. Shu, J. Wang, and L. Zhong, *Adaptive-multilevel BDDC algorithm for three-dimensional plane wave Helmholtz systems*, Journal of Computational and Applied Mathematics, 381 (2021), p. 113011.

[84] D. D. Pietro and A. Ern, *A hybrid high-order locking-free method for linear elasticity on general meshes*, Comput. Methods Appl. Mech. Engrg., 283 (2015), pp. 1–21.

[85] A. Rand, A. Gillette, and C. Bajaj, *Interpolation error estimates for mean value coordinates over convex polygons*, Adv. Comput. Math., 39 (2013), pp. 327–347.

[86] S. Rjasanow and S. Weisser, *Higher order bem-based fem on polygonal meshes*, SIAM J. Numer. Anal., 50 (2012), pp. 2357–2378.

[87] J. Rudi, G. Stadler, and O. Ghattas, *Weighted BFBT preconditioner for Stokes flow problems with highly heterogeneous viscosity*, SIAM Journal on Scientific Computing, 39 (2017), pp. S272–S297.

[88] B. Sousedík, *Inexact and primal multilevel FETI-DP methods: a multilevel extension and interplay with BDDC*, International Journal for Numerical Methods in Engineering, 123 (2022), pp. 4844–4858.

[89] B. Sousedík, J. Sistek, and J. Mandel, *Adaptive-multilevel BDDC and its parallel implementation*, Computing, 95 (2013), pp. 1087–1119.

[90] L. Tobiska and R. Verfürth, *Analysis of a streamline diffusion finite element method for the Stokes and Navier–Stokes equations*, SIAM Journal on Numerical Analysis, 33 (1996), pp. 107–127.

[91] A. Toselli and O. B. Widlund, *Domain decomposition methods-algorithms and theory*, vol. 34, Springer Science & Business Media, 2006.

[92] X. Tu, *Three-level BDDC in three dimensions*, SIAM Journal on Scientific Computing, 29 (2007), pp. 1759–1780.

[93] X. Tu, *Three-level BDDC in two dimensions*, International journal for numerical methods in engineering, 69 (2007), pp. 33–59.

[94] X. Tu, *A three-level BDDC algorithm for a saddle point problem*, Numerische Mathematik, 119 (2011), pp. 189–217.

[95] X. Tu and J. Li, *BDDC for Nonsymmetric positive definite and symmetric indefinite problems*, in Domain Decomposition Methods in Science and Engineering XVIII, Springer, 2009, pp. 75–86.

[96] X. Tu and B. Wang, *A BDDC algorithm for the Stokes problem with weak Galerkin discretizations*, Comput. Math. Appl., 76 (2018), pp. 377–392.

[97] X. Tu and J. Zhang, *BDDC Algorithms for Oseen problems with HDG Discretizations*, IMA Journal of Numerical Analysis, (2022), p. drac074.

[98] O. B. Widlund, S. Zampini, S. Scacchi, and L. F. Pavarino, *Block FETI–DP/BDDC preconditioners for mixed isogeometric discretizations of three-dimensional almost incompressible elasticity*, Math. Comp., 90 (2021), pp. 1773–1797.

[99] S. Zampini, *Dual-primal methods for the cardiac bidomain model*, Mathematical Models and Methods in Applied Sciences, 24 (2014), pp. 667–696.

[100] S. Zampini, *PCBDDC: a class of robust dual-primal methods in PETSc*, SIAM Journal on Scientific Computing, 38 (2016), pp. S282–S306.

[101] S. Zampini and X. Tu, *Multilevel balancing domain decomposition by constraints deluxe algorithms with adaptive coarse spaces for flow in porous media*, SIAM Journal on Scientific Computing, 39 (2017), pp. A1389–A1415.