Università degli Studi di Pavia

Department of Mathematics
Ph.D. Program in Mathematics
Cycle XXXVII

A family of Virtual Element Methods for advection-dominated problems

Supervisor:
Prof. Carlo Lovadina
Tutor:
Prof. Giancarlo Sangalli

Ph.D. Thesis of:
Manuel Trezzi

Registration number: 508244

Academic year: 2023/2024

Ringraziamenti

Prima di iniziare la discussione di quella che è la mia tesi di dottorato, ci tenevo a ringraziare alcune persone che ritengo essere state fondamentali in questi tre anni.

In primo luogo, la persona a cui va il ringraziamento più grosso è il mio relatore Carlo Lovadina. Queste mie parole non vogliono essere qualcosa di dovuto, scritte per riverenza. Sebbene sulla carta sia stato il mio superiore, ciò che ho visto in lui non è un capo bensì una sorta di padre accademico. Carlo è stato in grado di guidarmi in questo percorso senza la strafottenza di chi è più esperto di te e senza la pretesa di avere da me risultati che servissero ad arricchire il suo curriculum. Nella nostra collaborazione, ha sempre cercato di scegliere la strada migliore per quella che era la mia carriera. Si è sempre fidato silenziosamente delle mie capacità e ciò mi ha trasmesso quel po' di sicurezza di cui avevo bisogno quando credevo di non avere sufficiente talento. È tra le persone che più ammiro dal punto di vista umano, prima ancora che accademico. Sono estremamente grato della possibilità di aver lavorato con lui, potessi tornare indietro rifarei questa scelta un milione di volte.

In secondo luogo volevo ringraziare tutti gli altri professori che mi hanno accompagnato in questo percorso, in particolare Lourenço Beirão da Veiga e Ilaria Perugia. Il primo mi ha accolto senza esitazioni, accettando di collaborare con me nonostante la mia inesperienza. Come Carlo, ha sempre riposto molta fiducia nelle mie capacità e non ha esitato a darmi la possibilità di continuare il mio percorso accademico. La seconda mi ha dato il benvenuto a Vienna durante il mio periodo all'estero. L'esperienza con lei mi ha arricchito molto dal punto di vista accademico, anche lei è sempre stata disponibilissima nei miei confronti. Custodisco con piacere i sinceri complimenti che mi ha fatto una volta terminato il mio periodo da lei. Infine ci tenevo a ringraziare il professor Giancarlo Sangalli che mi ha sostenuto più volte dal punto di vista burocratico ed anche economico.

Un ringraziamento particolare va fatto a quelli che sono stati i miei compagni di dottorato. Se questi tre anni sono stati degli anni speciali è perché ho trovato in voi degli amici fantastici. Mi porto dietro un sacco di ricordi felici legati ai momenti passati in dipartimento con voi. Mi dispiace che le nostre carriere porteranno inevitabilmente a dividerci, so che difficilmente troverò un ambiente come quello pavese. In particolare ci tenevo a menzionare in maniera esplicita i miei compagni di ufficio: Francesca, Giulia e Paolo con i quali ho trascorso la gran parte del mio tempo. Nomino anche i ragazzi del mio stesso ciclo: Ambrogio, Edoardo, Enrico, Matteo, Jonathan e Simone (ed è anche un fantastico coinquilino) coi quali ho legato maggiormente. Ringrazio anche Umberto con il quale è nato un bel lavoro.

Ci tenevo a ringraziare anche mia madre, mio padre e mia nonna. È innegabile che se oggi sono la persona che sono, ottengo certi risultati, il merito è anche di chi mi ha cresciuto e ha costruito intorno a me un ambiente sano. Nonostante le incomprensioni che ogni tanto capitano a tutte le famiglie, ho sempre trovato del terreno fertile e soprattutto felice dal quale far germogliare senza troppi pensieri le mie ricerche.

Infine, un grossissimo ringraziamento deve andare ad Anna che è la cosa più bella nata da questa esperienza pavese. Non ti ringrazio solo per essere la mia compagna, ti ringrazio per tutto ciò che sei e hai rappresentato. Ti ringrazio di permettermi di essere me stesso, ti ringrazio per essere il porto sicuro al quale fare ritorno quando sono stanco e ti ringrazio soprattutto per sostenere ogni mia scelta, spronandomi a non arrendermi. Hai reso una vita che già era bella qualcosa di unico e speciale. Inoltre, scusa se ogni tanto ti faccio tornare da judo con qualche livido in più, fa parte del mestiere. Ringrazio anche la sua famiglia che mi ha accettato come se fossi il loro terzo figlio.

Contents

Introduction

The Virtual Element Method (VEM), introduced in 2013 [8], represents an advanced evolution of the classical Finite Element Method (FEM) to solve Partial Differential Equations (PDE). One of its most attractive features is its ability to manage complex polygonal/polyhedral meshes, allowing each element to adopt arbitrary shapes. This flexibility is particularly advantageous in addressing mechanical problems involving intricate geometries; see for instance [4] and [12] and the references therein. The VEM has found applications in numerous fields, including structural mechanics, fluid dynamics, and geophysics, owing to its ability to handle complex domain geometries and its robustness in numerical simulations. The term *virtual* in VEM highlights that it does not necessitate explicit knowledge of the basis functions; instead, only certain information is needed to construct the stiffness matrix.

Let us now mention other polygonal methods that have been adopted to solve PDEs, without claiming to be exhaustive:

- the Extend Finite Element Method (E-FEM) [44],
- the Hybrid High-Order method (HHO) [40, 38],
- the Mimetic Finite Differences method (MFD) [29],
- the Polygonal Discontinuous Galerkin Method (polyDG) [3],
- the Polygonal Finite Element Method (P-FEM) [56],
- the weak Galerkin method (wG) [58],
- the Recovered FEM (R-FEM) [42], in particular this method has some contact point with our analysis in the definition of an Oswald interpolant.

The core concept of the VEM is that each local discrete space includes polynomials up to a specified degree k. Additionally, the local spaces contain functions defined implicitly as solutions to a PDE related to the one being solved. When constructing the stiffness matrix, special attention is given to cases where one of the two entries is a polynomial of degree up to k. For the remaining part of the stiffness matrix, it suffices to approximate it with a second bilinear form that scales similarly to the original bilinear form. This approximation term is known as the *stabilization term.*

This thesis introduces a family of VEM specifically designed for advection-dominated problems, which are well-documented in the literature for producing unsatisfactory numerical solutions when the Péclet number is not sufficiently small. There is extensive research in the context of FEM that addresses this issue, offering various stabilized methods that are quasi-robust in the hyperbolic limit. Theoretically, a method is considered quasi-robust if, given a sufficiently regular solution and data, it provides error estimates that are uniform with respect to the diffusion parameter in a norm that also directly controls the convective term. Some of the well-known approaches in this area include:

- upwind Discontinuous Galerkin schemes [41, 54, 30],
- Streamline Upwind Petrov-Galerkin (SUPG) and variants [31],

- Continuous Interior Penalty (CIP) [43, 35, 33],
- Local Projection Stabilization (LPS) [47, 50].

The Virtual Element Method is particularly suitable for advection-dominated problems due to its flexible mesh construction and handling. VEM allows for more localized refinement procedures and the seamless integration of fine meshes with coarser ones, which is especially useful in the presence of layers. However, the presence of projection operators in VEM can alter the structure of the convective term, making it challenging to devise and analyze quasi-robust VEM schemes. Notable exceptions include the SUPG and LPS approaches detailed in [21, 23, 14] and [48], respectively. Since three of the most popular stabilization techniques, namely SUPG, LPS and CIP, have their own strongly defined set of assets/drawbacks, broadening the available approaches with CIP schemes is important for the VEM technology. Specifically, the thesis focuses on a family of VEM based on the CIP stabilized technique for two key equations on a domain $\Omega \subset \mathbb{R}^2$: the advection-diffusion-reaction equation and the Oseen equations. These equations are often used as model problems for more complex equations, such as the Navier-Stokes equations. In this context, our current contribution can be seen as a first step toward the design of quasi-robust methods for incompressible fluid problems.

For the advection-diffusion equation, we begin by presenting a conforming VEM as detailed in [17], which incorporates a CIP-like term into the weak formulation. Specifically, we include a bilinear form that measures the jumps in the gradients of the projection of the virtual functions. In [32], it was proved that it is a “minimal stabilization” as it adds the minimal positive term guaranteeing control on piecewise polynomial convection, Assuming, as is common in most publications on the subject, the presence of a uniformly positive reaction term, we are able to develop quasi-robust error estimates for our method with respect to a norm that effectively controls the convective term when quasi-uniform mesh families are employed. In the absence of a reaction term, we demonstrate improved error estimates compared to a non-stabilized scheme, but these improvements are shown only under the assumption of piecewise polynomial convection data. In particular, the well-posedness of the discrete problem is established by proving the inf-sup condition with respect to a norm that includes both “symmetric terms” related to the diffusion bilinear form, the reaction bilinear form, and the jumps, as well as a “skew-symmetric term” arising from the convection term. To control this part of the norm, we devise an Oswald-like interpolant that maps piecewise polynomial functions into the virtual element space by leveraging the DoFs from each element. Due to the definition of this interpolant, we require the quasi-uniformity of the mesh rather than just local quasi-uniformity. Finally, to achieve the quasi-robustness of the method, we impose the boundary condition weakly using a Nitsche-type technique. This approach ensures the stability and accuracy of the method, even in challenging advection-dominated regimes.

For the same equation, we also present a non-conforming VEM as described in [49]. Our interest in the non-conforming setting stems from the fact that the design, implementation, and analysis of non-conforming VEMs are independent of the spatial dimension, making the analysis straightforwardly extendable to the 3D case. Additionally, non-conforming methods are often employed to avoid locking phenomena in the simulation of incompressible fluids, such as using the Crouzeix-Raviart element in FEM. The analysis for the non-conforming VEM is very similar to the conforming method; we need to account some modifications due to the nonconformity of the discrete space. The main differences between [49] and [17] are:

- we discretize the variational formulation of the problem without skew-symmetrizing the discrete advection form,
- we use Nitsche method to impose Dirichlet boundary conditions as in [17], but we propose a symmetric version of it, which is robust in the vanishing advection limit.

Indeed, in situations where the advection term vanishes, only the symmetric version of the Nitsche method, which leads to an adjoint-consistent discrete scheme, can maintain optimal L^2-error estimates for regular problems. A preliminary comparison between conforming and non-conforming VEMs for the advection-diffusion-reaction equation was conducted in [36]. This paper focuses solely on the elliptic case, without considering the hyperbolic limit. The authors show that the two methods are equivalent in a diffusion-dominated regime. However, in a reaction-dominated regime, the non-conforming method provides a better approximation of the solution near the corners of the domain. The authors do not introduce any stabilization to address the hyperbolic limit. At the end of the presentation of our two methods, we compare their performance in the advection-dominated case. We show that two methods are equivalent by comparing the numerical results.

The Nitsche method [52, 46] is a popular technique for weakly imposing boundary conditions in finite element methods. It allows for the implementation of boundary conditions in a weak sense without requiring a Lagrange multiplier. A conforming VEM version of the Nitsche term was already devised in [25] in a more general setting. For the best of our knowledge the Nitsche method for the nonconforming VEM has never been discussed before [49]. Since both of the previous cited methods require the imposition of boundary conditions using a Nitsche-type technique, we will first discuss how to devise a VEM version (conforming and nonconforming) of the Nitsche method before presenting the specific details of the conforming and non-conforming VEMs for advection-diffusion-reaction. The development of the Nitsche method within the VEM follows the standard approach used in VEM implementations. Taking the usual Nitsche bilinear form, some polynomial projections are introduced where the direct computation of virtual functions is not possible. Specifically, in the nonconforming version, because the trace of a virtual function on an edge is unknown, a polynomial projection orthogonal to the L^2-norm of each boundary edge must be used. Other polynomial projections are inserted in the normal derivative of the virtual function in both of the methods. We will prove that both of the versions are well-posed and they produce numerical solutions that converge with the optimal rate of convergence.

Finally, we introduce a VEM with a CIP stabilization for the Oseen equations. This problem is the linearized version of the Navier-Stokes equations and represents a step towards developing stable VEMs for problems in fluid dynamics. A first FEM with CIP stabilization was presented in [34]. However, this method has the disadvantage of not being pressure-robust, meaning that the error estimate for the velocity depends on the pressure. To address this issue, a pressure-robust FEM with a three-level CIP stabilization is introduced in [7]. Inspired by [7], our objective is to develop a pressure robust VEM that achieves stability solely through jump operators applied to the interior edges of the mesh. We recall that, in [18], a VEM for the Stokes problem was proposed that achieves divergence-free conditions by ensuring that the divergence of a virtual velocity is included in the space of the pressures. However, this requirement does not entirely eliminate the dependence on pressure in the error analysis of the velocity. A slight dependence on the pressure still exists due to the approximation of the right-hand side. As a result, the discrete scheme does not fully remove the pressure dependence but significantly reduces it. The method that we propose controls the jumps of $(\boldsymbol{\nabla}\mathbf{u})\boldsymbol{\beta}$ through a three-level CIP-form. Specifically, it controls the polynomial part of the jumps of $(\boldsymbol{\nabla}\mathbf{u})\boldsymbol{\beta}$, the jump of the curl and the jump of the gradient of the curl. Unfortunately, the theoretical analysis of this method is not yet fully developed and will be the focus of our future research. Currently, we can demonstrate the well-posedness of the discrete method by proving coercivity and presenting the numerical results. Although an error analysis has not been formulated, we have some ideas on how to approach it.

The thesis is organized as follows: the first introductory chapter presents both the conforming and nonconforming VEMs for the Poisson problem. In the second chapter, we discuss the implementation of the Nitsche method in both the conforming and nonconforming settings. The third chapter introduces the conforming CIP-stabilized VEM for the advection-diffusion-reaction equation, while the fourth chapter focuses on the nonconforming VEM. The fifth chapter discusses the VEM for the Oseen equations. Finally, the last chapter provides implementation notes on the methods presented.

Chapter 1

The Virtual Element Method

In this chapter, we introduce the basic concepts of the VEM. Specifically, we will focus our attention on two different VEMs: the conforming method, originally presented in [8], and the nonconforming method devised in [5]. Both methods are introduced for the Poisson equation. The difference between the conforming VEM and the nonconforming VEM lies in the boundary conditions of the local virtual element spaces. In the conforming VEM, the virtual functions solve a Poisson problem with Dirichlet boundary conditions. In contrast, the nonconforming virtual functions solve a Poisson problem with Neumann boundary conditions. Additionally, the conforming VEM requires the continuity of virtual functions across element boundaries, whereas in the nonconforming VEM, the virtual functions are allowed to have discontinuities. However, the jumps of these functions are controlled when tested against certain polynomials.

This chapter is structured as follows: in the first section, we introduce the model problem, in the second section, we discuss the domain tessellation and some notation. The conforming method is presented in the third section, while the nonconforming method is introduced in the fourth section.

1.1 The Poisson Equation

As a model problem, we present the Poisson problem, the easiest elliptic PDE to consider. Given a domain $\Omega \subset \mathbb{R}^2$ with polygonal boundary Γ, the strong form of the Poisson problem with homogeneous Dirichlet boundary conditions reads as:

$$\begin{cases} -\Delta u = f & \text{in } \Omega \,, \\ \quad u = 0 & \text{on } \Gamma \,. \end{cases} \tag{1.1}$$

A possible variational formulation of (1.1) reads as:

$$\begin{cases} \text{find } u \in H_0^1(\Omega) \text{ such that:} \\ a(u,v) = \mathcal{F}(v) \quad \forall v \in H_0^1(\Omega) \,, \end{cases} \tag{1.2}$$

where the bilinear form $a(\cdot,\cdot) : H_0^1(\Omega) \times H_0^1(\Omega) \to \mathbb{R}$ and the linear form $\mathcal{F} : H_0^1(\Omega) \to \mathbb{R}$ are defined as

$$a(u,v) := \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}\Omega \,, \quad \text{and} \quad \mathcal{F}(v) := \int_\Omega f \, v \,\mathrm{d}\Omega \,, \tag{1.3}$$

and the load term satisfies $f \in L^2(\Omega)$. The problem (1.2) is obtained by multiplying the first equation in (1.1) by a sufficiently smooth function v, integrating over the domain Ω, using integration by parts on the left-hand side, and noting that v is zero on the boundary

$$-\int_\Omega \Delta u \, v \,\mathrm{d}\Omega = \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}\Omega - \int_\Gamma (\nabla u \cdot \boldsymbol{n}) \, v \,\mathrm{d}s = \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}\Omega = \int_\Omega f \, v \,\mathrm{d}\Omega \,,$$

where $\boldsymbol{n}$ is the outward unit normal vector to the boundary. Thanks to the Poincarè inequality, we recall that the space $H^1_0(\Omega)$ equipped with the norm

$$\|v\| \coloneqq |v|_{1,\Omega}\,,$$

is an Hilbert space. On $H^1_0(\Omega)$, the following estimates hold

$$\begin{aligned} |a(u,v)| &\leq \|u\|\,\|v\| && \forall u,v \in H^1_0(\Omega)\,,\\ a(u,u) &\geq \|u\|^2 && \forall u \in H^1_0(\Omega)\,. \end{aligned}$$

The existence and uniqueness of a solution $u \in H^1_0(\Omega)$ for problem (1.2) is guaranteed by the Lax-Milgram lemma. Here, we present the Lax-Milgram lemma in a very general setting.

Lemma 1.1 (Lax-Milgram)**.** *Let H be an Hilbert space equipped with the norm $\|\cdot\|_H$ and let $b(\cdot,\cdot) : H \times H \to \mathbb{R}$ be a bilinear form on H that satisfies*

$$\begin{aligned} |b(u,v)| &\leq c_*\|u\|_H\,\|v\|_H && \forall u,v \in H\,,\\ b(u,u) &\geq c^*\|u\|^2_H && \forall u \in H\,, \end{aligned}$$

for two positive constants c_*, $c^* \in \mathbb{R}$. Given a continuous linear form $\mathcal{F}_H : H \to \mathbb{R}$, the problem

$$\begin{cases} \text{find } u \in H \text{ such that:}\\ b(u,v) = \mathcal{F}_H(v) \quad \forall v \in H\,, \end{cases}$$

admits a unique solution.

1.2 Tesselation of the domain

We consider a sequence $\{\,\Omega_h\,\}_h$ of decompositions of the domain Ω. We assume that these tessellations are composed of non-overlapping (open) polygons $E \in \Omega_h$. Here, h denotes the maximum of the diameters of the elements in Ω_h

$$h \coloneqq \max_{E\in\Omega_h} h_E\,,$$

where h_E is the diameter of the element $E \in \Omega_h$. The key assumption in any VEM is that every element $E \in \Omega_h$ is star-shaped with respect to a disk. This assumption allows us to extend the approximation results to polygonal elements. In Figure 1.1, an element that is not star-shaped with respect to any disk is depicted alongside an element that satisfies this condition. We suppose that $\{\,\Omega_h\,\}_h$ satisfies the following assumption:
(A1) Mesh assumption. There exists a positive constant ρ such that for any $E \in \{\,\Omega_h\,\}_h$:

- E is star-shaped with respect to a ball B_E of radius grater or equal than $\rho\, h_E$,
- any edge e of E has length grater or equal than $\rho\, h_E$.

Remark 1.2. *In the original VEM work [8], the assumptions differ from those described. In that paper, the second condition concerns to the distances between vertices, specifically requiring that the distance $d_{i,j}$ between the i-th and j-th vertices satisfies*

$$d_{i,j} \geq p h_E\,.$$

*This condition is more stringent than the one described in **(A1)**, which allows for the existence of elements such as highly elongated rectangles. The assumption in **(A1)** first appears in [1].*

Remark 1.3. *It is possible to relax the conditions in **(A1)** further. In addition to maintaining the first condition in **(A1)**, a weaker version of the second condition is considered in [16] and [28]. This relaxed assumption requires that there exists a positive integer N such that the number of edges of every polygon $E \in \Omega_h$ is bounded by N. Importantly, the integer N does*

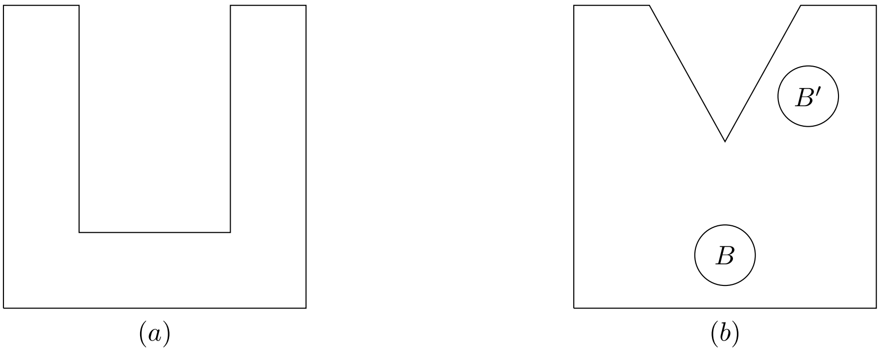


Figure 1.1: The element depicted in (a) is not star shaped with respect any ball, while the element in (b) is star shaped with respect to the ball B but not with respect to ball B'.

*not depend on the mesh size h. An even weaker assumption is utilized in [19], allowing for the existence of meshes that are formed by agglomerating, gluing, or cracking existing meshes. However, since our focus is not on minimizing mesh assumptions, we work with the conditions **(A1)**. Interested readers can find further details on these aspects in [4]. For completeness, it is worth noting that similar relaxation of conditions can also apply to the nonconforming method. Specifically, one can require that the ratio between the lengths of two adjacent edges is bounded, as discussed in [4] and [24].*

From now on, we denote by $|E|$ and $\boldsymbol{n}^E$ the area and the unit outward normal of the polygon E, respectively. The restriction of the unit normal to an edge $e \subset \partial E$ is denoted with $\boldsymbol{n}^e$. The intersection between the boundary of a polygon and the boundary of the domain is denoted with $\Gamma_E = \Gamma \cap \partial E$. The set of edges of a tessellation Ω_h is denoted by $\mathcal{E}_h$. This set is divide into internal edges and boundary edges

$$\mathcal{E}_h = \mathcal{E}_h^o \cup \mathcal{E}_h^\partial \,.$$

Let's introduce some basic spaces that will be useful later on. Given two positive integers n and m, and $p \in [0, +\infty]$, for any $E \in \Omega_h$ we define

- $\mathbb{P}_n(E)$: the set of polynomials on E of degree lesser or equal than n (with $\mathbb{P}_{-1}(E) = \{0\}$),
- $\mathbb{P}_n(e)$: the set of polynomials on e of degree lesser or equal than n (with $\mathbb{P}_{-1}(e) = \{0\}$),
- $\mathbb{P}_n(\Omega_h) := \{q \in L^2(\Omega) \quad \text{such that} \quad q|_E \in \mathbb{P}_n(E) \quad \text{for all } E \in \Omega_h\}$,
- $W^m_p(\Omega_h) := \{v \in L^2(\Omega) \text{ such that } v|_E \in W^m_p(E) \text{ for all } E \in \Omega_h\}$ equipped with the broken norm and seminorm

  $$\|v\|^p_{W^m_p(\Omega_h)} := \sum_{E \in \Omega_h} \|v\|^p_{W^m_p(E)}\,, \qquad |v|^p_{W^m_p(\Omega_h)} := \sum_{E \in \Omega_h} |v|^p_{W^m_p(E)}\,, \qquad \text{if } 1 \leq p < \infty\,,$$
  $$\|v\|_{W^m_p(\Omega_h)} := \max_{E \in \Omega_h} \|v\|_{W^m_p(E)}\,, \qquad |v|_{W^m_p(\Omega_h)} := \max_{E \in \Omega_h} |v|_{W^m_p(E)}\,, \qquad \text{if } p = \infty\,,$$

  In the case $p = 2$, we set

  $$\|v\|^2_{m,\Omega_h} := \|v\|^2_{W^m_p(\Omega_h)}\,, \qquad |v|^2_{m,\Omega_h} := |v|^2_{W^m_p(\Omega_h)}\,.$$

We also introduce the following polynomial projections, which are essential in any VEM implementation:

- the $\boldsymbol{L^2}$**-projection** $\Pi^{0,E}_n \colon L^2(E) \to \mathbb{P}_n(E)$, given by

  $$\int_E q_n (v - \Pi^{0,E}_n v)\,\mathrm{d}E = 0 \qquad \text{for all } v \in L^2(E) \text{ and } q_n \in \mathbb{P}_n(E)\,,$$

  with obvious extensions for functions defined on an edge $\Pi^{0,e}_n \colon L^2(e) \to \mathbb{P}_n(e)$, and for vector-valued functions $\boldsymbol{\Pi}^{0,E}_n \colon [L^2(E)]^2 \to [\mathbb{P}_n(E)]^2$ and $\boldsymbol{\Pi}^{0,e}_n \colon [L^2(e)]^2 \to [\mathbb{P}_n(e)]^2$,

- the ***H***$\boldsymbol{H^1}$**-seminorm projection** $\Pi_n^{\nabla,E}\colon H^1(E) \to \mathbb{P}_n(E)$, defined by

$$\begin{cases} \displaystyle\int_E \nabla q_n \cdot \nabla(v - \Pi_n^{\nabla,E} v)\,\mathrm{d}E = 0 \quad \text{for all } v \in H^1(E) \text{ and } q_n \in \mathbb{P}_n(E), \\ P_0(v - \Pi_n^{\nabla,E} v) = 0\,, \end{cases}$$

  here $P_0 : H^1(E) \to \mathbb{R}$ is any projection operator onto constants. More details on the choice of P_0 will be discussed later.

The global piecewise counterparts of these operators are defined as $\Pi_n^0\colon L^2(\Omega_h) \to \mathbb{P}_n(\Omega_h)$, $\Pi_n^\nabla\colon H^1(\Omega_h) \to \mathbb{P}_n(\Omega_h)$, and $\boldsymbol{\Pi}_n^0\colon [L^2(\Omega_h)]^2 \to [\mathbb{P}_n(\Omega_h)]^2$ defined by

$$(\Pi_n^0 v)|_E = \Pi_n^{0,E} v\,, \qquad (\Pi_n^\nabla v)|_E = \Pi_n^{\nabla,E} v\,, \qquad (\boldsymbol{\Pi}_n^0 v)|_E = \boldsymbol{\Pi}_n^{0,E} v\,, \qquad \text{for all } E \in \Omega_h\,.$$

We recall a classical result for polynomials on star-shaped domains (see, for instance, [27]).

Lemma 1.4 (Polynomial approximation)**.** *Under assumption **(A1)**, for any $E \in \Omega_h$ and for any sufficiently smooth function ϕ defined on E, we have that*

$$\begin{aligned} &\|\phi - \Pi_n^{0,E}\phi\|_{W_p^m(E)} \lesssim h_E^{s-m}|\phi|_{W_p^s(E)} && s,m \in \mathbb{N},\ m \le s \le n+1,\ p = 1,\dots,\infty\,, \\ &\|\phi - \Pi_n^{\nabla,E}\phi\|_{m,E} \lesssim h_E^{s-m}|\phi|_{s,E} && s,m \in \mathbb{N},\ m \le s \le n+1,\ s \ge 1\,, \\ &\|\nabla\phi - \boldsymbol{\Pi}_n^{0,E}\nabla\phi\|_{m,E} \lesssim h_E^{s-1-m}|\phi|_{s,E} && s,m \in \mathbb{N},\ m+1 \le s \le n+2\,. \end{aligned}$$

We also have a counterpart of this result for projections defined on an edge.

Lemma 1.5 (Polynomial approximation on edges)**.** *Let $e \in \mathcal{E}_h^o$ be an internal edge and let $E, K \in \Omega_h$ be such that $e \subset \partial E \cap \partial K$. Then, for every sufficiently smooth function φ defined on $E \cup K$, we have that*

$$\|\varphi - \Pi_n^{0,e}\varphi\|_{0,e} \lesssim h_e^{m-\frac{1}{2}}(|\varphi|_{m,E} + |\varphi|_{m,K}) \qquad m,n \in \mathbb{N},\ 1 \le m \le n+1\,.$$

Proof. Considering the element E, for $m, n \in \mathbb{N}$, with $1 \le m \le n+1$, we have

$$\begin{aligned} \|\varphi - \Pi_n^{0,e}\varphi\|_{0,e} \le \|\varphi - \Pi_n^{0,E}\varphi\|_{0,e} &\lesssim \left(h_E^{-1/2}\|\varphi - \Pi_n^{0,E}\varphi\|_{0,E} + h_E^{1/2}|\varphi - \Pi_n^{0,E}\varphi|_{1,E}\right) \\ &\lesssim h_E^{m-\frac{1}{2}}|\varphi|_{m,E}\,. \end{aligned}$$

Since the same estimate applies also to the element K, the thesis follows. □

1.3 The conforming VEM

In this section, we present the conforming VEM, following the approach outlined in [8]. Therefore, we avoid introducing the enhanced virtual element space, which will be discussed in the following chapters.

1.3.1 The virtual element spaces

Given a positive integer k and an element $E \in \Omega_h$ with n_E edges, we define

$$V_h^k(E) \coloneqq \left\{ v_h \in H^1(E) \text{ such that } \Delta v_h \in \mathbb{P}_{k-2}(E),\ v_h|_{\partial E} \in \mathbb{B}_k(\partial E) \right\}, \tag{1.4}$$

where

$$\mathbb{B}_k(\partial E) \coloneqq \left\{ v_h \in \mathcal{C}^0(\partial E) \text{ such that } \forall e \subset \partial E,\ v_h|_e \in \mathbb{P}_k(e) \right\}\,.$$

This is the space originally introduced in [8]. In details, we can note that:

- for $k = 1$, the space $V_h^1(E)$ contains harmonic functions whose trace is a piecewise linear polynomial. Therefore, a function $v_h \in V_h^1(E)$ is uniquely determined only by its values at the vertices of the polygon,

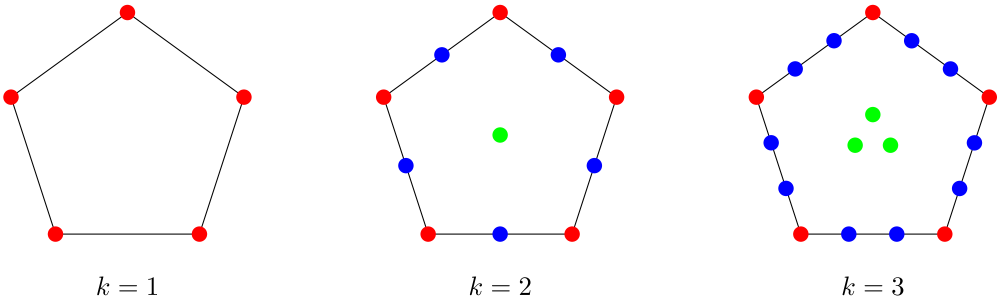


Figure 1.2: Degrees of freedom on a pentagon for $k = 1, 2, 3$. In red the DoFs associated to $\mathcal{V}_c^{E,k}$, in blue the ones associated to $\mathcal{E}_c^{E,k}$ and in green the ones associated to $\mathcal{P}_c^{E,k}$.

- for $k = 2$, the trace of a function $v_h \in V_h^2(E)$ is a quadratic polynomial. The trace is uniquely determined by the values at the vertices and at a midpoint of each edge. Moreover, the laplacian of v_h is constant. Consequently, the dimension of the space $V_h^2(E)$ is $2n_E + 1$.

A very important property is that the space of polynomials up to degree k is contained in the virtual element space $V_h^k(E)$. In general, the following set of linear operators constitutes a set of degrees of freedom (DoFs) for the space $V_h^k(E)$:

- $\mathcal{V}_c^{E,k}$: the pointwise values of v_h at the vertexes of the polygon E,
- $\mathcal{E}_c^{E,k}$: the values of v_h at $k-1$ internal points of a Gauss-Lobatto quadrature formula for every edge $e \subset \partial E$,
- $\mathcal{P}_c^{E,k}$: the moments up to the order $k-2$ on E:

$$\mu_E^{\boldsymbol{\alpha}}(v_h) := \frac{1}{|E|}\int_E v_h \left(\frac{\mathbf{x}-\mathbf{x}_E}{h_E}\right)^{\boldsymbol{\alpha}} \mathrm{d}E \quad |\boldsymbol{\alpha}| \leq k-2\,, \tag{1.5}$$

  where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T$ is a multi-index and $\mathbf{x}_E$ denotes the centroid of the polygon E.

In fact, we have the following unisolvence result:

Proposition 1.6 (Unisolvence)**.** *Let $E \in \Omega_h$, the linear operators $\mathcal{V}_c^{E,k}$ plus $\mathcal{E}_c^{E,k}$ plus $\mathcal{P}_c^{E,k}$ are unisolvent for the space $V_h^k(E)$.*

Proof. We want to show that if a function $v_h \in V_h^k(E)$ satisfies all operators in $\mathcal{V}_c^{E,k}$, $\mathcal{E}_c^{E,k}$ and $\mathcal{P}_c^{E,k}$ being equal to zero, then $v_h \equiv 0$. By integrating by parts, we observe

$$|v_h|_{1,E}^2 = \int_E \nabla v_h \cdot \nabla v_h \,\mathrm{d}E = -\int_E v_h\, \Delta v_h \,\mathrm{d}E + \int_{\partial E} (\nabla v_h \cdot \boldsymbol{n}^E)\, v_h \,\mathrm{d}s = 0\,.$$

The first integral is zero because $\Delta v_h \in \mathbb{P}_{k-2}(E)$ and the operators in $\mathcal{P}_c^{E,k}$ are zero. The second integral is zero since the trace of v_h is zero on ∂E. Therefore, we conclude that $\nabla v_h \equiv 0$ in E, hence v_h is a constant function. Since the trace of v_h is zero, this constant must be 0. Thus $v_h \equiv 0$. □

Remark 1.7. *It is not necessary for the points in $\mathcal{E}_c^{E,k}$ to come from a Gauss-Lobatto quadrature formula. We can select $k-1$ uniformly distributed points. However, the assumption that these points belong to a Gauss-Lobatto quadrature formula is useful in the implementation of the code because it speeds up execution.*

The DoFs are depicted in Figure 1.2. The dimension N_E of the local virtual element space $V_h^k(E)$ is given by

$$N_E := k\, n_E + \frac{(k-1)(k-2)}{2}\,.$$

Thanks to the definition of the DoFs, we are able to compute the projection operator $\Pi_k^{\nabla,E} : V_h^k(E) \to \mathbb{P}_k(E)$ introduced in the previous section. Given $v_h \in V_h^k(E)$ and $p_k \in \mathbb{P}_k(E)$, integrating by parts, we obtain

$$\int_E \nabla v_h \cdot \nabla p_k \,\mathrm{d}E = -\int_E v_h \,\Delta p_k \,\mathrm{d}E + \int_{\partial E} (\nabla p_k \cdot \boldsymbol{n}^E)\, v_h \,\mathrm{d}s$$

The first integral is computable thanks to the DoFs in $\mathcal{P}_c^{E,k}$, while the second is computable since we know the trace on the boundary of a virtual function (see $\mathcal{V}_c^{E,k}$ and $\mathcal{E}_c^{E,k}$).

Remark 1.8. *This projector operator is defined up to a projection onto constants $P_0 : V_h^k(E) \to \mathbb{R}$. A reasonable choice for this operator is*

$$\begin{aligned} P_0(v_h - \Pi_k^{\nabla,E} v_h) &= \int_{\partial E} v_h - \Pi_k^{\nabla,E} v_h \,\mathrm{d}s = 0 \qquad \text{if } k = 1\,, \\ P_0(v_h - \Pi_k^{\nabla,E} v_h) &= \int_E v_h - \Pi_k^{\nabla,E} v_h \,\mathrm{d}E = 0 \qquad \text{if } k \geq 2\,. \end{aligned} \tag{1.6}$$

In [10], it was proved that this choice guarantees that $\Pi_k^{\nabla,E} = \Pi_k^{0,E}$ if $k = 1, 2$.

If $k \geq 2$, we can compute the projection operator $\Pi_{k-2}^{0,E} : V_h^k(E) \to \mathbb{P}_{k-2}(E)$ that is orthogonal with respect to the $L^2(E)$-inner product. Exploiting the degrees of freedom in $\mathcal{P}_c^{E,k}$, we impose

$$\int_E \left(v_h - \Pi_{k-2}^{0,E} v_h\right) p_k \,\mathrm{d}E = 0 \quad \forall p_k \in \mathbb{P}_{k-2}\,.$$

We have developed the virtual element space corresponding to an element $E \in \Omega_h$. Next, we aim to define the virtual element space associated with the entire domain Ω. Considering a tessellation Ω_h composed of n_p polygons and a positive integer k, we define the space

$$V_h^k(\Omega_h) \coloneqq \left\{ v_h \in H_0^1(\Omega) \quad \text{such that} \quad v_h|_E \in V_h^k(E)\,, \quad \text{for all } E \in \Omega_h \right\}\,. \tag{1.7}$$

The space is thus constructed by assembling the local spaces with continuity across the element boundaries. The following set of linear operators is a set of DoFs for the global space $V_h^k(\Omega_h)$:

- $\mathcal{V}_c^k$: the values of v_h at the interior vertices,
- $\mathcal{E}_c^k$: the values of v_h at $k-1$ points on each interior edge e,
- $\mathcal{P}_c^k$: the moments up to order $k-2$ for each element $E \in \Omega_h$.

Following the same procedure in Proposition 1.6, we can prove that these linear operators are unisolvent for the space $V_h^k(\Omega_h)$. Therefore, the dimension N_{Ω_h} of the space $V_h^k(\Omega_h)$ is given by

$$N_{\Omega_h} = n_v^o + (k-1)|\mathcal{E}_h^o| + n_p \frac{(k-2)(k-1)}{2}\,,$$

where n_v^o represents the number of internal vertices in the decomposition Ω_h, $|\mathcal{E}_h^o|$ is the cardinality of $\mathcal{E}_h^o$ which is the number of internal edges. We conclude the presentation of the virtual element space by recalling an optimal approximation result for $V_h^k(\Omega_h)$. More details can be found in [27].

Lemma 1.9 (Approximation with conforming virtual element functions)**.** *Under assumption **(A1)**, for any $v \in H_0^1(\Omega) \cap H^{s+1}(\Omega_h)$, there exists $v_{\mathcal{I}} \in V_h^k(\Omega_h)$, such that for all $E \in \Omega_h$,*

$$\|v - v_{\mathcal{I}}\|_{0,E} + h_E \|\nabla(v - v_{\mathcal{I}})\|_{0,E} \lesssim h_E^{s+1} |v|_{s+1,E}\,,$$

where $0 < s \leq k$.

1.3.2 The virtual element forms and the discrete problem

We observe that the forms introduced in (1.3) can be decomposed in local contributions

$$a(u,v) =: \sum_{E\in\Omega_h} a^E(u,v)\,, \quad \text{and} \quad \mathcal{F}(v) =: \sum_{E\in\Omega_h} \mathcal{F}^E(v)\,,$$

by restricting the integral over an element E. Given an element $E \in \Omega_h$, we observe that, for two virtual functions $u_h, v_h \in V_h^k(E)$, it is not possible to compute the quantity

$$a^E(u_h, v_h) = \int_E \nabla u_h \cdot \nabla v_h \,\mathrm{d}E\,.$$

To discretize the problem (1.2), we have to devise a computable counterpart $a_h^E(\cdot,\cdot) : V_h^k(E) \times V_h^k(E) \to \mathbb{R}$ for the bilinear form $a^E(\cdot,\cdot)$. Using the $\Pi_k^{\nabla,E}$ projection, we can decompose a virtual function $v_h \in V_h^k(E)$ as follows

$$v_h = \Pi_k^{\nabla,E} v_h + (I - \Pi_k^{\nabla,E})v_h\,.$$

Inserting this into $a^E(\cdot,\cdot)$ and using the linearity of the form, we obtain:

$$\begin{aligned} a^E(u_h,v_h) &= a^E\big(\Pi_k^{\nabla,E}u_h, \Pi_k^{\nabla,E}v_h\big) + a^E\big((I-\Pi_k^{\nabla,E})u_h, (I-\Pi_k^{\nabla,E})v_h\big) \\ &\quad + a^E\big(\Pi_k^{\nabla,E}u_h, (I-\Pi_k^{\nabla,E})v_h\big) + a^E\big((I-\Pi_k^{\nabla,E})u_h, \Pi_k^{\nabla,E}v_h\big)\,. \end{aligned}$$

Thanks to the orthogonality of the projection $\Pi_k^{\nabla,E}$, the last two terms are equal to zero. The first term involves two polynomials and is therefore computable. This term is usually called *consistency term.* What remains is the second term, which is not computable. The idea is to replace this term with a second computable bilinear form $\mathcal{S}^E : V_h^k(E) \times V_h^k(E) \to \mathbb{R}$ that satisfies

$$\alpha_*|v_h|^2_{1,E} \leq \mathcal{S}^E(v_h,v_h) \leq \alpha^*|v_h|^2_{1,E} \quad \text{for all } v_h \in V_h^k(E)\,, \tag{1.8}$$

for two positive uniform constants α_* and α^*. A typical example in VEM approach is the the `dofi-dofi` stabilization (cf. [8, 16], for instance). It is defined as

$$\mathcal{S}^E(u_h,v_h) := \sum_{i=1}^{N_E} \mathrm{dof}_i(u_h)\,\mathrm{dof}_i(v_h)\,.$$

Remark 1.10. *It is possible to relax* (1.8) *such that it holds only for $v_h \in$ Ker$(\Pi_k^{\nabla,E})$. To simplify some steps, we kept the form* (1.8)*.*

We define the (local) bilinear form $a_h^E(\cdot,\cdot) : V_h^k(E) \times V_h^k(E) \to \mathbb{R}$ as

$$a_h^E(u_h,v_h) := a^E(\Pi_k^{\nabla,E}u_h, \Pi^{\nabla,E}v_h) + \mathcal{S}^E\big((I-\Pi_k^{\nabla,E})u_h, (I-\Pi_k^{\nabla,E})v_h\big)\,. \tag{1.9}$$

Remark 1.11. *Another possible choice in devising the bilinear form $a_h^E(\cdot,\cdot)$ is defining the consistency term as*

$$\int_E \mathbf{\Pi}_{k-1}^{0,E}\nabla u_h \cdot \mathbf{\Pi}_{k-1}^{0,E}\nabla v_h \,\mathrm{d}E\,.$$

Similarly to the projection $\Pi_k^{\nabla,E}$, it is straightforward to show that the projection operator $\mathbf{\Pi}_{k-1}^{0,E}$ is computable for a virtual function $v_h \in V_h^k(E)$. In [11], it was shown that this choice guarantees optimal convergence rates when a non-constant coefficient appears in front of the diffusive term in the PDE. Since we are considering the classical laplacian problem, we keep following the original approach presented in [8].

We introduce the (global) virtual element form $a_h(\cdot,\cdot) : V_h^k(\Omega_h) \times V_h^k(\Omega_h) \to \mathbb{R}$ as

$$a_h(u_h,v_h) := \sum_{E\in\Omega_h} a_h^E(u_h,v_h)\,.$$

This discrete bilinear form satisfies the following properties:

- *k-Consistency*: for all the virtual functions $v_h \in V_h^k(\Omega_h)$ and all the polynomials $p_k \in \mathbb{P}_k(\Omega_h)$, it holds
$$a(v_h, p_k) = a_h(v_h, p_k)\,,$$
- *Stability*: there exist two uniform constants α^* and α_* such that
$$\alpha_* a(v_h, v_h) \leq a_h(v_h, v_h) \leq \alpha^* a(v_h, v_h) \quad \forall v_h \in V_h^k(\Omega_h)\,. \tag{1.10}$$

It remains to discretize the right-hand side of (1.2). We distinguish two different cases:

- if $k = 1$, we approximate f by a piecewise constant $\Pi_0^0 f$ and then we define
$$\mathcal{F}_h(v_h) \coloneqq \sum_{E\in\Omega_h} \int_E \Pi_0^{0,E} f\, \bar{v}_h \,\mathrm{d}E = \sum_{E\in\Omega_h} |E|\, \Pi_0^{0,E} f\, \bar{v}_h \eqqcolon \sum_{E\in\Omega_h} \mathcal{F}_h^E(v_h),$$
where
$$\bar{v}_h \coloneqq \frac{1}{n_E}\sum_{i=1}^{n_E} v_h(V_i)\,, \quad V_i = \text{ vertices of } E\,.$$
We obtain
$$\begin{aligned}\mathcal{F}(v_h) - \mathcal{F}_h(v_h) &= \sum_{E\in\Omega_h} \int_E (f - \Pi_0^{0,E} f)\, \bar{v}_h + f\,(v_h - \bar{v}_h)\,\mathrm{d}E \\ &\lesssim \sum_{E\in\Omega_h} h_E \left(|f|_{1,E}\, \|v_h\|_{0,E} + \|f\|_{0,E}\, |v_h|_{1,E}\right) \\ &\lesssim \sum_{E\in\Omega_h} h_E\, \|f\|_{1,E}\, |v_h|_{1,\Omega_h}\,.\end{aligned} \tag{1.11}$$

- if $k \geq 2$, we define $\mathcal{F}_h(v_h)$ on each polygon E as the $L^2(E)$-orthogonal projection of f into $\mathbb{P}_{k-2}(E)$
$$\mathcal{F}_h(v_h) \coloneqq \sum_{E\in\Omega_h} \int_E (\Pi_{k-2}^{0,E} f)\, v_h \,\mathrm{d}E = \sum_{E\in\Omega_h} \int_E f\,(\Pi_{k-2}^{0,E} v_h)\,\mathrm{d}E \eqqcolon \sum_{E\in\Omega_h} \mathcal{F}_h^E(v_h)\,.$$
Thanks to the orthogonality of the $\Pi_{k-2}^{0,E}$ and Lemma 1.4, it holds
$$\begin{aligned}\mathcal{F}(v_h) - \mathcal{F}_h(v_h) &= \sum_{E\in\Omega_h} \int_E (f - \Pi_{k-2}^{0,E} f)\,(v_h - \Pi_0^{0,E} v_h)\,\mathrm{d}E \\ &\lesssim \sum_{E\in\Omega_h} h_E^k\, |f|_{k-1,E}\, |v_h|_{1,\Omega_h}\,.\end{aligned} \tag{1.12}$$

The discrete problems reads as
$$\begin{cases}\text{find } u_h \in V_h^k(\Omega_h) \text{ such that:} \\ a_h(u_h, v_h) = \mathcal{F}_h(v_h) \quad \forall v_h \in V_h^k(\Omega_h)\,.\end{cases} \tag{1.13}$$

Problem (1.13) is well-posed thanks to Lax-Milgram Lemma 1.1. In fact, exploiting (1.10), we have that the bilinear form $a_h(v_h, v_h)$ is coercive
$$a_h(v_h, v_h) \geq \tilde{\alpha}_* a(v_h, v_h) \geq \tilde{\alpha}_* |v_h|^2_{1,\Omega_h} \quad \forall v_h \in V_h^k(\Omega_h)\,,$$
where $\tilde{\alpha}_* \coloneqq \min\{1, \alpha_*\}$.

1.3.3 Error analysis

This section aims to estimate the rate of convergence of the discrete solution to the analytic solution. Let u be the solution of (1.2) and u_h the virtual function that solves (1.13). We introduce the following quantities

$$e_{\mathcal{I}} = u - u_{\mathcal{I}}\,, \quad e_\pi = u - \Pi_k^{\nabla} u\,, \quad e_h = u_h - u_{\mathcal{I}}\,.$$

To properly estimate the optimal rate of convergence, we make the following assumptions on the regularity of u and f. The solution u and the load term f in (1.2) satisfy:

$$u \in H^{k+1}(\Omega_h)\,, \quad f \in H^k(\Omega_h)\,.$$

Theorem 1.12 (Error estimate for the conforming method)**.** *Let u be the solution of* (1.2) *and u_h be the solution of* (1.13)*. Under assumptions* ***(A1)****, it holds*

$$|u - u_h|_{1,\Omega_h} \lesssim \sum_{E \in \Omega_h} \Theta^E\, h_E^k\,,$$

where the constant Θ_E depends on $\|u\|_{k+1,E}$ and $\|f\|_{k,E}$.

Proof. Adding and subtracting $u_{\mathcal{I}}$, we obtain

$$|u - u_h|_{1,\Omega_h} \leq |u - u_{\mathcal{I}}|_{1,\Omega_h} + |u_{\mathcal{I}} - u_h|_{1,\Omega_h} = |e_{\mathcal{I}}|_{1,\Omega_h} + |e_h|_{1,\Omega_h}\,.$$

The first term is controlled thanks to Lemma 1.9. We obtain

$$|e_{\mathcal{I}}|_{1,\Omega_h} = |u - u_{\mathcal{I}}|_{1,\Omega_h} \leq \sum_{E \in \Omega_h} h_E^k\, |u|_{k+1,E}\,.$$

To bound the second term, we exploit the coercivity of the bilinear form $a_h(\cdot,\cdot)$ and the fact that u and u_h solve (1.2) and (1.13), respectively. We have

$$\begin{aligned} |e_h|^2_{1,\Omega_h} = |u_{\mathcal{I}} - u_h|^2_{1,\Omega_h} &\lesssim a_h(u_{\mathcal{I}} - u_h, e_h) = a_h(u_{\mathcal{I}}, e_h) - \mathcal{F}_h(e_h) \\ &= a_h(u_{\mathcal{I}}, e_h) - a(u, e_h) + \mathcal{F}(e_h) - \mathcal{F}_h(e_h) \\ &= T_a + T_{\mathcal{F}}\,. \end{aligned} \tag{1.14}$$

Estimate for T_a : We exploit the orthogonality of the $\Pi_k^{\nabla,E}$ projection, Cauchy-Schwarz inequality, the stability property (1.10), triangular inequality, Lemma 1.9 and Lemma 1.4. We obtain

$$\begin{aligned} a_h(u_{\mathcal{I}}, e_h) - a(u, e_h) &= \sum_{E \in \Omega_h} a_h^E(u_{\mathcal{I}}, e_h) - a^E(u, e_h) \\ &= \sum_{E \in \Omega_h} a_h^E(u_{\mathcal{I}} - \Pi_k^{\nabla,E} u, e_h) - a^E(u - \Pi_k^{\nabla,E} u, e_h) \\ &\lesssim \sum_{E \in \Omega_h} \left(|u_{\mathcal{I}} - \Pi_k^{\nabla,E} u|_{1,E} + |u - \Pi_k^{\nabla,E} u|_{1,E}\right) |e_h|_{1,E} \\ &\lesssim \sum_{E \in \Omega_h} \left(|e_{\mathcal{I}}|_{1,E} + 2|e_\pi|_{1,E}\right) |e_h|_{1,E} \\ &\lesssim \sum_{E \in \Omega_h} h_E^k\, |u|_{k+1,E}\, |e_h|_{1,\Omega_h}\,. \end{aligned}$$

Estimate for $T_{\mathcal{F}}$: Depending on the choice of k, we use (1.11) or (1.12).
By combining the estimates of T_a and $T_{\mathcal{F}}$ in (1.14), and dividing by $|e_h|_{1,\Omega_h}$, we conclude. □

This theorem demonstrates the optimal rate of convergence in the (broken) H^1-seminorm. To derive an error estimate for the L^2-norm, a more accurate approximation of $\mathcal{F}^E$ for $k = 1, 2$ is necessary. By employing an Aubin-Nitsche duality argument, we can then obtain the desired estimate, see [9].

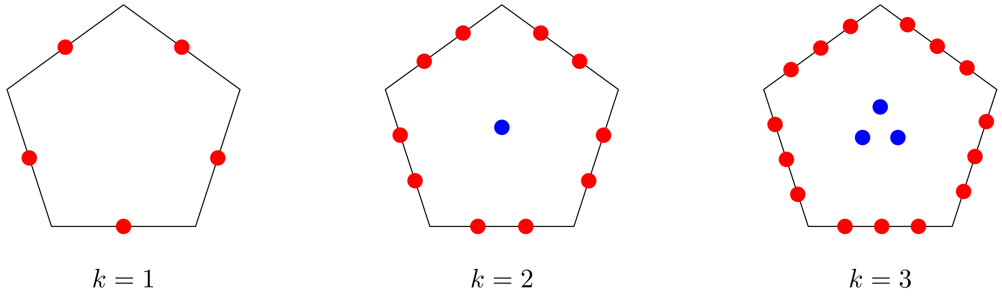


Figure 1.3: Degrees of freedom for a pentagon.

1.4 The nonconforming VEM

In this section, we introduce the non-conforming VEM as originally presented in [5]. Unlike the prevuois case, the nonconforming functions are not continuous across element boundaries; instead, only the jump, when tested against certain polynomials, is preserved. This necessitates the consideration of a term in the error analysis that measures the nonconformity of the method.

1.4.1 The non conforming virtual element spaces

Given a positive integer k and an element $E \in \Omega_h$ with n_E edges, we define the nonconforming virtual element space as

$$V_h^{k,nc}(E) := \Big\{ v_h \in H^1(E) \text{ such that } \Delta v_h \in \mathbb{P}_{k-2}(E),\ \nabla v_h \cdot \boldsymbol{n}^E \in \mathbb{P}_{k-1}(e) \text{ for all } e \subset \partial E \Big\}\,. \tag{1.15}$$

We note that the functions in $V_h^{k,nc}(E)$ solves a Poisson problem with a Neumann boundary condition. Unlike functions in (1.4), nonconforming functions remain unknown even on the boundaries of elements. Due to the absence of function traces, we cannot employ the same DoFs as in the conforming case. For the nonconforming space $V_h^{k,nc}(E)$, the following set of linear operators constitutes a set of DoFs:

- $\mathcal{E}_{nc}^{E,k}$: the moments up to the order $k-1$ on each edge $e \subset \partial E$:

$$\mu_{E,e}^l(v_h) := \frac{1}{|e|}\int_e v_h \left(\frac{s - s_e}{h_e}\right)^l \mathrm{d}s\,, \quad l = 0, \ldots, k-1\,, \tag{1.16}$$

  where s denotes a local coordinate on e, and s_e is the coordinate value corresponding to the edge midpoint,

- $\mathcal{P}_{nc}^{E,k}$: the moments up to the order $k-2$ on E:

$$\mu_E^{\boldsymbol{\alpha}}(v_h) := \frac{1}{|E|}\int_E v_h \left(\frac{\mathbf{x} - \mathbf{x}_E}{h_E}\right)^{\boldsymbol{\alpha}} \mathrm{d}E\,, \quad |\boldsymbol{\alpha}| \leq k-2\,, \tag{1.17}$$

  where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T$ is a multi-index.

The DoFs are illustred in Figure 1.3. It can be demonstrated that these linear operators constitute a complete set of degrees of freedom for the space $V_h^{k,nc}(E)$ following a procedure similar to Proposition 1.6. The dimension N_E^{nc} of the space $V_h^{k,nc}(E)$ is identical to the one of $V_h^k(E)$, thus

$$N_E^{nc} := k\, n_E + \frac{(k-1)(k-2)}{2}\,.$$

Similarly to the conforming case, for the nonconforming space, it holds that $\mathbb{P}_k(E) \subseteq V_h^{k,nc}(E)$. The preliminary global space is obtained by

$$\begin{aligned}\tilde{V}_h^{k,nc}(\Omega_h) := \big\{ v_h \in L^2(\Omega) \ \text{such that}\ v_h|_E \in V_h^{k,nc}(E) \quad \forall E \in \Omega_h\,, \\ \int_e [\![v_h]\!] \cdot \boldsymbol{n}^e\, q = 0 \quad \forall q \in \mathbb{P}_{k-1}(e) \quad \forall e \in \mathcal{E}_h^o \big\}\,.\end{aligned} \tag{1.18}$$

where

$$[\![v]\!] := v^E \boldsymbol{n}^E + v^K \boldsymbol{n}^K \,, \tag{1.19}$$

and e is such that $e \subseteq E \cap K$. We emphasize that full continuity across the element boundaries is not imposed. On each interior edge, we only require preservation of moments up to order $k-1$. Therefore, $V_h^{k,nc}(\Omega_h) \not\subset H^1(\Omega)$. The global nonconforming virtual element space is defined as

$$V_h^{k,nc}(\Omega_h) := \left\{ v_h \in \tilde{V}_h^{k,nc}(\Omega_h) \text{ such that } \int_e v_h \left(\frac{s-s_e}{h_e}\right)^l = 0 \,, \quad \forall e \in \mathcal{E}_h^\partial \,, \quad l = 0, \dots, k-1 \right\} . \tag{1.20}$$

The space $V_h^{k,nc}(\Omega_h)$ is obtained by imposing a boundary condition in the preliminary space $\tilde{V}_h^{k,nc}(\Omega_h)$. Specifically, we enforce that the DoFs associated with the boundary edges are set to zero. The DoFs for the space $V_h^{k,nc}(\Omega_h)$ are defined as follows:

- $\mathcal{E}_{nc}^k$: the moments up to the order $k-1$ on each internal edge $e \in \mathcal{E}_h^o$:

$$\mu_e^l(v_h) := \frac{1}{|e|} \int_e v_h \left(\frac{s-s_e}{h_e}\right)^l \mathrm{d}s \quad l = 0, \dots, k-1 \,,$$

- $\mathcal{P}_{nc}^k$: the moments up to the order $k-2$ on each $E \in \Omega_h$:

$$\mu_E^{\boldsymbol{\alpha}}(v_h) := \frac{1}{|E|} \int_E v_h \left(\frac{\boldsymbol{x}-\boldsymbol{x}_E}{h_E}\right)^{\boldsymbol{\alpha}} \mathrm{d}E \quad |\boldsymbol{\alpha}| \le k-2 \,,$$

  where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T$ is a multi-index.

The dimension $N_{\Omega_h}^{nc}$ of the space $V_h^{k,nc}(\Omega_h)$ is given by

$$N_{\Omega_h}^{nc} = k\,|\mathcal{E}_h^o| + \frac{(k-1)(k-2)}{2}\, n_p \,.$$

We can note that given a tessellation Ω_h, it holds $N_{\Omega_h} \neq N_{\Omega_h}^{nc}$. The analogous of Lemma 1.9 holds for nonconforming functions

Lemma 1.13 (Approximation with nonconforming virtual element functions)**.** *Under assumption **(A1)**, for any $v \in H_0^1(\Omega) \cap H^{s+1}(\Omega_h)$, there exists $v_{\mathcal{I}} \in V_h^{k,nc}(\Omega_h)$, such that for all $E \in \Omega_h$,*

$$\|v - v_{\mathcal{I}}\|_{0,E} + h_E \|\nabla(v - v_{\mathcal{I}})\|_{0,E} \lesssim h_E^{s+1} |v|_{s+1,E} \,,$$

where $0 < s \le k$.

1.4.2 The nonconforming discrete problem

Similarly to the conforming method, our goal is to discretize the problem (1.2). Since the procedure is quite analogous, we skip over certain details. The discretization of the bilinear form $a(\cdot,\cdot)$ is constructed in a manner similar to that presented in Section 1.3.2. We define a bilinear form $a_h(\cdot,\cdot) : V_h^{k,nc}(\Omega_h) \times V_h^{k,nc}(\Omega_h) \to \mathbb{R}$ such that

$$a_h(u_h, v_h) := \sum_{E \in \Omega_h} a_h^E(u_h, v_h) \,,$$

and the local forms are defined as

$$a_h^E(u_h, v_h) = a^E(\Pi_k^{\nabla,E} u_h, \Pi^{\nabla,E} v_h) + \mathcal{S}^E\big((I - \Pi_k^{\nabla,E}) u_h, (I - \Pi_k^{\nabla,E}) v_h\big) \,,$$

where $\mathcal{S}^E(\cdot,\cdot) : V_h^{k,nc}(E) \times V_h^{k,nc}(E) \to \mathbb{R}$ is a stabilization term as in (1.8). For $k \ge 2$, the discretization of the right-hand side follows the same procedure of the conforming method. If $k = 1$, given an element $E \in \Omega_h$, we set

$$\bar{v}_h|_E := \frac{1}{n_E} \sum_{e \subset \partial E} \frac{1}{|e|} \int_e v_h \,\mathrm{d}s \,,$$

and then we define

$$\mathcal{F}_h(v_h) := \sum_{E\in\Omega_h} \int_E \Pi_0^{0,E} f\, \bar{v}_h \,\mathrm{d}E = \sum_{E\in\Omega_h} |E|\, \Pi_0^{0,E} f\, \bar{v}_h \;=: \sum_{E\in\Omega_h} \mathcal{F}_h^E(v_h)\,.$$

In the same way of (1.11), it can be proved that

$$\begin{aligned}\mathcal{F}(v_h) - \mathcal{F}_h(v_h) &= \sum_{E\in\Omega_h} \int_E (f - \Pi_0^{0,E} f)\, \bar{v}_h + f\,(v_h - \bar{v}_h)\,\mathrm{d}E \\ &\lesssim \sum_{E\in\Omega_h} h_E \left(|f|_{1,E}\, \|v_h\|_{0,E} + \|f\|_{0,E}\, |v_h|_{1,E}\right) \\ &\lesssim \sum_{E\in\Omega_h} h_E\, \|f\|_{1,E}\, |v_h|_{1,\Omega_h}\,.\end{aligned} \tag{1.21}$$

The discrete problem reads as

$$\begin{cases}\text{find } u_h \in V_h^{k,nc}(\Omega_h) \text{ such that:} \\ a_h(u_h, v_h) = \mathcal{F}_h(v_h) \quad \forall v_h \in V_h^{k,nc}(\Omega_h)\,.\end{cases} \tag{1.22}$$

1.4.3 Error Analysis

The error analysis for the nonconforming problem follows a procedure similar to the conforming method, with a key difference arising from the nonconformity of the method. We assume that the solution u and the load term f in (1.2) satisfy:

$$u \in H^{k+1}(\Omega_h)\,, \quad f \in H^k(\Omega_h)\,,$$

Because $V_h^{k,nc}(\Omega_h) \not\subset H^1(\Omega)$, we have to consider in the analysis a consistency error. Let u be the solution of (1.2). Through integration by parts, we obtain

$$\begin{aligned}a(u, v_h) &= -\sum_{E\in\Omega_h} \int_E \Delta u\, v_h \,\mathrm{d}E + \sum_{E\in\Omega_h} \int_{\partial E} (\nabla u \cdot \boldsymbol{n}^E)\, v_h \,\mathrm{d}s \\ &=: \mathcal{F}(v_h) + \mathcal{B}_h(u, v_h) \quad \forall v_h \in V_h^{k,nc}(\Omega_h)\,.\end{aligned}$$

We can rewrite the form $\mathcal{B}_h(u, v_h)$ as

$$\mathcal{B}_h(u, v_h) = \sum_{e\in\mathcal{E}_h^o} \int_e \nabla u \cdot [\![v_h]\!] \,\mathrm{d}s\,. \tag{1.23}$$

This version shows that the form $\mathcal{B}_h(u, v_h)$ quantifies how much the analytical solution u deviates from fitting within the virtual element space $V_h^{k,nc}(\Omega_h)$. An estimate of this consistency error is provided in the following Proposition.

Proposition 1.14 (Consistency error)**.** *Under assumption **(A1)**, let $u \in H^2(\Omega)$ be the solution of* (1.2)*. It holds*

$$|\mathcal{B}(u, v_h)| \lesssim \sum_{E\in\Omega_h} h_E^k |u|_{k+1,E}\, |v_h|_{1,\Omega_h}\,.$$

Proof. Given an edge $e \in \mathcal{E}_h^o$, we introduce the projection operator $\Pi_{k-1}^{0,e}$ as the orthogonal projection with respect to the $L^2(e)$-inner product. Using the orthogonality of this projection and Lemma 1.5, it holds

$$\begin{aligned}
|\mathcal{B}(u, v_h)| &= \left|\sum_{e\in\mathcal{E}_h^o} \int_e (\nabla u - \mathbf{\Pi}^{0,e}_{k-1}\nabla u)\cdot [\![v_h]\!]\,\mathrm{d}s\right| \\
&= \left|\sum_{e\in\mathcal{E}_h^o} \int_e (\nabla u - \mathbf{\Pi}^{0,e}_{k-1}\nabla u)\cdot ([\![v_h]\!] - \Pi_0^{0,e}[\![v_h]\!])\,\mathrm{d}s\right| \\
&\lesssim \sum_{e\in\mathcal{E}_h^o} \|\nabla u - \mathbf{\Pi}^{0,e}_{k-1}\nabla u\|_{0,e}\, \|[\![v_h]\!] - \Pi_0^{0,e}[\![v_h]\!]\|_{0,e} \\
&\lesssim \sum_{E\in\Omega_h} h_E^k |u|_{k+1,E}\, |v_h|_{1,\Omega_h}\,.
\end{aligned}$$

The final step relies on assumption **(A1)**, which guarantees that each polygon is encountered a finite number of times and $h_e \leq h_E \leq \rho\, h_e$. □

Remark 1.15. *We note that we are requiring more regularity compared to the conforming method, as previously observed in [5]. In Proposition 1.14, it is necessary to estimate the gradient of u restricted to an edge. If $u \in H^1(\Omega)$, the restriction of the gradient to the boundary is not defined.*

Thanks to this estimate, we can conclude this section with the following result.

Theorem 1.16 (Error estimate for the nonconforming method)**.** *Under assumption **(A1)**, let $u \in H^2(\Omega)$ be the solution of* (1.2) *and u_h be the solution of* (1.13)*. It holds*

$$|u - u_h|_{1,\Omega_h} \lesssim \sum_{E\in\Omega_h} \Theta^E\, h_E^k\,,$$

where the constant Θ_E depends on $\|u\|_{k+1,E}$ and $\|f\|_{k,E}$.

Proof. It's sufficient to combine the proof of Theorem 1.12 with Proposition 1.14. The error is divided as

$$|u - u_h|_{1,\Omega_h} \leq |e_{\mathcal{I}}|_{1,\Omega_h} + |e_h|_{1,\Omega_h}\,.$$

Using Lemma 1.13, it holds

$$|e_{\mathcal{I}}|_{1,\Omega_h} \leq \sum_{E\in\Omega_h} h_E^k\, |u|_{k+1,E}\,.$$

The difference with Theorem 1.12 is

$$\begin{aligned}
|e_h|^2_{1,\Omega_h} = |u_{\mathcal{I}} - u_h|^2_{1,\Omega_h} &\lesssim a_h(u_{\mathcal{I}} - u_h, e_h) = a_h(u_{\mathcal{I}}, e_h) - \mathcal{F}_h(e_h) \\
&= a_h(u_{\mathcal{I}}, e_h) - a(u, e_h) + \mathcal{F}(e_h) - \mathcal{F}_h(e_h) + \mathcal{B}_h(u, e_h).
\end{aligned}$$

The term $a_h(u_{\mathcal{I}}, e_h) - a(u, e_h)$ is estimated exactly as in Theorem 1.12. The error on the right-hand side is controlled using (1.21) or (1.12), depending on $k = 1$ or $k \geq 2$. The nonconformity error is estimated in Proposition 1.14. □

.

Chapter 2

Nitsche method for VEM

In this chapter, we discuss the implementation of the Nitsche method in a virtual element context. This technique was first introduced in [52]. The Nitsche method allows for the implementation of boundary conditions in a weak sense without requiring a Lagrange multiplier. This is achieved by incorporating certain terms into the formulation of the problem to enforce the boundary conditions. The first term arises from integrating the Laplacian by parts. The second term, a symmetric version of the first, is added to restore the symmetry of the problem. The final symmetric term is included to ensure coercivity and requires tuning a parameter δ that depends on the geometry of the mesh.

A conforming VEM version of the Nitsche term was already devised in [25] in a more general setting. For the best of our knowledge the Nitsche method for the nonconforming VEM has never been discussed.

This chapter is essential for the following two chapters, as the methods we will present require the imposition of the boundary condition in a weak sense for their analysis. The chapter is organized as follows: the first section presents the Nitsche method in a very general setting. The second section discusses how to implement the Nitsche method for the conforming VEM, and the last section covers its implementation for the nonconforming VEM.

2.1 Introduction to the Nitsche method

In this section, we discuss how to implement the boundary condition in a weak sense using a Nitsche type technique [52, 46] in a very general setting. As model problem, we consider again the Poisson equation already introduced in Section 1.1 with non-homogeneous boundary condition

$$\begin{cases} -\Delta u = f & \text{in } \Omega\,, \\ \quad u = g & \text{on } \Gamma\,, \end{cases} \tag{2.1}$$

where $g \in H^{\frac{1}{2}}(\Gamma)$. The general idea of the Nitsche method is to choose a discrete space V_h associated to a mesh Ω_h and consider a problem with the following form

$$\begin{cases} \text{find } u_h \in V_h \text{ such that:} \\ a(u_h, v_h) + \mathcal{N}(u_h, v_h) = \mathcal{F}^{\mathcal{N}}(v_h) \quad \forall v_h \in V_h\,, \end{cases} \tag{2.2}$$

where the bilinear form $a(\cdot,\cdot)$ is the same that appears in (1.3) and the bilinear form $\mathcal{N}(\cdot,\cdot) : V_h \times V_h \to \mathbb{R}$ is defined as

$$\mathcal{N}(u_h, v_h) := -\sum_{e \in \mathcal{E}_h^\partial} \langle \nabla u_h \cdot \boldsymbol{n}^e, v_h \rangle_e - \sum_{e \in \mathcal{E}_h^\partial} \langle u_h, \nabla v_h \cdot \boldsymbol{n}^e \rangle_e + \sum_{e \in \mathcal{E}_h^\partial} \frac{1}{\delta h_e} \langle u_h, v_h \rangle_e\,, \tag{2.3}$$

here $\langle \cdot,\cdot \rangle_e$ denotes the $L^2(e)$-inner product. The right-hand side is defined as

$$\mathcal{F}^{\mathcal{N}}(v_h) := \int_\Omega f\, v_h \,\mathrm{d}\Omega - \sum_{e \in \mathcal{E}_h^\partial} \langle g, \nabla v_h \cdot \boldsymbol{n}^e \rangle_e + \sum_{e \in \mathcal{E}_h^\partial} \frac{1}{\delta h_e} \langle g, v_h \rangle_e\,. \tag{2.4}$$

Here, δ is a parameter that will be set to ensure coercivity and $\langle\cdot,\cdot\rangle_e$ denotes $L^2(e)$-inner product. The first term in (2.3) comes from the integration by part of the laplacian operator and the fact that v_h is not zero on the boundary. The second term is coupled with the second term in (2.4) and is included to restore the symmetry of the problem. The last terms in (2.3) and (2.4) ensure the coercivity of the problem. Unlike in (1.13), where the virtual space inherently satisfies the boundary condition by definition, problem (2.2) imposes the boundary condition within the formulation of the forms and the right-hand side. The next two sections will discuss how to extend this technique to the conforming and nonconforming virtual element spaces introduced in the previous chapter.

Remark 2.1. *Depending on the context and the method, the definitions* (2.3) *and* (2.4) *may vary. Similar to the previous chapter, within a VEM context, it will be necessary to incorporate some polynomial projections into the definitions of* (2.3) *and* (2.4)*.*

2.2 Nitsche method for conforming VEM

In this section, we extend the Nitsche technique to the conforming virtual element method. This approach was initially explored in [25] within a broader context. In that study, the authors introduced a Lagrange multiplier method inspired by the work in [6], from which they derived the Nitsche method as a specific case. Given a positive integer k and a decomposition Ω_h of the domain Ω, we consider the (global) virtual element space

$$V_h^k(\Omega_h) \coloneqq \left\{ v_h \in H^1(\Omega) \text{ such that } v_h|_E \in V_h^k(E) \quad \text{for all } E \in \Omega_h \right\} . \tag{2.5}$$

Here, $V_h^k(E)$ are the same local spaces already introduced in (1.4). Contrary to the definition (1.7), the global space (2.5) does not impose any boundary condition to the virtual functions v_h. Due to this difference, the set of DoFs for the space (2.5) is slightly different:

- $\mathcal{V}_c^k$: the values of v_h at the vertices,
- $\mathcal{E}_c^k$: the values of v_h at $k-1$ points on each edge $e \in \mathcal{E}_h$,
- $\mathcal{P}_c^k$: the moments up to order $k-2$ for each element $E \in \Omega_h$.

The dimension of the space $V_h^k(\Omega_h)$ is

$$N_{\Omega_h} = n_v + (k-1)|\mathcal{E}_h| + n_p \frac{(k-2)(k-1)}{2} ,$$

where n_v is the number of vertices in the decomposition Ω_h.

2.2.1 Conforming Nitsche virtual element forms

The form $a(\cdot,\cdot)$ is discretized in the same manner as in Section 1.3.2. To discretize the form (2.3) and the right-hand side, we adopt the same principles as the standard virtual element implementation. We define the local bilinear form $\mathcal{N}_h^E : V_h^k(E) \times V_h^k(E) \to \mathbb{R}$ as follows:

$$\mathcal{N}_h^E(u_h, v_h) \coloneqq -\langle \nabla \Pi_k^{\nabla,E} u_h \cdot \boldsymbol{n}^e, v_h \rangle_{\Gamma_E} - \langle u_h, \nabla \Pi_k^{\nabla,E} v_h \cdot \boldsymbol{n}^e \rangle_{\Gamma_E} + \frac{1}{\delta h_E} \langle u_h, v_h \rangle_{\Gamma_E} . \tag{2.6}$$

Since we know the virtual functions on the boundary, there is no need to insert additional projections in the terms of (2.6) except for the projection on the normal derivative. Then, we define the global bilinear form $\mathcal{N}_h(\cdot,\cdot) : V_h^k(\Omega_h) \times V_h^k(\Omega_h) \to \mathbb{R}$ as

$$\mathcal{N}_h(u_h, v_h) \coloneqq \sum_{E \in \Omega_h} \mathcal{N}_h^E(u_h, v_h) .$$

After that, the right-hand side $\mathcal{F}_h^{\mathcal{N}} : V_h^k(\Omega_h) \to \mathbb{R}$ is constucted in a very similar way and it is defined as

$$\mathcal{F}_h^{\mathcal{N}}(v_h) \coloneqq \sum_{E \in \Omega_h} \mathcal{F}_h^{\mathcal{N},E}(v_h) , \tag{2.7}$$

and

$$\mathcal{F}_h^{\mathcal{N},E}(v_h) := \mathcal{F}_h^E(v_h) - \langle g, \nabla \Pi_k^{\nabla,E} v_h \cdot \boldsymbol{n}^e \rangle_{\Gamma_E} + \frac{1}{\delta h_E} \langle g, v_h \rangle_{\Gamma_E} \,,$$

where $\mathcal{F}_h^E$ is the linear form introduced in Section 1.3.2.

Remark 2.2. *As suggested in Remark 1.11, another possibility is to define the bilinear form $\mathcal{N}_h^E(\cdot,\cdot)$ as*

$$\mathcal{N}_h^E(u_h, v_h) := -\langle \boldsymbol{\Pi}_{k-1}^{0,E} \nabla u_h \cdot \boldsymbol{n}^e, v_h \rangle_{\Gamma_E} - \langle u_h, \boldsymbol{\Pi}_{k-1}^{0,E} \nabla v_h \cdot \boldsymbol{n}^e \rangle_{\Gamma_E} + \frac{1}{\delta h_E} \langle u_h, v_h \rangle_{\Gamma_E} \,.$$

Therefore, the local right-hand side should be defined as

$$\mathcal{F}_h^{\mathcal{N},E}(v_h) := \mathcal{F}_h^E(v_h) - \langle g, \boldsymbol{\Pi}_{k-1}^{0,E} \nabla v_h \cdot \boldsymbol{n}^e \rangle_{\Gamma_E} + \frac{1}{\delta h_E} \langle g, v_h \rangle_{\Gamma_E} \,.$$

To avoid confusion, we prefer to keep the same projection that appears in the definition of the bilinear form $a_h^E(\cdot,\cdot)$.

We introduce the bilinear form

$$\mathcal{A}(u_h, v_h) := \sum_{E \in \Omega_h} \mathcal{A}^E(u_h, v_h) \,, \quad \text{and} \quad \mathcal{A}^E(u_h, v_h) := a_h^E(u_h, v_h) + \mathcal{N}_h^E(u_h, v_h) \,.$$

Finally, the discrete linear problem reads as

$$\begin{cases} \text{find } u_h \in V_h^k(\Omega_h) \text{ such that:} \\ \mathcal{A}(u_h, v_h) = \mathcal{F}_h^{\mathcal{N}}(v_h) \quad \forall v_h \in V_h^k(\Omega_h) \,. \end{cases} \tag{2.8}$$

2.2.2 Consistency of the method

Due to the polynomials projections entering in (2.8), the analytic solution u of (2.1) does not satisfy the discrete scheme (2.8). This is a typical situation in VEM. Despite that, assuming that $u \in H^2(\Omega) \cap H_g^1(\Omega)$, it satisfies the problem

$$\tilde{\mathcal{A}}(u, v_h) = \tilde{\mathcal{F}}_h^{\mathcal{N}}(v_h) \qquad \text{for all } v_h \in V_h^k(\Omega_h) \,. \tag{2.9}$$

where

$$\tilde{\mathcal{A}}(u, v_h) := \sum_{E \in \Omega_h} \tilde{\mathcal{A}}^E(u, v_h) \,, \quad \tilde{\mathcal{F}}_h^{\mathcal{N}}(v_h) := \sum_{E \in \Omega_h} \tilde{\mathcal{F}}_h^{\mathcal{N},E}(v_h) \,.$$

The local forms are defined as

$$\tilde{\mathcal{A}}^E(u, v_h) := a^E(u, v_h) + \tilde{\mathcal{N}}_h^E(u, v_h) \,,$$

with

$$\tilde{\mathcal{N}}_h^E(u, v_h) := -\langle \nabla u \cdot \boldsymbol{n}^E, v_h \rangle_{\Gamma_E} - \langle u, \nabla \Pi_k^{\nabla,E} v_h \cdot \boldsymbol{n}^E \rangle_{\Gamma_E} + \frac{1}{\delta h_E} \langle u, v_h \rangle_{\Gamma_E} \,,$$

and

$$\tilde{\mathcal{F}}_h^{\mathcal{N},E}(v_h) := \int_E f \, v_h \, \mathrm{d}E - \langle g, \nabla \Pi_k^{\nabla,E} v_h \cdot \boldsymbol{n}^E \rangle_{\Gamma_E} + \frac{1}{\delta h_E} \langle g, v_h \rangle_{\Gamma_E} \,.$$

We also set

$$\tilde{\mathcal{N}}_h(u, v_h) := \sum_{E \in \Omega_h} \tilde{\mathcal{N}}_h^E(u, v_h) \,.$$

We emphasize that, since $u = g$ on the boundary, it is assured that

$$\langle u - g, \nabla \Pi_k^{\nabla,E} v_h \cdot \boldsymbol{n}^E \rangle_{\Gamma_E} = 0 \quad \forall E \in \Omega_h \,,$$

despite the presence of the projection $\Pi_k^{\nabla,E}$ in the normal derivative of v_h.

2.2.3 Coercivity of the bilinear form

Given a virtual function $v_h \in V_h^k(\Omega_h)$, we introduce the local norm

$$\|v_h\|_{\mathcal{N},E}^2 \coloneqq |v_h|_{1,E}^2 + \frac{1}{\delta h_E}\|v_h\|_{0,\Gamma_E}^2 \,, \tag{2.10}$$

with global counterpart

$$\|v_h\|_{\mathcal{N}} \coloneqq \left(\sum_{E\in\Omega_h} \|v_h\|_{\mathcal{N},E}^2\right)^{\frac{1}{2}} . \tag{2.11}$$

The following coercivity result guarantees the well posedness of problem (2.8).

Proposition 2.3 (Coercivity)**.** *Under assumptions **(A1)**, the bilinear form $\mathcal{A}(u_h, v_h)$ is coercive with respect to $\|\cdot\|_{\mathcal{N}}$. It holds*

$$\mathcal{A}(v_h, v_h) \gtrsim \|v_h\|_{\mathcal{N}}^2 \quad \forall v_h \in V_h^k(\Omega_h)\,,$$

for a suitable choice of the parameter δ.

Proof. For each element $E \in \Omega_h$, we have

$$\mathcal{A}^E(v_h, v_h) = a_h^E(v_h, v_h) + \mathcal{N}_h^E(v_h, v_h)\,.$$

Thanks to the property (1.8) and the orthogonality of the projection $\Pi_k^{\nabla,E}$, it holds

$$a_h^E(v_h, v_h) \geq |\Pi_k^{\nabla,E} v_h|_{1,E}^2 + \alpha_* \, |(I - \Pi_k^{\nabla,E})v_h|_{1,E}^2 \geq \tilde{\alpha}_* \, |v_h|_{1,E}^2 \,,$$

with $\tilde{\alpha}_* = \min\{1, \alpha_*\}$. For the Nitsche term, it holds

$$\mathcal{N}_h^E(v_h, v_h) = -2\langle \nabla \Pi_k^{\nabla,E} v_h \cdot \boldsymbol{n}^e, v_h\rangle_{\Gamma_E} + \frac{1}{\delta h_E}\|v_h\|_{0,\Gamma_E}^2 \,.$$

We must control the first term in this equality. Using Cauchy-Schwarz inequality, Young inequality and the polynomial trace inequality, we obtain

$$\begin{aligned} 2\langle \nabla \Pi_k^{\nabla,E} v_h \cdot \boldsymbol{n}^E, v_h\rangle_{\Gamma_E} &\leq 2\delta h_E \|\nabla \Pi_k^{\nabla,E} v_h \cdot \boldsymbol{n}^E\|_{0,e}^2 + \frac{1}{2\delta h_E}\|v_h\|_{0,\Gamma_E}^2 \\ &\leq 2\delta C_{\mathrm{tr}} \|\nabla \Pi_k^{\nabla,E} v_h\|_{0,E}^2 + \frac{1}{2\delta h_E}\|v_h\|_{0,\Gamma_E}^2 \\ &\leq 2\delta C_{\mathrm{tr}} \|\nabla v_h\|_{0,E}^2 + \frac{1}{2\delta h_E}\|v_h\|_{0,\Gamma_E}^2 \,, \end{aligned}$$

where C_{tr} is the inverse trace inequality constant for polynomials. Choosing $\delta = \tilde{\alpha}_*/(4C_{\mathrm{tr}})$, we obtain

$$\mathcal{A}^E(v_h, v_h) \geq \frac{\tilde{\alpha}_*}{2}|v_h|_{1,E}^2 + \frac{1}{2\delta h_E}\|v_h\|_{0,\Gamma_E}^2 \gtrsim \|v_h\|_{\mathcal{N},E}^2 \,.$$

The proof is completed by summing over all the elements $E \in \Omega_h$. □

2.2.4 Error analysis

In this section we prove that the conforming VEM with the Nitsche bilinear form converges to the analytic solution with optimal rate of convergence. We recall the definitions given in Section 1.3.3

$$e_{\mathcal{I}} = u - u_{\mathcal{I}}\,, \quad e_h = u_h - u_{\mathcal{I}}\,.$$

Again, we assume that the solution u and the load term f in (1.2) satisfy:

$$u \in H^2(\Omega) \cap H^{k+1}(\Omega_h)\,, \quad f \in H^k(\Omega_h)\,.$$

Theorem 2.4 (Error estimate)**.** *Let u be the solution of* (1.2) *and u_h be the solution of* (2.8)*. Under assumption **(A1)**, it holds*

$$\|u - u_h\|_{\mathcal{N}} \lesssim \sum_{E\in\Omega_h} \Theta^E \, h_E^k \,,$$

where the constant Θ_E depends on $\|u\|_{k+1,E}$ and $\|f\|_{k,E}$.

Proof. Since the proof is very similar to the one of theorem 1.12, we skip some steps. Using triangular inequality, we have

$$\|u - u_h\|_{\mathcal{N}} \leq \|u - u_{\mathcal{I}}\|_{\mathcal{N}} + \|u_{\mathcal{I}} - u_h\|_{\mathcal{N}} = \|e_{\mathcal{I}}\|_{\mathcal{N}} + \|e_h\|_{\mathcal{N}} \,.$$

Again, the interpolation error is controlled using Lemma 1.9 and a trace inequality

$$\begin{aligned}
\|e_{\mathcal{I}}\|_{\mathcal{N}}^2 &= |e_{\mathcal{I}}|_{1,\Omega_h}^2 + \sum_{E\in\Omega_h} \frac{1}{\delta h_E} \|e_{\mathcal{I}}\|_{0,\Gamma_E}^2 \\
&\lesssim |e_{\mathcal{I}}|_{1,\Omega_h}^2 + \sum_{E\in\Omega_h} h_E^{-2} \|e_{\mathcal{I}}\|_{0,E}^2 \\
&\lesssim \sum_{E\in\Omega_h} h_E^{2k} \, |u|_{k+1,E}^2 \,.
\end{aligned}$$

Exploiting the coercivity of the bilinear form and the observations in Section 2.2.2

$$\begin{aligned}
\|e_h\|_{\mathcal{N}}^2 = \|u_{\mathcal{I}} - u_h\|_{\mathcal{N}}^2 &\lesssim \mathcal{A}(u_{\mathcal{I}} - u_h, e_h) = \mathcal{A}(u_{\mathcal{I}}, e_h) - \mathcal{F}_h(e_h) \\
&= \mathcal{A}(u_{\mathcal{I}}, e_h) - \tilde{\mathcal{A}}(u, e_h) + \tilde{\mathcal{F}}_h^{\mathcal{N}}(e_h) - \mathcal{F}_h^{\mathcal{N}}(e_h) \qquad (2.12) \\
&= T_1 + T_2 \,.
\end{aligned}$$

The term T_2 was already estimated in (1.11) and (1.12). We point out that, since $u = g$ on the boundary, the boundary terms do not appear in the estimate. The term in T_1 is split into two parts. We have

$$T_1 = a_h(u_{\mathcal{I}}, e_h) - a(u, e_h) + \mathcal{N}_h(u_{\mathcal{I}}, e_h) - \tilde{\mathcal{N}}_h(u, e_h) =: T_a + T_{\mathcal{N}} \,.$$

Following the same steps in Theorem 1.12, we obtain for T_a that

$$T_a \lesssim \sum_{E\in\Omega_h} h_E^k |u|_{k+1,E} |e_h|_{1,\Omega_h} \,.$$

It remains to bound $T_{\mathcal{N}}$. This term is split as

$$\begin{aligned}
T_{\mathcal{N}} &= \mathcal{N}_h^E(u_{\mathcal{I}}, e_h) - \tilde{\mathcal{N}}_h^E(u, e_h) \\
&= \sum_{E\in\Omega_h} -\langle \nabla\Pi_k^{\nabla,E} u_{\mathcal{I}} \cdot \boldsymbol{n}^e, e_h\rangle_{\Gamma_E} + \langle \nabla u \cdot \boldsymbol{n}^e, e_h\rangle_{\Gamma_E} \\
&\quad + \sum_{E\in\Omega_h} -\langle u_{\mathcal{I}}, \nabla\Pi_k^{\nabla,E} e_h \cdot \boldsymbol{n}^e\rangle_{\Gamma_E} + \langle u, \nabla\Pi_k^{\nabla,E} e_h \cdot \boldsymbol{n}^e\rangle_{\Gamma_E} \\
&\quad + \sum_{E\in\Omega_h} \frac{1}{\delta h_E} \langle u_{\mathcal{I}}, e_h\rangle_{\Gamma_E} - \frac{1}{\delta h_E} \langle u, e_h\rangle_{\Gamma_E} \\
&=: \sum_{E\in\Omega_h} T_{\mathcal{N},1}^E + T_{\mathcal{N},2}^E + T_{\mathcal{N},3}^E \,.
\end{aligned}$$

On the first one, we use Cauchy-Schwarz inequality and trace inequality, to derive

$$\begin{aligned}
T_{\mathcal{N},1}^E &= \langle \nabla u \cdot \boldsymbol{n}^e, e_h\rangle_{\Gamma_E} - \langle \nabla\Pi_k^{\nabla,E} u_{\mathcal{I}} \cdot \boldsymbol{n}^e, e_h\rangle_{\Gamma_E} \\
&= \langle \nabla(u - \nabla\Pi_k^{\nabla,E} u_{\mathcal{I}}) \cdot \boldsymbol{n}^e, e_h\rangle_{\Gamma_E} \\
&\lesssim (h_E^{-\frac{1}{2}} \|\nabla u - \nabla\Pi_k^{\nabla,E} u_{\mathcal{I}}\|_{0,E} + h_E^{\frac{1}{2}} |\nabla u - \nabla\Pi_k^{\nabla,E} u_{\mathcal{I}}|_{1,E}) \|e_h\|_{0,\Gamma_E} \\
&\lesssim (\|\nabla u - \nabla\Pi_k^{\nabla,E} u_{\mathcal{I}}\|_{0,E} + h_E |\nabla u - \nabla\Pi_k^{\nabla,E} u_{\mathcal{I}}|_{1,E}) \|e_h\|_{\mathcal{N},E} \,.
\end{aligned}$$

Adding and subtracting $\Pi_k^{\nabla,E} u$, using triangular inequality and Lemma 1.4, we obtain

$$T_{\mathcal{N},1}^E \lesssim h_E^k |u|_{k+1,E} \|v_h\|_{\mathcal{N}} .$$

The second term is treated in a very similar way. The main difference is that we have to use a polynomial trace inequality on the normal derivative of v_h

$$\begin{aligned}
T_{\mathcal{N},2}^E &= \langle u, \nabla \Pi_k^{\nabla,E} e_h \cdot \boldsymbol{n}^E \rangle_{\Gamma_E} - \langle u_{\mathcal{I}}, \nabla \Pi_k^{\nabla,E} e_h \cdot \boldsymbol{n}^E \rangle_{\Gamma_E} \\
&= \langle u - u_{\mathcal{I}}, \nabla \Pi_k^{\nabla,E} e_h \cdot \boldsymbol{n}^E \rangle_{\Gamma_E} \\
&\lesssim \|u - u_{\mathcal{I}}\|_{0,\Gamma_E} \|\nabla \Pi_k^{\nabla,E} e_h \cdot \boldsymbol{n}^E\|_{0,\Gamma_E} \\
&\lesssim (h_E^{-\frac{1}{2}} \|u - u_{\mathcal{I}}\|_{0,E} + h_E^{-\frac{1}{2}} |u - u_{\mathcal{I}}|_{1,E}) h_E^{\frac{1}{2}} \|\nabla e_h\|_{0,E} \\
&\lesssim h_E^k |u|_{k+1,E} \|e_h\|_{\mathcal{N}} .
\end{aligned}$$

Finally, using again the trace inequality, we bound the last term as

$$\begin{aligned}
T_{\mathcal{N},3}^E &= \langle u, e_h \rangle_{\Gamma_E} - \langle u_{\mathcal{I}}, e_h \rangle_{\Gamma_E} = \langle u - u_{\mathcal{I}}, e_h \rangle_{\Gamma_E} \\
&\lesssim \|u - u_{\mathcal{I}}\|_{0,\Gamma_E} \|e_h\|_{0,\Gamma_E} \\
&\lesssim (\|u - u_{\mathcal{I}}\|_{0,E} + h_E |u - u_{\mathcal{I}}|_{1,E})\, h_E^{-\frac{1}{2}} \|e_h\|_{0,\Gamma_E} \\
&\lesssim h_E^k |u|_{k+1,E} \|e_h\|_{\mathcal{N}} .
\end{aligned}$$

Combining the last three estimates with the one for T_a and then dividing by $\|e_h\|_{\mathcal{N}}$, we conclude. □

2.3 Nitsche method for nonconforming VEM

2.3.1 Virtual element spaces and forms

We introduce the nonconforming virtual element space as

$$\begin{aligned}
V_h^{k,nc}(\Omega_h) := \big\{ v_h \in L^2(\Omega) \quad \text{such that} \quad & v_h|_E \in V_h^{k,nc}(E) \quad \forall E \in \Omega_h , \\
& \int_e [\![v_h]\!] \cdot \boldsymbol{n}^e \, q = 0 \quad \forall q \in \mathbb{P}_{k-1}(e) , \quad \forall e \in \mathcal{E}_h^o \big\} .
\end{aligned} \tag{2.13}$$

This space is equivalent to the space (1.18). Similarly to the conforming space, we take the spaces defined in the first chapter without imposing any boundary condition. Hence, the set of DoFs is bigger with respect to the one presented in Section 1.4 because the set $\mathcal{E}^k$ contains also the moments of the boundary edges. The dimension of the space is

$$N_{\Omega_h}^{nc} = k\,|\mathcal{E}_h| + \frac{(k-1)(k-2)}{2}\,|\Omega_h| .$$

Contrary to the conforming case, the trace of a nonconforming virtual function is unknown. Hence, when we consider the consider the term

$$\langle u_h, v_h \rangle_{\Gamma_E} ,$$

we have to insert a projection operator. Hence we consider the bilinear form

$$\begin{aligned}
\mathcal{N}_{nc,h}^E(u_h, v_h) := & -\langle \nabla \Pi_k^{\nabla,E} u_h \cdot \boldsymbol{n}^e, v_h \rangle_{\Gamma_E} - \langle u_h, \nabla \Pi_k^{\nabla,E} v_h \cdot \boldsymbol{n}^e \rangle_{\Gamma_E} \\
& + \sum_{e \subset \Gamma_E} \frac{1}{\delta h_E} \langle \Pi_{k-1}^{0,e} u_h, \Pi_{k-1}^{0,e} v_h \rangle_e ,
\end{aligned} \tag{2.14}$$

where the projection $\Pi_{k-1}^{0,e} : V_h^{k,nc}(E) \to \mathbb{P}_{k-1}(e)$ is the $L^2(e)$-orthogonal projection into the space of polynomial on the boundary edge e. Consequentially, we have to change also the definition of the load term

$$\mathcal{F}_{nc,h}^{\mathcal{N},E}(v_h) := \mathcal{F}_h^E(v_h) - \langle g, \nabla \Pi_k^{\nabla,E} v_h \cdot \boldsymbol{n}^e \rangle_{\Gamma_E} + \sum_{e \subset \Gamma_E} \frac{1}{\delta h_E} \langle g, \Pi_{k-1}^{0,e} v_h \rangle_e .$$

After introducing the local bilinear form

$$\mathcal{A}^E_{nc}(u_h, v_h) := a^E_h(u_h, v_h) + \mathcal{N}^{E,nc}_h(u_h, v_h)\,,$$

we define the local forms

$$\mathcal{A}_{nc}(u_h, v_h) =: \sum_{E\in\Omega_h} \mathcal{A}^E_{nc}(u_h, v_h)\,, \quad \mathcal{N}_{nc,h}(u_h, v_h) =: \sum_{E\in\Omega_h} N^E_{nc,h}(u_h, v_h)\,,$$

$$\mathcal{F}^{\mathcal{N}}_{nc,h}(v_h) =: \sum_{E\in\Omega_h} \mathcal{F}^{\mathcal{N},E}_{nc,h}(v_h)\,.$$

The discrete problem reads as

$$\begin{cases} \text{find } u \in V^{k,nc}_h(\Omega_h) \text{ such that:} \\ \mathcal{A}_{nc}(u_h, v_h) = \mathcal{F}^{\mathcal{N}}_{nc,h}(v_h) \quad \forall v_h \in V^{k,nc}_h(\Omega_h)\,. \end{cases} \tag{2.15}$$

2.3.2 Theoretical analysis

In this section, we extend Section 2.2.2, Section 2.2.3 and Section 2.2.4 to the nonconforming method. Since the steps are almost completely identical, we will omit some details. In addition to nonconformity, the main difference is the presence of the projection operator $\Pi^{0,e}_{k-1}$ in the Nitsche term. As stated in Section 2.2.2, the solution u of (1.1) does not fit into the discrete scheme. Despite this, it solves a problem analogous to the one presented in Section 2.2.2 except for the presence of the $\Pi^{0,e}_{k-1}$ projection in the last term of the Nitsche bilinear form. It solves

$$\tilde{\mathcal{A}}_{nc}(u, v_h) = \tilde{\mathcal{F}}^{\mathcal{N}}_{nc,h}(v_h) \qquad \text{for all } v_h \in V^{k,nc}_h(\Omega_h)\,,$$

where

$$\tilde{\mathcal{A}}_{nc}(u, v_h) := \sum_{E\in\Omega_h} \tilde{\mathcal{A}}^E_{nc}(u, v_h) := \sum_{E\in\Omega_h} a^E(u, v_h) + \tilde{\mathcal{N}}^E_{nc,h}(u, v_h)\,, \quad \tilde{\mathcal{F}}^{\mathcal{N}}_{nc,h}(v_h) := \sum_{E\in\Omega_h} \tilde{\mathcal{F}}^{\mathcal{N},E}_{nc,h}(v_h)\,.$$

The local Nitsche term is defined as

$$\tilde{\mathcal{N}}^E_h(u, v_h) := -\langle \nabla u \cdot \boldsymbol{n}^E, v_h \rangle_{\Gamma_E} - \langle u, \nabla \Pi^{\nabla,E}_k v_h \cdot \boldsymbol{n}^E \rangle_{\Gamma_E} + \sum_{e\subset\Gamma_E} \frac{1}{\delta h_E} \langle \Pi^{0,e}_{k-1} u, \Pi^{0,e}_{k-1} v_h \rangle_e\,.$$

The right-hand side is defined as

$$\tilde{\mathcal{F}}^{\mathcal{N},E}_h(v_h) := \int_E f\, v_h \,\mathrm{d}E - \langle g, \nabla \Pi^{\nabla,E}_k v_h \cdot \boldsymbol{n}^E \rangle_{\Gamma_E} + \sum_{e\subset\Gamma_E} \frac{1}{\delta h_E} \langle \Pi^{0,e}_{k-1} g, \Pi^{0,e}_{k-1} v_h \rangle_e\,.$$

We also set

$$\tilde{\mathcal{N}}_{nc,h}(u, v_h) := \sum_{E\in\Omega_h} \tilde{\mathcal{N}}^E_{nc,h}(u, v_h)\,.$$

Also the definition of the norm (2.10) is changed. Here, we define the local norm as

$$\|v_h\|^2_{\mathcal{N},nc,E} := |v_h|^2_{1,E} + \sum_{e\subset\Gamma_E} \frac{1}{\delta h_E} \|\Pi^{0,e}_{k-1} v_h\|^2_{0,e}\,.$$

Obviously, the global norm is obtained by summing all the local norms

$$\|v_h\|_{\mathcal{N},nc} := \left(\sum_{E\in\Omega_h} \|v_h\|^2_{\mathcal{N},nc,E} \right)\,.$$

We can prove the following coercivity result.

Proposition 2.5 (Coercivity)**.** *Under assumptions **(A1)**, the bilinear form $\mathcal{A}_{nc}(\cdot,\cdot)$ is coercive with respect to $\|\cdot\|_{\mathcal{N},nc}$. It holds*

$$\mathcal{A}_{nc}(v_h, v_h) \gtrsim \|v_h\|^2_{\mathcal{N},nc}\,, \quad \forall v_h \in V_h^{k,nc}(\Omega_h)\,,$$

for a suitable choice of the parameter δ.

Proof. Thanks to the orthogonality of the projection operator $\Pi^{0,e}_{k-1}$, we have

$$\langle \nabla \Pi^{\nabla,E}_k u_h \cdot \boldsymbol{n}^e, v_h\rangle_{\Gamma_E} = \langle \nabla \Pi^{\nabla,E}_k u_h \cdot \boldsymbol{n}^e, \Pi^{0,e}_{k-1} v_h\rangle_{\Gamma_E}$$

and

$$\langle u_h, \nabla \Pi^{\nabla,E}_k v_h \cdot \boldsymbol{n}^e\rangle_{\Gamma_E} = \langle \Pi^{0,e}_{k-1} u_h, \nabla \Pi^{\nabla,E}_k v_h \cdot \boldsymbol{n}^e\rangle_{\Gamma_E}$$

The proof is completed by mimic the same procedure of Proposition 2.3. □

Finally, we can prove the following error estimate under the assumption

$$u \in H^2(\Omega) \cap H^{k+1}(\Omega_h)\,, \quad f \in H^k(\Omega_h)\,.$$

Theorem 2.6 (Error estimate)**.** *Let u be the solution of* (1.2) *and u_h be the solution of* (1.22)*. Under assumption **(A1)**, it holds*

$$\|u - u_h\|_{\mathcal{N},nc} \lesssim \sum_{E \in \Omega_h} \Theta^E\, h_E^k\,,$$

where the constant Θ_E depends on $\|u\|_{k+1,E}$ and $\|f\|_{k,E}$.

Proof. We repeat some steps of Theorem 1.16 and Theorem 2.4. Adding and subtracting $u_{\mathcal{I}}$, we obtain

$$\|u - u_h\|_{\mathcal{N},nc} \le \|u - u_{\mathcal{I}}\|_{\mathcal{N},nc} + \|u_{\mathcal{I}} - u_h\|_{\mathcal{N},nc} = \|e_{\mathcal{I}}\|_{\mathcal{N},nc} + \|e_h\|_{\mathcal{N},nc}\,.$$

Following the steps in Theorem 2.4, and using the fact that $\|\Pi^{0,e}_{k-1} v_h\|_{0,e} \le \|v_h\|_{0,e}$ for every boundary edge $e \in \mathcal{E}_h^{\partial}$ and every virtual function $v_h \in V_h^{k,nc}(\Omega_h)$, it holds that

$$\|e_{\mathcal{I}}\|_{\mathcal{N},nc} \le \sum_{E \in \Omega_h} h_E^k\, |u|_{k+1,E}\,.$$

To bound the second term, we exploit the coercivity of the bilinear form $\mathcal{A}_{nc}(\cdot,\cdot)$

$$\begin{aligned}\|e_h\|^2_{\mathcal{N},nc} &\lesssim \mathcal{A}_{nc}(u_{\mathcal{I}} - u_h, e_h) = \mathcal{A}_{nc}(u_{\mathcal{I}}, e_h) - \mathcal{F}^{\mathcal{N}}_{nc,h}(e_h)\\ &= \mathcal{A}_{nc}(u_{\mathcal{I}}, e_h) - \tilde{\mathcal{A}}_{nc}(u_{\mathcal{I}}, e_h) + \tilde{\mathcal{F}}^{\mathcal{N}}_{nc,h}(e_h) - \mathcal{F}^{\mathcal{N}}_{nc,h}(e_h) + \mathcal{B}(u, v_h)\,.\end{aligned}$$

Here, $\mathcal{B}(u, v_h)$ is the same bilinear form that appears in (1.23) and it can be estimated as in Proposition 1.14

$$|\mathcal{B}(u, v_h)| \lesssim \sum_{E \in \Omega_h} h_E^k |u|_{k+1,E}\, |v_h|_{1,\Omega_h} \lesssim \sum_{E \in \Omega_h} h_E^k |u|_{k+1,E}\, \|v_h\|_{\mathcal{N},nc}\,.$$

Depending on whether $k = 1$ or $k \ge 2$, the term $\tilde{\mathcal{F}}^{\mathcal{N}}_{nc,h}(e_h) - \mathcal{F}^{\mathcal{N}}_{nc,h}(e_h)$ is estimated as (1.21) or (1.12) respectively. We remark that, since $u = g$ on the boundary, the terms containing g do not appear in the estimate. It remains

$$\mathcal{A}_{nc}(u_{\mathcal{I}}, e_h) - \tilde{\mathcal{A}}_{nc}(u_{\mathcal{I}}, e_h) = a_h(u_{\mathcal{I}}, e_h) - a(u, e_h) + \mathcal{N}_{nc,h}(u_{\mathcal{I}}, e_h) - \tilde{\mathcal{N}}_{nc,h}(u, e_h) =: T_a + T_{\mathcal{N}}\,,$$

where

$$T_a \lesssim \sum_{E \in \Omega_h} h_E^k |u|_{k+1,E} \|e_h\|_{\mathcal{N},nc}\,.$$

Following the proof of Theorem 2.4, we have

$$\begin{aligned}
T_{\mathcal{N}} &= \mathcal{N}^E_{nc,h}(u_{\mathcal{I}}, e_h) - \tilde{\mathcal{N}}^E_{nc,h}(u, e_h) \\
&= \sum_{E\in\Omega_h} -\langle \nabla\Pi^{\nabla,E}_k u_{\mathcal{I}} \cdot \boldsymbol{n}^e, e_h\rangle_{\Gamma_E} + \langle \nabla u \cdot \boldsymbol{n}^e, e_h\rangle_{\Gamma_E} \\
&\quad + \sum_{E\in\Omega_h} -\langle u_{\mathcal{I}}, \nabla\Pi^{\nabla,E}_k e_h \cdot \boldsymbol{n}^e\rangle_{\Gamma_E} + \langle u, \nabla\Pi^{\nabla,E}_k e_h \cdot \boldsymbol{n}^e\rangle_{\Gamma_E} \\
&\quad + \sum_{E\in\Omega_h} \sum_{e\subset\Gamma_E} \frac{1}{\delta h_E} \langle \Pi^{0,e}_{k-1} u_{\mathcal{I}}, \Pi^{0,e}_{k-1} e_h\rangle_e - \langle \Pi^{0,e}_{k-1} u, \Pi^{0,e}_{k-1} e_h\rangle_e \\
&=: \sum_{E\in\Omega_h} T^E_{\mathcal{N},1} + T^E_{\mathcal{N},2} + T^E_{\mathcal{N},3}\,.
\end{aligned}$$

The term $T^E_{\mathcal{N},2}$ can be estimated as in Theorem 2.4 and we have

$$T^E_{\mathcal{N},2} \lesssim h^k_E |u|_{k+1,E} \|e_h\|_{\mathcal{N},nc,E}\,.$$

To estimate $T^E_{\mathcal{N},1}$, we use the trace inequality, Lemma 1.9, the orthogonality of the projection $\Pi^{0,e}_0$ and the fact that the norm of $\Pi^{0,e}_{k-1}$ is bigger than the norm of $\Pi^{0,e}_0$

$$\begin{aligned}
T^E_{\mathcal{N},1} &= \langle \nabla u \cdot \boldsymbol{n}^e, e_h\rangle_{\Gamma_E} - \langle \nabla\Pi^{\nabla,E}_k u_{\mathcal{I}} \cdot \boldsymbol{n}^e, e_h\rangle_{\Gamma_E} \\
&= \langle \nabla(u - \nabla\Pi^{\nabla,E}_k u_{\mathcal{I}}) \cdot \boldsymbol{n}^e, e_h\rangle_{\Gamma_E} \\
&\lesssim \big(h_E^{-\frac{1}{2}} \|\nabla u - \nabla\Pi^{\nabla,E}_k u_{\mathcal{I}}\|_{0,E} + h_E^{\frac{1}{2}} |\nabla u - \nabla\Pi^{\nabla,E}_k u_{\mathcal{I}}|_{1,E} \big) \|e_h\|_{0,\Gamma_E} \\
&\lesssim h^k_E |u|_{k+1,E} h_E^{-\frac{1}{2}} \|e_h\|_{0,e} \\
&\lesssim h^k_E |u|_{k+1,E} \sum_{e\subset\Gamma_E} h_E^{-\frac{1}{2}} \big(\|e_h - \Pi^{0,e}_0 v_h\|_{0,e} + \|\Pi^{0,e}_0 e_h\|_{0,e} \big) \\
&\lesssim h^k_E |u|_{k+1,E} \big(|e_h|_{1,E} + \sum_{e\subset\Gamma_E} h_E^{-\frac{1}{2}} \|\Pi^{0,e}_{k-1} e_h\|_{0,e} \big) \lesssim h^k_E |u|_{k+1,E} \, \|e_h\|_{\mathcal{N},nc,E}\,.
\end{aligned}$$

It remains

$$\begin{aligned}
T^E_{\mathcal{N},3} &= \sum_{e\subset\Gamma_E} \langle \Pi^{0,e}_{k-1} u, \Pi^{0,e}_{k-1} e_h\rangle_e - \langle \Pi^{0,e}_{k-1} u_{\mathcal{I}}, \Pi^{0,e}_{k-1} e_h\rangle_e \\
&= \sum_{e\subset\Gamma_E} \langle u - u_{\mathcal{I}}, \Pi^{0,e}_{k-1} e_h\rangle_e \\
&\lesssim \sum_{e\subset\Gamma_E} \|u - u_{\mathcal{I}}\|_{0,e} \|e_h\|_{0,e} \\
&\lesssim (\|u - u_{\mathcal{I}}\|_{0,E} + h_E |u - u_{\mathcal{I}}|_{1,E}) \sum_{e\subset\Gamma_E} h_E^{-\frac{1}{2}} \|\Pi^{0,e}_{k-1} e_h\|_{0,e} \\
&\lesssim h^k_E |u|_{k+1,E} \|e_h\|_{\mathcal{N},nc}\,.
\end{aligned}$$

□

Chapter 3

CIP-stabilized Virtual Elements for diffusion-convection-reaction problems

In this chapter, we consider the classical diffusion-reaction-advection scalar problem. Under appropriate assumptions on the data, this is a standard "textbook" elliptic problem without significant difficulties. However, it is well known that when the advective term becomes dominant, especially over the diffusive term, a classical FEM approach results in substantial errors and oscillations in the discrete solution unless an extremely fine mesh is used. It is important to note that the diffusion-reaction-advection problem also serves as a model for more complex problems in fluid mechanics, such as the Navier-Stokes equations or the Oseen equations.

The Virtual Element Method is particularly suitable for advection-dominated problems due to its flexible mesh construction and handling. VEM allows for more localized refinement procedures and the seamless integration of fine meshes with coarser ones, which is especially useful in the presence of layers. Additionally, VEM offers efficient discretization of complex domains, which is invaluable in applications such as reservoir [2] and fracture-network simulations [22, 20], where diffusion-reaction-advection equations are critical. As mentioned in the introduction, the SUPG and LPS approaches were already extended to VEM in [21, 23, 14] and [48], respectively. The purpose of the present contribution is exactly to fill this gap and develop CIP stabilized VEM method, providing also a theoretical error analysis. Of course, our method combines VEM stabilization terms (to deal with polygonal meshes) and CIP-like terms (to deal with the advection-dominated regime). Furthermore, it is worth noticing that the backstage complex nature of CIP, which is a "minimal stabilization" as it adds the minimal positive term guaranteeing control on piecewise polynomial convection, makes the analysis in the VEM setting particularly interesting and challenging. We are able to develop quasi-robust error estimates for our method, when quasi-uniform mesh families are employed and a uniform reactive term is included. In the absence of reaction, we are able to show some improved error estimates if the convective term is a piecewise polynomial.

The chapter is organized as follows. After presenting the continuous and discrete problems in Section 3.1, we develop the stability and convergence analysis in Section 3.2. The chapter ends with a set of numerical tests showing the actual robustness of the method and comparing it with the non-stabilized approach.

3.1 The continuous and the discrete problems

3.1.1 Model problem

Instead of considering the Poisson problem (1.1), we consider the steady advection-diffusion-reaction equation with homogeneous boundary conditions:

$$\begin{cases} \text{find } u : \Omega \to \mathbb{R} \text{ such that:} \\ -\epsilon\,\Delta u + \boldsymbol{\beta}\cdot\nabla u + \sigma\, u = f & \text{in } \Omega\,, \\ u = 0 & \text{on } \Gamma\,, \end{cases} \tag{3.1}$$

where $\Omega \subset \mathbb{R}^2$ is a polygonal domain of boundary Γ. Here, $\epsilon > 0$ is the diffusion coefficient, assumed to be constant. The advection field $\boldsymbol{\beta} \in [W^1_\infty(\Omega)]^2$ is such that $\nabla\cdot\beta = 0$. Additionally, $\sigma > 0$ is the reaction constant. It is worth noting that we assume σ to be a positive constant, although the extension to the case where $0 < \sigma \in L^\infty(\Omega)$ with $\sigma^{-1} \in L^\infty(\Omega)$ is straightforward. Finally, $f \in L^2(\Omega)$ represents the source term. The domain boundary will be divided into two non-overlapping regions:

$$\Gamma_{\text{in}} := \{\boldsymbol{x} \in \Gamma \mid (\boldsymbol{\beta}(\boldsymbol{x})\cdot\boldsymbol{n}) < 0\}\,,$$

and

$$\Gamma_{\text{out}} := \{\boldsymbol{x} \in \Gamma \mid (\boldsymbol{\beta}(\boldsymbol{x})\cdot\boldsymbol{n}) \geq 0\}\,.$$

A variational formulation of problem (3.1) reads as follows:

$$\begin{cases} \text{find } u \in V(\Omega) := H^1_0(\Omega) \text{ such that:} \\ \epsilon\, a(u,v) + b^{\text{skew}}(u,v) + \sigma\, c(u,v) = \displaystyle\int_\Omega f\, v \,\mathrm{d}\Omega \quad \forall v \in V(\Omega)\,. \end{cases} \tag{3.2}$$

The bilinear forms $a(\cdot,\cdot)\colon V(\Omega)\times V(\Omega) \to \mathbb{R}$, $b^{\text{skew}}(\cdot,\cdot)\colon V(\Omega)\times V(\Omega) \to \mathbb{R}$ and $c(\cdot,\cdot)\colon V(\Omega)\times V(\Omega) \to \mathbb{R}$ are defined as

$$a(u,v) := \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}\Omega \qquad \forall u, v \in V(\Omega)\,,$$

$$b^{\text{skew}}(u,v) := \frac{1}{2}\big(b(u,v) - b(v,u)\big) \quad \text{with} \quad b(u,v) := \int_\Omega (\boldsymbol{\beta}\cdot\nabla u)\, v \,\mathrm{d}\Omega \qquad \forall u, v \in V(\Omega)\,,$$

$$c(u,v) := \int_\Omega u\, v \,\mathrm{d}\Omega \qquad \forall u, v \in V(\Omega)\,.$$

It is well known that when ϵ is small compared to $\boldsymbol{\beta}$ and/or σ, standard discretizations of equation (3.2) often produce unsatisfactory numerical solutions with spurious oscillations. To address these issues, several strategies have been proposed in the literature. In this chapter, we employ the Continuous Interior Penalty (CIP) method, as introduced in [35] within a Finite Element framework. Given our primary interest in the advection-dominated case, we will assume from this point onward that the material parameters are scaled as follows:

$$\|\boldsymbol{\beta}\|_{[L^\infty(\Omega)]^2} = 1\,, \tag{3.3}$$

and we may consider that $\epsilon \ll 1$.

3.1.2 Mesh assumptions

In this section, we revisit the mesh assumptions presented in Section 1.2. The key change compared to Section 1.2 is the requirement for quasi-uniformity of the mesh. We consider the following mesh assumptions:

(A-C) Mesh assumption. There exists a positive constant ρ such that for any $E \in \{\,\Omega_h\,\}_h$:

- E is star-shaped with respect to a ball B_E of radius greater or equal than $\rho\, h_E$,
- any edge e of E has length greater or equal than $\rho\, h_E$,
- the mesh is quasi-uniform, any polygon has diameter $h_E \geq \rho\, h$.

The reason for requiring quasi-uniformity of the mesh is due to the Oswald interpolant construction (3.19) and the definition (3.34). We will provide more details later in this chapter.

3.1.3 Virtual Element spaces

Given a polygon $E \in \Omega_h$ and a positive integer k, we define the local "enhanced" virtual element space as

$$\begin{aligned} V_h^k(E) = \big\{ v_h \in & H^1(E) \cap C^0(\partial E) \quad \text{such that} \quad v_h|_{\partial E} \in \mathbb{B}_k(\partial E), \\ & \Delta v_h \in \mathbb{P}_k(E)\,, \quad (v_h - \Pi_k^{\nabla,E} v_h,\, \widehat{p}_k)_{0,E} = 0 \quad \text{for all } \widehat{p}_k \in \mathbb{P}_k(E)/\mathbb{P}_{k-2}(E) \big\}\,. \end{aligned} \tag{3.4}$$

The definition of this virtual element space differs from the one presented in (1.4). The distinction lies in the Laplacian of the virtual functions. In (1.4), the Laplacian is required to be a polynomial of degree $k-2$. Here, however, we require the Laplacian to be a polynomial of degree k. To manage the larger resulting space, we introduce a constraint on the $\Pi_k^{\nabla,E}$ projection of the virtual functions. This new definition allows the construction of the projection operator $\Pi_k^{0,E} : V_h^k(E) \to \mathbb{P}_k(E)$. Despite the change in the definition of the local virtual element space, we can prove that the same set of DoFs of the space (1.4) also serves as a set of DoFs for (3.4). Here, we recall the definition of the DoFs as it will be instrumental in the future:

- $\mathcal{V}_c^{E,k}$: the pointwise values of v_h at the vertexes of the polygon E,
- $\mathcal{E}_c^{E,k}$: the values of v_h at $k-1$ internal points of a Gauss-Lobatto quadrature for every edge $e \subset \partial E$,
- $\mathcal{P}_c^{E,k}$: the moments up to the order $k-2$ on E:

$$\mu_E^{\boldsymbol{\alpha}}(v_h) \coloneqq \frac{1}{|E|} \int_E v_h \left(\frac{\mathbf{x} - \mathbf{x}_E}{h_E} \right)^{\boldsymbol{\alpha}} \mathrm{d}E \quad |\boldsymbol{\alpha}| \leq k-2, \tag{3.5}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T$ is a multi-index.

By gluing together the local spaces, we define the global virtual element space as

$$V_h^k(\Omega_h) = \{ v_h \in H^1(\Omega) \quad \text{such that} \quad v_h|_E \in V_h^k(E) \quad \text{for all } E \in \Omega_h \}\,,$$

with the associated set of degrees of freedom:

- $\mathcal{V}_c^k$: the values of v_h at the vertices,
- $\mathcal{E}_c^k$: the values of v_h at $k-1$ points on each edge $e \in \mathcal{E}_h$,
- $\mathcal{P}_c^k$: the moments up to order $k-2$ for each element $E \in \Omega_h$.

Similarly to the space presented in Section 1.3.1, the following interpolation estimate holds:

Lemma 3.1 (Approximation with enhanced conforming virtual element functions)**.** *Under assumption **(A1)**, for any $v \in H_0^1(\Omega) \cap H^{s+1}(\Omega_h)$, there exists $v_{\mathcal{I}} \in V_h^k(\Omega_h)$, such that for all $E \in \Omega_h$,*

$$\|v - v_{\mathcal{I}}\|_{0,E} + h_E \|\nabla(v - v_{\mathcal{I}})\|_{0,E} \lesssim h_E^{s+1} |v|_{s+1,E}\,,$$

where $0 < s \leq k$.

3.1.4 Virtual Element Forms and the Discrete Problem

We begin by noting that the bilinear forms $a(\cdot,\cdot)$, $b^{\text{skew}}(\cdot,\cdot)$ and $c(\cdot,\cdot)$ can clearly be decomposed into local contributions

$$a(u,v) =: \sum_{E \in \Omega_h} a^E(u,v)\,, \quad b^{\text{skew}}(u,v) =: \sum_{E \in \Omega_h} b^{\text{skew},E}(u,v)\,, \quad c(u,v) =: \sum_{E \in \Omega_h} c^E(u,v)\,. \tag{3.6}$$

As in the previous chapters, we need to devise some computable counterparts for these bilinear forms. The diffusion term is replace by the bilinear form $a_h^E(\cdot,\cdot)\colon V_h^k(E) \times V_h^k(E) \to \mathbb{R}$ defined as

$$a_h^E(u_h, v_h) := \int_E \boldsymbol{\Pi}_{k-1}^{0,E} \nabla u_h \cdot \boldsymbol{\Pi}_{k-1}^{0,E} \nabla v_h \,\mathrm{d}E + \mathcal{S}^E\big((I - \Pi_k^{\nabla,E}) u_h,\, (I - \Pi_k^{\nabla,E}) v_h\big)\,. \tag{3.7}$$

Here, the term $\mathcal{S}^E\colon V_h^k(E) \times V_h^k(E) \to \mathbb{R}$ is a stabilization term introduced in (1.8). In our numerical example we have always adopt the `dofi-dofi` stabilization term.

Remark 3.2. *Since in this chapter we are considering a coefficient in front of the Laplacian, we choose to discretize the bilinear form $a(\cdot,\cdot)$ in this way. Although we are currently dealing with a constant coefficient, this approach is more convenient if we intend to address the case with a variable diffusive coefficient in the future.*

The convective form is replaced by $b_h^E(\cdot,\cdot)\colon V_h^k(E)\times V_h^k(E)\to\mathbb{R}$ defined as

$$b_h^E(u_h,v_h) := \int_E \boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}u_h\,\Pi_k^{0,E}v_h\,\mathrm{d}E + \int_{\partial E}(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)(I-\Pi_k^{0,E})u_h\,\Pi_k^{0,E}v_h\,\mathrm{d}s\,. \tag{3.8}$$

In the numerical scheme, we will employ the skew-symmetrized form:

$$b_h^{\mathrm{skew},E}(u_h,v_h) = \frac{1}{2}\left(b_h^E(u_h,v_h) - b_h^E(v_h,u_h)\right)\,.$$

Remark 3.3. *At the continuous level, there is no distinction between $b(\cdot,\cdot)$ and $b^{\mathrm{skew}}(\cdot,\cdot)$. However, when we instert the projections $\Pi_k^{0,E}$ into the discretization of $b(\cdot,\cdot)$, we lose the skew-symmetry property.*

Finally, the reaction form is locally replaced by $c_h^E(\cdot,\cdot)\colon V_h^k(E)\times V_h^k(E)\to\mathbb{R}$, defined as

$$c_h^E(u_h,v_h) := \int_E \Pi_k^{0,E}u_h\,\Pi_k^{0,E}v_h\,\mathrm{d}E + |E|\,\mathcal{S}^E\big((I-\Pi_k^{0,E})u_h,\,(I-\Pi_k^{0,E})v_h\big)\,.$$

Based on [35, 33], we now present a VEM version of the local CIP-stabilization form, defined as

$$J_h^E(u_h,v_h) := \sum_{e\subset\partial E}\frac{\gamma_e}{2}\int_e h_e^2\,[\![\nabla\Pi_k^0 u_h]\!]\cdot[\![\nabla\Pi_k^0 v_h]\!]\,\mathrm{d}s + \gamma_E\,h_E\,\mathcal{S}^E\big((I-\Pi_k^{\nabla,E})u_h,(I-\Pi_k^{\nabla,E})v_h\big)\,, \tag{3.9}$$

where $[\![\nabla\cdot]\!]$ denotes the gradient jump in the normal direction across e defined in (1.19). If e is a boundary edge we set $[\![\nabla\cdot]\!]=0$. The parameters γ_e and γ_E are defined as

$$\gamma_e := \kappa_e\|\boldsymbol{\beta}\|_{[L^\infty(e)]^2}\;, \qquad \gamma_E := \kappa_E\|\boldsymbol{\beta}\|_{[L^\infty(\partial E)]^2}\;, \tag{3.10}$$

where κ_e and κ_E are positive constants to be chosen. Specifically, within an element E where $\boldsymbol{\beta}=0$, both γ_e and γ_E vanish, resulting in the absence of CIP stabilization.

Remark 3.4. *The presence of two different parameters, γ_e and γ_E, is for implementation reasons. The first part is computed using a for loop over the internal edges, while the second part is computed using a for loop over the elements. Moreover, the first part requires knowledge of the Π_k^0 projection. Therefore, a loop over the elements must have already been completed.*

Remark 3.5. *In [35], the authors set $\gamma_e=\gamma_E=0.025$ in their numerical results. In [33], they prove that the correct scaling for γ_e and γ_E is $k^{-\frac{7}{2}}$. Although this result applies to FEM, it underscores that as the discrete local space increases, the need for a CIP-stabilization term decreases.*

Moreover, we impose the Dirichlet boundary conditions using a Nitsche-type technique (see Chapter 2). To this end, we define the local forms

$$\begin{aligned}\mathcal{N}_h^E(u_h,v_h) := &-\epsilon\,\langle u_h,\boldsymbol{\Pi}_{k-1}^{0,E}\nabla v_h\cdot\boldsymbol{n}^E\rangle_{\Gamma_E} - \epsilon\,\langle\boldsymbol{\Pi}_{k-1}^{0,E}\nabla u_h\cdot\boldsymbol{n}^E,v_h\rangle_{\Gamma_E}\\ &+\frac{\epsilon}{\delta h_E}\,\langle u_h,v_h\rangle_{\Gamma_E} + \frac{1}{2}\,\langle|\boldsymbol{\beta}\cdot\boldsymbol{n}^E|u_h,v_h\rangle_{\Gamma_E}\,,\end{aligned}$$

where δ is a parameter to be chosen according to Proposition 2.3.

Remark 3.6. *The bilinear form we are discussing here differs from the one presented in [17]. Here, we also consider the term*

$$-\epsilon\,\langle u_h,\boldsymbol{\Pi}_{k-1}^{0,E}\nabla v_h\cdot\boldsymbol{n}^E\rangle_{\Gamma_E}\,.$$

Additionally, we have modified the projection operator in the normal derivative to make it consistent with the one used in the definition of $a_h(\cdot,\cdot)$.

Remark 3.7. *The last term arises due to the presence of the convection form. Typically, it is locally defined as*

$$-\langle(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)u_h, v_h\rangle_{\Gamma_{E,in}}\,,$$

with $\Gamma_{E,in} := \partial E \cap \Gamma_{in}$*. By integration by parts, in the definition of* $b^{\text{skew}}(\cdot,\cdot)$*, we should consider also*

$$\frac{1}{2}\langle(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)u_h, v_h\rangle_{\Gamma_E}\,.$$

Summing the last two terms, we recover our definition of $\mathcal{N}_h^E(\cdot,\cdot)$*.*

Summing all of these contributions, we construct the discrete bilinear form $\mathcal{A}_{\text{cip}}^E \colon V_h^k(E)\times V_h^k(E)\to\mathbb{R}$ as

$$\mathcal{A}_{\text{cip}}^E(u_h,v_h) = \epsilon\, a_h^E(u_h,v_h) + b_h^{\text{skew},E}(u_h,v_h) + \sigma\, c_h^E(u_h,v_h) + \mathcal{N}_h^E(u_h,v_h) + J_h^E(u_h,v_h)\,, \quad (3.11)$$

and summing over all the polygons, we obtain the global versions of the bilinear forms

$$a_h(u_h,v_h) := \sum_{E\in\Omega_h} a_h^E(u_h,v_h)\,, \qquad b_h^{\text{skew}}(u_h,v_h) := \sum_{E\in\Omega_h} b_h^{\text{skew},E}(u_h,v_h)\,,$$
$$c_h(u_h,v_h) := \sum_{E\in\Omega_h} c_h^E(u_h,v_h)\,, \qquad J_h(u_h,v_h) := \sum_{E\in\Omega_h} J_h^E(u_h,v_h)\,,$$
$$\mathcal{N}_h(u_h,v_h) := \sum_{E\in\Omega_h} \mathcal{N}_h^E(u_h,v_h)\,,$$

and

$$\mathcal{A}_{\text{cip}}(u_h,v_h) := \sum_{E\in\Omega_h} \mathcal{A}_{\text{cip}}^E(u_h,v_h)\,. \quad (3.12)$$

The discrete local and global right-hand side $\mathcal{F}_h^E \colon V_h^k(E)\to\mathbb{R}$ and $\mathcal{F}_h \colon V_h^k(\Omega_h)\to\mathbb{R}$ are defined as

$$\mathcal{F}_h^E(v_h) := \int_E f\,\Pi_k^{0,E} v_h \,\mathrm{d}E\,, \qquad \mathcal{F}_h(v_h) := \sum_{E\in\Omega_h}\mathcal{F}_h^E(v_h)\,. \quad (3.13)$$

Unlike the discretizations of the right-hand side presented in the previous chapter, the enhanced virtual element space allows for the construction of the Π_k^0 projection. Consequently, we can achieve a more accurate discretization of the load term. The discrete virtual element problem reads as:

$$\begin{cases} \text{find } u_h \in V_h^k(\Omega_h) \text{ such that} \\ \mathcal{A}_{\text{cip}}(u_h,\, v_h) = \mathcal{F}_h(v_h) \quad \forall v_h \in V_h^k(\Omega_h)\,. \end{cases} \quad (3.14)$$

3.1.5 Consistency of the method

Due to the polynomial projections involved in (3.14), it is evident that, as usual for VEMs, the solution u of the continuous problem (3.2) does not satisfy the discrete scheme (3.14) (thus, strong consistency does not hold). However, if u is more regular, say $u \in H^2(\Omega)\cap H_0^1(\Omega)$, then it holds:

$$\widetilde{\mathcal{A}}_{\text{cip}}(u,\, v_h) = \tilde{\mathcal{F}}(v_h) \qquad \forall v_h \in V_h^k(\Omega_h)\,. \quad (3.15)$$

where

$$\widetilde{\mathcal{A}}_{\text{cip}}(u,v_h) := \sum_{E\in\Omega_h}\widetilde{\mathcal{A}}_{\text{cip}}^E(u,v_h)\,, \quad \tilde{\mathcal{F}}(v_h) := \sum_{E\in\Omega_h}\tilde{\mathcal{F}}^E(v_h)\,, \quad (3.16)$$

and the local forms are defined as follows:

- $\widetilde{\mathcal{A}}_{\text{cip}}^E(u,v_h) := \epsilon\, a^E(u,v_h) + b^{\text{skew},E}(u,v_h) + \sigma\, c^E(u,v_h) + \tilde{\mathcal{N}}_h^E(u,v_h) + \tilde{J}_h^E(u,v_h)\,,$ (3.17) with

$$\tilde{\mathcal{N}}_h^E(u,v_h) := -\epsilon\langle \nabla u \cdot \boldsymbol{n}^E, v_h\rangle_{\Gamma_E} - \epsilon\langle u, \boldsymbol{\Pi}_{k-1}^{0,E}\nabla v_h \cdot \boldsymbol{n}^E\rangle_{\Gamma_E} + \frac{\epsilon}{\delta h_E}\langle u, v_h\rangle_{\Gamma_E} + \frac{1}{2}\langle|\boldsymbol{\beta}\cdot\boldsymbol{n}^E| u, v_h\rangle_{\Gamma_E}\,,$$

and

$$\tilde{\mathcal{J}}_h^E(u,v_h) := \frac{1}{2}\sum_{e\subset\partial E}\gamma_e\int_e h_e^2\,[\![\nabla u]\!]\cdot[\![\nabla v_h]\!]\,\mathrm{d}s\,.$$

Note that since $u \in H^2(\Omega)$, the jump of the gradient in the normal derivative is zero. Hence we have $\tilde{\mathcal{J}}_h^E(u,v_h) = 0$.

- $$\tilde{\mathcal{F}}^E(v_h) := \int_E f\,v_h\,\mathrm{d}E\,. \tag{3.18}$$

3.2 Stability and convergence analysis

3.2.1 Preliminary results

Before proving the stability of the discrete problem, we will discuss some preliminary results that are useful for our purposes. The first result is a standard inverse estimate for the virtual element functions. This result can be found in [57].

Lemma 3.8 (Inverse estimate)**.** *Under the assumption **(A1)**, for any $E \in \Omega_h$, there exists a uniform positive constant such that*

$$|v_h|_{1,E} \lesssim h_E^{-1}\|v_h\|_{0,E} \quad \forall v_h \in V_h^k(\Omega_h)\,.$$

We also recall, see [16, 26], the following inverse trace inequality.

Lemma 3.9 (Inverse trace inequality)**.** *Under the assumption **(A1)**, for any $E \in \Omega_h$ and for every $v_h \in V_h^k(E)$ such that $\Pi_{k-2}^{0,E}v_h \equiv 0$, it holds that*

$$\|v_h\|_{0,E} \lesssim h_E^{\frac{1}{2}}\|v_h\|_{0,\partial E}\,.$$

We now construct a VEM version of the Oswald interpolation operator, similar to what is discussed in [35, 33] for the FEM framework.. Consider a point ν associated with a degree of freedom in $\mathcal{E}_c^k$ or $\mathcal{V}_c^k$. We define $E_\nu := \bigcup\{E \in \Omega_h \text{ such that } \nu \in \partial E\}$, which is the union of all elements containing the point ν. The quasi-interpolation operator π for a sufficiently regular function v is defined as follows:

$$\pi v = \sum_{\nu\in\mathcal{V}_c^k\cup\mathcal{E}_c^k}\lambda_\nu(v)\varphi_\nu + \sum_{E\in\Omega_h}\sum_{|\boldsymbol{\alpha}|\le k-2}\mu_E^{\boldsymbol{\alpha}}(v)\varphi_E^{\boldsymbol{\alpha}}\,, \tag{3.19}$$

where $\{\varphi_\nu\}_{\nu\in\mathcal{V}_c^k\cup\mathcal{E}_c^k}$ are the canonical basis functions associated to the degree of freedom pointed at $\{\nu\}_{\nu\in\mathcal{V}_c^k\cup\mathcal{E}_c^k}$ and the coefficients $\{\lambda_\nu(v)\}$ are defined as

$$\lambda_\nu(v) := \frac{1}{|E_\nu|}\sum_{E\subseteq E_\nu} v^E(\nu)\,|E|\,. \tag{3.20}$$

Above, and from now on in this section, a superscript E for a function denotes the restriction of that function to the element E. Similarly, above $\{\varphi_E^{\boldsymbol{\alpha}}\}$ denote the basis functions associated to the degrees of freedom $\mathcal{P}_c^{E,k}$, and $\{\mu_E^{\boldsymbol{\alpha}}(v)\}$ are the associated coefficients corresponding to v, defined as (cf. (3.5)):

$$\mu_E^{\boldsymbol{\alpha}}(v) := \frac{1}{|E|}\int_E v\left(\frac{\mathbf{x}-\mathbf{x}_E}{h_E}\right)^{\boldsymbol{\alpha}}\mathrm{d}E \quad |\boldsymbol{\alpha}| \le k-2\,. \tag{3.21}$$

Remark 3.10. *In simple terms, the Oswald interpolant works by averaging the DoFs from each polygon $E \in \Omega_h$. The DoFs in $\mathcal{P}_c^{E,k}$ belong to just one element, so the Oswald interpolant keeps them unchanged. For a DoF on the interior of an edge, which belongs to two elements, the interpolant takes the average of the values from both elements. Similarly, for a vertex DoF that belongs to more than two elements, the interpolant averages the values from all the connected elements.*

We are ready to prove the following estimate concerning the interpolation error for piecewise polynomial functions. A FEM version of this result can be found in [35, 33].

Proposition 3.11 (Error of the Oswald interpolant)**.** *Under assumption **(A-C)**, for every $E \in \Omega_h$ it holds*

$$\|(I-\pi)\,p\|_{0,E} \lesssim h^{\frac{1}{2}} \sum_{e\in\mathcal{F}_E} \|[\![p]\!]\|_{0,e} \qquad \forall p \in \mathbb{P}_k(\Omega_h)\,,$$

where $\mathcal{F}_E := \{e \in \mathcal{E}_h$ such that $e \cap \partial E \neq \emptyset\}$ is the set of the edges with at least one endpoint which is a vertex of E.

Proof. We introduce the difference

$$d := (I-\pi)\,p\,.$$

We restrict our attention to an element $E \in \Omega_h$, and consider d^E. Since the DoFs in $\mathcal{P}_c^{E,k}$ belong to only one element, we observe that for d^E only the DoFs arising from $\mathcal{V}_c^{E,k}$ and $\mathcal{E}_c^{E,k}$ (i.e. the ones on the mesh skeleton), are involved. Hence, noting that $d^E \in V_h^k(E)$ and $\Pi_{k-2}^{0,E} d^E = 0$ we can apply Lemma 3.9:

$$\|d^E\|_{0,E} \lesssim h^{\frac{1}{2}} \|d^E\|_{0,\partial E} \lesssim h \|d^E\|_{\infty,\partial E}\,. \tag{3.22}$$

Since the basis function associated to $\mathcal{V}_c^k$ and $\mathcal{E}_c^k$ are scaled in a way that their L^∞−norm is equal to 1, we have that

$$h\|d^E\|_{\infty,\partial E} \lesssim h \max_{\nu\in\mathcal{E}_c^{E,k}\cup\mathcal{V}_c^{E,k}} |d^E(\nu)|\,. \tag{3.23}$$

Exploiting the definition of the Oswald interpolant, we observe that if $\nu \in \mathcal{E}_c^{E,k}$ is not on the boundary, we have that

$$\begin{aligned} d^E(\nu) = p^E(\nu) - (\pi p)^E(\nu) &= \frac{1}{|E\cup K|}\big(|E\cup K|\,p^E(\nu) - |E|\,p^E(\nu) - |K|\,p^K(\nu)\big) \\ &= c\,(p^E(\nu) - p^K(\nu)) = c\,([\![p]\!]\cdot \boldsymbol{n}^E)(\nu)\,, \end{aligned}$$

where K is the second element that shares the node ν. Thanks to the mesh assumptions **(A1)**, all the values

$$c = \frac{|E\cup K| - |E|}{|E\cup K|} = \frac{|K|}{|E\cup K|} \approx \frac{1}{2} > 0\,.$$

are uniformly bounded from below and they do not depend on h; hence it holds

$$\max_{\nu\in\mathcal{E}_c^{E,k}} |d^E(\nu)| \lesssim \max_{\nu\in\mathcal{E}_c^{E,k}} |([\![p]\!]\cdot\boldsymbol{n}^E)(\nu)|\,. \tag{3.24}$$

If $\nu \in \mathcal{V}_c^{E,k}$, a similar computation allows to bound $|d^E(\nu)|$ by means of the jumps of p at the nodes on the edges containing ν (this set is denoted by $\mathcal{N}_\nu$ here below):

$$|d^E(\nu)| \lesssim \max_{\nu'\in\mathcal{N}_\nu} |([\![p]\!]\cdot\boldsymbol{n}^E)(\nu')|\,. \tag{3.25}$$

Combining (3.24) and (3.25), we get

$$h \max_{\nu\in\mathcal{E}_c^{E,k}\cup\mathcal{V}_c^{E,k}} |d^E(\nu)| \lesssim h \max_{\nu\in e\,,e\in\mathcal{F}_E} |([\![p]\!]\cdot\boldsymbol{n}^E)(\nu)| \lesssim h\,\|[\![p]\!]\cdot\boldsymbol{n}^E\|_{\infty,\mathcal{E}(E)} \lesssim h\,\|[\![p]\!]\|_{\infty,\mathcal{E}(E)}\,, \tag{3.26}$$

where $\mathcal{E}(E) := \bigcup_{e\in\mathcal{F}_E} e$. Since an inverse estimate gives

$$h\,\|[\![p]\!]\|_{\infty,\mathcal{E}(E)} \lesssim h^{\frac{1}{2}}\,\|[\![p]\!]\|_{0,\mathcal{E}(E)} \lesssim h^{\frac{1}{2}} \sum_{e\in\mathcal{F}_E} \|[\![p]\!]\|_{0,e}\,, \tag{3.27}$$

from (3.22), (3.23), (3.26) and (3.27) we obtain

$$\|(I-\pi)\,p\|_{0,E} = \|d^E\|_{0,E} \lesssim h^{\frac{1}{2}} \sum_{e\in\mathcal{F}_E} \|[\![p]\!]\|_{0,e} \,.$$

□

Remark 3.12. *In this proof, it was sufficient to require the local quasi uniformity of the mesh.*

We can control the norm of the Oswald interpolant using the norm of the original function in the neighborhood of the element E, as demonstrated in the following result.

Lemma 3.13 (Stability of the Oswald interpolant)**.** *Under assumption **(A-C)**, for every $E \in \Omega_h$ it holds*

$$\|\pi\,p\|_{0,E} \lesssim \|p\|_{0,\mathcal{D}(E)} \quad \forall p \in \mathbb{P}_k(\Omega_h)\,,$$

where $\mathcal{D}(E) \coloneqq \bigcup\{K \in \Omega_h$ such that $\bar{E}\cap\bar{K} \neq \emptyset\}$.

Proof. Using triangular inequality, we obtain

$$\|\pi\,p\|_{0,E} \leq \|p\|_{0,E} + \|(I-\pi)\,p\|_{0,E} \,.$$

Thanks to Proposition 3.11, we control the second term with the jumps

$$\|(I-\pi)\,p\|_{0,E} \lesssim h^{\frac{1}{2}} \sum_{e\in\mathcal{F}_E} \|[\![p]\!]\|_{0,e} \,.$$

Thanks to the polynomial trace inequality, we conclude

$$\|(I-\pi)\,p\|_{0,E} \lesssim \|p\|_{0,\mathcal{D}(E)} \,,$$

hence

$$\|\pi\,p\|_{0,E} \leq \|p\|_{0,E} + \|(I-\pi)\,p\|_{0,E} \lesssim \|p\|_{0,\mathcal{D}(E)} \,.$$

□

3.2.2 Stability of the discrete problem

We start the theoretical analysis for the proposed method by introducing the local VEM-CIP norm

$$\begin{aligned} \|v_h\|^2_{\mathrm{cip},E} \coloneqq{}& \epsilon\,\|\nabla v_h\|^2_{0,E} + h\,\|\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h\|^2_{0,E} + \sigma\,\|v_h\|^2_{0,E} \\ &+ \|\xi(\epsilon,\boldsymbol{\beta})v_h\|^2_{0,E} + J^E_h(v_h,v_h)\,, \end{aligned} \tag{3.28}$$

where

$$\xi(\epsilon,\boldsymbol{\beta}) \coloneqq \left(\frac{\epsilon}{\delta h} + \frac{1}{2}|\boldsymbol{\beta}\cdot\boldsymbol{n}|\right)^{\frac{1}{2}}\,, \tag{3.29}$$

with global counterpart

$$\|v_h\|^2_{\mathrm{cip}} \coloneqq \sum_{E\in\Omega_h} \|v_h\|^2_{\mathrm{cip},E} \,. \tag{3.30}$$

Contrary to the previous two chapters, we now include the term

$$h\,\|\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h\|^2_{0,E}$$

in the definition of the local norm. This term arises from a skew-symmetric term in the bilinear form $\mathcal{A}_{\mathrm{cip}}$. Consequently, we cannot expect to prove a coercivity result as was done previously. To demonstrate the stability of the method, instead of proving the coercivity of the bilinear form, we establish that the inf-sup condition holds. The next two lemmas will be instrumental in proving the stability of the method.

Lemma 3.14. *Under assumptions **(A1)**, given $v_h \in V_h^k(\Omega_h)$, it holds*

$$\mathcal{A}_{\mathrm{cip}}(v_h, v_h) \gtrsim \epsilon \, \|\nabla v_h\|_{0,\Omega_h}^2 + J_h(v_h, v_h) + \sigma \, \|v_h\|_{0,\Omega_h}^2 + \sum_{e \in \mathcal{E}_h^\partial} \|\xi(\epsilon, \boldsymbol{\beta}) v_h\|_{0,e}^2 \, . \tag{3.31}$$

Proof. We begin by fixing an element $E \in \Omega_h$. Due to the skew-symmetry property of $b_h^{skew,E}(\cdot,\cdot)$, testing the quadratic form $\mathcal{A}_{\mathrm{cip}}^E(\cdot,\cdot)$ with v_h in both entries yields

$$\begin{aligned} \mathcal{A}_{\mathrm{cip}}^E(v_h, v_h) \gtrsim \epsilon \, \|\nabla v_h\|_{0,E}^2 - 2\epsilon \, \langle \mathbf{\Pi}_{k-1}^{0,E} \nabla v_h \cdot \boldsymbol{n}^E, v_h \rangle_{\Gamma_E} + \|\xi(\epsilon, \boldsymbol{\beta}) v_h\|_{0,\Gamma_E}^2 \\ + J_h^E(v_h, v_h) + \sigma \|v_h\|_{0,E}^2 \, . \end{aligned} \tag{3.32}$$

We now handle the non-symmetric term in (3.32). The procedure is the same that appears in Proposition 2.3. It holds

$$\begin{aligned} 2\epsilon \langle \mathbf{\Pi}_{k-1}^{0,E} \nabla v_h \cdot \boldsymbol{n}^E, v_h \rangle_{\Gamma_E} &\leq 2\epsilon\delta h_E \, \|\mathbf{\Pi}_{k-1}^{0,E} \nabla v_h \cdot \boldsymbol{n}^E\|_{0,\Gamma_E}^2 + \frac{\epsilon}{2\delta h_E} \, \|v_h\|_{0,\Gamma_E}^2 \\ &\leq 2\epsilon\delta C_{\mathrm{tr}} \, \|\mathbf{\Pi}_{k-1}^{0,E} \nabla v_h\|_{0,E}^2 + \frac{\epsilon}{2\delta h_E} \, \|v_h\|_{0,\Gamma_E}^2 \\ &\leq 2\epsilon\delta C_{\mathrm{tr}} \, \|\nabla v_h\|_{0,E}^2 + \frac{\epsilon}{2\delta h_E} \, \|v_h\|_{0,\Gamma_E}^2 \, , \end{aligned}$$

where C_{tr} is the inverse trace inequality constant for polynomials. Therefore, with the choice, e.g., $\delta = \widetilde{\alpha}_*/(4C_{\mathrm{tr}})$, inserting this in (3.32), we obtain

$$\epsilon \|\nabla v_h\|_{0,E}^2 + J_h^E(v_h, v_h) + \sigma \|v_h\|_{0,E}^2 + \|\xi(\epsilon, \beta) v_h\|_{0,\Gamma_E}^2 \lesssim \mathcal{A}_{\mathrm{cip}}^E(v_h, v_h) \, .$$

Summing over all to elements $E \in \Omega_h$, we get the control of the symmetric terms in $\| \cdot \|_{\mathrm{cip}}$:

$$\epsilon \|\nabla v_h\|_{0,\Omega_h}^2 + J_h(v_h, v_h) + \sigma \|v_h\|_{0,\Omega_h}^2 + \sum_{e \in \mathcal{E}_h^\partial} \|\xi(\epsilon, \beta) v_h\|_{0,e}^2 \lesssim \mathcal{A}_{\mathrm{cip}}(v_h, v_h) \, . \tag{3.33}$$

□

Lemma 3.15. *Given $v_h \in V_h^k(\Omega_h)$, let us set*

$$w_h := h \pi(\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h) \, , \tag{3.34}$$

*where $\boldsymbol{\beta}_h$ is the L^2-projection of $\boldsymbol{\beta}$ onto the space of piecewise linear functions $[\mathbb{P}_1(\Omega_h)]^2$. Then, under assumptions **(A-C)**, if $\epsilon < h$ it holds*

$$\mathcal{A}_{\mathrm{cip}}(v_h, w_h) \geq C_1 \, h \|\boldsymbol{\beta} \cdot \nabla \Pi_k^0 v_h\|_{0,\Omega_h}^2 - C_2 \, \mathcal{A}_{\mathrm{cip}}(v_h, v_h) \, . \tag{3.35}$$

Proof. Thanks to Lemma 3.13, we first notice that

$$\|\pi(\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h)\|_{0,E} \lesssim \|\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h\|_{0,\mathcal{D}(E)} \, , \tag{3.36}$$

an estimate which will be frequently used in the sequel.

Recalling (3.34), we locally have

$$\begin{aligned} \mathcal{A}_{\mathrm{cip}}^E(v_h, w_h) &= \epsilon \, a_h^E\big(v_h, h\pi(\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h)\big) + J_h^E\big(v_h, h\pi(\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h)\big) \\ &\quad + \sigma \, c_h^E\big(v_h, h\pi(\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h)\big) + \mathcal{N}_h^E\big(v_h, h\pi(\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h)\big) \\ &\quad + b_h^{\mathrm{skew},E}\big(v_h, h\pi(\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h)\big) \\ &= T_1 + T_2 + T_3 + T_4 + T_5 \, . \end{aligned} \tag{3.37}$$

We consider each of the five terms in this equation.

Estimate for $(\mathbf{T_1})$. Using Cauchy-Schwarz inequality, Lemma 3.8, estimate (3.36) and recalling that $\epsilon < h$, we get

$$\begin{aligned} \epsilon \, a_h^E\big(v_h, h\pi(\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h)\big) &\geq -\epsilon \, a_h^E(v_h, v_h)^{\frac{1}{2}} \, a_h^E\big(h\pi(\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h), h\pi(\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h)\big)^{\frac{1}{2}} \\ &\gtrsim -\epsilon^{\frac{1}{2}} \|\nabla v_h\|_{0,E} \, \epsilon^{\frac{1}{2}} |h\pi(\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h)|_{1,E} \\ &\gtrsim -\epsilon^{\frac{1}{2}} \|\nabla v_h\|_{0,E} \, \epsilon^{\frac{1}{2}} h^{-1} \|h\pi(\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h)\|_{0,E} \\ &\gtrsim -\epsilon^{\frac{1}{2}} \|\nabla v_h\|_{0,E} \, h^{\frac{1}{2}} \|\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h\|_{0,\mathcal{D}(E)} \, . \end{aligned} \tag{3.38}$$

Estimate for $(\mathbf{T_2})$. For the jump operator $J_h^E(\cdot,\cdot)$, we use again Cauchy-Schwarz inequality

$$J_h^E\big(v_h, h\pi(\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h)\big) \geq -J_h^E(v_h,v_h)^{\frac{1}{2}}\, J_h^E\big(h\pi(\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h), h\pi(\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h)\big)^{\frac{1}{2}}\,.$$

Thanks to the trace inequality for polynomials, Lemma 3.8 and estimate (3.36), we obtain ($w_h = h\pi(\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h)$):

$$\begin{aligned}
J_h^E(w_h,w_h) &= \frac{1}{2}\sum_{e\subset\partial E}\gamma_e \int_e h_e^2\, [\![\nabla\Pi_k^0 w_h]\!]^2\,\mathrm{d}s + \gamma_E\, h_E\, \mathcal{S}_J^E\big((I-\Pi_k^{\nabla,E})w_h, (I-\Pi_k^{\nabla,E})w_h\big)\\
&\lesssim h\,\|\nabla\Pi_k^0 w_h\|^2_{0,\mathcal{D}(E)} + h_E\,|h\pi(\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h)|^2_{1,E}\\
&\lesssim h^{-1}\,\|\Pi_k^0 w_h\|^2_{0,\mathcal{D}(E)} + h^{-1}\|h\pi(\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h)\|^2_{0,E}\\
&\lesssim h^{-1}\|h\pi(\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h)\|^2_{0,\mathcal{D}(E)}\\
&\lesssim h\|\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h\|^2_{0,\mathcal{D}(\mathcal{D}(E))}\,,
\end{aligned}\tag{3.39}$$

where $\mathcal{D}(\mathcal{D}(E)) := \cup_{E'\subseteq\mathcal{D}(E)}\mathcal{D}(E')$. Therefore, it holds

$$J_h^E\big(v_h, h\pi(\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h)\big) \gtrsim -J_h^E(v_h,v_h)^{\frac{1}{2}}\, h^{\frac{1}{2}}\, \|\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h\|_{0,\mathcal{D}(\mathcal{D}(E))}\,. \tag{3.40}$$

Estimate for $(\mathbf{T_3})$. Using a similar procedure, we control the bilinear form $c_h(\cdot,\cdot)$ in this way

$$\begin{aligned}
\sigma\, c_h^E\big(v_h, h\pi(\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h)\big) &\gtrsim -\sigma\,\|v_h\|_{0,E}\,\|h\pi(\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h)\|_{0,E}\\
&\gtrsim -\|v_h\|_{0,E}\, h^{\frac{1}{2}}\,\|\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h\|_{0,\mathcal{D}(E)}\,.
\end{aligned}\tag{3.41}$$

where we used $h^{\frac{1}{2}} \lesssim 1$ to simplify later developments.
Estimate for $(\mathbf{T_4})$. For the Nitsche term, we have to control four different terms:

$$\begin{aligned}
T_4 = \mathcal{N}_h^E(v_h,w_h) &= -\epsilon\,\langle \boldsymbol{\Pi}_{k-1}^{0,E}\nabla v_h \cdot \boldsymbol{n}^E, w_h\rangle_{\Gamma_E} - \epsilon\,\langle v_h, \boldsymbol{\Pi}_{k-1}^{0,E}\nabla w_h \cdot \boldsymbol{n}^E\rangle_{\Gamma_E}\\
&\quad + \frac{\epsilon}{\delta h_E}\langle v_h, w_h\rangle_{\Gamma_E} + \frac{1}{2}\langle|\boldsymbol{\beta}\cdot\boldsymbol{n}^E| v_h, w_h\rangle_{\Gamma_E}\\
&=: \eta_{\mathcal{N}_1} + \eta_{\mathcal{N}_2} + \eta_{\mathcal{N}_3} + \eta_{\mathcal{N}_4}\,.
\end{aligned}$$

We consider each of the four terms above. Using Cauchy-Schwarz inequality, trace inequality, $\epsilon < h$ and inverse estimate, the first term is estimated by

$$\begin{aligned}
\eta_{\mathcal{N}_1} &\gtrsim -\epsilon\,\|\boldsymbol{\Pi}_{k-1}^{0,E}\nabla v_h\|_{0,\Gamma_E}\|w_h\|_{0,\Gamma_E}\\
&\gtrsim -\epsilon\, h^{-1}\|\boldsymbol{\Pi}_{k-1}^{0,E}\nabla v_h\|_{0,E}\|w_h\|_{0,E}\\
&\gtrsim -\epsilon^{\frac{1}{2}}\,\|\nabla v_h\|_{0,E}\, h^{\frac{1}{2}}\,\|\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h\|_{0,\mathcal{D}(E)}\,.
\end{aligned}\tag{3.42}$$

For $\eta_{\mathcal{N}_2}$, we use the Cauchy-Schwarz inequality, inverse trace and inverse inequalities, and $\epsilon < h$, to obtain

$$\begin{aligned}
\eta_{\mathcal{N}_2} &= -\epsilon\,\langle v_h, \boldsymbol{\Pi}_{k-1}^{0,E}\nabla w_h \cdot \boldsymbol{n}^E\rangle_{\Gamma_E}\\
&\gtrsim -\epsilon\,\|v_h\|_{0,\Gamma_E}\|\boldsymbol{\Pi}_{k-1}^{0,E}\nabla w_h\|_{0,\Gamma_E}\\
&\gtrsim -\epsilon\, h^{-\frac{1}{2}}\,\|v_h\|_{0,\Gamma_E}\|\boldsymbol{\Pi}_{k-1}^{0,E}\nabla w_h\|_{0,E}\\
&\gtrsim -\epsilon\, h^{-\frac{3}{2}}\|v_h\|_{0,\Gamma_E}\|w_h\|_{0,E}\\
&\gtrsim -\left(\frac{\epsilon}{\delta h}\right)^{\frac{1}{2}}\|v_h\|_{0,\Gamma_E}\, h^{\frac{1}{2}}\,\|\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h\|_{0,\mathcal{D}(E)}\,.
\end{aligned}\tag{3.43}$$

For $\eta_{\mathcal{N}_3}$, we proceed similarly to the previous cases:

$$\begin{aligned}
\eta_{\mathcal{N}_3} &\gtrsim -\frac{\epsilon}{\delta h}\|v_h\|_{0,\Gamma_E}\|w_h\|_{0,\Gamma_E}\\
&\gtrsim -h^{-1}\left(\frac{\epsilon}{\delta h}\right)^{\frac{1}{2}}\|v_h\|_{0,\Gamma_E}\|w_h\|_{0,E}\\
&\gtrsim -\left(\frac{\epsilon}{\delta h}\right)^{\frac{1}{2}}\|v_h\|_{0,\Gamma_E}\, h^{\frac{1}{2}}\|\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h\|_{0,\mathcal{D}(E)}\,.
\end{aligned}\tag{3.44}$$

For the last one, using the same estimates, we get

$$\eta_{\mathcal{N}_4} \gtrsim -\|\xi(\epsilon,\boldsymbol{\beta})v_h\|_{0,\Gamma_E}\, h^{\frac{1}{2}}\, \|\boldsymbol{\beta}_h \cdot \nabla\Pi^0_k v_h\|_{0,\mathcal{D}(E)}\,. \tag{3.45}$$

Hence it holds

$$\mathcal{N}^E_h(v_h, w_h) \gtrsim -\left(\epsilon^{\frac{1}{2}}\|\nabla v_h\|_{0,E} + \|\xi(\epsilon,\boldsymbol{\beta})v_h\|_{0,\Gamma_E}\right) h^{\frac{1}{2}}\, \|\boldsymbol{\beta}_h \cdot \nabla\Pi^0_k v_h\|_{0,\mathcal{D}(E)}\,. \tag{3.46}$$

Estimate for $(\mathbf{T_5})$. It is the most involved term. The skew term $b_h^{\text{skew},E}(v_h, w_h)$ is composed by two parts

$$b_h^{\text{skew},E}(v_h, w_h) = \frac{1}{2}\big(b^E_h(v_h,w_h) - b^E_h(w_h,v_h)\big)\,, \tag{3.47}$$

and we consider each of these two terms separately. The first term is defined as

$$b^E_h(v_h,w_h) = \big(\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h, \Pi^{0,E}_k w_h\big)_{0,E} + \langle(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)(I - \Pi^{0,E}_k)v_h, \Pi^{0,E}_k w_h\rangle_{\partial E}\,. \tag{3.48}$$

We split the first term of (3.48) as

$$\begin{aligned}
\big(\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h,\, \Pi^{0,E}_k w_h\big)_{0,E} &= \big(\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h,\, w_h\big)_{0,E} + \big(\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h,\, (\Pi^{0,E}_k - I)w_h\big)_{0,E}\\
&= \big(\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h,\, h\boldsymbol{\beta}_h\cdot\nabla\Pi^{0,E}_k v_h\big)_{0,E}\\
&\quad + \big(\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h,\, w_h - h\boldsymbol{\beta}_h\cdot\nabla\Pi^{0,E}_k v_h\big)_{0,E}\\
&\quad + \big(\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h,\, (\Pi^{0,E}_k - I)w_h\big)_{0,E}\\
&=: \eta_{\boldsymbol{\beta}_1} + \eta_{\boldsymbol{\beta}_2} + \eta_{\boldsymbol{\beta}_3}\,.
\end{aligned} \tag{3.49}$$

We estimate each of these three quantities. For the first term we have

$$\begin{aligned}
\eta_{\boldsymbol{\beta}_1} &= (\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h, h\boldsymbol{\beta}_h\cdot\nabla\Pi^0_k v_h)_{0,E}\\
&= h\,\|\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h\|^2_{0,E} + (\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h, h(\boldsymbol{\beta}_h - \boldsymbol{\beta})\cdot\nabla\Pi^{0,E}_k v_h)_{0,E}\\
&\geq h\,\|\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h\|^2_{0,E} - C\,h^{\frac{1}{2}}\,\|\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h\|_{0,E}\, h^{\frac{1}{2}}\,|\boldsymbol{\beta}|_{[W^1_\infty(E)]^2} h\|\nabla\Pi^{0,E}_k v_h\|_{0,E}\\
&\geq h\,\|\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h\|^2_{0,E} - C\,h^{\frac{1}{2}}\,\|\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h\|_{0,E}\, h^{\frac{1}{2}}\,|\boldsymbol{\beta}|_{[W^1_\infty(E)]^2}\|v_h\|_{0,E}\,.
\end{aligned} \tag{3.50}$$

Recalling (3.34) and by Young's inequality we get:

$$\begin{aligned}
\eta_{\boldsymbol{\beta}_2} &= h\big(\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h, (\pi - I)(\boldsymbol{\beta}_h\cdot\nabla\Pi^{0,E}_k v_h)\big)_{0,E}\\
&\geq -\frac{h}{2}\|\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h\|^2_{0,E} - \frac{h}{2}\|(\pi - I)(\boldsymbol{\beta}_h\cdot\nabla\Pi^0_k v_h)\|^2_{0,E}\,.
\end{aligned} \tag{3.51}$$

Since $\boldsymbol{\beta}_h$ is piecewise linear, for the second term we can use Proposition 3.11 and obtain

$$\begin{aligned}
h\|(\pi - I)(\boldsymbol{\beta}_h\cdot\nabla\Pi^0_k v_h)\|^2_{0,E} &\lesssim h^2 \sum_{e\subset\mathcal{F}_E} \|[\![\boldsymbol{\beta}_h\cdot\nabla\Pi^0_k v_h]\!]\|^2_{0,e}\\
&\lesssim h^2 \sum_{e\subset\mathcal{F}_E} \|[\![(\boldsymbol{\beta}_h - \boldsymbol{\beta})\cdot\nabla\Pi^0_k v_h]\!]\|^2_{0,e} + h^2 \sum_{e\subset\mathcal{F}_E} \|[\![\boldsymbol{\beta}\cdot\nabla\Pi^0_k v_h]\!]\|^2_{0,e}\\
&\lesssim h^2 \sum_{e\subset\mathcal{F}_E} \|[\![(\boldsymbol{\beta}_h - \boldsymbol{\beta})\cdot\nabla\Pi^0_k v_h]\!]\|^2_{0,e} + h^2 \sum_{e\subset\mathcal{F}_E} \gamma_e^2\|[\![\nabla\Pi^0_k v_h]\!]\|^2_{0,e}\\
&\lesssim h^2 \sum_{e\subset\mathcal{F}_E} \|[\![(\boldsymbol{\beta}_h - \boldsymbol{\beta})\cdot\nabla\Pi^0_k v_h]\!]\|^2_{0,e} + J_h^{\mathcal{D}(E)}(v_h, v_h)\,,
\end{aligned} \tag{3.52}$$

where in the last inequality we used (3.9) and (3.10), together with $\gamma_e^2 \leq \gamma_e$ (since $\gamma_e \leq 1$, see (3.3)). On each e, we control the first term in the previous inequality as

$$\begin{aligned}
h^2\|[\![(\boldsymbol{\beta}_h - \boldsymbol{\beta})\cdot\nabla\Pi^0_k v_h]\!]\|^2_{0,e} &\lesssim h^4\,|\boldsymbol{\beta}|^2_{[W^1_\infty(E\cup K)]^2}\, h^{-1}\|\nabla\Pi^0_k v_h\|^2_{0,E\cup K}\\
&\lesssim h\,|\boldsymbol{\beta}|^2_{[W^1_\infty(E\cup K)]^2}\,\|\Pi^0_k v_h\|^2_{0,E\cup K}\\
&\lesssim h\,|\boldsymbol{\beta}|^2_{[W^1_\infty(E\cup K)]^2}\,\|v_h\|^2_{0,E\cup K}\,,
\end{aligned} \tag{3.53}$$

where E and K are the two elements sharing the edge e. Combining (3.51) with (3.52) and (3.53), we obtain

$$\eta_{\boldsymbol{\beta}_2} \geq -\frac{h}{2}\|\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}v_h\|_{0,E}^2 - C\Big(h\,|\boldsymbol{\beta}|^2_{[W^1_\infty(\mathcal{D}(E))]^2}\|v_h\|^2_{0,\mathcal{D}(E)} + J_h^{\mathcal{D}(E)}(v_h,v_h)\Big)\,. \tag{3.54}$$

It remains to control $\eta_{\boldsymbol{\beta}_3}$. Since $\boldsymbol{\beta}_h \in \mathbb{P}_1(E)$, it holds $\big(\boldsymbol{\beta}_h\cdot\nabla\Pi_k^{0,E}v_h, (\Pi_k^{0,E}-I)w_h\big)_{0,E} = 0$.

Hence we have

$$\begin{aligned}
\eta_{\boldsymbol{\beta}_3} &= \big((\boldsymbol{\beta}-\boldsymbol{\beta}_h)\cdot\nabla\Pi_k^{0,E}v_h, (\Pi_k^{0,E}-I)w_h\big)_{0,E} \\
&\gtrsim -\|(\boldsymbol{\beta}-\boldsymbol{\beta}_h)\cdot\nabla\Pi_k^{0,E}v_h\|_{0,E}\,\|h\pi(\boldsymbol{\beta}_h\cdot\nabla\Pi_k^0 v_h)\|_{0,E} \\
&\gtrsim -|\boldsymbol{\beta}|_{[W^1_\infty(E)]^2} h\|\nabla\Pi_k^{0,E}v_h\|_{0,E}\, h\|\boldsymbol{\beta}_h\cdot\nabla\Pi_k^0 v_h\|_{0,\mathcal{D}(E)} \\
&\gtrsim -|\boldsymbol{\beta}|_{[W^1_\infty(E)]^2}\|v_h\|^2_{0,\mathcal{D}(E)}\,.
\end{aligned} \tag{3.55}$$

Collecting (3.50), (3.54) and (3.55), from (3.49) we get

$$\begin{aligned}
\big(\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}v_h,\, \Pi_k^{0,E}w_h\big)_{0,E} \geq \frac{h}{2}\,\|\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}u_h\|^2_{0,E} &- C\Big(J_h^{\mathcal{D}(E)}(v_h,v_h) \\
&+ h^{\frac{1}{2}}\|\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}v_h\|_{0,E}\, h^{\frac{1}{2}}|\boldsymbol{\beta}|_{[W^1_\infty(\mathcal{D}(E))]^2}\|v_h\|_{0,E} \\
&+ h|\boldsymbol{\beta}|^2_{[W^1_\infty(\mathcal{D}(E))]^2}\|v_h\|^2_{0,\mathcal{D}(E)} + |\boldsymbol{\beta}|_{[W^1_\infty(\mathcal{D}(E))]^2}\|v_h\|^2_{0,\mathcal{D}(E)}\Big)\,.
\end{aligned} \tag{3.56}$$

Returning to (3.48), we have to control the boundary term. We first notice that, due to Agmon inequality and Poincaré inequality, it holds

$$\|(I-\Pi_k^{0,E})v_h\|_{0,\partial E} \lesssim h^{\frac{1}{2}}|(I-\Pi_k^{0,E})v_h|_{1,E}\,.$$

Together with an inverse inequality for the polynomial $\Pi_k^{0,E}w_h$, the definition of w_h (cf. (3.34)), Lemma 3.13 and recalling (3.10), we thus get:

$$\begin{aligned}
\big\langle(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)(I-\Pi_k^{0,E})v_h, \Pi_k^{0,E}w_h\big\rangle_{\partial E} &\gtrsim -\gamma_E\|(I-\Pi_k^{0,E})v_h\|_{0,\partial E}\,\|\Pi_k^{0,E}w_h\|_{0,\partial E} \\
&\gtrsim -h^{\frac{1}{2}}\gamma_E|(I-\Pi_k^{0,E})v_h|_{1,E}\, h^{-\frac{1}{2}}\|\Pi_k^{0,E}w_h\|_{0,E} \\
&\gtrsim -\gamma_E|(I-\Pi_k^{0,E})v_h|_{1,E}\,\|w_h\|_{0,E} \\
&\gtrsim -\gamma_E^{\frac{1}{2}}|(I-\Pi_k^{0,E})v_h|_{1,E}\,\|w_h\|_{0,E} \\
&\gtrsim -\,J_h^E(v_h,v_h)^{\frac{1}{2}}\, h^{\frac{1}{2}}\|\boldsymbol{\beta}_h\cdot\nabla\Pi_k^0 v_h\|_{0,\mathcal{D}(E)}\,.
\end{aligned} \tag{3.57}$$

Above, we have used $\gamma_E \leq \gamma_E^{\frac{1}{2}}$ (since $\gamma_E \leq 1$, see (3.3) and (3.10)), together with the estimate, see (3.9):

$$\gamma_E\, h\,|(I-\Pi_k^{0,E})v_h|^2_{1,E} \lesssim J_h^E(v_h,v_h)\,.$$

From (3.48), (3.56) and (3.57) we get

$$\begin{aligned}
b_h^E(v_h,w_h) \geq \frac{h}{2}\,\|\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}u_h\|^2_{0,E} &- C\Big(J_h^{\mathcal{D}(E)}(v_h,v_h) \\
&+ h^{\frac{1}{2}}\|\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}v_h\|_{0,E}\, h^{\frac{1}{2}}|\boldsymbol{\beta}|_{[W^1_\infty(E)]^2}\|v_h\|_{0,E} \\
&+ h|\boldsymbol{\beta}|^2_{[W^1_\infty(\mathcal{D}(E))]^2}\|v_h\|^2_{0,\mathcal{D}(E)} + |\boldsymbol{\beta}|_{[W^1_\infty(\mathcal{D}(E))]^2}\|v_h\|^2_{0,\mathcal{D}(E)} \\
&+\ J_h^E(v_h,v_h)^{\frac{1}{2}}\, h^{\frac{1}{2}}\|\boldsymbol{\beta}_h\cdot\nabla\Pi_k^0 v_h\|_{0,\mathcal{D}(E)}\Big)\,.
\end{aligned} \tag{3.58}$$

Finally, we need to control $-b_h^E(w_h,v_h)$, see (3.47). Integrating by parts, we obtain

$$\begin{aligned}
-b_h^E(w_h,v_h) &= -\big(\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}w_h, \Pi_k^{0,E}v_h\big)_{0,E} - \langle(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)(I-\Pi_k^{0,E})w_h, \Pi_k^{0,E}v_h\rangle_{\partial E} \\
&= \big(\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}v_h, \Pi_k^{0,E}w_h\big)_{0,E} - \langle(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)w_h, \Pi_k^{0,E}v_h\rangle_{\partial E} \\
&= \big(\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}v_h, \Pi_k^{0,E}w_h\big)_{0,E} - \langle(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)w_h, (\Pi_k^{0,E}-I)v_h\rangle_{\partial E} \\
&\quad - \langle(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)w_h, v_h\rangle_{\partial E}\,.
\end{aligned} \tag{3.59}$$

The first two terms are similar to the case $b_h(v_h, w_h)$. The last one vanishes on the interior edges when we sum over all $E \in \Omega_h$. Hence, we need to consider the elements E sharing with $\partial\Omega$ at least an edge. Using Cauchy-Schwarz inequality, trace inequality, inverse estimates and the continuity of π, we obtain on these boundary edges

$$\begin{aligned} -\langle(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)w_h, v_h\rangle_{\partial E} &\geq -\|\xi(\epsilon,\boldsymbol{\beta})v_h\|_{0,\Gamma_E}\,\|\xi(\epsilon,\boldsymbol{\beta})w_h\|_{0,\Gamma_E} \\ &\gtrsim -\|\xi(\epsilon,\boldsymbol{\beta})v_h\|_{0,\Gamma_E}\,h^{-\frac{1}{2}}\|w_h\|_{0,E} \\ &\gtrsim -\|\xi(\epsilon,\boldsymbol{\beta})v_h\|_{0,\Gamma_E}\,h^{\frac{1}{2}}\|\boldsymbol{\beta}_h\cdot\nabla\Pi_k^0 v_h\|_{0,\mathcal{D}(E)}\,. \end{aligned} \tag{3.60}$$

Therefore, from (3.47), (3.58), (3.59) and (3.60) we get

$$\begin{aligned} b_h^{\text{skew},E}(v_h, w_h) \geq \frac{h}{2}\|\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}u_h\|_{0,E}^2 &- C\Big(J_h^{\mathcal{D}(E)}(v_h,v_h) \\ &+ h^{\frac{1}{2}}\|\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}v_h\|_{0,E}\,h^{\frac{1}{2}}|\boldsymbol{\beta}|_{[W_\infty^1(E)]^2}\|v_h\|_{0,E} \\ &+ h|\boldsymbol{\beta}|^2_{[W_\infty^1(E)]^2}\|v_h\|^2_{0,\mathcal{D}(E)} + |\boldsymbol{\beta}|_{[W_\infty^1(E)]^2}\|v_h\|^2_{0,\mathcal{D}(E)} \\ &+ \big(J_h^E(v_h,v_h)^{\frac{1}{2}} + \|\xi(\epsilon,\boldsymbol{\beta})v_h\|_{0,\Gamma_E}\big)\,h^{\frac{1}{2}}\|\boldsymbol{\beta}_h\cdot\nabla\Pi_k^0 v_h\|_{0,\mathcal{D}(E)}\Big)\,. \end{aligned} \tag{3.61}$$

We now consider the five local estimates (3.38), (3.40), (3.41), (3.46) and (3.61). From (3.37), summing over all the elements $E \in \Omega_h$, we obtain

$$\begin{aligned} \mathcal{A}_{\text{cip}}(v_h, w_h) \geq \frac{h}{2}\|\boldsymbol{\beta}\cdot\nabla\Pi_k^0 v_h\|^2_{0,\Omega_h} &- C\Big(\sum_{E\in\Omega_h}\big(\epsilon^{\frac{1}{2}}\|\nabla v_h\|_{0,E} \\ &+ J_h^E(v_h,v_h)^{\frac{1}{2}} + \|v_h\|_{0,E} + \|\xi(\epsilon,\beta)v_h\|_{0,\Gamma_E}\big)h^{\frac{1}{2}}\|\boldsymbol{\beta}_h\cdot\nabla\Pi_k^{0,E}v_h\|_{0,E} \\ &+ J_h(v_h,v_h) + \sum_{E\in\Omega_h}\Big(h|\boldsymbol{\beta}|^2_{[W_\infty^1(E)]^2} + |\boldsymbol{\beta}|_{[W_\infty^1(E)]^2}\Big)\|v_h\|^2_{0,E} \\ &+ \sum_{E\in\Omega_h} h^{\frac{1}{2}}\|\boldsymbol{\beta}_h\cdot\nabla\Pi_k^{0,E}v_h\|_{0,E}\,h^{\frac{1}{2}}|\boldsymbol{\beta}|_{[W_\infty^1(E)]^2}\|v_h\|_{0,E}\Big)\,. \end{aligned} \tag{3.62}$$

Above, we have also used the property that, due to assumption **(A-C)**, summing over the elements each polygon is counted only a uniformly bounded number of times, even when the terms involve norms on $\mathcal{D}(E)$ or $\mathcal{D}(\mathcal{D}(E))$.

We now notice that the triangular inequality, standard approximation results and an inverse estimate give

$$h^{\frac{1}{2}}\|\boldsymbol{\beta}_h\cdot\nabla\Pi_k^{0,E}v_h\|_{0,E} \lesssim h^{\frac{1}{2}}\Big(\|\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}v_h\|_{0,E} + |\boldsymbol{\beta}|_{[W_\infty^1(E)]^2}\|v_h\|_{0,E}\Big)\,. \tag{3.63}$$

Hence, from (3.62), using also Young's inequality (with suitable constants) for the first and the last summations in the right-hand side, we get

$$\begin{aligned} \mathcal{A}_{\text{cip}}(v_h, w_h) \geq C_1\,h\|\boldsymbol{\beta}\cdot\nabla\Pi_k^0 v_h\|^2_{0,\Omega_h} - C_2\Big(\epsilon\,\|\nabla v_h\|^2_{0,\Omega_h} + J_h(v_h,v_h) \\ + \|v_h\|^2_{0,\Omega_h} + \sum_{e\in\mathcal{E}_h^\partial}\|\xi(\epsilon,\beta)v_h\|^2_{0,e}\Big)\,. \end{aligned}$$

From Lemma 3.14, we now obtain

$$\mathcal{A}_{\text{cip}}(v_h, w_h) \geq C_1\,h\|\boldsymbol{\beta}\cdot\nabla\Pi_k^0 v_h\|^2_{0,\Omega_h} - C_2\,\mathcal{A}_{\text{cip}}(v_h, v_h)\,.$$

□

Remark 3.16. *Equation* (3.51) *motivates the imposition of boundary conditions in a weak sense. If we were to impose the boundary conditions strongly, by requiring that the functions $V_h^k(\Omega_h)$ be zero on the boundary, then the function $\pi\boldsymbol{\beta}_h\cdot\nabla\Pi_k^0 v_h$ would have a zero trace. In contrast, the function $\boldsymbol{\beta}_h\cdot\nabla\Pi_k^0 v_h$ is not necessarily zero on the boundary. Due to this discrepancy between the two functions at the boundary, we cannot expect the estimate in Proposition 3.11, which involves only internal edges, to hold.*

Remark 3.17. *Equation* (3.52) *motivates why we need the quasi uniformity of the mesh Ω_h. If we define* (3.34) *as*

$$w_h = \pi\big(\tilde{h}\,\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h\big)\,,$$

where tildeh is such that $\tilde{h}|_E = h_E$, we can not take $\tilde{h}$ outside the brackets by linearity since in general $h_E \neq h_K$. This implies that when we use Proposition 3.11, the diameter h_E of each element appears inside the jump. Since the diameter $\tilde{h}$ is inside the jump, we are not able to recover the optimal rate of converge.

With Lemmas 3.14 and 3.15 at our disposal, the inf-sup condition easily follows.

Proposition 3.18. *Under assumptions **(A-C)**, it holds:*

$$\|v_h\|_{\text{cip}} \lesssim \sup_{z_h \in V_h^k(\Omega_h)} \frac{\mathcal{A}_{\text{cip}}(v_h, z_h)}{\|z_h\|_{\text{cip}}} \qquad \forall v_h \in V_h^k(\Omega_h).$$

Proof. We split the proof into two cases.
First case. We first consider $\varepsilon < h$. Given $v_h \in V_h^k(\Omega_h)$, we take $z_h = w_h + \kappa v_h$, where w_h is defined as in Lemma 3.15. From Lemmas 3.14 and 3.15, for κ sufficiently large we have

$$\mathcal{A}_{\text{cip}}(v_h, z_h) = \mathcal{A}_{\text{cip}}(v_h, w_h + \kappa v_h) \gtrsim \|v_h\|^2_{\text{cip}}\,.$$

In order to conclude the proof of the inf-sup condition, we have to prove the estimate

$$\|w_h\|_{\text{cip}} \lesssim \|v_h\|_{\text{cip}}\,,$$

which obviously implies $\|z_h\|_{\text{cip}} \lesssim \|v_h\|_{\text{cip}}$. Recalling the norm definition (3.28)-(3.30) and that $w_h := h\pi(\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h)$, the above continuity estimate follows from Lemma 3.13, estimate (3.39) and observing that

$$h\|\boldsymbol{\beta} \cdot \nabla \Pi_k^{0,E} w_h\|^2_{0,E} \lesssim h^{-1}\|\Pi_k^{0,E} w_h\|^2_{0,E} \lesssim h\|\boldsymbol{\beta}_h \cdot \nabla \Pi_k^{0,E} v_h\|^2_{0,\mathcal{D}(E)}\,, \tag{3.64}$$

and

$$\begin{aligned} \|w_h\|^2_{0,\Gamma_E} &= \|h\pi(\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h)\|^2_{0,\Gamma_E} \lesssim h\,\|\pi(\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h)\|^2_{0,E} + h^3\,|\pi(\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h)|^2_{1,E} \\ &\lesssim h\,\|\pi(\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h)\|^2_{0,E} \lesssim h\,\|\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h\|^2_{0,\mathcal{D}(E)}\,. \end{aligned} \tag{3.65}$$

The above bounds (3.64) and (3.65) are to be combined with (3.63).
Second case. We now consider the case $\varepsilon \geq h$. In such case the proof simply follows from Lemma 3.14 and the observation that

$$h\|\boldsymbol{\beta} \cdot \nabla \Pi_k^{0,E} v_h\|^2_{0,E} \lesssim \varepsilon\|\nabla \Pi_k^{0,E} v_h\|^2_{0,E} \lesssim \varepsilon\|\nabla v_h\|^2_{0,E}\,,$$

which allows to control also convection with $\mathcal{A}_{\text{cip}}(v_h, v_h)$.

$$h\|\boldsymbol{\beta} \cdot \nabla \Pi_k^{0,E} v_h\|^2_{0,E} \lesssim \varepsilon\|\nabla \Pi_k^{0,E} v_h\|^2_{0,E} \lesssim \varepsilon\|\nabla v_h\|^2_{0,E}\,,$$

which allows to control also convection with $\mathcal{A}_{\text{cip}}(v_h, v_h)$. □

3.2.3 Error estimates

In this section we use the assumption that the solution of the continuous problem u, the right-hand side f, and the advective field $\boldsymbol{\beta}$ in (3.2) satisfy

$$u \in H^2(\Omega) \cap H^1_0(\Omega) \cap H^{k+1}(\Omega_h)\,, \quad f \in H^{k+\frac{1}{2}}(\Omega_h)\,, \quad \boldsymbol{\beta} \in [W^{k+1}_\infty(\Omega_h)]^2\,.$$

We begin our error analysis, which follows the steps of [14], with the following result.

Proposition 3.19. *Let u and u_h be the solutions of problem* (3.15) *and problem* (3.14), *respectively. Furthermore, let us define*

$$e_{\mathcal{I}} := u - u_{\mathcal{I}},$$

*is the interpolant function of u defined in Lemma 3.1. Then under assumption **(A-C)**, it holds that*

$$\|u - u_h\|_{\text{cip}} \lesssim \|e_{\mathcal{I}}\|_{\text{cip}} + \eta_{\mathcal{F}} + \eta_a + \eta_b + \eta_c + \eta_{\mathcal{N}} + \eta_J\,, \tag{3.66}$$

where we have defined

$$\begin{aligned} \eta_{\mathcal{F}} &:= \|\tilde{\mathcal{F}} - \mathcal{F}_h\|_{\text{cip}^*}\,, & \eta_a &:= \epsilon\,\|a(u,\cdot) - a_h(u_{\mathcal{I}},\cdot)\|_{\text{cip}^*}\,, \\ \eta_b &:= \|b(u,\cdot) - b_h(u_{\mathcal{I}},\cdot)\|_{\text{cip}^*}\,, & \eta_c &:= \sigma\|c(u,\cdot) - c_h(u_{\mathcal{I}},\cdot)\|_{\text{cip}^*}\,, \\ \eta_{\mathcal{N}} &:= \|\tilde{\mathcal{N}}_h(u,\cdot) - \mathcal{N}_h(u_{\mathcal{I}},\cdot)\|_{\text{cip}^*}\,, & \eta_J &:= \|\tilde{J}_h(u,\cdot) - J_h(u_{\mathcal{I}},\cdot)\|_{\text{cip}^*}\,, \end{aligned}$$

where $\|\cdot\|_{\text{cip}^*}$ denotes the dual norm of the norm $\|\cdot\|_{\text{cip}}$.

Proof. Setting $e_h := u_h - u_{\mathcal{I}}$, thanks to the triangular inequality, we have

$$\|u - u_h\|_{\text{cip}} \le \|e_{\mathcal{I}}\|_{\text{cip}} + \|e_h\|_{\text{cip}}\,.$$

Therefore, we only need to bound the second term on the right-hand side. Using the inf-sup condition, (3.14) and (3.15), we have that

$$\begin{aligned} \|e_h\|_{\text{cip}} &= \sup_{v_h \in V_h^k(\Omega_h)} \frac{\mathcal{A}_{\text{cip}}(u_h - u_{\mathcal{I}}, v_h)}{\|v_h\|_{\text{cip}}} = \sup_{v_h \in V_h^k(\Omega_h)} \frac{\mathcal{F}_h(v_h) - \mathcal{A}_{\text{cip}}(u_{\mathcal{I}}, v_h)}{\|v_h\|_{\text{cip}}} \\ &= \sup_{v_h \in V_h^k(\Omega_h)} \frac{\mathcal{F}_h(v_h) - \tilde{\mathcal{F}}(v_h) + \widetilde{\mathcal{A}}_{\text{cip}}(u, v_h) - \mathcal{A}_{\text{cip}}(u_{\mathcal{I}}, v_h)}{\|v_h\|_{\text{cip}}}\,. \end{aligned} \tag{3.67}$$

Estimate (3.66) now follows from considering the definitions of $\mathcal{A}_{\text{cip}}(\cdot,\cdot)$ and $\widetilde{\mathcal{A}}_{\text{cip}}(\cdot,\cdot)$ given in (3.12) and (3.16)-(3.17), respectively. □

Lemma 3.20 (Estimate of $\|e_{\mathcal{I}}\|_{\text{cip}}$)**.** *Under assumptions **(A-C)**, the term $\|e_{\mathcal{I}}\|^2_{\text{cip}}$ can be bounded as follows*

$$\|e_{\mathcal{I}}\|_{\text{cip}} \lesssim \epsilon^{\frac{1}{2}}\,h^k \left(\sum_{E \in \Omega_h} |u|^2_{k+1,E}\right)^{\frac{1}{2}} + h^{k+\frac{1}{2}} \left(\sum_{E \in \Omega_h} |u|^2_{k+1,E}\right)^{\frac{1}{2}}\,.$$

Proof. We start by fixing an element $E \in \Omega_h$. By definition of $\|\cdot\|_{\text{cip},E}$, we have that

$$\|e_{\mathcal{I}}\|^2_{\text{cip},E} = \epsilon\,\|\nabla e_{\mathcal{I}}\|^2_{0,E} + h\,\|\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E} e_{\mathcal{I}}\|^2_{0,E} + \sigma\,\|e_{\mathcal{I}}\|^2_{0,E} + \|\xi(\epsilon,\boldsymbol{\beta}) e_{\mathcal{I}}\|^2_{\Gamma_E} + J_h^E(e_{\mathcal{I}}, e_{\mathcal{I}})\,.$$

Using Lemma 3.1, we have that

$$\epsilon\,\|\nabla e_{\mathcal{I}}\|^2_{0,E} + h\,\|\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E} e_{\mathcal{I}}\|^2_{0,E} \lesssim (\epsilon + h)\,\|\nabla e_{\mathcal{I}}\|^2_{0,E} \lesssim (\epsilon + h)\,h^{2k}\,|u|^2_{k+1,E}\,,$$

and

$$\|e_{\mathcal{I}}\|^2_{0,E} \lesssim h^{2k+2}\,|u|^2_{k+1,E}\,.$$

For the boundary term we have that

$$\|\xi(\epsilon,\boldsymbol{\beta}) e_{\mathcal{I}}\|^2_{0,\Gamma_E} = \frac{\epsilon}{\delta h}\langle e_{\mathcal{I}}, e_{\mathcal{I}}\rangle_{\Gamma_E} + \langle|\boldsymbol{\beta}\cdot\boldsymbol{n}| e_{\mathcal{I}}, e_{\mathcal{I}}\rangle_{\Gamma_E}\,.$$

Using trace inequality and interpolation estimate, we obtain

$$\frac{\epsilon}{\delta h}\langle e_{\mathcal{I}}, e_{\mathcal{I}}\rangle_{\Gamma_E} \lesssim \frac{\epsilon}{\delta h^2}\|e_{\mathcal{I}}\|^2_{0,E} + |e_{\mathcal{I}}|^2_{1,E} \lesssim \epsilon\,h^{2k}\,|u|^2_{k+1,E}\,,$$

and

$$\langle|\boldsymbol{\beta}\cdot\boldsymbol{n}^E| e_{\mathcal{I}}, e_{\mathcal{I}}\rangle_{\Gamma_E} \lesssim h^{-1}\,\|e_{\mathcal{I}}\|^2_{0,E} \lesssim h^{2k+1}\,|u|^2_{k+1,E}\,.$$

It remains to control the jump operator. We have

$$\begin{aligned}
J_h^E(e_{\mathcal{I}}, e_{\mathcal{I}}) &= \frac{1}{2} \sum_{e \subset \partial E} \gamma_e \int_e h_e^2 \, [\![\nabla \Pi_k^0 e_{\mathcal{I}}]\!] \cdot [\![\nabla \Pi_k^0 e_{\mathcal{I}}]\!] \, \mathrm{d}s + \gamma_E \, h_E \, \mathcal{S}^E\big((I - \Pi_k^{\nabla,E}) e_{\mathcal{I}}, (I - \Pi_k^{\nabla,E}) e_{\mathcal{I}}\big) \\
&\lesssim h^2 \, (h^{-1} \|\nabla \Pi_k^0 e_{\mathcal{I}}\|^2_{0,\mathcal{D}(E)} + h \, |\nabla \Pi_k^0 e_{\mathcal{I}}|^2_{1,\mathcal{D}(E)}) + h \, |(I - \Pi_k^{\nabla,E}) e_{\mathcal{I}}|^2_{1,E} \\
&\lesssim h^2 \, (h^{-1} \|\nabla \Pi_k^0 e_{\mathcal{I}}\|^2_{0,\mathcal{D}(E)}) + h \, |(I - \Pi_k^{\nabla,E}) e_{\mathcal{I}}|^2_{1,E} \\
&\lesssim h \, |e_{\mathcal{I}}|^2_{1,\mathcal{D}(E)} \lesssim h^{2k+1} \, |u|^2_{k+1,\mathcal{D}(E)} \, .
\end{aligned}$$

□

Lemma 3.21 (Estimate of $\eta_{\mathcal{F}}$)**.** *Under the assumptions* ***(A-C)****, the term $\eta_{\mathcal{F}}$ can be bounded as follows*

$$\eta_{\mathcal{F}} \lesssim h^{k+1} \left(\sum_{E \in \Omega_h} |f|^2_{k+1,E} \right)^{\frac{1}{2}} .$$

Proof. Using the orthogonality of $\Pi_k^{0,E}$, Cauchy-Schwarz inequality, Poincaré inequality and Lemma 1.4, we obtain

$$\begin{aligned}
\eta_{\mathcal{F}}^E &= \tilde{\mathcal{F}}^E(v_h) - \mathcal{F}_h^E(v_h) \\
&= \big(f, v_h - \Pi_k^{0,E} v_h\big)_{0,E} = \big((I - \Pi_k^{0,E}) f, (I - \Pi_k^{0,E}) v_h\big)_{0,E} \\
&\leq \|(I - \Pi_k^{0,E}) f\|_{0,E} \, \|(I - \Pi_k^{0,E}) v_h\|_{0,E} \\
&\lesssim \|(I - \Pi_k^{0,E}) f\|_{0,E} \, \|v_h\|_{0,E} \\
&\lesssim h^{k+1} \, |f|_{k+1,E} \, \|v_h\|_{\mathrm{cip},E} \, .
\end{aligned}$$

□

Lemma 3.22 (Estimate of η_a)**.** *Under the assumptions* ***(A-C)****, the term η_a can be bounded as follows*

$$\eta_a \lesssim \epsilon^{\frac{1}{2}} \, h^k \left(\sum_{E \in \Omega_h} |u|^2_{k+1,\mathcal{D}(E)} \right)^{\frac{1}{2}} .$$

Proof. We fix an element $E \in \Omega_h$. Adding and subtracting $\Pi_k^{\nabla,E} u$, using Cauchy-Schwarz inequality, we obtain

$$\begin{aligned}
\epsilon \, a^E(u, v_h) - \epsilon \, a_h^E(u_{\mathcal{I}}, v_h) &= \epsilon \, a^E(u - \Pi_k^{\nabla,E} u, v_h) + \epsilon \, a_h^E(\Pi_k^{\nabla,E} u - u_{\mathcal{I}}, v_h) \\
&\leq \epsilon \, (\|\nabla e_\pi\|_{0,E} + (1 + \alpha^*) \|\nabla(\Pi_k^{\nabla,E} u - u_{\mathcal{I}})\|_{0,E}) \, \|\nabla v_h\|_{0,E} \\
&\lesssim \epsilon \, (\|\nabla e_\pi\|_{0,E} + \|\nabla e_{\mathcal{I}}\|_{0,E}) \, \|\nabla v_h\|_{0,E} \\
&\lesssim \epsilon^{\frac{1}{2}} \, h^k \, |u|_{k+1,E} \, \|v_h\|_{\mathrm{cip},E} \, ,
\end{aligned}$$

where $e_\pi := u - \Pi_k^{\nabla,E} u$. □

Lemma 3.23 (Estimate of η_b)**.** *Under the assumptions* ***(A-C)****, the term η_b can be bounded as follows*

$$\eta_b \lesssim h^{k+\frac{1}{2}} \left(\sum_{E \in \Omega_h} \|u\|^2_{k+1,E} \right)^{\frac{1}{2}} + h^{k+1} \left(\sum_{E \in \Omega_h} \|\boldsymbol{\beta}\|^2_{[W^k_\infty(E)]^2} \|u\|^2_{k+1,E} \right)^{\frac{1}{2}} + \int_{\partial E} (\boldsymbol{\beta} \cdot \boldsymbol{n}^E) \, e_{\mathcal{I}} \, v_h \, \mathrm{d}s \, .$$

Proof. Recalling the definition, we need to estimate on each element $E \in \Omega_h$

$$\eta_{b,A}^E := \big(\boldsymbol{\beta} \cdot \nabla u, v_h\big)_{0,E} - \big(\boldsymbol{\beta} \cdot \nabla \Pi_k^{0,E} u_{\mathcal{I}}, \Pi_k^{0,E} v_h\big)_{0,E} - \int_{\partial E} (\boldsymbol{\beta} \cdot \boldsymbol{n}^E) \, (I - \Pi_k^{0,E}) \, u_{\mathcal{I}} \, \Pi_k^{0,E} v_h \, \mathrm{d}s \, ,$$

$$\eta_{b,B}^E := \big(\Pi_k^{0,E} u_{\mathcal{I}}, \boldsymbol{\beta} \cdot \nabla \Pi_k^{0,E} v_h\big)_{0,E} - \big(u, \boldsymbol{\beta} \cdot \nabla v_h\big)_{0,E} + \int_{\partial E} (\boldsymbol{\beta} \cdot \boldsymbol{n}^E) \, (I - \Pi_k^{0,E}) \, v_h \, \Pi_k^{0,E} u_{\mathcal{I}} \, \mathrm{d}s \, .$$

By integration by parts, we have

$$\begin{aligned}\eta_{b,A}^E &= \big(\boldsymbol{\beta}\cdot\nabla u, (I-\Pi_k^{0,E})\, v_h\big)_{0,E} + \big(\boldsymbol{\beta}\cdot\nabla(u-\Pi_k^{0,E}u_{\mathcal{I}}), \Pi_k^{0,E}v_h\big)_{0,E} \\ &\quad - \int_{\partial E}(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)(I-\Pi_k^{0,E})\, u_{\mathcal{I}}\, \Pi_k^{0,E}v_h \,\mathrm{d}s \\ &= \big(\boldsymbol{\beta}\cdot\nabla u, (I-\Pi_k^{0,E})v_h\big)_{0,E} - \big(u-\Pi_k^{0,E}u_{\mathcal{I}}, \boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}v_h\big)_{0,E} \\ &\quad + \int_{\partial E}(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)(u-u_{\mathcal{I}})\,\Pi_k^{0,E}v_h\,\mathrm{d}s \\ &= \big((I-\Pi_k^{0,E})\boldsymbol{\beta}\cdot\nabla u, (I-\Pi_k^{0,E})v_h\big)_{0,E} + \big(\Pi_k^{0,E}u_{\mathcal{I}}-u, \boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}v_h\big)_{0,E} \\ &\quad + \int_{\partial E}(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)\, e_{\mathcal{I}}\,\Pi_k^{0,E}v_h\,\mathrm{d}s \\ &=: \eta_{b,1}^E + \eta_{b,2}^E + \eta_{b,3}^E\,,\end{aligned}$$

and

$$\begin{aligned}\eta_{b,B}^E &= \big(\Pi_k^{0,E}u_{\mathcal{I}}-u, \boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}v_h\big)_{0,E} - \big(u, \boldsymbol{\beta}\cdot\nabla(I-\Pi_k^{0,E})v_h\big)_{0,E} \\ &\quad + \int_{\partial E}(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)\,(I-\Pi_k^{0,E})\,v_h\,\Pi_k^{0,E}u_{\mathcal{I}}\,\mathrm{d}s \\ &= \big(\Pi_k^{0,E}u_{\mathcal{I}}-u, \boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}v_h\big)_{0,E} + \big(\boldsymbol{\beta}\cdot\nabla u, (I-\Pi_k^{0,E})v_h\big)_{0,E} \\ &\quad + \int_{\partial E}(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)\,(I-\Pi_k^{0,E})\,v_h\,(\Pi_k^{0,E}u_{\mathcal{I}}-u)\,\mathrm{d}s \\ &= \big(\Pi_k^{0,E}u_{\mathcal{I}}-u, \boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}v_h\big)_{0,E} + \big((I-\Pi_k^{0,E})\boldsymbol{\beta}\cdot\nabla u, (I-\Pi_k^{0,E})v_h\big)_{0,E} \\ &\quad + \int_{\partial E}(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)\,(I-\Pi_k^{0,E})\,v_h\,(\Pi_k^{0,E}u_{\mathcal{I}}-u)\,\mathrm{d}s \\ &=: \eta_{b,2}^E + \eta_{b,1}^E + \eta_{b,4}^E\,.\end{aligned}$$

yielding the following expression for η_b^E

$$2\eta_b^E = 2\eta_{b,1}^E + 2\eta_{b,2}^E + \eta_{b,3}^E + \eta_{b,4}^E\,. \tag{3.68}$$

We now analyze each term in the sum above.

- $\eta_{b,1}^E$: using Cauchy-Schwarz, the continuity in $\Pi_k^{0,E}$ in L^2 and standard estimates, we obtain

$$\begin{aligned}\eta_{b,1}^E &= \big((I-\Pi_k^{0,E})\boldsymbol{\beta}\cdot\nabla u, (I-\Pi_k^{0,E})v_h\big)_{0,E} \\ &\leq \|(I-\Pi_k^{0,E})\boldsymbol{\beta}\cdot\nabla u\|_{0,E}\,\|v_h\|_{0,E} \\ &\leq \|(I-\Pi_k^{0,E})\boldsymbol{\beta}\cdot\nabla u\|_{0,E}\,\|v_h\|_{\mathrm{cip},E} \\ &\lesssim h^k|\boldsymbol{\beta}\cdot\nabla u|_{k,E}\|v_h\|_{\mathrm{cip},E} \\ &\lesssim h^k\,\|u\|_{k+1,E}\,\|\beta\|_{[W_\infty^k(E)]^2}\,\|v_h\|_{\mathrm{cip},E}.\end{aligned}$$

- $\eta_{b,2}^E$: we have that

$$\begin{aligned}\eta_{b,2}^E &= \big(\Pi_k^{0,E}u_{\mathcal{I}}-u, \boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}v_h\big)_{0,E} \\ &\leq \|\Pi_k^{0,E}u_{\mathcal{I}}-u\|_{0,E}\,\|\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}v_h\|_{0,E} \\ &\leq \big(\|(I-\Pi_k^{0,E})u\|_{0,E}+\|e_{\mathcal{I}}\|_{0,E}\big)\|\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}v_h\|_{0,E} \\ &\lesssim h^{k+\frac{1}{2}}\|u\|_{k+1,E}\|v_h\|_{\mathrm{cip},E}\,.\end{aligned}$$

- $\eta_{b,3}^E + \eta_{b,4}^E$: we use a scaled trace inequality making use of the scaled norm

$$\forall v\in H^1(E), \quad |\!|\!|v|\!|\!|_{1,E}^2 := \|v\|_{L^2(E)}^2 + h_E^2\,|v|_{H^1(E)}^2\,.$$

We obtain

$$
\begin{aligned}
\eta^E_{b,3} + \eta^E_{b,4} &= \int_{\partial E} (\boldsymbol{\beta} \cdot \boldsymbol{n}^E)\, e_{\mathcal{I}}\, \Pi^{0,E}_k v_h \,\mathrm{d}s + \int_{\partial E} (\boldsymbol{\beta} \cdot \boldsymbol{n}^E)\, (I - \Pi^{0,E}_k)\, v_h\, (\Pi^{0,E}_k u_{\mathcal{I}} - u) \,\mathrm{d}s \\
&= \int_{\partial E} (\boldsymbol{\beta} \cdot \boldsymbol{n}^E)\, (\Pi^{0,E}_k - I)\, v_h\, (e_{\mathcal{I}} + u - \Pi^{0,E}_k u_{\mathcal{I}}) \,\mathrm{d}s + \int_{\partial E} (\boldsymbol{\beta} \cdot \boldsymbol{n}^E)\, e_{\mathcal{I}}\, v_h \,\mathrm{d}s \\
&\lesssim \gamma_E^{\frac{1}{2}} \left(\|e_{\mathcal{I}}\|_{L^2(\partial E)} + \|u - \Pi^{0,E}_k u_{\mathcal{I}}\|_{L^2(\partial E)}\right) \gamma_E^{\frac{1}{2}} \|(I - \Pi^{0,E}_k) v_h\|_{L^2(\partial E)} \\
&\quad + \int_{\partial E} (\boldsymbol{\beta} \cdot \boldsymbol{n}^E)\, e_{\mathcal{I}}\, v_h \,\mathrm{d}s \\
&\lesssim h_E^{-1} \left(|\!|\!| e_{\mathcal{I}} |\!|\!|_{1,E} + |\!|\!| u - \Pi^{0,E}_k u_{\mathcal{I}} |\!|\!|_{1,E}\right) \gamma_E^{\frac{1}{2}}\, \|(I - \Pi^{0,E}_k) v_h\|_{0,E} \\
&\quad + \int_{\partial E} (\boldsymbol{\beta} \cdot \boldsymbol{n}^E)\, e_{\mathcal{I}}\, v_h \mathrm{d}s \\
&\lesssim h^{-\frac{1}{2}} (|\!|\!| e_{\mathcal{I}} |\!|\!|_{1,E} + |\!|\!| u - \Pi^{0,E}_k u_{\mathcal{I}} |\!|\!|_{1,E}) \gamma_E^{\frac{1}{2}}\, h^{\frac{1}{2}} \|\nabla(I - \Pi^{\nabla,E}_k) v_h\|_{0,E} \\
&\quad + \int_{\partial E} (\boldsymbol{\beta} \cdot \boldsymbol{n}^E)\, e_{\mathcal{I}}\, v_h \mathrm{d}s \\
&\lesssim h^{k+\frac{1}{2}} |u|_{k+1,E} \|v_h\|_{\mathrm{cip},E} + \int_{\partial E} (\boldsymbol{\beta} \cdot \boldsymbol{n}^E)\, e_{\mathcal{I}}\, v_h \,\mathrm{d}s\,,
\end{aligned}
$$

where in the last step we used the $J_h(v_h, v_h)$ term in the definition of $\|v_h\|_{\mathrm{cip},E}$.

The thesis now follows gathering the last three inequalities in (3.68). □

Lemma 3.24 (Estimate of η_c). *Under the assumptions **(A-C)**, the term η_c can be bounded as follows*

$$
\eta_c \lesssim h^{k+1} \left(\sum_{E \in \Omega_h} |u|^2_{k+1,E} \right)^{\frac{1}{2}} .
$$

Proof. Similarly to Lemma 3.22, we have that

$$
\begin{aligned}
c^E(u, v_h) - c^E_h(u_{\mathcal{I}}, v_h) &= c^E(u - \Pi^{0,E}_k u, v_h) + c^E_h(\Pi^{0,E}_k u - u_{\mathcal{I}}, v_h) \\
&\leq (\|e_\pi\|_{0,E} + (1 + \alpha^*)\|\Pi^{0,E}_k u - u_{\mathcal{I}}\|_{0,E})\, \|v_h\|_{0,E} \\
&\lesssim (\|e_\pi\|_{0,E} + \|e_{\mathcal{I}}\|_{0,E})\, \|v_h\|_{0,E} \\
&\lesssim h^{k+1}\, |u|_{k+1,E}\, \|v_h\|_{\mathrm{cip},E} \,.
\end{aligned}
$$

□

Lemma 3.25 (Estimate of $\eta_{\mathcal{N}}$). *Under the assumptions **(A-C)**, the term $\eta_{\mathcal{N}}$ can be bounded as follows*

$$
\eta_{\mathcal{N}} \lesssim (\epsilon^{\frac{1}{2}}\, h^k + h^{k+\frac{1}{2}}) \left(\sum_{E \in \Omega_h} |u|^2_{k+1,E} \right)^{\frac{1}{2}} .
$$

Proof. We consider an element $E \in \Omega_h$. By definition of the two bilinear forms, we have that

$$
\begin{aligned}
&\epsilon\, \langle \boldsymbol{\Pi}^{0,E}_{k-1} \nabla u_{\mathcal{I}} \cdot \boldsymbol{n} - \nabla u \cdot \boldsymbol{n}, v_h \rangle_{\Gamma_E} + \epsilon\, \langle u_{\mathcal{I}} - u, \boldsymbol{\Pi}^{0,E}_{k-1} \nabla v_h \cdot \boldsymbol{n} \rangle_{\Gamma_E} \\
&\quad + \frac{\epsilon}{\delta h_E} \langle u - u_{\mathcal{I}}, v_h \rangle_{\Gamma_E} + \frac{1}{2} \langle |\boldsymbol{\beta} \cdot \boldsymbol{n}| (u - u_{\mathcal{I}}), v_h \rangle_{\Gamma_E} \\
&= \eta^E_{\mathcal{N},a} + \eta^E_{\mathcal{N},b} + \eta^E_{\mathcal{N},c} + \eta^E_{\mathcal{N},d} \,.
\end{aligned}
\tag{3.69}
$$

Now, we estimate each of the four terms. Using trace inequality, the first returns

$$
\begin{aligned}
\eta^E_{\mathcal{N},a} &= \epsilon\, \langle \boldsymbol{\Pi}^{0,E}_{k-1} \nabla u_{\mathcal{I}} \cdot \boldsymbol{n} - \nabla u \cdot \boldsymbol{n}^E, v_h \rangle_{\Gamma_E} \\
&\lesssim \epsilon\, (h^{-\frac{1}{2}} \|\nabla u - \boldsymbol{\Pi}^{0,E}_{k-1} \nabla u_{\mathcal{I}}\|_{0,E} + h^{\frac{1}{2}} |\nabla u - \boldsymbol{\Pi}^{0,E}_{k-1} \nabla u_{\mathcal{I}}|_{1,E}) \|v_h\|_{0,\Gamma_E} \\
&\lesssim \epsilon^{\frac{1}{2}}\, (\|\nabla u - \boldsymbol{\Pi}^{0,E}_{k-1} \nabla u_{\mathcal{I}} u_{\mathcal{I}}\|_{0,E} + h |\nabla u - \boldsymbol{\Pi}^{0,E}_{k-1} \nabla u_{\mathcal{I}}|_{1,E}) \|v_h\|_{\mathrm{cip},E} \,.
\end{aligned}
$$

Adding and subtracting $\Pi_k^{\nabla,E} u$, using triangular inequality and Lemma 1.4, we obtain

$$\eta^E_{\mathcal{N},a} \lesssim \epsilon^{\frac{1}{2}} h^k |u|_{k+1,E} \|v_h\|_{\mathrm{cip},E} \,.$$

On the second one we have

$$\begin{aligned}
\eta^E_{\mathcal{N},c} &= \epsilon \, \langle u_{\mathcal{I}} - u, \boldsymbol{\Pi}^{0,E}_{k-1} \nabla v_h \cdot \boldsymbol{n} \rangle_{0,\Gamma_E} \\
&\lesssim \epsilon \, \|u - u_{\mathcal{I}}\|_{0,\Gamma_E} \|\boldsymbol{\Pi}^{0,E}_{k-1} \nabla v_h\|_{0,\Gamma_E} \\
&\lesssim \epsilon \, \|u - u_{\mathcal{I}}\|_{0,\Gamma_E} \, h^{-\frac{1}{2}} \|\boldsymbol{\Pi}^{0,E}_{k-1} \nabla v_h\|_{0,E} \\
&\lesssim \epsilon \, h^{-\frac{1}{2}} \, \|u - u_{\mathcal{I}}\|_{0,\Gamma_E} |v_h|_{1,E} \\
&\lesssim \epsilon^{\frac{1}{2}} \, \left(|u - u_{\mathcal{I}}|_{1,E} + h^{-1} \|u - u_{\mathcal{I}}\|_{0,E}\right) \|v_h\|_{\mathrm{cip},E} \\
&\lesssim \epsilon^{\frac{1}{2}} h^k |u|_{k+1,E} \|v_h\|_{\mathrm{cip},E} \,.
\end{aligned} \tag{3.70}$$

For the third term, using trace inequality and interpolation estimate, we have that

$$\begin{aligned}
\eta^E_{\mathcal{N},c} &= \frac{\epsilon}{\delta h_E} \langle u, v_h \rangle_{\Gamma_E} - \frac{\epsilon}{\delta h_E} \langle u_{\mathcal{I}}, v_h \rangle_{\Gamma_E} \\
&\lesssim \frac{\epsilon}{\delta h} \|u - u_{\mathcal{I}}\|_{0,\Gamma_E} \|v_h\|_{0,\Gamma_E} \\
&\lesssim \left(\frac{\epsilon}{\delta h}\right)^{\frac{1}{2}} \|u - u_{\mathcal{I}}\|_{0,\Gamma_E} \|v_h\|_{\mathrm{cip},E} \\
&\lesssim \epsilon^{\frac{1}{2}} h^k |u|_{k+1,E} \|v_h\|_{\mathrm{cip},E} \,.
\end{aligned}$$

Finally, the last one is treated in a very similar way with respect to the previous one, it gives

$$\begin{aligned}
\eta^E_{\mathcal{N},d} &= -\frac{1}{2} \langle |\boldsymbol{\beta} \cdot \boldsymbol{n}^E| u, v_h \rangle_{\Gamma_E} + \frac{1}{2} \langle |\boldsymbol{\beta} \cdot \boldsymbol{n}^E| u_{\mathcal{I}}, v_h \rangle_{\Gamma_E} \\
&\lesssim h^{-\frac{1}{2}} \|u - u_{\mathcal{I}}\|_{0,E} \|v_h\|_{\mathrm{cip},E} \\
&\lesssim h^{k+\frac{1}{2}} |u|_{k+1,E} \|v_h\|_{\mathrm{cip},E} \,.
\end{aligned} \tag{3.71}$$

□

Lemma 3.26 (Estimate of η_J)**.** *Under the assumptions **(A-C)**, the term η_J can be bounded as follows*

$$\eta_J \lesssim h^{k+\frac{1}{2}} \left(\sum_{E \in \Omega_h} |u|^2_{k+1,E} \right)^{\frac{1}{2}} .$$

Proof. Using Cauchy-Schwarz inequality, we have that

$$\begin{aligned}
J_h^E(u_{\mathcal{I}}, v_h) &\leq J_h^E(u_{\mathcal{I}}, u_{\mathcal{I}})^{\frac{1}{2}} J_h^E(v_h, v_h)^{\frac{1}{2}} \\
&\leq J_h^E(u_{\mathcal{I}}, u_{\mathcal{I}})^{\frac{1}{2}} \|v_h\|_{\mathrm{cip},E} \,.
\end{aligned} \tag{3.72}$$

Since the solution u is sufficiently smooth, we have that

$$\begin{aligned}
J_h^E(u_{\mathcal{I}}, u_{\mathcal{I}}) &= \sum_{e \subset \partial E} \int_e \frac{\gamma_e}{2} \, h_e^2 \, [\![\nabla \Pi_k^0 u_{\mathcal{I}}]\!] \cdot [\![\nabla \Pi_k^0 u_{\mathcal{I}}]\!] \, \mathrm{d}s + h_E \, \gamma_E \, \mathcal{S}_j^E((I - \Pi_k^{\nabla,E}) u_{\mathcal{I}}, (I - \Pi_k^{\nabla,E}) u_{\mathcal{I}}) \\
&= \sum_{e \subset \partial E} \int_e \frac{\gamma_e}{2} \, h_e^2 \, [\![\nabla (\Pi_k^0 u_{\mathcal{I}} - u)]\!]^2 \, \mathrm{d}s + h_E \, \gamma_E \, \mathcal{S}_j^E((I - \Pi_k^{\nabla,E}) u_{\mathcal{I}}, (I - \Pi_k^{\nabla,E}) v_h) \\
&\lesssim \sum_{K \in \mathcal{D}(E)} h^2 \, \|\nabla (\Pi_k^0 u_{\mathcal{I}} - u)\|^2_{0,\partial K} + h \, |(I - \Pi_k^{\nabla,E}) u_{\mathcal{I}}|^2_{1,E} \,.
\end{aligned}$$

Using trace inequality, we obtain for the first term

$$\|\nabla \Pi_k^0 u_{\mathcal{I}} - \nabla u\|_{0,\partial K} \lesssim \left(h^{-1} \, \|\nabla \Pi_k^0 u_{\mathcal{I}} - \nabla u\|^2_{0,K} + h \, |\nabla \Pi_k^0 u_{\mathcal{I}} - \nabla u|^2_{1,K}\right)^{\frac{1}{2}} .$$

Adding and subtracting $\nabla\Pi_k^{0,E}u$, using lemma 1.4 and interpolation estimate, we obtain

$$\begin{aligned}h^{-\frac{1}{2}}\|\nabla\Pi_k^{0,E}u_{\mathcal{I}}-\nabla u\|_{0,K} &\lesssim h^{-\frac{1}{2}}\|\nabla\Pi_k^{0,E}u-\nabla u\|_{0,K}+h^{-\frac{1}{2}}\|\nabla\Pi_k^{0,E}u_{\mathcal{I}}-\nabla\Pi_k^{0,E}u\|_{0,K}\\ &\lesssim h^{k-\frac{1}{2}}|u|_{k+1,K}\,,\end{aligned}$$

and similarly, we have that

$$h^{\frac{1}{2}}|\nabla\Pi_k^{0,E}u_{\mathcal{I}}-\nabla u|_{1,K}\lesssim h^{k-\frac{1}{2}}|u|_{k+1,K}\,.$$

Using Lemma 1.4, we have that

$$h^{\frac{1}{2}}\,|(I-\Pi_k^{\nabla,E})u_{\mathcal{I}}|_{1,E}\lesssim h^{\frac{1}{2}}\left(\|\nabla e_{\mathcal{I}}\|_{0,E}+\|\nabla e_{\pi}\|_{0,E}\right)\lesssim h^{k+\frac{1}{2}}|u|_{k+1,E}\,.$$

We conclude

$$J_h^E(u_{\mathcal{I}},v_h)\lesssim h^{k+\frac{1}{2}}\,|u|_{k+1,\mathcal{D}(E)}\,\|v_h\|_{\text{cip},\mathcal{D}(E)}\,.$$

□

We thus have the following proposition.

Theorem 3.27 (Error estimate). *Under the assumptions **(A-C)**, let u be the solution of equation* (3.2) *and $u_h\in V_h^k(\Omega_h)$ be the solution of equation* (3.14). *Then it holds that*

$$\|u-u_h\|_{\text{cip}}^2\lesssim\sum_{E\in\Omega_h}\Theta^E\left(\epsilon\,h^{2k}+h^{2k+1}\right)\,,$$

where the constant Θ^E depends on $\|u\|_{k+1,E}$, $\|f\|_{k+1,E}$, $\|\beta\|_{[W_\infty^{k+1}(E)]^2}$.

Proof. It it sufficient to use Proposition 3.19 combined with Lemmas 3.20, 3.21, 3.22, 3.23, 3.24, 3.25 3.26, noting that

$$\sum_{E\in\Omega_h}\int_{\partial E\setminus\partial\Omega}(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)\,e_{\mathcal{I}}\,v_h\mathrm{d}s=0\,,$$

and the contributions stemming from $\partial\Omega$ are controlled as in (3.71). □

3.2.4 A special case: advection-diffusion problem with $\boldsymbol{\beta}\in[\mathbb{P}_1(\Omega)]^2$

We consider problem (3.1) in a particular situation: we assume an advection term $\boldsymbol{\beta}\in[\mathbb{P}_1(\Omega)]^2$, i.e. globally linear, and we allow the reaction coefficient $\sigma=0$. We do not make further assumptions on the diffusion coefficient ϵ and on the load term f. Thus, the advection-diffusion problem reads as (cf. (3.1))

$$\begin{cases}\text{find } u\in V(\Omega)\text{ such that:}\\ \epsilon\,a(u,v)+b^{\text{skew}}(u,v)=\mathcal{F}(v)\qquad\forall v\in V(\Omega).\end{cases}\tag{3.73}$$

In this case, even without the reaction term, we are able to prove robust estimates for the approximation of problem (3.73). Using the same approach as before, the discrete version of problem (3.73) reads as

$$\begin{cases}\text{find } u_h\in V_h^k(\Omega_h)\text{ such that:}\\ \mathcal{A}_{\text{cip}}^{\text{ad}}(u_h,\,v_h)=\mathcal{F}_h(v_h)\qquad\forall v_h\in V_h^k(\Omega_h),\end{cases}\tag{3.74}$$

where

$$\mathcal{A}_{\text{cip}}^{\text{ad}}(u_h,v_h):=\sum_{E\in\Omega_h}\mathcal{A}_{\text{cip}}^{\text{ad},E}(u_h,v_h)\,,$$

and

$$\mathcal{A}_{\text{cip}}^{\text{ad},E}(u_h,v_h):=\epsilon\,a_h^E(u_h,v_h)+b_h^{\text{skew},E}(u_h,v_h)+\mathcal{N}_h(u_h,v_h)+J_h^E(u_h,v_h)\,.$$

The key observation is that a suitable inf-sup condition still holds true without the help of the L^2-type norm stemming from the reaction term. In fact, introducing the local norm

$$\|v_h\|^2_{\mathrm{cip,ad},E} := \epsilon \, \|\nabla v_h\|^2_{0,E} + h \, \|\boldsymbol{\beta} \cdot \nabla \Pi^{0,E}_k v_h\|^2_{0,E} + \sum_{e \subset \Gamma_E} \|\xi(\epsilon, \boldsymbol{\beta}) v_h\|^2_{0,e} + J^E_h(v_h, v_h) \, ,$$

and its global counterpart

$$\|v_h\|^2_{\mathrm{cip,ad}} := \sum_{E \in \Omega_h} \|v_h\|^2_{\mathrm{cip,ad},E} \, ,$$

similarly to Proposition 3.18, we have the following result.

Proposition 3.28. *Under assumptions **(A-C)**, it holds that*

$$\|v_h\|_{\mathrm{cip,ad}} \lesssim \sup_{z_h \in V_h(\Omega_h)} \frac{\mathcal{A}^{\mathrm{ad}}_{\mathrm{cip}}(v_h, z_h)}{\|z_h\|_{\mathrm{cip,ad}}} \qquad \forall v_h \in V^k_h(\Omega_h), \tag{3.75}$$

for a constant that does not depend on h and ϵ.

Proof. The proof of (3.75) is analogous to the one of proposition 3.18, with the simplification that in Lemma 3.15 it holds $\boldsymbol{\beta}_h = \boldsymbol{\beta}$. The main difference stands in the treatment of $\eta_{\boldsymbol{\beta}_1}$, $\eta_{\boldsymbol{\beta}_2}$ and $\eta_{\boldsymbol{\beta}_3}$ in ($\mathbf{T_5}$), see (3.49). These terms are the only ones requiring the help of the L^2 norm in the general case (apart, of course, the reaction term itself). Regarding the term $\eta_{\boldsymbol{\beta}_1}$, in our present case we immediately have the advective norm:

$$\eta_{\boldsymbol{\beta}_1} = h \, \|\boldsymbol{\beta} \cdot \nabla \Pi^{0,E}_k v_h\|^2_{0,E} \tag{3.76}$$

Since $\boldsymbol{\beta} \in \mathbb{P}_1(\Omega)$, it follows that $\boldsymbol{\beta} \cdot \nabla \Pi^{0,E}_k v_h \in \mathbb{P}_k(E)$, so that we can directly bound $\eta_{\boldsymbol{\beta}_2}$ using Young's inequality and proposition 3.11:

$$\begin{aligned}
\eta_{\boldsymbol{\beta}_2} &= h\big(\boldsymbol{\beta} \cdot \nabla \Pi^{0,E}_k v_h, (\pi - I)(\boldsymbol{\beta} \cdot \nabla \Pi^{0,E}_k v_h)\big)_{0,E} \\
&\geq -\frac{h}{2} \|\boldsymbol{\beta} \cdot \nabla \Pi^{0,E}_k v_h\|^2_{0,E} - \frac{h}{2} \|(\pi - I)(\boldsymbol{\beta} \cdot \nabla \Pi^{0,E}_k v_h)\|^2_{0,E} \\
&\geq -\frac{h}{2} \|\boldsymbol{\beta} \cdot \nabla \Pi^{0,E}_k v_h\|^2_{0,E} - C \, h^2 \left(\sum_{e \in \mathcal{F}_E} \|[\![\boldsymbol{\beta} \cdot \nabla \Pi^{0,E}_k v_h]\!]\|_{0,e} \right)^2 \\
&\geq -\frac{h}{2} \|\boldsymbol{\beta} \cdot \nabla \Pi^{0,E}_k v_h\|^2_{0,E} - C \, h^2 \sum_{e \in \mathcal{F}_E} \gamma_e^2 \|[\![\nabla \Pi^{0,E}_k v_h]\!]\|^2_{0,e} \\
&\geq -\frac{h}{2} \|\boldsymbol{\beta} \cdot \nabla \Pi^{0,E}_k v_h\|^2_{0,E} - C \, h^2 \sum_{e \in \mathcal{F}_E} \gamma_e \|[\![\nabla \Pi^{0,E}_k v_h]\!]\|^2_{0,e} \\
&\geq -\frac{h}{2} \|\boldsymbol{\beta} \cdot \nabla \Pi^{0,E}_k v_h\|^2_{0,E} - C \, J^{\mathcal{D}(E)}_h(v_w, v_h) \, ,
\end{aligned} \tag{3.77}$$

which is the counterpart of (3.54). Furthermore, again since $\boldsymbol{\beta} \cdot \nabla \Pi^{0,E}_k v_h \in \mathbb{P}_k(E)$, it follows that

$$\eta_{\boldsymbol{\beta}_3} := \big(\boldsymbol{\beta} \cdot \nabla \Pi^{0,E}_k v_h, \, (\Pi^{0,E}_k - I) w_h\big)_{0,E} = 0 \, .$$

□

Once the above stability result has been established, the next error estimate can be proved using the same arguments of theorem 3.27. The only difference is handling the terms $\eta^E_{\mathcal{F}}$ and $\eta^E_{b,1}$ which now must be bounded using diffusion (since reaction is not available) and therefore paying a price in terms of ε but with a better rate in terms of h. For the sake of conciseness we here omit the simple alternative derivations for such terms.

Theorem 3.29. *Under the assumptions **(A-C)**, let u be the solution of equation* (3.73) *and $u_h \in V^k_h(\Omega_h)$ be the solution of equation* (3.74). *Then it holds that*

$$\|u - u_h\|^2_{\mathrm{cip,ad}} \lesssim \sum_{E \in \Omega_h} \Theta^E \left(\epsilon \, h^{2k} + h^{2k+1} + \frac{h^{2(k+2)}}{\epsilon} \right) \, ,$$

where the constant Θ^E depends on $\|u\|_{k+1,E}$, $\|f\|_{k+1,E}$, $\|\beta\|_{[W^{k+1}_\infty(E)]^2}$.

3.3 Numerical Experiment

In this section, we investigate the actual computational behavior of the proposed method.

Model problem. We consider a family of problems in the unit square $\Omega = (0,\,1) \times (0,\,1)$. We choose the boundary conditions and the source term (which turns out to depend on ϵ, σ and $\boldsymbol{\beta}$) in such a way that the analytical solution is always the function

$$u(x,\,y) := \sin(\pi\,x)\sin(\pi\,y)\,.$$

Different choices of the parameters σ, ϵ and of the advective term $\boldsymbol{\beta}(x, y)$ will be selected. Since the pointwise values of the numerical solution u_h are unknown, the following error quantities will be considered:

- H^1**−seminorm error**

$$e_{H^1} := \sqrt{\sum_{E\in\Omega_h} \left\|\nabla(u - \Pi_k^{\nabla,E} u_h)\right\|_{0,E}^2}\,;$$

- L^2**−norm error**

$$e_{L^2} := \sqrt{\sum_{E\in\Omega_h} \left\|(u - \Pi_k^{0,E} u_h)\right\|_{0,E}^2}\,.$$

We will consider two different mesh families :

- `octag:` a mesh composed by convex and non-convex quadrilaterals obtained perturbing a mesh composed of structured squares. After displacing randomly the vertexes, each edge in the mesh is split into two parts making the elements octagons with aligned edges;
- `voro:` a centroidal Voronoi tessellation of the unit square.

These two families are represented in Figure 3.1. In all the following numerical experiments, accordingly with [35], we select $\kappa_e = \kappa_E = 0.025$, see (3.10).

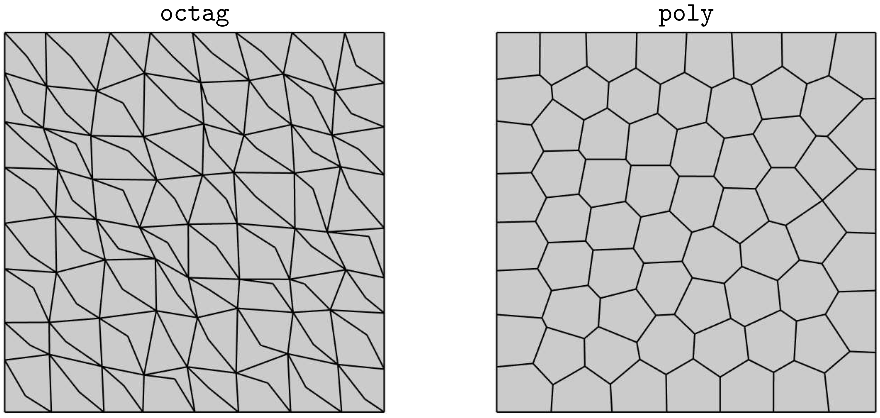


Figure 3.1: Example of meshes used for the present test cases.

Effects of the CIP stabilization. The first aspect we investigate is the benefits of inserting the CIP term in the variational formulation of the problem. We thus consider an advection-dominated regime and choose the parameters $\epsilon = 10^{-5}$, $\sigma = 0$, along with a constant advection term

$$\boldsymbol{\beta}(x,\,y) := \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}.$$

We consider a centroidal Voronoi tesselation of the domain Ω into 256 polygons. The order of the method is set to $k = 1$.

In Figure 3.2 we observe that by inserting the bilinear form $J_h(\cdot, \cdot)$ in the variational formulation, we are able to accurately approximate the analytic solution $u(x, y)$ of the model problem. If we omit the CIP term, we obtain (as expected) a definitely unsatisfactory numerical solution, which exhibits non-physical oscillations all over the computational domain.

As a second example, we consider a solution with an internal layer. The parameters are set to $\epsilon = 10^{-5}$, $\sigma = 1$, along with a non-constant advection term defined as:

$$\boldsymbol{\beta}(x, y) := \begin{bmatrix} -2\,\pi\,\sin(\pi\,(x + 2\,y)) \\ \pi\,\sin(\pi\,(x + 2\,y)) \end{bmatrix}.$$

The order of the method is again $k = 1$. As expected, in Figure 3.3 we note nonphysical oscillations concentrated in particular near the layer. Inserting the CIP stabilization, the oscillations are highly reduced.

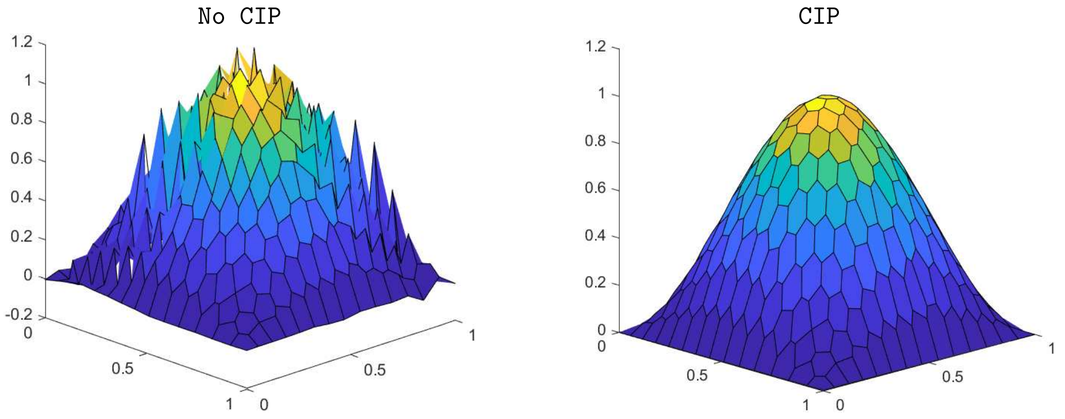


Figure 3.2: Effects of the CIP Stabilizing Term. On the left, a numerical solution without the CIP stabilization term is depicted, while on the right, a solution with the CIP stabilization term is shown.

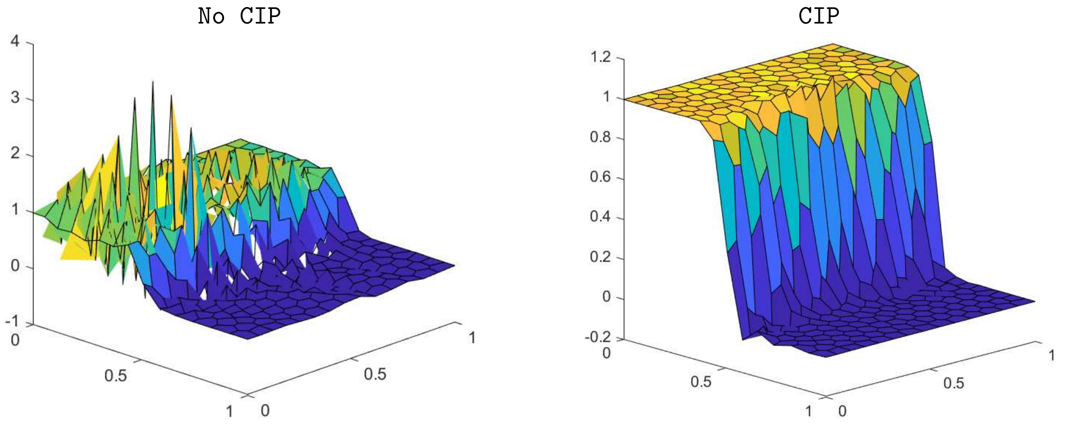


Figure 3.3: Effects of the CIP stabilizing term on a solution with an internal layer. On the left, a numerical solution without the CIP stabilization term is depicted, while on the right, a solution with the CIP stabilization term is shown.

Convergence analysis. We now investigate the convergence of the numerical method by means of the norms introduced above, and choosing a different consistency order, that is,

$k = 1, 2, 3$. We choose the constant advective field

$$\boldsymbol{\beta}(x,\, y) := \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}.$$

We consider a diffusion-dominated case ($\epsilon = 1$), and two advection-dominated cases corresponding to $\epsilon = 10^{-5}$ and $\epsilon = 10^{-9}$. Thus, we are in the framework of section 3.2.4. Accordingly, we neglect the reaction term (hence, $\sigma = 0$) and the theoretical error bound of Theorem 3.29 holds. We compare the method with and without the jump term $J_h(\cdot, \cdot)$. The results are obtained using the Voronoi mesh family.

In Figure 3.4, we observe that in the case $\epsilon = 1$ the two methods behave in the same way, with the lines corresponding to the two cases overlapping completely. Instead, in the advection-dominated regime we observe that the optimal convergences are attained when inserting the stabilising jump term; without it, as expected, the method displays unsatisfactory results, especially for the low-order case. The advantages of the CIP stabilization term become more apparent as the diffusion coefficient decreases.

Effect of the reaction term. We now consider an advection-dominated problem with a variable advection term not in $\mathbb{P}_1(\Omega)$. In particular, we select

$$\boldsymbol{\beta}(x,\, y) := \begin{bmatrix} -2\,\pi\, \sin(\pi\,(x + 2\,y)) \\ \pi\, \sin(\pi\,(x + 2\,y)) \end{bmatrix}.$$

We recall that for this case we are able to prove robust error bounds only with the aid of the reaction term, see Theorem 3.27. The diffusive coefficient is set to $\epsilon = 10{-}5$. We consider both mesh families (`voro` and `octag`). We selected two different values for the reaction term: $\sigma = 1$ and $\sigma = 0$. Figure 3.5 shows that there is no significant difference between the cases $\sigma = 1$ and $\sigma = 0$. As already mentioned, for this latter case Theorem 3.27 does not apply, and no satisfactory theoretical analysis is available, yet. However, the numerical outcomes seem to suggest that it could be possible to drop the reaction term even if the advection term is not globally linear. We note also that we achieve a good convergence also in the case that the mesh is composed of unstructured quadrilaterals.

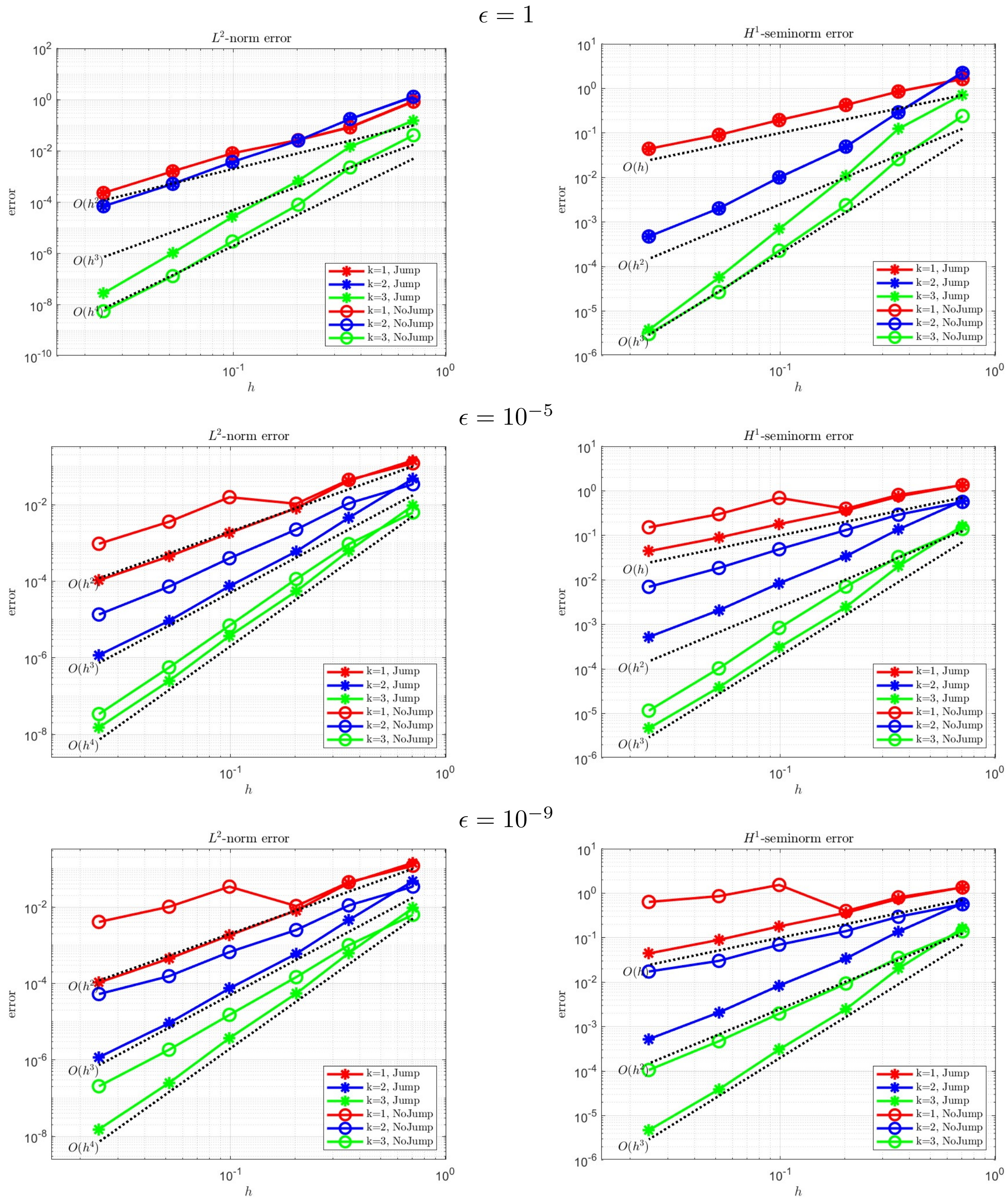
Strong boundary conditions. While the method's theoretical analysis necessitates applying boundary conditions using a Nitsche-type technique, we have experimented with enforcing strong boundary conditions. Our numerical tests show that the solutions converge to the analytical solution with the same convergence rate (specific results not shown here). The primary difference is visual, particularly noticeable when dealing with a solution featuring a boundary layer. We select the function

$$u(x, y) := \left(x - \frac{\exp(\frac{x-1}{\epsilon}) - \exp(\frac{-1}{\epsilon})}{1 - \exp(\frac{-1}{\epsilon})} \right) (y - y^2)\,.$$

as the analytical solution of (3.1). The parameters are set as follows: $\epsilon = 10^{-5}$, $\boldsymbol{\beta} = (1, 2)^T$, $\sigma = 1$, $\kappa_e = \kappa_E = 0.025$ and $\delta = 0.1$. The equation is solved with $k = 1$ on a Voronoi mesh consisting in 256 polygons. In Figure 3.6, we observe that while the boundary conditions on the right are correctly imposed, some oscillations persist near the boundary layer. In contrast, using the Nitsche method, no oscillations are present, but the boundary conditions at $x = 1$ are not accurately enforced.

Other options for the CIP term. To simplify the notation, in this paragraph we denote with $[q]$ the jump of a function q across an interior edge $e \in \mathcal{E}_h^o$. Contrary to the definition (1.19), we are not multiplying by the normal. At the continuous level, given two sufficiently smooth functions u and v, and an interior edge $e \in \mathcal{E}_h^o$, it holds that

$$\int_e h_e^2\, [\nabla u] \cdot [\nabla v]\, \mathrm{d}s = \int_e h_e^2\, [\nabla u \cdot \boldsymbol{n}^e][\nabla u \cdot \boldsymbol{n}^e]\, \mathrm{d}s$$



$\epsilon = 1$

$\epsilon = 10^{-5}$

$\epsilon = 10^{-9}$

Figure 3.4: Convergences in the cases $\epsilon = 1$, $\epsilon = 10^{-5}$ and $\epsilon = 10^{-9}$. The red lines correspond to the case $k = 1$, the blue lines to the case $k = 2$, and the green lines to the case $k = 3$. The circles represent the cases without CIP stabilization, while the stars indicate the cases with CIP stabilization. The left column correspond to the L^2-norm of the error and the right column correspond to the H^1-seminorm of the error.

In our method, we choose to discretize the jump with the normal derivative but it is possible to consider a discretization of the form

$$K_h^E(u_h, v_h) := \sum_{e \subset \partial E} \frac{\gamma_e}{2} \int_e h_e^2 \, [\nabla \Pi_k^0 u_h] \cdot [\nabla \Pi_k^0 v_h] \, \mathrm{d}s + \gamma_E \, h_E \, \mathcal{S}^E\big((I - \Pi_k^{\nabla,E}) u_h, (I - \Pi_k^{\nabla,E}) v_h\big) \, ,$$

in which we discretize the jump of the full gradient. Another possibility is to decompose the gradient as

$$\nabla u = (\nabla u \cdot \boldsymbol{\beta})\boldsymbol{\beta} + (\nabla u \cdot \boldsymbol{\beta}^{\perp})\boldsymbol{\beta}^{\perp} \, ,$$

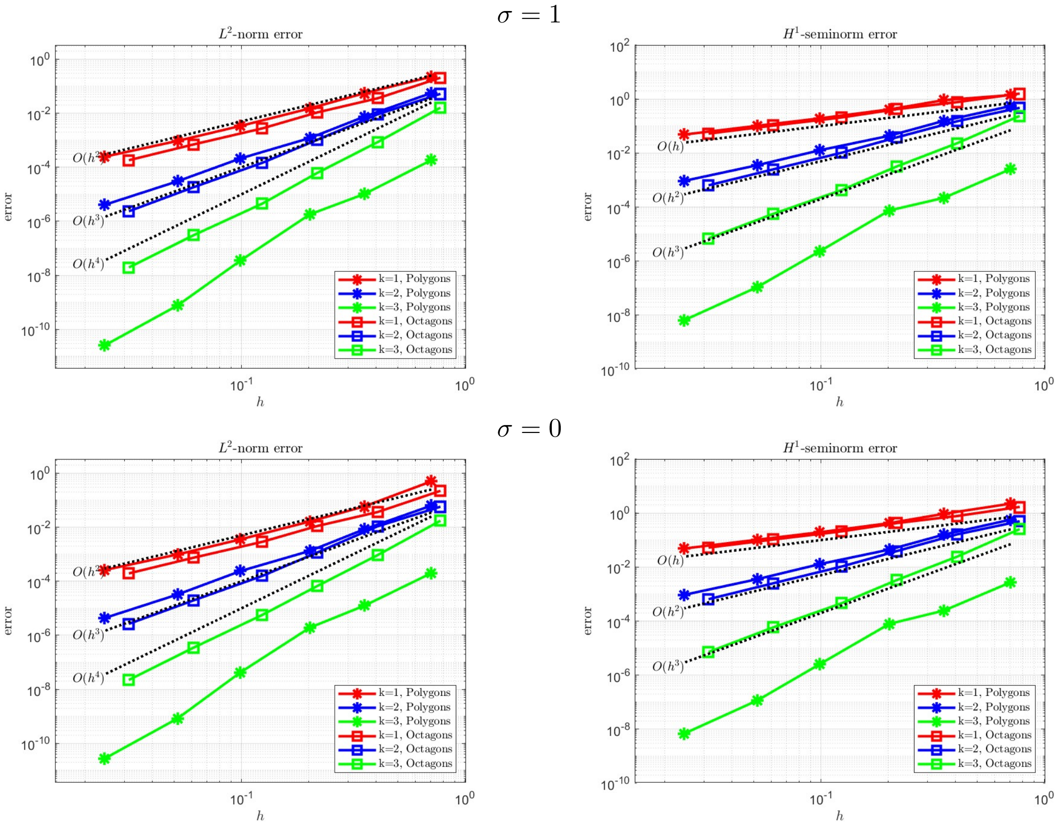
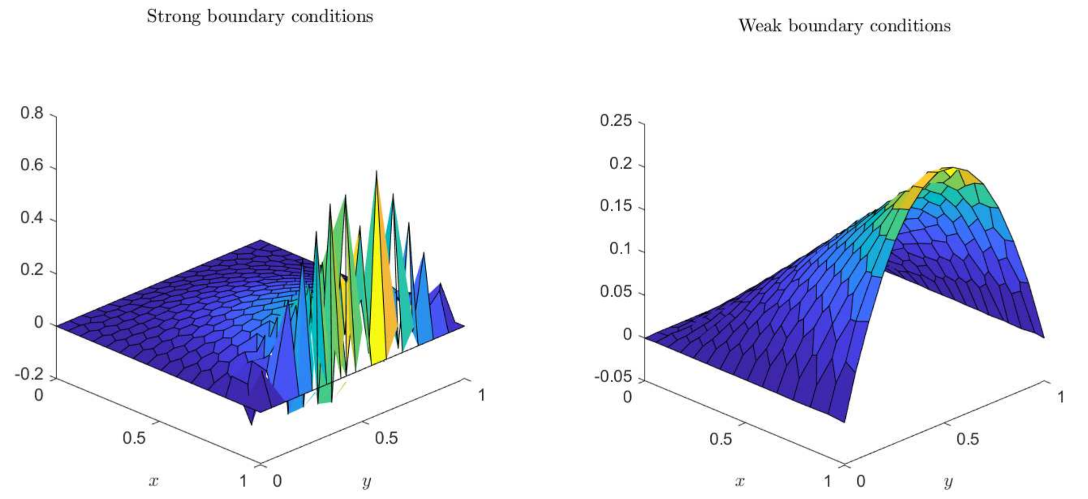


Figure 3.5: Convergences in the case $\sigma = 0$ and $\sigma = 1$. The red lines correspond to the case $k = 1$, the blue lines to the case $k = 2$, and the green lines to the case $k = 3$. The stars refer to the Voronoi mesh family and the squares correspond to octagonal mesh family. The left column correspond to the L^2-norm of the error and the right column correspond to the H^1-seminorm of the error.



Strong boundary conditions

Weak boundary conditions

Figure 3.6: On the left, it represented the solution obtained with the strong boundary conditions, while on the right the one obtained with the weakly imposed boundary conditions.

where $\boldsymbol{\beta}^{\perp}$ is the orthogonal direction with respect to $\boldsymbol{\beta}$, and consider the following discretization

$$\begin{aligned} L_h^E(u_h, v_h) := & \sum_{e \subset \partial E} \frac{\gamma_e}{2} \int_e h_e^2 \, [\boldsymbol{\beta} \cdot \nabla \Pi_k^0 u_h] \cdot [\boldsymbol{\beta} \cdot \nabla \Pi_k^0 v_h] \, \mathrm{d}s \\ & + \sum_{e \subset \partial E} \frac{\gamma_e^{\perp}}{2} \int_e h_e^2 \, [\boldsymbol{\beta}^{\perp} \cdot \nabla \Pi_k^0 u_h] \cdot [\boldsymbol{\beta}^{\perp} \cdot \nabla \Pi_k^0 v_h] \, \mathrm{d}s \\ & + \gamma_E \, h_E \, \mathcal{S}^E\big((I - \Pi_k^{\nabla,E}) u_h, (I - \Pi_k^{\nabla,E}) v_h\big) \, , \end{aligned} \tag{3.78}$$

where $\gamma_e^{\perp}$ is a parameter that controls the gradient in the direction $\boldsymbol{\beta}^{\perp}$. If we choose $\gamma_e = \gamma_e^{\perp}$, we have that $K_h^E(u_h, v_h) = L_h^E(u_h, v_h)$. Now, we consider the situation of the second numerical test with $\epsilon = 10^{-5}$, $k = 1$ on Voronoi meshes, and we verify that the three discretizations $J_h(\cdot,\cdot), K_h(\cdot,\cdot)$ and $L_h(\cdot,\cdot)$ are equivalent by comparing the error in the H^1-seminorm. In $L_h(\cdot,\cdot)$, we set $\gamma_e^{\perp} = 0.01$. The results are depicted in Table 3.1 and it is easy to note the equivalences of the three discretization. We have also tested different polynomial projections like $\nabla \Pi_h^{\nabla}$ or $\boldsymbol{\Pi}_{k-1}^0 \nabla$ and we do not see any substantial difference (results not reported here).

n_P	J_h	K_h	L_h
4	1.3590	1.3598	1.3268
16	0.7491	0.7361	0.7453
64	0.3631	0.3632	0.3629
256	0.1795	0.1797	0.1795
1024	0.0895	0.0895	0.0895
4096	0.0443	0.0443	0.0443
16384	0.0222	0.0222	0.0222

Table 3.1: Results for the three different choices of the stabilization term. We denote with n_P the number of elements in the mesh Ω_h.

Simulation of a fluid inside a channel with two pipes. In this test, we aim to consider a more realistic scenario. We assume that a pollutant, such as oil, is moving through a water-filled channel. This channel contains two cylindrical obstacles, modeled as pipes positioned at different points along its length. The channel is represented by the domain $\Omega = [0, 5] \times$

$[-0.5, 0.5]$, while the pipes are modeled as two circles with centers at $(1.5, 0)$ and $(3.5, 0)$, respectively, and each with a radius of 0.3. For simplicity, we assume that the flow velocity in the channel is uniform and directed along the horizontal axis, given by $\boldsymbol{\beta} = (1, 0)^T$. We consider a scenario with low diffusion, setting the diffusion coefficient ϵ to 10^{-5}, while neglecting the reaction term. We choose as CIP parameter $\kappa_e = \kappa_E = 0.025$, while the Nitsche parameter is set to $\delta = 0.1$. The order of the method is $k = 1$. Homogeneous Dirichlet boundary conditions are imposed along the top and bottom boundaries of the channel, as well as on the surfaces of the pipes. On the left boundary, we prescribe an inflow condition with a parabolic velocity profile of the form

$$-10\,(y - 0.5)\,(y + 0.5)\,.$$

Finally, on the right boundary, we impose homogeneous Neumann boundary conditions to allow the fluid to exit the domain without further interaction. In Figure 3.7, we observe that the pollutant moves in a straight path until it encounters the first pipe. After this point, only the pollutant concentrated near the top and bottom boundaries continues to move toward the right boundary, while the central portion is blocked beneath the pipe.

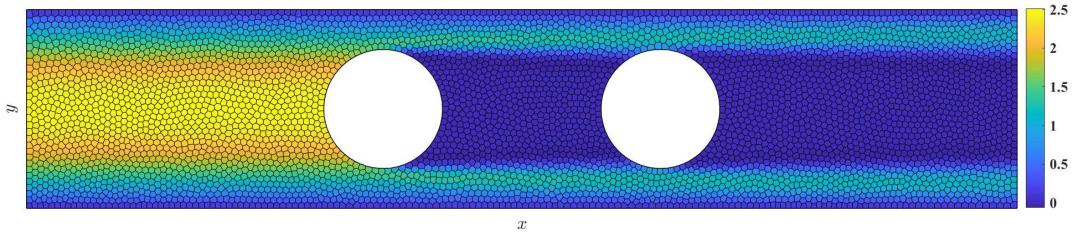


Figure 3.7: Numerical representation of a pollutant moves inside a channel with two pipes. The color represent the concentration of the pollutant.

Chapter 4

A Nonconforming Virtual Element Method for Advection-Diffusion-Reaction Problems with CIP Stabilization

In this chapter we present a nonconforming method for the advection-diffusion-equation with a CIP stabilization term. This work has been developed in [49]. As mentioned in the introduction, the main reason for our interest in the nonconforming setting is that the design, implementation and analysis of nonconforming VEMs are independent of the spatial dimension.

Our nonconforming methods exhibit the following distinctive features compared to the original work [17], which also entail deviations within the stability and error analysis guidelines developed there.

- the convective term is not skew-symmetrized,
- we use a symmetric version of the Nitsche's method to impose Dirichlet boundary conditions.

The chapter is organized as follows. The continuous and discrete problems are presented in the first section. The stability and convergence analysis are developed in the second section. The last section contains a set of numerical tests showing the actual robustness of the method and comparing it with the conforming method described in the previous chapter.

4.1 The continuous and the discrete problems

4.1.1 Model problem

This chapter aims to extend the results present in the previous chapter to the nonconforming method. We consider a polytopal domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, with boundary Γ. In this chapter we consider again the advection-diffusion-reaction equation with non-homogeneous boundary conditions

$$\begin{cases} -\epsilon \Delta u + \boldsymbol{\beta} \cdot \nabla u + \sigma u = f & \text{in } \Omega \,, \\ u = g & \text{on } \Gamma \,. \end{cases} \tag{4.1}$$

where $f \in L^2(\Omega)$ and $g \in H^{\frac{1}{2}}(\Gamma)$. As before, we assume that the diffusion and the reaction coefficient ϵ and σ, respectively, are two positive constants, while the advection coefficient $\boldsymbol{\beta} \in [W^1_\infty(\Omega)]^d$ is such that $\mathrm{div}(\boldsymbol{\beta}) = 0$. The variational formulation is the same that appears in (3.2) with the difference that the trial space is the sobolev space

$$H^1_g(\Omega) := \{v \in H^1(\Omega) \quad \text{such that} \quad v|_\Gamma = g\} \,.$$

The variational problem reads as

$$\begin{cases}\text{find } u \in V_g(\Omega) \coloneqq H^1_g(\Omega) \text{ such that:}\\ \epsilon\, a(u,v) + b(u,v) + \sigma\, c(u,v) = \mathcal{F}(v) \quad \forall v \in V(\Omega) \coloneqq H^1_0(\Omega)\,.\end{cases} \tag{4.2}$$

The forms $a(\cdot,\cdot)\colon V_g(\Omega)\times V(\Omega) \to \mathbb{R}$, $b(\cdot,\cdot)\colon V_g(\Omega)\times V(\Omega)\to\mathbb{R}$ and $c(\cdot,\cdot)\colon V_g(\Omega)\times V(\Omega)\to\mathbb{R}$ are the same that appears in Section 3.1.1. We note that, since the trial space is $H^1_g(\Omega)$, these forms cannot be called "bilinear".

4.1.2 Mesh assumptions

The mesh assumptions for the nonconforming method are the same as those presented in the previous chapter. However, we must exercise caution because this chapter also addresses the case where $d = 3$, necessitating additional considerations.

(A-NC) Mesh assumption. There exists a positive constant ρ such that, for any element $E \in \{\,\Omega_h\,\}_h$,

- E is star-shaped with respect to a ball B_E of radius $\geq\ \rho\, h_E$,
- any facet e of E has diameter $h_e\ \geq\ \rho\, h_E$,
- for $d = 3$, every face e is star-shaped with respect to a ball B_e of radius $\geq\ \rho\, h_e$,
- the mesh is quasi-uniform, i.e., any element has diameter $h_E \geq \rho h$.

Given an interior edge $e \in \mathcal{E}^o_h$, we also introduce the scalar-valued average $\{v\}$ for sufficiently smooth functions v as

$$\{v\} \coloneqq \frac{v^E + v^K}{2}\,, \tag{4.3}$$

where $E, K \in \Omega_h$ are such that $e \subset \partial E \cap \partial K$.

4.1.3 Virtual Element Spaces

Given an element $E \in \Omega_h$ with n_E facets, and a positive integer k, we define the space

$$\begin{aligned} V_h^{k,nc}(E) \coloneqq \big\{ v_h \in & H^1(E) \quad \text{such that} \quad \nabla v_h \cdot \boldsymbol{n}^E \in \mathbb{P}_{k-1}(e) \quad \text{for all } e \subset \partial E,\\ & \Delta v_h \in \mathbb{P}_k(E)\,, \quad (v_h - \Pi^{\nabla,E}_k v_h,\, \widehat{p}_k)_{0,E} = 0 \quad \text{for all } \widehat{p}_k \in \mathbb{P}_k(E)/\mathbb{P}_{k-2}(E) \big\}\,. \end{aligned} \tag{4.4}$$

Similarly to (3.4), this space represents the enhanced version of the virtual element space presented in (1.15). It allows the construction of the projection operator $\Pi^{0,E}_k : V_h^{k,nc}(E) \to \mathbb{P}_k(E)$. We present a set of DoFs for the space (4.4):

- $\mathcal{E}^{E,k}_{nc}$: the moments up to the order $k-1$ on each facet $e \subset \partial E$:

$$\mu^{\boldsymbol{\ell}}_{E,e}(v_h) \coloneqq \frac{1}{|e|}\int_e v_h \left(\frac{\mathbf{s}-\mathbf{s}_e}{h_e}\right)^{\boldsymbol{\ell}} \mathrm{d}s \quad |\boldsymbol{\ell}| \leq k-1, \tag{4.5}$$

  where $\mathbf{s}$ is expressed in the local $d-1$ coordinates on e, $\mathbf{s}_e$ is the barycenter of e, and $\boldsymbol{\ell} \in \mathbb{N}^{d-1}$ is a multi-index with $d-1$ components,

- $\mathcal{P}^{E,k}_{nc}$: the moments up to the order $k-2$ on E:

$$\mu^{\boldsymbol{\alpha}}_E(v_h) \coloneqq \frac{1}{|E|}\int_E v_h \left(\frac{\boldsymbol{x}-\boldsymbol{x}_E}{h_E}\right)^{\boldsymbol{\alpha}} \mathrm{d}E \quad |\boldsymbol{\alpha}| \leq k-2, \tag{4.6}$$

  where $\boldsymbol{x}_E$ is the barycenter of E, and $\boldsymbol{\alpha} \in \mathbb{N}^d$ is a multi-index with d components.

Moving from 2D to 3D in the nonconforming setting is straightforward. In $\mathbb{R}^3$, the DoFs are defined in the same way as in 2D. The key difference is that the interior DoFs involve 3D integrals, while the boundary DoFs involve 2D integrals. In the conforming method this step is more involved.

The dimension N_E^{nc} of the space $V_h^{k,nc}(E)$ is

$$N_E^{nc} = \begin{cases} k\, n_E + \dfrac{(k-1)(k-2)}{2} & \text{if } d = 2\,, \\ \dfrac{(k+1)k}{2} n_E + \dfrac{(k+1)k(k-1)}{6} & \text{if } d = 3\,. \end{cases}$$

For every mesh Ω_h, we introduce the global virtual element space as

$$\begin{aligned} V_h^{k,nc}(\Omega_h) := \big\{ v_h \in W_2^1(\Omega_h) \quad \text{such that} \quad & v_h|_E \in V_h^{k,nc}(E) \quad \forall E \in \Omega_h\,, \\ & \int_e [\![v_h]\!] \cdot \boldsymbol{n}^e \, q \, \mathrm{d}s = 0 \quad \forall q \in \mathbb{P}_{k-1}(e) \quad \forall e \in \mathcal{E}_h^o \big\}\,. \end{aligned} \tag{4.7}$$

We remark that we do not impose full continuity across the element boundaries. On each interior facet, we only require that the moments up to order $k-1$ are preserved. Therefore, $V_h^{k,nc}(\Omega_h) \not\subset H^1(\Omega)$. The global DoFs are as follows:

- $\mathcal{E}_{nc}^k$: the moments up to the order $k-1$ on each $e \in \mathcal{E}_h$:

$$\mu_e^{\boldsymbol{\ell}}(v_h) := \frac{1}{|e|} \int_e v_h \left(\frac{\mathbf{s} - \mathbf{s}_e}{h_e} \right)^{\boldsymbol{\ell}} \mathrm{d}s \quad |\boldsymbol{\ell}| \le k-1,$$

  where $\boldsymbol{\ell} \in \mathbb{N}^{d-1}$ is a multi-index with $d-1$ components,

- $\mathcal{P}_{nc}^k$: the moments up to the order $k-2$ on each $E \in \Omega_h$:

$$\mu_E^{\boldsymbol{\alpha}}(v_h) := \frac{1}{|E|} \int_E v_h \left(\frac{\boldsymbol{x} - \boldsymbol{x}_E}{h_E} \right)^{\boldsymbol{\alpha}} \mathrm{d}E \quad |\boldsymbol{\alpha}| \le k-2,$$

  where $\boldsymbol{\alpha} \in \mathbb{N}^d$ is a multi-index with d components.

The dimension of $V_h^{k,nc}(\Omega_h)$ is

$$N_{\Omega_h}^{nc} = \begin{cases} k\,|\mathcal{E}_h| + \dfrac{(k-1)(k-2)}{2} n_p & \text{if } d = 2\,, \\ \dfrac{(k+1)k}{2} |\mathcal{E}_h| + \dfrac{(k+1)k(k-1)}{6} n_p & \text{if } d = 3\,. \end{cases}$$

If $d = 2$, the dimensions of the global and local spaces are the same that the one presented in Section 2.3.1. Similarly to the conforming enhanced virtual element space, the following interpolation result holds

Lemma 4.1 (Approximation with nonconforming virtual element functions)**.** *Under assumption **(A1)**, for any $v \in H^1(\Omega) \cap H^{s+1}(\Omega_h)$, there exists $v_{\mathcal{I}} \in V_h(\Omega_h)$, such that for all $E \in \Omega_h$,*

$$\|v - v_{\mathcal{I}}\|_{0,E} + h_E \|\nabla(v - v_{\mathcal{I}})\|_{0,E} \lesssim h_E^{s+1} |v|_{s+1,E}\,,$$

where $0 < s \le k$.

4.1.4 Virtual Element Forms and the Discrete Problem

In this section, we discretize the problem and the forms presented in Section 4.1.1. Some of the bilinear forms that will be included in the discrete problem are identical to those appearing in Section 3.1.4. Here, we simply recall their definitions.

$$a_h^E(u_h, v_h) := \int_E \boldsymbol{\Pi}_{k-1}^{0,E} \nabla u_h \cdot \boldsymbol{\Pi}_{k-1}^{0,E} \nabla v_h \, \mathrm{d}E + \mathcal{S}^E\big((I - \Pi_k^{\nabla,E}) u_h,\, (I - \Pi_k^{\nabla,E}) v_h \big)\,, \tag{4.8}$$

$$c_h^E(u_h, v_h) := \int_E \Pi_k^{0,E} u_h \, \Pi_k^{0,E} v_h \mathrm{d}E + |E| \, \mathcal{S}^E\big((I - \Pi_k^{0,E})u_h, \, (I - \Pi_k^{0,E})v_h\big) \, . \tag{4.9}$$

We recall that $\mathcal{S}^E(\cdot, \cdot)$ is a stabilization form, see (1.8). The discretization of the convective term is different. We recall the definition that appears in (3.8)

$$b_h^E(u_h, v_h) = \int_E \boldsymbol{\beta} \cdot \nabla \Pi_k^{0,E} u_h \, \Pi_k^{0,E} v_h \, \mathrm{d}E + \int_{\partial E} (\boldsymbol{\beta} \cdot \boldsymbol{n}^E)(I - \Pi_k^{0,E}) u_h \, \Pi_k^{0,E} v_h \, \mathrm{d}s \, .$$

Since the trace of a virtual function is unknown, it is clear that the boundary integral is not computable if $u_h \in V_h^{k,nc}(E)$. Hence, we define

$$b_h^E(u_h, v_h) := \int_E (\boldsymbol{\beta} \cdot \nabla \Pi_k^{0,E} u_h) \Pi_k^{0,E} v_h \mathrm{d}E \, . \tag{4.10}$$

Another important difference is that we do not skew-symmetrize the bilinear form $b_h^E(\cdot, \cdot)$. Since the method is nonconforming, skew-symmetrizing the bilinear form $b(\cdot, \cdot)$ would require including the boundary integral

$$\frac{1}{2} \int_{\partial E} (\boldsymbol{\beta} \cdot \boldsymbol{n}^E) \, u_h \, v_h \, \mathrm{d}s \, ,$$

because this term does not cancel when summed over all elements. If we opt for this choice, we should insert in the discrete problem

$$\frac{1}{2} \int_{\partial E} (\boldsymbol{\beta} \cdot \boldsymbol{n}^E) \, \Pi_k^{0,E} u_h \, \Pi_k^{0,E} v_h \, \mathrm{d}s \, .$$

At this point, it is clear that

$$b_h^E(u_h, v_h) = \frac{1}{2}\big(b_h^E(u_h, v_h) - b_h^E(v_h, u_h)\big) + \frac{1}{2} \int_{\partial E} (\boldsymbol{\beta} \cdot \boldsymbol{n}^E) \, \Pi_k^{0,E} u_h \, \Pi_k^{0,E} v_h \, \mathrm{d}s \, .$$

In order to ensure stability, we also need to introduce an extra terms. We introduce the bilinear form $d_h^E(\cdot, \cdot) \colon V_h^{k,nc}(E) \times V_h^{k,nc}(E) \to \mathbb{R}$

$$d_h^E(u_h, v_h) := -\frac{1}{2} \int_{\partial E \setminus \Gamma} \boldsymbol{\beta} \cdot [\![\Pi_k^0 u_h]\!] \{\Pi_k^0 v_h\} \mathrm{d}s \, . \tag{4.11}$$

The reasons behind this bilinear form will become clear in Lemma 4.8 below and are related to the nonconformity of the method. We recall the definition of the CIP term which is equivalent to the conforming method. It is defined as

$$J_h^E(u_h, v_h) := \sum_{e \subset \partial E \setminus \Gamma} \frac{\gamma_e}{2} \int_e h_e^2 \, [\![\nabla \Pi_k^0 u_h]\!] \, [\![\nabla \Pi_k^0 v_h]\!] \, \mathrm{d}s + \gamma_E \, h_E \, \mathcal{S}^E\big((I - \Pi_k^{0,E})u_h, (I - \Pi_k^{0,E})v_h\big) \, . \tag{4.12}$$

where the parameters γ_e and γ_E are defined as

$$\gamma_e := \kappa_e \|\boldsymbol{\beta}\|_{[L^\infty(e)]^2} \, , \qquad \gamma_E := \kappa_E \|\boldsymbol{\beta}\|_{[L^\infty(\partial E)]^2} \, , \tag{4.13}$$

see Remarks 3.4 and 3.5.

Remark 4.2. *Despite the fact that the normal derivative of nonconforming virtual functions is a polynomial, we still need to insert a polynomial projection operator in the CIP term. This is necessary because the normal derivative of the virtual function is not explicitly known.*

In order to complete the definition of our method, we need to impose the boundary conditions. We do this using a Nitsche-type technique, which locally consists of adding the form

$$\begin{aligned} \mathcal{N}_h^E(u_h, v_h) :=& -\epsilon \langle \boldsymbol{\Pi}_{k-1}^{0,E} \nabla u_h \cdot \boldsymbol{n}^E, v_h \rangle_{\Gamma_E} - \epsilon \langle u_h, \boldsymbol{\Pi}_{k-1}^{0,E} \nabla v_h \cdot \boldsymbol{n}^E \rangle_{\Gamma_E} \\ &+ \frac{\epsilon}{\delta h_E} \sum_{e \subset \Gamma_E} \langle \Pi_{k-1}^{0,e} u_h, \Pi_{k-1}^{0,e} v_h \rangle_e + \langle |\boldsymbol{\beta} \cdot \boldsymbol{n}^E| \, \Pi_k^{0,E} u_h, \Pi_k^{0,E} v_h \rangle_{\Gamma_{E,\mathrm{in}}} \, ; \end{aligned} \tag{4.14}$$

see [52, 46] for more details. By adding these terms together, we define the complete local bilinear form $\mathcal{A}^E_{\mathrm{cip}} \colon V_h^{k,nc}(E) \times V_h^{k,nc}(E) \to \mathbb{R}$ as

$$\mathcal{A}^E_{\mathrm{cip}}(u_h, v_h) = \epsilon a_h^E(u_h, v_h) + b_h^E(u_h, v_h) + \sigma c_h^E(u_h, v_h) + d_h^E(u_h, v_h) + \mathcal{N}_h^E(u_h, v_h) + J_h^E(u_h, v_h)\,. \tag{4.15}$$

By summing over all the mesh elements, we obtain the global versions of the bilinear forms

$$\begin{aligned} a_h(u_h, v_h) &:= \sum_{E \in \Omega_h} a_h^E(u_h, v_h)\,, & b_h(u_h, v_h) &:= \sum_{E \in \Omega_h} b_h^E(u_h, v_h)\,, \\ c_h(u_h, v_h) &:= \sum_{E \in \Omega_h} c_h^E(u_h, v_h)\,, & d_h(u_h, v_h) &:= \sum_{E \in \Omega_h} d_h^E(u_h, v_h)\,, \\ J_h(u_h, v_h) &:= \sum_{E \in \Omega_h} J_h^E(u_h, v_h)\,, & \mathcal{N}_h(u_h, v_h) &:= \sum_{E \in \Omega_h} \mathcal{N}_h^E(u_h, v_h)\,, \end{aligned}$$

and

$$\mathcal{A}_{\mathrm{cip}}(u_h, v_h) := \sum_{E \in \Omega_h} \mathcal{A}^E_{\mathrm{cip}}(u_h, v_h)\,. \tag{4.16}$$

The local load term is defined as

$$\begin{aligned} \mathcal{F}_h^E(v_h) := &\int_E f\, \Pi_k^{0,E} v_h \mathrm{d}E - \epsilon\, \langle g, \boldsymbol{\Pi}_{k-1}^{0,E} \nabla v_h \cdot \boldsymbol{n}^E \rangle_{\Gamma_E} \\ &+ \frac{\epsilon}{\delta h_E} \sum_{e \subset \Gamma_E} \langle g, \Pi_{k-1}^{0,e} v_h \rangle_e + \langle |\boldsymbol{\beta} \cdot \boldsymbol{n}^E|\, g, \Pi_k^{0,E} v_h \rangle_{\Gamma_{E,\mathrm{in}}}\,, \end{aligned} \tag{4.17}$$

and the global one as

$$\mathcal{F}_h(v_h) := \sum_{E \in \Omega_h} \mathcal{F}_h^E(v_h)\,.$$

Eventually, the discrete problem reads as follows:

$$\begin{cases} \text{find } u_h \in V_h^{k,nc}(\Omega_h) \text{ such that} \\ \mathcal{A}_{\mathrm{cip}}(u_h,\, v_h) = \mathcal{F}_h(v_h) \quad \forall v_h \in V_h^{k,nc}(\Omega_h)\,. \end{cases} \tag{4.18}$$

4.2 Theoretical Analysis

Similarly to Section 3.1.5, due to the nonconformity of the virtual element space and due to the projections entering formulation (4.18), the solution u of the continuous problem (4.2) does not solve the discrete problem (4.18). However, if $u \in H^2(\Omega) \cap H^1_g(\Omega)$, it solves the following problem, which is strictly connected to (4.18):

$$\widetilde{\mathcal{A}}_{\mathrm{cip}}(u,\, v_h) = \tilde{\mathcal{F}}(v_h) + \tilde{\mathcal{B}}(u, v_h) \qquad \text{for all } v_h \in V_h^{k,nc}(\Omega_h)\,. \tag{4.19}$$

The form on the left-hand side of (4.19) is defined as

$$\widetilde{\mathcal{A}}_{\mathrm{cip}}(u, v_h) := \epsilon\, a(u, v_h) + b(u, v_h) + \sigma\, c(u, v_h) + \tilde{\mathcal{N}}_h(u, v_h) + \tilde{J}_h(u, v_h)\,, \tag{4.20}$$

where $\tilde{\mathcal{N}}_h(u, v_h)$ and $\tilde{J}_h(u, v_h)$ are sums over all mesh elements E of the local contributions

$$\begin{aligned} \tilde{\mathcal{N}}_h^E(u, v_h) := &- \epsilon \langle \nabla u \cdot \boldsymbol{n}^E, v_h \rangle_{\Gamma_E} - \epsilon \langle u, \boldsymbol{\Pi}_{k-1}^{0,E} \nabla v_h \cdot \boldsymbol{n}^E \rangle_{\Gamma_E} \\ &+ \frac{\epsilon}{\delta h_E} \sum_{e \subset \Gamma_E} \langle u, \Pi_{k-1}^{0,e} v_h \rangle_e + \langle |\boldsymbol{\beta} \cdot \boldsymbol{n}^E| u, \Pi_k^{0,E} v_h \rangle_{\Gamma_{E,\mathrm{in}}}\,, \end{aligned} \tag{4.21}$$

and

$$\tilde{J}_h^E(u, v_h) := \frac{1}{2} \sum_{e \subset \partial E \setminus \Gamma} \gamma_e \int_e h_e^2\, [\![\nabla u]\!]\, [\![\nabla \Pi_k^0 v_h]\!]\, \mathrm{d}s\,, \tag{4.22}$$

respectively. On the right-hand side of (4.19), the load term $\tilde{\mathcal{F}}(v_h)$ is defined as the sum over all mesh elements E of the local contributions

$$\begin{aligned}\tilde{\mathcal{F}}_h^E(v_h) := \int_E f\, v_h \mathrm{d}E - \epsilon\, \langle g, \boldsymbol{\Pi}_{k-1}^{0,E} \nabla v_h \cdot \boldsymbol{n}^E \rangle_{\Gamma_E} \\ + \frac{\epsilon}{\delta h_E} \sum_{e \subset \Gamma_E} \langle g, \Pi_{k-1}^{0,e} v_h \rangle_e + \langle |\boldsymbol{\beta} \cdot \boldsymbol{n}^E| g, \Pi_k^{0,E} v_h \rangle_{\Gamma_{E,\mathrm{in}}}\,,\end{aligned} \tag{4.23}$$

and form $\tilde{\mathcal{B}}(u, v_h)$, which arises from the nonconformity of the method, is defined as

$$\tilde{\mathcal{B}}(u, v_h) := \sum_{e \in \mathcal{E}_h^o} \epsilon \int_e \nabla u \cdot [\![v_h]\!] \mathrm{d}s\,.$$

4.2.1 Preliminary results

In this section, we present some results that are useful in the following analysis. Some of them are already known for conforming virtual element spaces. The first one is an inverse inequality for functions in $V_h^{k,nc}(E)$.

Proposition 4.3 (Inverse inequality)**.** *Under assumption* ***(A1)****, for any $E \in \Omega_h$ we have*

$$|v_h|_{1,E} \lesssim h_E^{-1} \|v_h\|_{0,E} \quad \forall v_h \in V_h^{k,nc}(E).$$

Proof. An integration by parts gives

$$|v_h|_{1,E}^2 = \int_E \nabla v_h \cdot \nabla v_h \,\mathrm{d}E = -\int_E v_h\, \Delta v_h \mathrm{d}E + \int_{\partial E} v_h \left(\nabla v_h \cdot \boldsymbol{n}^E\right) \mathrm{d}s. \tag{4.24}$$

Since Δv_h is polynomial we have (see for instance [16][1])

$$\| \Delta v_h \|_{0,E} \leq C_\Delta h_E^{-1} |v_h|_{1,E}, \tag{4.25}$$

for a constant $C_\Delta > 0$ independent of h_E. Therefore, for the first term on the right-hand side of (4.24), we get

$$-\int_E v_h\, \Delta v_h \,\mathrm{d}E \leq \|v_h\|_{0,E} \| \Delta v_h \|_{0,E} \leq C_\Delta h_E^{-1} \, \|v_h\|_{0,E} |v_h|_{1,E}\,. \tag{4.26}$$

For the second term, using the Cauchy-Schwarz inequality and the multiplicative trace inequality, we obtain

$$\begin{aligned}\int_{\partial E} v_h \left(\nabla v_h \cdot \boldsymbol{n}^E\right) \mathrm{d}s &\leq \|v_h\|_{0,\partial E} \, \left\|\nabla v_h \cdot \boldsymbol{n}^E\right\|_{0,\partial E} \\ &\leq C_{t_1} \|v_h\|_{0,E}^{\frac{1}{2}} \left(h_E^{-1} \|v_h\|_{0,E} + |v_h|_{1,E}\right)^{\frac{1}{2}} \, \left\|\nabla v_h \cdot \boldsymbol{n}^E\right\|_{0,\partial E}\,,\end{aligned} \tag{4.27}$$

for a positive constant $C_{t_1} > 0$ independent of h_E. The multiplicative trace inequality and (4.25) give

$$\begin{aligned}\left\|\nabla v_h \cdot \boldsymbol{n}^E\right\|_{0,\partial E} &\leq C_{t_2} |v_h|_{1,E}^{\frac{1}{2}} \left(h_E^{-1} |v_h|_{1,E} + \|\Delta v_h\|_{0,E}\right)^{\frac{1}{2}} \\ &\leq C_{t_2} |v_h|_{1,E}^{\frac{1}{2}} \left(h_E^{-1} |v_h|_{1,E} + C_\Delta h_E^{-1} |v_h|_{1,E}\right)^{\frac{1}{2}} \\ &\leq C_{t_2} \sqrt{1 + C_\Delta} h_E^{-\frac{1}{2}} |v_h|_{1,E},\end{aligned} \tag{4.28}$$

for $C_{t_2} > 0$ independent of h_E. Hence, with $C_t := \max\{C_{t_1}, C_{t_2}\}$, we obtain

$$\int_{\partial E} v_h \nabla v_h \cdot \boldsymbol{n}^E \,\mathrm{d}s \leq C_t^2 \sqrt{1 + C_\Delta}\, h_E^{-\frac{1}{2}} \|v_h\|_{0,E}^{\frac{1}{2}} \left(h_E^{-1} \|v_h\|_{0,E} + |v_h|_{1,E}\right)^{\frac{1}{2}} |v_h|_{1,E}. \tag{4.29}$$

[1] Several of the results we quote have been proven in the references for $d = 2$. Nevertheless, these results can be extended to $d = 3$ in a straightforward manner.

From (4.24), (4.26), and (4.29), using the Young inequality $ab \leq \frac{1}{2\eta}a^2 + \frac{\eta}{2}b^2$ for any $\eta > 0$, we get

$$\begin{aligned}|v_h|_{1,E} &\leq C_\Delta h_E^{-1}\|v_h\|_{0,E} + C_t^2\sqrt{1+C_\Delta}\, h_E^{-\frac{1}{2}}\|v_h\|_{0,E}^{\frac{1}{2}}\left(h_E^{-1}\|v_h\|_{0,E} + |v_h|_{1,E}\right)^{\frac{1}{2}} \\ &\leq C_\Delta h_E^{-1}\|v_h\|_{0,E} + C_t^2\sqrt{1+C_\Delta}\left(\frac{1}{2\eta}h_E^{-1}\|v_h\|_{0,E} + \frac{\eta}{2}h_E^{-1}\|v_h\|_{0,E} + \frac{\eta}{2}|v_h|_{1,E}\right).\end{aligned}$$

Choosing $\eta = 1/(C_t^2\sqrt{1+C_\Delta})$, we conclude

$$|v_h|_{1,E} \leq \left(2C_\Delta + C_t^4(1+C_\Delta) + 1\right) h_E^{-1}\|v_h\|_{0,E},$$

and the proof is complete. □

Similarly to Lemma 3.9, the second result is an inverse trace inequality for functions in $V_h^{k,nc}(E)$ with internal DoFs equal to zero.

Lemma 4.4 (Inverse trace inequality)**.** *Under assumption **(A1)**, for any $E \in \Omega_h$ we have*

$$\|v_h\|_{0,E} \lesssim \left(h_E \sum_{e\subset\partial E}\|\Pi_{k-1}^{0,e}v_h\|_{0,e}^2\right)^{\frac{1}{2}} \qquad \forall v_h \in V_h^{k,nc}(E) \text{ such that } \Pi_{k-2}^{0,E}v_h \equiv 0.$$

Proof. Thanks to the orthogonality of the $\Pi_k^{0,E}$ projection in $L^2(E)$, we have

$$\|v_h\|_{0,E}^2 = \|(I-\Pi_k^{0,E})v_h\|_{0,E}^2 + \|\Pi_k^{0,E}v_h\|_{0,E}^2\,. \tag{4.30}$$

We now remark that the techniques leading to [28, Lemma 2.18] also apply to the nonconforming space $V_h^{k,nc}(E)$. Therefore, using $\Pi_{k-2}^{0,E}v_h = 0$, we have

$$\|\Pi_k^{0,E}v_h\|_{0,E}^2 \lesssim h_E \sum_{e\subset\partial E}\|\Pi_{k-1}^{0,e}v_h\|_{0,e}^2\,. \tag{4.31}$$

In addition, Lemma 1.4 gives

$$\|(I-\Pi_k^{0,E})v_h\|_{0,E}^2 \lesssim h_E^2\,|v_h|_{1,E}^2\,. \tag{4.32}$$

Using integration by parts, recalling that $\Delta v_h \in \mathbb{P}_k(E)$ and $(\nabla v_h \cdot \boldsymbol{n}^E)_{|e} \in \mathbb{P}_{k-1}(e)$, exploiting estimates (4.25), (4.31) and (4.28) we obtain

$$\begin{aligned}|v_h|_{1,E}^2 &= -\int_E v_h\,\Delta v_h \mathrm{d}E + \int_{\partial E} v_h\,(\nabla v_h\cdot\boldsymbol{n}^E)\,\mathrm{d}s \\ &= -\int_E \Pi_k^{0,E}v_h\,\Delta v_h\mathrm{d}E + \sum_{e\subset\partial E}\int_e \Pi_{k-1}^{0,e}v_h\,(\nabla v_h\cdot\boldsymbol{n}^E)\,\mathrm{d}s \\ &\leq \|\Pi_k^{0,E}v_h\|_{0,E}\|\Delta v_h\|_{0,E} + \sum_{e\subset\partial E}\|\Pi_{k-1}^{0,e}v_h\|_{0,e}\left\|\nabla v_h\cdot\boldsymbol{n}^E\right\|_{0,e} \\ &\lesssim \left(h_E^{-1}\sum_{e\subset\partial E}\|\Pi_{k-1}^{0,e}v_h\|_{0,e}^2\right)^{\frac{1}{2}}|v_h|_{1,E}.\end{aligned} \tag{4.33}$$

It follows that

$$|v_h|_{1,E}^2 \lesssim h_E^{-1}\sum_{e\subset\partial E}\|\Pi_{k-1}^{0,e}v_h\|_{0,e}^2. \tag{4.34}$$

Hence, from (4.32) and (4.34), we get

$$\|(I-\Pi_k^{0,E})v_h\|_{0,E}^2 \lesssim h_E\sum_{e\subset\partial E}\|\Pi_{k-1}^{0,e}v_h\|_{0,e}^2\,. \tag{4.35}$$

Collecting (4.30), (4.31), and (4.35) concludes the proof. □

We now construct an Oswald-type interpolation operator π that maps piecewise (sufficiently) smooth functions into the nonconforming space $V_h^{k,nc}(\Omega_h)$. Recall the definitions (4.6) and (4.5) of the local DoFs. To define the global DoFs, we thus average the local DoFs at inter-element boundaries. More precisely, the

$$\pi v = \sum_{e \in \mathcal{E}_h} \sum_{|\boldsymbol{\ell}| \leq k-1} \mu_e^{\boldsymbol{\ell}}(v) \varphi_e^{\boldsymbol{\ell}} + \sum_{E \in \Omega_h} \sum_{|\boldsymbol{\alpha}| \leq k-2} \mu_E^{\boldsymbol{\alpha}}(v) \varphi_E^{\boldsymbol{\alpha}} \,, \tag{4.36}$$

where $\{\varphi_e^{\boldsymbol{\ell}}\}$ is the set of basis functions associated to the DoFs at the mesh skeleton $\mathcal{E}_h$, and $\{\varphi_E^{\boldsymbol{\alpha}}\}$ is the set of basis functions associated to the interior DoFs. For an interior facet $e \in \mathcal{E}_h^o$, $e \subset \partial E \cap \partial K$, the coefficient $\mu_e^{\boldsymbol{\ell}}(v)$ is defined as

$$\mu_e^{\boldsymbol{\ell}}(v) := \frac{1}{2} \left(\mu_{E,e}^{\boldsymbol{\ell}}(v) + \mu_{K,e}^{\boldsymbol{\ell}}(v) \right) , \tag{4.37}$$

while for a boundary edge $e \in \mathcal{E}_h^\partial$, it is simply defined as

$$\mu_e^{\boldsymbol{\ell}}(v) := \mu_{E,e}^{\boldsymbol{\ell}}(v). \tag{4.38}$$

Therefore, considering (4.3) and (4.5), on any edge e we get

$$\Pi_{k-1}^{0,e}(\pi v) = \{\Pi_{k-1}^{0,e} v\} = \Pi_{k-1}^{0,e}(\{v\}). \tag{4.39}$$

The main result of this section is the following proposition, that is the analogous of Proposition 3.11.

Proposition 4.5. *Let $p \in \mathbb{P}_k(\Omega_h)$ be a (discontinuous) piecewise polynomial, and let $\pi : \mathbb{P}_k(\Omega_h) \to V_h^{k,nc}(\Omega_h)$ denotes the Oswald's interpolant. Under assumption **(A-NC)**, we have that*

$$\|(I - \pi)\, p\|_{0,E}^2 \lesssim h_E \sum_{e \subset \partial E \setminus \Gamma} \|[\![p]\!]\|_{0,e}^2 \qquad \forall E \in \Omega_h \,, \quad \forall p \in \mathbb{P}_k(\Omega_h) \,.$$

Proof. We introduce the difference

$$d := (I - \pi) p \,.$$

We restrict our attention to an element $E \in \Omega_h$, and consider $d^E := d|_E$. From (4.39) we get

$$\Pi_{k-1}^{0,e} d^E = \begin{cases} \dfrac{1}{2} \Pi_{k-1}^{0,e}([\![p]\!] \cdot \boldsymbol{n}^E) & \text{if } e \not\subset \Gamma, \\ 0 & \text{if } e \subset \Gamma. \end{cases} \tag{4.40}$$

We now observe that the interior DoFs of d^E are equal to zero, so that $\Pi_{k-2}^{0,E} d^E = 0$, see (4.6). Hence, from Lemma 4.4 and (4.40), we obtain

$$\begin{aligned} \|d^E\|_{0,E} &\lesssim \left(h_E \sum_{e \subset \partial E \setminus \Gamma} \|\Pi_{k-1}^{0,e} d^E\|_{0,e}^2 \right)^{\frac{1}{2}} \lesssim \left(h_E \sum_{e \subset \partial E \setminus \Gamma} \|\Pi_{k-1}^{0,e}([\![p]\!] \cdot \boldsymbol{n}^e)\|_{0,e}^2 \right)^{\frac{1}{2}} \\ &\lesssim \left(h_E \sum_{e \subset \partial E \setminus \Gamma} \|[\![p]\!] \cdot \boldsymbol{n}^e\|_{0,e}^2 \right)^{\frac{1}{2}} = \left(h_E \sum_{e \subset \partial E \setminus \Gamma} \|[\![p]\!]\|_{0,e}^2 \right)^{\frac{1}{2}} . \end{aligned} \tag{4.41}$$

□

Lemma 4.6. *Under assumption **(A1)**, for every $E \in \Omega_h$, we have*

$$\|\pi p\|_{0,E} \lesssim \|p\|_{0,\mathcal{D}(E)} \quad \textit{for all}\, p \in \mathbb{P}_k(\Omega_h) \,,$$

where $\mathcal{D}(E) := \bigcup \{K \in \Omega_h \quad s.t. \quad |\partial E \cap \partial K| > 0\}$.

Proof. As in Lemma 3.13, the proof easily follows from the triangle inequality $\|\pi p\|_{0,E} \leq \|\pi p - p\|_{0,E} + \|p\|_{0,E}$, together with Proposition 4.5 and a polynomial trace inequality. □

Remark 4.7. *We remark that Proposition 4.5 and Lemma 4.6 are actually valid also for $v \in L^2(\Omega)$ such that $v_{|E} \in V_h^{k,nc}(E)$ for every $E \in \Omega_h$. In particular, in this case, the proof of Lemma 4.6 requires the application of the Agmon inequality and Proposition 4.3, instead of the polynomial trace inequality.*

4.2.2 Inf-Sup condition

The goal of this section is to prove the following inf-sup condition for the discrete problem (4.18), namely

$$\|v_h\|_{\mathrm{cip}} \lesssim \sup_{z_h \in V_h^{k,nc}(\Omega_h)} \frac{\mathcal{A}_{\mathrm{cip}}(v_h, z_h)}{\|z_h\|_{\mathrm{cip}}} \qquad \forall v_h \in V_h^{k,nc}(\Omega_h)\,, \tag{4.42}$$

where the norm $\|\cdot\|_{\mathrm{cip}}$ in $V_h^{k,nc}(\Omega_h)$ is defined by

$$\|v_h\|^2_{\mathrm{cip}} := \sum_{E\in\Omega_h} \|v_h\|^2_{\mathrm{cip},E}\,, \tag{4.43}$$

with

$$\begin{aligned}\|v_h\|^2_{\mathrm{cip},E} :=&\epsilon\,\|\nabla v_h\|^2_{0,E} + h\,\|\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E} v_h\|^2_{0,E} + \sigma\,\|v_h\|^2_{0,E}\\ &+\frac{\epsilon}{\delta h}\sum_{e\subset\Gamma_E}\|\Pi_{k-1}^{0,e} v_h\|^2_{0,e} + \||\boldsymbol{\beta}\cdot\boldsymbol{n}^E|^{\frac12}\Pi_k^{0,E} v_h\|^2_{0,\Gamma_{E,\mathrm{in}}} + J_h^E(v_h,v_h)\,.\end{aligned} \tag{4.44}$$

Similarly to the previous chapter, we divide the proof of (4.42) in two parts. In the first part (Lemma 4.8), we estimate the diffusion, reaction, and inflow boundary terms in $\|v_h\|^2_{\mathrm{cip}}$ with $\mathcal{A}_{\mathrm{cip}}(v_h,v_h)$. In the second part (Lemma 4.9), we estimate the convective term in $\|v_h\|^2_{\mathrm{cip}}$. In order to do so, we would like to take $\widetilde{w}_h$ locally defined by

$$\widetilde{w}_h|_E := h\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E} v_h$$

as the second argument in $\mathcal{A}_{\mathrm{cip}}(\cdot,\cdot)$. This is not possible, since $\widetilde{w}_h \notin V_h^{k,nc}(\Omega_h)$, and we take its Oswald interpolant $\pi\widetilde{w}_h$ instead. Then, the difference between $\widetilde{w}_h$ and $\pi\widetilde{w}_h$ is controlled thanks to the jump term. We combine these two results and conclude the proof of the inf-sup condition (4.42) in Theorem 4.11.

Lemma 4.8. *Under assumptions **(A1)**, given $v_h \in V_h^{k,nc}(\Omega_h)$, we have*

$$\begin{aligned}\mathcal{A}_{\mathrm{cip}}(v_h,v_h) \gtrsim&\epsilon\,\|\nabla v_h\|^2_{0,\Omega_h} + \sigma\,\|v_h\|^2_{0,\Omega_h} + \sum_{E\in\Omega_h}\frac{\epsilon}{\delta h}\sum_{e\subset\Gamma_E}\|\Pi_{k-1}^{0,e} v_h\|^2_{0,e}\\ &+\sum_{E\in\Omega_h}\||\boldsymbol{\beta}\cdot\boldsymbol{n}^E|^{\frac12}\Pi_k^{0,E} v_h\|^2_{0,\Gamma_{E,\mathrm{in}}} + J_h(v_h,v_h)\,.\end{aligned}$$

Proof. We take the same v_h in both arguments of $\mathcal{A}_{\mathrm{cip}}(\cdot,\cdot)$ and obtain

$$\mathcal{A}_{\mathrm{cip}}(v_h,v_h) = \epsilon\, a_h(v_h,v_h) + b_h(v_h,v_h) + \sigma\, c_h(v_h,v_h) + d_h(v_h,v_h) + \mathcal{N}_h(v_h,v_h) + J_h(v_h,v_h)\,.$$

The diffusion term is easily estimated using the orthogonality of the projectors and property (1.8) of the stabilization form:

$$\epsilon a_h(v_h,v_h) \geq \widetilde{\alpha}_*\epsilon\,\|\nabla v_h\|^2_{0,\Omega_h},$$

where $\widetilde{\alpha}_* = \min\{1,\alpha_*\}$. For the reaction term, using (1.8) and the Poincaré inequality for $(I-\Pi_k^{0,E})v_h$, we have

$$\sigma c_h(v_h,v_h) \geq C_0\widetilde{\alpha}_*\sigma\|v_h\|^2_{0,\Omega_h}.$$

for a positive constant C_0 independent of σ and h. Therefore, we get

$$\epsilon a_h(v_h,v_h) + \sigma c_h(v_h,v_h) \geq \widetilde{\alpha}_*\epsilon\,\|\nabla v_h\|^2_{0,,\Omega_h} + C_0\widetilde{\alpha}_*\sigma\|v_h\|^2_{0,\Omega_h}\,. \tag{4.45}$$

For the convective term, by integrating by parts, we obtain

$$\begin{aligned}b_h(v_h,v_h) =&\frac12\int_\Omega(\boldsymbol{\beta}\cdot\nabla\Pi_k^0 v_h)\Pi_k^0 v_h\mathrm{d}E - \frac12\int_\Omega\Pi_k^0 v_h(\boldsymbol{\beta}\cdot\nabla\Pi_k^0 v_h)\mathrm{d}E\\ &+\frac12\sum_{E\in\Omega_h}\int_{\partial E}(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)\Pi_k^{0,E} v_h\Pi_k^{0,E} v_h\mathrm{d}s\\ =&\frac12\sum_{E\in\Omega_h}\int_{\Gamma_E}(\boldsymbol{\beta}\cdot\boldsymbol{n}^E)\Pi_k^{0,E} v_h\Pi_k^{0,E} v_h\mathrm{d}s - d_h(v_h,v_h),\end{aligned}$$

where the last step follows from simple algebraic manipulations and from the definition of $d_h(\cdot,\cdot)$. Combining the terms on Γ_{in} in the expression above with those in $\mathcal{N}_h(\cdot,\cdot)$ gives

$$\begin{aligned} b_h(v_h,v_h)+d_h(v_h,v_h)+\mathcal{N}_h(v_h,v_h) &= \frac{1}{2}\sum_{E\in\Omega_h} \| |\boldsymbol{\beta}\cdot\boldsymbol{n}^E|^{\frac{1}{2}}\Pi_k^{0,E}v_h\|^2_{0,\Gamma_E} \\ &+\sum_{E\in\Omega_h}\frac{\epsilon}{\delta h_E}\sum_{e\subset\Gamma_E}\|\Pi_{k-1}^{0,e}v_h\|^2_{0,e} - 2\epsilon\sum_{E\in\Omega_h}\langle \boldsymbol{\Pi}_{k-1}^{0,E}\nabla v_h\cdot\boldsymbol{n}^E, v_h\rangle_{\Gamma_E}\,. \end{aligned}$$

For the last term on the right-hand side, we use the estimate

$$\begin{aligned} 2\epsilon\langle \boldsymbol{\Pi}_{k-1}^{0,E}\nabla v_h\cdot\boldsymbol{n}^E, v_h\rangle_{\Gamma_E} &= 2\epsilon\sum_{e\subset\Gamma_E}\langle \boldsymbol{\Pi}_{k-1}^{0,E}\nabla v_h\cdot\boldsymbol{n}^E, \Pi_{k-1}^{0,e}v_h\rangle_e \\ &\leq 2\epsilon\delta h_E\|\boldsymbol{\Pi}_{k-1}^{0,E}\nabla v_h\cdot\boldsymbol{n}^E\|^2_{\Gamma_E}+\frac{\epsilon}{2\delta h_E}\sum_{e\subset\Gamma_E}\|\Pi_{k-1}^{0,e}v_h\|^2_{0,e} \\ &\leq 2\epsilon\delta C_{\text{tr}}\|\boldsymbol{\Pi}_{k-1}^{0,E}\nabla v_h\|^2_{0,E}+\frac{\epsilon}{2\delta h_E}\sum_{e\subset\Gamma_E}\|\Pi_{k-1}^{0,e}v_h\|^2_{0,e} \\ &\leq 2\epsilon\delta C_{\text{tr}}\|\nabla v_h\|^2_{0,E}+\frac{\epsilon}{2\delta h_E}\sum_{e\subset\Gamma_E}\|\Pi_{k-1}^{0,e}v_h\|^2_{0,e}\,, \end{aligned}$$

where C_{tr} is the inverse trace inequality constant for polynomials. Therefore, with the choice, e.g., $\delta=\widetilde{\alpha}_*/(4C_{\text{tr}})$, we obtain

$$\begin{aligned} b_h(v_h,v_h)+d_h(v_h,v_h)+\mathcal{N}_h(v_h,v_h) \geq& \frac{1}{2}\sum_{E\in\Omega_h}\| |\boldsymbol{\beta}\cdot\boldsymbol{n}^E|^{\frac{1}{2}}\Pi_k^{0,E}v_h\|^2_{0,\Gamma_E} \\ &+\frac{1}{2}\sum_{E\in\Omega_h}\frac{\epsilon}{\delta h_E}\sum_{e\subset\Gamma_E}\|\Pi_{k-1}^{0,e}v_h\|^2_{0,e}-\frac{\widetilde{\alpha}_*}{2}\epsilon\|\nabla v_h\|^2_{0,\Omega_h}\,. \end{aligned}$$

This, together with (4.45), and noting that

$$\sum_{E\in\Omega_h}\| |\boldsymbol{\beta}\cdot\boldsymbol{n}^E|^{\frac{1}{2}}\Pi_k^{0,E}v_h\|^2_{0,\Gamma_E} \geq \sum_{E\in\Omega_h}\| |\boldsymbol{\beta}\cdot\boldsymbol{n}^E|^{\frac{1}{2}}\Pi_k^{0,E}v_h\|^2_{0,\Gamma_{E,\text{in}}},$$

gives the result. □

Lemma 4.9. *Given $v_h\in V_h^{k,nc}(\Omega_h)$, let us define the function*

$$w_h := h\pi(\boldsymbol{\beta}_h\cdot\nabla\Pi_k^0 v_h)\,, \tag{4.46}$$

*where $\boldsymbol{\beta}_h$ is the L^2-projection of $\boldsymbol{\beta}$ in the space of piecewise linear functions $[\mathbb{P}_1(\Omega_h)]^d$. Then, under assumptions **(A-NC)**, if the mesh size satisfies $h>\varepsilon$, we have*

$$\mathcal{A}_{\text{cip}}(v_h,w_h)\geq C_1\,h\|\boldsymbol{\beta}\cdot\nabla\Pi_k^0 v_h\|^2_{0,\Omega_h}-C_2\,\mathcal{A}_{\text{cip}}(v_h,v_h)\,, \tag{4.47}$$

for $C_1, C_2>0$ independent of ϵ and h.

Proof. From Lemma 4.6, w_h defined in (4.46) satisfies the following estimate, which will be used throughout the rest of the proof:

$$\|w_h\|_{0,E}\lesssim h\|\boldsymbol{\beta}_h\cdot\nabla\Pi_k^0 v_h\|_{0,\mathcal{D}(E)}\,. \tag{4.48}$$

We proceed element by element. By definition of the local bilinear form $\mathcal{A}^E_{\text{cip}}(\cdot,\cdot)$, we have that

$$\begin{aligned} \mathcal{A}^E_{\text{cip}}(v_h,w_h) &= \epsilon\, a_h^E(v_h,w_h)+J_h^E(v_h,w_h)+\sigma\, c_h^E(v_h,w_h) \\ &\quad+\mathcal{N}_h^E(v_h,w_h)+b_h^E(v_h,w_h)+d_h^E(v_h,w_h) \\ &=: T_1+T_2+T_3+T_4+T_5+T_6\,. \end{aligned} \tag{4.49}$$

We estimate each of these six terms separately.
Estimate of $(\mathbf{T_1})$. Using the Cauchy-Schwarz inequality, the properties of $a_h(\cdot,\cdot)$, the inverse inequality for virtual functions, estimate (4.48), and recalling that $\epsilon < h$, we get

$$\begin{aligned} T_1 = \epsilon\, a_h^E(v_h, w_h) &\geq -\epsilon\, a_h^E(v_h, v_h)^{\frac{1}{2}}\, a_h^E(w_h, w_h)^{\frac{1}{2}} \\ &\gtrsim -\epsilon^{\frac{1}{2}}\, \|\nabla v_h\|_{0,E}\, \epsilon^{\frac{1}{2}}\, \|\nabla w_h\|_{0,E} \\ &\gtrsim -\epsilon^{\frac{1}{2}}\, \|\nabla v_h\|_{0,E}\, \epsilon^{\frac{1}{2}} h^{-1} \|w_h\|_{0,E} \\ &\gtrsim -\epsilon^{\frac{1}{2}}\, \|\nabla v_h\|_{0,E}\, h^{\frac{1}{2}}\, \|\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h\|_{0,\mathcal{D}(E)}\,. \end{aligned} \tag{4.50}$$

Estimate of $(\mathbf{T_2})$. We first split $J_h^E(\cdot,\cdot)$ using the Cauchy-Schwarz inequality

$$T_2 = J_h^E(v_h, w_h) \geq -J_h^E(v_h, v_h)^{\frac{1}{2}}\, J_h^E(w_h, w_h)^{\frac{1}{2}}\,.$$

Using inverse and inverse trace inequalities for polynomials, the inverse inequality for virtual functions, the properties of the L^2 projection, and (4.48), we get

$$\begin{aligned} J_h^E(w_h, w_h) &= \frac{1}{2} \sum_{e \subset \partial E \setminus \Gamma} \gamma_e \int_e h_e^2\, |[\![\nabla \Pi_k^0 w_h]\!]|^2\, \mathrm{d}s + \gamma_E\, h_E\, \mathcal{S}_J^E\big((I - \Pi_k^{\nabla,E}) w_h, (I - \Pi_k^{\nabla,E}) w_h\big) \\ &\lesssim h\, \|\nabla \Pi_k^0 w_h\|_{0,\mathcal{D}(E)}^2 + h\, \|\nabla w_h\|_{0,E}^2 \\ &\lesssim h^{-1}\, \|\Pi_k^0 w_h\|_{0,\mathcal{D}(E)}^2 + h^{-1} \|w_h\|_{0,E}^2 \\ &\lesssim h^{-1} \|w_h\|_{0,\mathcal{D}(E)}^2 \lesssim h \|\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h\|_{0,\mathcal{D}(\mathcal{D}(E))}^2\,, \end{aligned}$$

from which we conclude

$$T_2 \gtrsim -J_h^E(v_h, v_h)^{\frac{1}{2}} h^{\frac{1}{2}}\, \|\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h\|_{0,\mathcal{D}(\mathcal{D}(E))}\,. \tag{4.51}$$

Estimate of $(\mathbf{T_3})$. From the properties of $c_h^E(\cdot,\cdot)$ and (4.48), we get

$$\begin{aligned} T_3 = \sigma c_h^E(v_h, w_h) &\gtrsim -\sigma \|v_h\|_{0,E}\, \|w_h\|_{0,E} \\ &\gtrsim -\|v_h\|_{0,E}\, h^{\frac{1}{2}}\, \|\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h\|_{0,\mathcal{D}(E)}\,. \end{aligned} \tag{4.52}$$

where we used $h^{\frac{1}{2}} \lesssim 1$ to simplify later developments.
Estimate of $(\mathbf{T_4})$. For the Nitsche term, we have to control four different terms:

$$\begin{aligned} T_4 = \mathcal{N}_h^E(v_h, w_h) &= -\epsilon\, \langle \boldsymbol{\Pi}_{k-1}^{0,E} \nabla v_h \cdot \boldsymbol{n}^E, w_h \rangle_{\Gamma_E} - \epsilon\, \langle v_h, \boldsymbol{\Pi}_{k-1}^{0,E} \nabla w_h \cdot \boldsymbol{n}^E \rangle_{\Gamma_E} \\ &\quad + \frac{\epsilon}{\delta h_E} \sum_{e \subset \Gamma_E} \langle \Pi_{k-1}^{0,e} v_h, \Pi_{k-1}^{0,e} w_h \rangle_e + \langle |\boldsymbol{\beta} \cdot \boldsymbol{n}^E| \Pi_k^{0,E} v_h, \Pi_k^{0,E} w_h \rangle_{\Gamma_{E,\mathrm{in}}} \\ &=: \eta_{\mathcal{N}_1} + \eta_{\mathcal{N}_2} + \eta_{\mathcal{N}_3} + \eta_{\mathcal{N}_4}\,. \end{aligned}$$

We remark that $\mathcal{N}_h^E(\cdot,\cdot)$ is different from zero only if E has at least one facet on Γ.
For $\eta_{\mathcal{N}_1}$, we use Chauchy-Schwarz inequality, the inverse trace inequality for polynomials and for virtual functions, esitmate (4.48), and the assumption $\epsilon < h$, and derive

$$\begin{aligned} \eta_{\mathcal{N}_1} &\gtrsim -\epsilon \|\boldsymbol{\Pi}_{k-1}^{0,E} \nabla v_h\|_{0,\Gamma_E} \|w_h\|_{0,\Gamma_E} \\ &\gtrsim -\epsilon\, h^{-1} \|\boldsymbol{\Pi}_{k-1}^{0,E} \nabla v_h\|_{0,E} \|w_h\|_{0,E} \\ &\gtrsim -\epsilon^{\frac{1}{2}}\, \|\nabla v_h\|_{0,E}\, h^{\frac{1}{2}}\, \|\boldsymbol{\beta}_h \cdot \nabla \Pi_k^0 v_h\|_{0,\mathcal{D}(E)}\,. \end{aligned} \tag{4.53}$$

For $\eta_{\mathcal{N}_2}$, we use the orthogonality of $\Pi_{k-1}^{0,e}$, the Cauchy-Schwarz inequality, inverse trace and inverse inequalities, and $\epsilon < h$, to obtain

$$\begin{aligned}\eta_{\mathcal{N}_2} &= -\epsilon \sum_{e\subset\Gamma_E} \langle \Pi^{0,e}_{k-1} v_h, \boldsymbol{\Pi}^{0,E}_{k-1}\nabla w_h \cdot \boldsymbol{n}^E\rangle_e \\ &\gtrsim -\epsilon \sum_{e\subset\Gamma_E} \|\Pi^{0,e}_{k-1} v_h\|_{0,e} \|\boldsymbol{\Pi}^{0,E}_{k-1}\nabla w_h\|_{0,\Gamma_E} \\ &\gtrsim -\epsilon\, h^{-\frac{1}{2}} \sum_{e\subset\Gamma_E} \|\Pi^{0,e}_{k-1} v_h\|_{0,e} \|\boldsymbol{\Pi}^{0,E}_{k-1}\nabla w_h\|_{0,E} \\ &\gtrsim -\epsilon\, h^{-\frac{3}{2}} \sum_{e\subset\Gamma_E} \|\Pi^{0,e}_{k-1} v_h\|_{0,e} \|w_h\|_{0,E} \\ &\gtrsim -\left(\frac{\epsilon}{\delta h}\right)^{\frac{1}{2}} \sum_{e\subset\Gamma_E} \|\Pi^{0,e}_{k-1} v_h\|_{0,e}\, h^{\frac{1}{2}} \|\boldsymbol{\beta}_h \cdot \nabla\Pi^0_k v_h\|_{0,\mathcal{D}(E)} \,.\end{aligned} \tag{4.54}$$

For $\eta_{\mathcal{N}_3}$, thanks to orthogonality, we remove the $\Pi^{0,e}_{k-1}$ projection on the second term and we proceed similarly to the previous cases:

$$\begin{aligned}\eta_{\mathcal{N}_3} &= \frac{\epsilon}{\delta h} \sum_{e\subset\Gamma_E} \langle \Pi^{0,e}_{k-1} v_h,\ w_h\rangle_e \gtrsim -\frac{\epsilon}{\delta h} \sum_{e\subset\Gamma_E} \|\Pi^{0,e}_{k-1} v_h\|_{0,e} \|w_h\|_{0,\Gamma_E} \\ &\gtrsim -h^{-1}\left(\frac{\epsilon}{\delta h}\right)^{\frac{1}{2}} \sum_{e\subset\Gamma_E} \|\Pi^{0,e}_{k-1} v_h\|_{0,e} \|w_h\|_{0,E} \\ &\gtrsim -\left(\frac{\epsilon}{\delta h}\right)^{\frac{1}{2}} \sum_{e\subset\Gamma_E} \|\Pi^{0,e}_{k-1} v_h\|_{0,e}\, h^{\frac{1}{2}} \|\boldsymbol{\beta}_h \cdot \nabla\Pi^0_k v_h\|_{0,\mathcal{D}(E)} \,,\end{aligned} \tag{4.55}$$

where we have used again $h \lesssim 1$.
For $\eta_{\mathcal{N}_4}$, using $|\boldsymbol{\beta}\cdot\boldsymbol{n}^E| \leq 1$, we have that

$$\begin{aligned}\eta_{\mathcal{N}_4} &\gtrsim -\| |\boldsymbol{\beta}\cdot\boldsymbol{n}^E|^{\frac{1}{2}} \Pi^{0,E}_k v_h\|_{0,\Gamma_{E,\text{in}}} \|\Pi^{0,E}_k w_h\|_{0,\Gamma_{E,\text{in}}} \\ &\gtrsim -h^{-\frac{1}{2}} \| |\boldsymbol{\beta}\cdot\boldsymbol{n}^E|^{\frac{1}{2}} \Pi^{0,E}_k v_h\|_{0,\Gamma_{E,\text{in}}} \|\Pi^{0,E}_k w_h\|_{0,E} \\ &\gtrsim -\| |\boldsymbol{\beta}\cdot\boldsymbol{n}^E|^{\frac{1}{2}} \Pi^{0,E}_k v_h\|_{0,\Gamma_{E,\text{in}}}\, h^{\frac{1}{2}} \|\boldsymbol{\beta}_h \cdot \nabla\Pi^0_k v_h\|_{0,\mathcal{D}(E)} \,.\end{aligned} \tag{4.56}$$

Gathering (4.53)–(4.56) gives

$$\begin{aligned}T_4 \gtrsim &-\left(\epsilon^{\frac{1}{2}} \|\nabla v_h\|_{0,E} + \left(\frac{\epsilon}{\delta h}\right)^{\frac{1}{2}} \sum_{e\subset\Gamma_E} \|\Pi^{0,e}_{k-1} v_h\|_{0,e} + \| |\boldsymbol{\beta}\cdot\boldsymbol{n}^E|^{\frac{1}{2}} \Pi^{0,E}_k v_h\|_{0,\Gamma_{E,\text{in}}}\right) \\ &\cdot h^{\frac{1}{2}} \|\boldsymbol{\beta}_h \cdot \nabla\Pi^0_k v_h\|_{0,\mathcal{D}(E)} \,.\end{aligned} \tag{4.57}$$

Estimate of $(\mathbf{T_5})$. The definition of the bilinear form $b_h(\cdot,\cdot)$ implies that

$$\begin{aligned}T_5 = b^E_h(v_h, w_h) &= \big(\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h,\ \Pi^{0,E}_k w_h\big)_{0,E} \\ &= \big(\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h,\ w_h\big)_{0,E} + \big(\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h,\ (\Pi^{0,E}_k - I) w_h\big)_{0,E} \\ &= \big(\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h,\ h\boldsymbol{\beta}_h\cdot\nabla\Pi^{0,E}_k v_h\big)_{0,E} \\ &\quad + \big(\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h,\ w_h - h\boldsymbol{\beta}_h\cdot\nabla\Pi^{0,E}_k v_h\big)_{0,E} \\ &\quad + \big(\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h,\ (\Pi^{0,E}_k - I) w_h\big)_{0,E} \\ &=: \eta_{\boldsymbol{\beta}_1} + \eta_{\boldsymbol{\beta}_2} + \eta_{\boldsymbol{\beta}_3} \,.\end{aligned} \tag{4.58}$$

For $\eta_{\boldsymbol{\beta}_1}$, we add and subtract $(\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h, h\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h)_{0,E}$ and we use the Cauchy-Schwarz inequality to obtain

$$\begin{aligned}\eta_{\boldsymbol{\beta}_1} &= (\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h, h\boldsymbol{\beta}_h\cdot\nabla\Pi^0_k v_h)_{0,E} \\ &= h\,\|\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h\|^2_{0,E} + (\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h, h(\boldsymbol{\beta}_h - \boldsymbol{\beta})\cdot\nabla\Pi^{0,E}_k v_h)_{0,E} \\ &\geq h\,\|\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h\|^2_{0,E} - C\,h^{\frac{1}{2}} \|\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h\|_{0,E}\, h^{\frac{1}{2}} |\boldsymbol{\beta}|_{[W^1_\infty(E)]^d} h \|\nabla\Pi^{0,E}_k v_h\|_{0,E} \\ &\geq h\,\|\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h\|^2_{0,E} - C\,h^{\frac{1}{2}} \|\boldsymbol{\beta}\cdot\nabla\Pi^{0,E}_k v_h\|_{0,E}\, h^{\frac{1}{2}} |\boldsymbol{\beta}|_{[W^1_\infty(E)]^d} \|v_h\|_{0,E} \,,\end{aligned} \tag{4.59}$$

where in the third step we have used $\|\boldsymbol{\beta}_h - \boldsymbol{\beta}\|_{[L^\infty(E)]^d} \lesssim h|\boldsymbol{\beta}|_{[W^1_\infty(E)]^d}$.
For $\eta_{\boldsymbol{\beta}_2}$, recalling the definition of w_h and using the Young inequality to split the two terms, we get

$$\begin{aligned} \eta_{\boldsymbol{\beta}_2} &= h\big(\boldsymbol{\beta} \cdot \nabla\Pi_k^{0,E} v_h, (\pi - I)(\boldsymbol{\beta}_h \cdot \nabla\Pi_k^{0,E} v_h)\big)_{0,E} \\ &\geq -\frac{h}{2}\|\boldsymbol{\beta} \cdot \nabla\Pi_k^{0,E} v_h\|^2_{0,E} - \frac{h}{2}\|(\pi - I)(\boldsymbol{\beta}_h \cdot \nabla\Pi_k^{0,E} v_h)\|^2_{0,E} \,. \end{aligned} \tag{4.60}$$

For the second term on the right-hand side, we use the fact that $\boldsymbol{\beta}_h \cdot \nabla\Pi^0_k v_h \in \mathbb{P}_k(\Omega_h)$. Proposition 4.5 gives

$$h\|(\pi - I)(\boldsymbol{\beta}_h \cdot \nabla\Pi^0_k v_h)\|^2_{0,E} \lesssim h^2 \sum_{e\subset\partial E\setminus\Gamma} \|[\![\boldsymbol{\beta}_h \cdot \nabla\Pi^0_k v_h]\!]\|^2_{0,e} \,.$$

The triangular inequality and the definition of the jump bilinear form give

$$\begin{aligned} h^2 \sum_{e\subset\partial E\setminus\Gamma} \|[\![\boldsymbol{\beta}_h \cdot \nabla\Pi^0_k v_h]\!]\|^2_{0,e} &\lesssim h^2 \sum_{e\subset\partial E\setminus\Gamma} \|[\![(\boldsymbol{\beta}_h - \boldsymbol{\beta}) \cdot \nabla\Pi^0_k v_h]\!]\|^2_{0,e} + h^2 \sum_{e\subset\partial E\setminus\Gamma} \|[\![\boldsymbol{\beta} \cdot \nabla\Pi^0_k v_h]\!]\|^2_{0,e} \\ &\lesssim h^2 \sum_{e\subset\partial E\setminus\Gamma} \|[\![(\boldsymbol{\beta}_h - \boldsymbol{\beta}) \cdot \nabla\Pi^0_k v_h]\!]\|^2_{0,e} + h^2 \sum_{e\subset\partial E\setminus\Gamma} \gamma_e^2\|[\![\nabla\Pi^0_k v_h]\!]\|^2_{0,e} \\ &\lesssim h^2 \sum_{e\subset\partial E\setminus\Gamma} \|[\![(\boldsymbol{\beta}_h - \boldsymbol{\beta}) \cdot \nabla\Pi^0_k v_h]\!]\|^2_{0,e} + \sum_{E'\subset\mathcal{D}(E)} J_h^{E'}(v_h, v_h) \,, \end{aligned}$$

where we used that $\gamma_e^2 \leq \gamma_e$ (since $\gamma_e \leq 1$, see (3.3)). On each e, the argument in the first sum on the right-hand side of the previous inequality is controlled using the trace inequality and standard estimates on $\boldsymbol{\beta} \in [W^1_\infty(\Omega)]^d$:

$$\begin{aligned} h^2\|[\![(\boldsymbol{\beta}_h - \boldsymbol{\beta}) \cdot \nabla\Pi^0_k v_h]\!]\|^2_{0,e} &\lesssim h^4|\boldsymbol{\beta}|^2_{[W^1_\infty(E\cup K)]^d} h^{-1}\|\nabla\Pi^0_k v_h\|^2_{0,E\cup E'} \\ &\lesssim h|\boldsymbol{\beta}|^2_{[W^1_\infty(E\cup K)]^d}\|\Pi^0_k v_h\|^2_{0,E\cup E'} \\ &\lesssim h|\boldsymbol{\beta}|^2_{[W^1_\infty(E\cup K)]^d}\|v_h\|^2_{0,E\cup E'} \,, \end{aligned} \tag{4.61}$$

where E and E' are the two elements sharing the edge e. Combining (4.60) with (4.61), we obtain for $\eta_{\boldsymbol{\beta}_2}$

$$\begin{aligned} \eta_{\boldsymbol{\beta}_2} \geq &-\frac{h}{2}\|\boldsymbol{\beta} \cdot \nabla\Pi_k^{0,E} v_h\|^2_{0,E} \\ &- C\Big(h|\boldsymbol{\beta}|^2_{[W^1_\infty(\mathcal{D}(E))]^d}\|v_h\|^2_{0,\mathcal{D}(E)} + \sum_{K\subset\mathcal{D}(E)} J_h^K(v_h, v_h)\Big) \,. \end{aligned} \tag{4.62}$$

It remains to control $\eta_{\boldsymbol{\beta}_3}$. Since $\boldsymbol{\beta}_h \in [\mathbb{P}_1(E)]^d$, it holds $\big(\boldsymbol{\beta}_h \cdot \nabla\Pi_k^{0,E} v_h, (\Pi_k^{0,E} - I)w_h\big)_{0,E} = 0$. Hence we have

$$\begin{aligned} \eta_{\boldsymbol{\beta}_3} &= \big((\boldsymbol{\beta} - \boldsymbol{\beta}_h) \cdot \nabla\Pi_k^{0,E} v_h, (\Pi_k^{0,E} - I)w_h\big)_{0,E} \\ &\gtrsim -\|(\boldsymbol{\beta} - \boldsymbol{\beta}_h) \cdot \nabla\Pi_k^{0,E} v_h\|_{0,E} \, \|h\pi(\boldsymbol{\beta}_h \cdot \nabla\Pi_k^{0,E} v_h)\|_{0,E} \\ &\gtrsim -|\boldsymbol{\beta}|_{[W^1_\infty(E)]^d} h\|\nabla\Pi_k^{0,E} v_h\|_{0,E} \, h\|\boldsymbol{\beta}_h \cdot \nabla\Pi^0_k v_h\|_{0,\mathcal{D}(E)} \\ &\gtrsim -|\boldsymbol{\beta}|_{[W^1_\infty(E)]^d}\|v_h\|^2_{0.\mathcal{D}(E)} \,. \end{aligned} \tag{4.63}$$

Collecting (4.59), (4.62) and (4.63), from (4.58) we get

$$\begin{aligned} T_5 \geq \frac{h}{2}\, &\|\boldsymbol{\beta} \cdot \nabla\Pi_k^{0,E} v_h\|^2_{0,E} - C\Big(\sum_{E'\subset\mathcal{D}(E)} J_h^{E'}(v_h, v_h) \\ &+ h^{\frac{1}{2}}\|\boldsymbol{\beta} \cdot \nabla\Pi_k^{0,E} v_h\|_{0,E}\, h^{\frac{1}{2}}|\boldsymbol{\beta}|_{[W^1_\infty(E)]^d}\|v_h\|_{0,E} \\ &+ h|\boldsymbol{\beta}|^2_{[W^1_\infty(\mathcal{D}(E))]^d}\|v_h\|^2_{0,\mathcal{D}(E)} + |\boldsymbol{\beta}|_{[W^1_\infty(E)]^d}\|v_h\|^2_{0,\mathcal{D}(E)}\Big) \,. \end{aligned} \tag{4.64}$$

Estimate of $(\mathbf{T_6})$. The last term that we have to estimate is related to $d_h(\cdot,\cdot)$. We use the Cauchy-Schwarz and the trace inequality for polynomials:

$$
\begin{aligned}
T_6 = d_h^E(v_h, w_h) &= -\frac{1}{2} \sum_{e\subset\partial E\setminus\Gamma} \int_e \boldsymbol{\beta}\cdot [\![\Pi_k^0 v_h]\!]\{\Pi_k^0 w_h\}\mathrm{d}s \\
&\gtrsim - \sum_{e\subset\partial E\setminus\Gamma} \|\boldsymbol{\beta}\cdot [\![\Pi_k^0 v_h]\!]\|_{0,e}\|\{\Pi_k^0 w_h\}\|_{0,e} \\
&\gtrsim - \sum_{e\subset\partial E\setminus\Gamma} \Big(\||\boldsymbol{\beta}\cdot \boldsymbol{n}^E|^{\frac12}\Pi_k^{0,E} v_h\|_{0,e} + \||\boldsymbol{\beta}\cdot \boldsymbol{n}^K|^{\frac12}\Pi_k^{0,K} v_h\|_{0,e}\Big)\, h^{-\frac12}\|w_h\|_{0,E\cup K} \\
&\gtrsim - \sum_{e\subset\partial E\setminus\Gamma} \Big(\||\boldsymbol{\beta}\cdot \boldsymbol{n}^E|^{\frac12}\Pi_k^{0,E} v_h\|_{0,e} + \||\boldsymbol{\beta}\cdot \boldsymbol{n}^K|^{\frac12}\Pi_k^{0,K} v_h\|_{0,e}\Big)\, h^{\frac12}\|\boldsymbol{\beta}_h\cdot\nabla\Pi_k^0 v_h\|_{0,\mathcal{D}(E\cup K)}\,.
\end{aligned}
$$

By collecting the estimates of all six terms and adding over all elements, we obtain

$$
\begin{aligned}
\mathcal{A}_{\mathrm{cip}}(v_h, w_h) \geq \frac{h}{2}\|\boldsymbol{\beta}\cdot\nabla\Pi_k^0 v_h\|^2_{0,\Omega_h} - C\Big(&\sum_{E\in\Omega_h} \big(\epsilon^{\frac12}\|\nabla v_h\|_{0,E} + J_h^E(v_h,v_h)^{\frac12} + \|v_h\|_{0,E} \\
&+ \Big(\frac{\epsilon}{\delta h}\Big)^{\frac12} \sum_{e\subset\Gamma_E} \|\Pi_{k-1}^{0,e} v_h\|_{0,e} + \||\boldsymbol{\beta}\cdot\boldsymbol{n}^E|^{\frac12}\Pi_k^{0,E} v_h\|_{0,\Gamma_{E,\mathrm{in}}}\big) h^{\frac12}\|\boldsymbol{\beta}_h\cdot\nabla\Pi_k^{0,E} v_h\|_{0,E} \\
&+ J_h(v_h,v_h) + \sum_{E\in\Omega_h} \Big(h|\boldsymbol{\beta}|^2_{[W^1_\infty(E)]^d} + |\boldsymbol{\beta}|_{[W^1_\infty(E)]^d}\Big)\|v_h\|^2_{0,E} \\
&+ \sum_{E\in\Omega_h} h^{\frac12}\|\boldsymbol{\beta}_h\cdot\nabla\Pi_k^{0,E} v_h\|_{0,E}\, h^{\frac12}|\boldsymbol{\beta}|_{[W^1_\infty(E)]^d}\|v_h\|_{0,E}\Big)\,.
\end{aligned}
\tag{4.65}
$$

Above, we have also used the property that, due to assumption **(A1)**, summing over the elements, each element is counted only a uniformly bounded number of times, even when the terms involve norms on $\mathcal{D}(E)$ or $\mathcal{D}(\mathcal{D}(E))$.

We now notice that the triangular inequality, standard approximation results, and an inverse estimate give

$$
h^{\frac12}\|\boldsymbol{\beta}_h\cdot\nabla\Pi_k^{0,E} v_h\|_{0,E} \lesssim h^{\frac12}\Big(\|\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E} v_h\|_{0,E} + |\boldsymbol{\beta}|_{[W^1_\infty(E)]^d}\|v_h\|_{0,E}\Big)\,. \tag{4.66}
$$

Hence, from (4.65), using (4.66) and the Young inequality (with suitable constants) for the first and the last summations on the right-hand side, we get

$$
\begin{aligned}
\mathcal{A}_{\mathrm{cip}}(v_h, w_h) \geq & C_1 h\|\boldsymbol{\beta}\cdot\nabla\Pi_k^0 v_h\|^2_{0,\Omega_h} - C_2\Big(\epsilon\|\nabla v_h\|^2_{0,\Omega_h} + \sum_{E\in\Omega_h}\frac{\epsilon}{\delta h_E}\sum_{e\subset\Gamma_E}\|\Pi_{k-1}^{0,e} v_h\|^2_{0,e} \\
&+ \sigma\|v_h\|^2_{0,\Omega_h} + \sum_{E\in\Omega_h}\||\boldsymbol{\beta}\cdot\boldsymbol{n}^E|^{\frac12}\Pi_k^{0,E} v_h\|^2_{0,\Gamma_{E,\mathrm{in}}} + J_h(v_h,v_h)\Big)\,.
\end{aligned}
$$

From Lemma 4.8, we now obtain

$$
\mathcal{A}_{\mathrm{cip}}(v_h, w_h) \geq C_1\, h\|\boldsymbol{\beta}\cdot\nabla\Pi_k^0 v_h\|^2_{0,\Omega_h} - C_2\ \mathcal{A}_{\mathrm{cip}}(v_h, v_h)\,.
$$

□

Whenever $h > \varepsilon$, we also need the following estimate.

Lemma 4.10. *Under assumptions **(A-NC)**, assume that $h > \varepsilon$. For any $v_h \in V_h^{k,nc}(\Omega_h)$, let w_h be defined as in* (4.46). *Then,*

$$
\|w_h\|_{\mathrm{cip}} \lesssim \|v_h\|_{\mathrm{cip}}\,,
$$

with hidden constant independent of ϵ, h, and v_h.

Proof. From the definition of w_h and $h > \varepsilon$, for the gradient term in the $\|\cdot\|_{\text{cip}}$ norm, we have

$$\epsilon\|\nabla w_h\|_{0,E}^2 \lesssim \epsilon h^{-2}\|w_h\|_{0,E}^2 \lesssim \epsilon\|\pi(\boldsymbol{\beta}_h \cdot \nabla\Pi_k^{0,E} v_h)\|_{0,E}^2 \lesssim h\|\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h\|_{0,\mathcal{D}(E)}^2 \,. \tag{4.67}$$

For the L^2 term, we have that

$$\sigma\|w_h\|_{0,E}^2 \lesssim \sigma\|h\pi(\boldsymbol{\beta}_h \cdot \nabla\Pi_k^{0,E} v_h)\|_{0,E}^2 \lesssim \sigma\|v_h\|_{0,\mathcal{D}(E)}^2 \,. \tag{4.68}$$

The convective term in the norm is estimated by

$$h\|\boldsymbol{\beta} \cdot \nabla\Pi_k^{0,E} w_h\|_{0,E}^2 \lesssim h^{-1}\|\Pi_k^{0,E} w_h\|_{0,E}^2 \lesssim h\|\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h\|_{0,\mathcal{D}(E)}^2 \,. \tag{4.69}$$

The boundary terms are estimated by

$$\sum_{e\subset\Gamma_E} \frac{\epsilon}{\delta h}\|\Pi_{k-1}^{0,e} w_h\|_{0,e}^2 \leq \frac{\epsilon}{\delta h}\|w_h\|_{0,\Gamma_E}^2 \lesssim \epsilon\|\pi(\boldsymbol{\beta}_h \cdot \nabla\Pi_k^{0,E} v_h)\|_{0,E}^2 \lesssim h\|\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h\|_{0,\mathcal{D}(E)}^2 \,, \tag{4.70}$$

and

$$\||\boldsymbol{\beta} \cdot \boldsymbol{n}^E|^{\frac{1}{2}}\Pi_k^{0,E} w_h\|_{0,\Gamma_{E,\text{in}}}^2 \lesssim h^{-1}\|\Pi_k^{0,E} w_h\|_{0,E}^2 \lesssim h\|\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h\|_{0,\mathcal{D}(E)}^2 \,. \tag{4.71}$$

We gather estimates (4.67), (4.68), (4.69), (4.70), (4.71), together with estimate (4.51) for the term $J_h^E(w_h, w_h)$, add over all elements, and obtain

$$\|w_h\|_{\text{cip}}^2 \lesssim \sigma\|v_h\|_{0,\Omega_h}^2 + \sum_{E\in\Omega_h} h\|\boldsymbol{\beta}_h \cdot \nabla\Pi_k^0 v_h\|_{0,E}^2 \,,$$

and the result follows from estimate (4.66). □

We are now able to prove the inf-sup condition (4.42).

Theorem 4.11 (Inf-sup condition)**.** *Under assumptions **(A-NC)**,*

$$\|v_h\|_{\text{cip}} \lesssim \sup_{z_h\in V_h^{k,nc}(\Omega_h)} \frac{\mathcal{A}_{\text{cip}}(v_h, z_h)}{\|z_h\|_{\text{cip}}} \qquad \textit{for all } v_h \in V_h^{k,nc}(\Omega_h). \tag{4.72}$$

Proof. We distinguish two cases: $h > \epsilon$ and $h \leq \epsilon$.

If $h > \epsilon$, given $v_h \in V^{k,nc}(\Omega_h)$, we define the function $z_h := w_h + \kappa v_h$, where w_h is given by (4.46), and κ is a sufficiently large constant. Combining Lemma 4.8 with Lemma 4.9 gives

$$\mathcal{A}_{\text{cip}}(v_h, z_h) = \mathcal{A}_{\text{cip}}(v_h, w_h + \kappa v_h) \gtrsim \|v_h\|_{\text{cip}}^2 \,.$$

This, together with Lemma 4.10, gives the inf-sup condition in the case $h > \epsilon$. If $h \leq \epsilon$, from Lemma 4.8, the definition of the norm $\|\cdot\|_{\text{cip}}$, and the estimate

$$h\|\boldsymbol{\beta} \cdot \nabla\Pi_k^{0,E} v_h\|_{0,E}^2 \lesssim \varepsilon\|\nabla\Pi_k^{0,E} v_h\|_{0,E}^2 \lesssim \varepsilon\|\nabla v_h\|_{0,E}^2 \,,$$

we obtain $\mathcal{A}_{\text{cip}}(v_h, v_h) \gtrsim \|v_h\|_{\text{cip}}^2$ and the proof is complete. □

4.2.3 Error Estimates

In this section, we derive error estimates under the following smoothness assumption on the terms appearing in (4.1). The solution of the continuous problem u, the right-hand side f, and the advective field $\boldsymbol{\beta}$ in (4.2) satisfy

$$u \in H^2(\Omega) \cap H_g^1(\Omega) \cap H^{k+1}(\Omega_h)\,, \quad f \in H^{k+\frac{1}{2}}(\Omega_h)\,, \quad \boldsymbol{\beta} \in [W_\infty^{k+1}(\Omega_h)]^d \,.$$

Let u_h be the discrete solution of (4.18). Then, thanks to the inf-sup condition (4.72), for the error $u - u_h$, we prove the following result.

Proposition 4.12. *Under assumptions (A1), we have*

$$\|u-u_h\|_{\mathrm{cip}} \lesssim \|e_{\mathcal{I}}\|_{\mathrm{cip}} + \eta_{\mathcal{F}} + \eta_{\tilde{\mathcal{B}}} + \eta_a + \eta_b + \eta_c + \eta_d + \eta_{\mathcal{N}} + \eta_J \,, \tag{4.73}$$

where $e_{\mathcal{I}} := u - u_{\mathcal{I}}$ and $u_{\mathcal{I}} \in V_h^{k,nc}(\Omega_h)$ is the interpolant function of u defined in Lemma 4.1. Moreover, in the right-hand side of (4.73) *we have defined*

$$\begin{aligned}
\eta_{\mathcal{F}} &:= \|\tilde{\mathcal{F}} - \mathcal{F}_h\|_{\mathrm{cip}^*} \,, \\
\eta_{\tilde{\mathcal{B}}} &:= \|\tilde{\mathcal{B}}(u,\cdot)\|_{\mathrm{cip}^*} \,, \\
\eta_a &:= \epsilon\, \|a(u,\cdot) - a_h(u_{\mathcal{I}},\cdot)\|_{\mathrm{cip}^*} \,, \\
\eta_b &:= \|b(u,\cdot) - b_h(u_{\mathcal{I}},\cdot)\|_{\mathrm{cip}^*} \,, \\
\eta_c &:= \sigma \|c(u,\cdot) - c_h(u_{\mathcal{I}},\cdot)\|_{\mathrm{cip}^*} \,, \\
\eta_d &:= \|d_h(u_{\mathcal{I}},\cdot)\|_{\mathrm{cip}^*} \,, \\
\eta_{\mathcal{N}} &:= \|\tilde{\mathcal{N}}_h(u,\cdot) - \mathcal{N}_h(u_{\mathcal{I}},\cdot)\|_{\mathrm{cip}^*} \,, \\
\eta_J &:= \|\tilde{J}_h(u,\cdot) - J_h(u_{\mathcal{I}},\cdot)\|_{\mathrm{cip}^*} = \|J_h(u_{\mathcal{I}},\cdot)\|_{\mathrm{cip}^*} \,,
\end{aligned} \tag{4.74}$$

where $\|\cdot\|_{\mathrm{cip}^*}$ denotes the dual norm of the norm $\|\cdot\|_{\mathrm{cip}}$.

Proof. Setting $e_h := u_h - u_{\mathcal{I}}$, thanks to the triangular inequality, we have

$$\|u-u_h\|_{\mathrm{cip}} \le \|e_{\mathcal{I}}\|_{\mathrm{cip}} + \|e_h\|_{\mathrm{cip}} \,.$$

Hence, we only have to bound the second term of the right-hand side. Using the inf-sup condition, (4.18) and (4.19), we have that

$$\begin{aligned}
\|e_h\|_{\mathrm{cip}} &= \sup_{v_h \in V_h^{k,nc}(\Omega_h)} \frac{\mathcal{A}_{\mathrm{cip}}(u_h - u_{\mathcal{I}}, v_h)}{\|v_h\|_{\mathrm{cip}}} = \sup_{v_h \in V_h^{k,nc}(\Omega_h)} \frac{\mathcal{F}_h(v_h) - \mathcal{A}_{\mathrm{cip}}(u_{\mathcal{I}}, v_h)}{\|v_h\|_{\mathrm{cip}}} \\
&= \sup_{v_h \in V_h^{k,nc}(\Omega_h)} \frac{\mathcal{F}_h(v_h) - \tilde{\mathcal{F}}(v_h) - \tilde{\mathcal{B}}(u,v_h) + \widetilde{\mathcal{A}}_{\mathrm{cip}}(u,v_h) - \mathcal{A}_{\mathrm{cip}}(u_{\mathcal{I}}, v_h)}{\|v_h\|_{\mathrm{cip}}} .
\end{aligned}$$

Estimate (4.73) now follows from considering the definitions of $\mathcal{A}_{\mathrm{cip}}(\cdot,\cdot)$ and $\widetilde{\mathcal{A}}_{\mathrm{cip}}(\cdot,\cdot)$ given in (4.16) and (4.20), respectively. □

We proceed by estimating each of the terms on the right-hand side of (4.73). We start with the interpolation error in the CIP norm.

Lemma 4.13. *Under assumptions (A-NC), we have the following estimate:*

$$\|e_{\mathcal{I}}\|_{\mathrm{cip}}^2 \lesssim \sum_{E \in \Omega_h} \epsilon\, h^{2k}\, |u|_{k+1,E}^2 + \sum_{E \in \Omega_h} h^{2k+1}\, |u|_{k+1,E}^2 \,.$$

Proof. The proof can be developed along the lines of [17]. The only slight differences lie in the treatment of the following Nitsche terms, for which we use a trace inequality and the interpolation estimate in Lemma 4.1 to obtain

$$\sum_{e \subset \Gamma_E} \frac{\epsilon}{\delta h_E} \langle \Pi_{k-1}^{0,e} e_{\mathcal{I}}, \Pi_{k-1}^{0,e} e_{\mathcal{I}} \rangle_e \lesssim \sum_{e \subset \Gamma_E} \frac{\epsilon}{\delta h} \|e_{\mathcal{I}}\|_{0,e}^2 \lesssim \frac{\epsilon}{\delta h^2} \|e_{\mathcal{I}}\|_{0,E}^2 + \frac{\epsilon}{\delta} |e_{\mathcal{I}}|_{1,E}^2 \lesssim \epsilon\, h^{2k}\, |u|_{k+1,E}^2 \,,$$

and

$$\sum_{e \subset \Gamma_{E,\mathrm{in}}} \langle |\boldsymbol{\beta} \cdot \boldsymbol{n}^E| \Pi_k^{0,E} e_{\mathcal{I}}, \Pi_k^{0,E} e_{\mathcal{I}} \rangle_e \lesssim h^{-1}\, \|\Pi_k^{0,E} e_{\mathcal{I}}\|_{0,E}^2 \le h^{-1}\, \|e_{\mathcal{I}}\|_{0,E}^2 \lesssim h^{2k+1}\, |u|_{k+1,E}^2 \,.$$

□

Now, we estimate the term $\eta_{\tilde{\mathcal{B}}}$ defined in (4.74), which enters into play because of the nonconformity of our method.

Lemma 4.14 (Estimate of $\eta_{\tilde{\mathcal{B}}}$). *Under assumptions **(A-NC)**, we have the following estimate:*

$$\eta_{\tilde{\mathcal{B}}} \lesssim \epsilon^{\frac{1}{2}} h^k \Big(\sum_{E \in \Omega_h} |u|^2_{k+1,E} \Big)^{1/2} . \tag{4.75}$$

Proof. Thanks to the definition of the space $V_h^{k,nc}(\Omega_h)$ and Lemma 1.5, we get

$$\begin{aligned}
\tilde{\mathcal{B}}(u, v_h) &= \sum_{e \in \mathcal{E}_h^0} \epsilon \int_e \nabla u \cdot [\![v_h]\!] \mathrm{d}s = \sum_{e \in \mathcal{E}_h^0} \epsilon \int_e (\mathbf{I} - \mathbf{\Pi}^{0,e}_{k-1}) \nabla u \cdot [\![v_h]\!] \mathrm{d}s \\
&= \sum_{e \in \mathcal{E}_h^0} \epsilon \int_e (\mathbf{I} - \mathbf{\Pi}^{0,e}_{k-1}) \nabla u \cdot ([\![v_h]\!] - [\![\Pi_0^{0,e} v_h]\!]) \mathrm{d}s \\
&\leq \sum_{e \in \mathcal{E}_h^0} \epsilon^{\frac{1}{2}} \| (\mathbf{I} - \mathbf{\Pi}^{0,e}_{k-1}) \nabla u \|_{0,e} \, \epsilon^{\frac{1}{2}} \| \, [\![v_h]\!] - [\![\Pi_0^{0,e} v_h]\!] \, \|_{0,e} \\
&\leq \sum_{E \in \Omega_h} \epsilon^{\frac{1}{2}} \, h^{k-\frac{1}{2}} \, |u|_{k+1,E} \, \epsilon^{\frac{1}{2}} \, h^{\frac{1}{2}} |v_h|_{1,E} \\
&\lesssim \epsilon^{\frac{1}{2}} h^k \sum_{E \in \Omega_h} |u|_{k+1,E} \|v_h\|_{\mathrm{cip},E} \\
&\lesssim \epsilon^{\frac{1}{2}} h^k \Big(\sum_{E \in \Omega_h} |u|^2_{k+1,E} \Big)^{1/2} \|v_h\|_{\mathrm{cip}} ,
\end{aligned} \tag{4.76}$$

where we have used that each element is counted a finite number of times. Estimate (4.75) follows from (4.76). □

Some of the other quantities in (4.74) can be estimated exactly as in the previous chapter. We group them in the following lemma.

Lemma 4.15 (Estimates of η_a, η_c, η_J and $\eta_{\mathcal{F}}$). *Under assumptions **(A-NC)**, we have the following estimates:*

$$\begin{aligned}
\eta_a &\lesssim \epsilon^{\frac{1}{2}} \, h^k \Big(\sum_{E \in \Omega_h} |u|^2_{k+1,\mathcal{D}(E)} \Big)^{1/2} , \\
\eta_c &\lesssim h^{k+1} \Big(\sum_{E \in \Omega_h} |u|^2_{k+1,E} \Big)^{1/2} , \\
\eta_J &\lesssim h^{k+\frac{1}{2}} \Big(\sum_{E \in \Omega_h} |u|^2_{k+1,E} \Big)^{1/2} , \\
\eta_{\mathcal{F}} &\lesssim h^{k+\frac{1}{2}} \, \Big(\sum_{E \in \Omega_h} |f|^2_{k+\frac{1}{2},E} \Big)^{1/2} .
\end{aligned}$$

Proof. We refer to Lemma 3.22, Lemma 3.24, Lemma 3.26 and Lemma 3.21. We observe that, for $\eta_{\mathcal{F}}$, the terms containing g cancel with each other (cf. (4.17) and (4.23)), therefore the estimate contains only f. □

In the following three lemmas, we detail the proofs of the estimates for the remaining terms in (4.74).

Lemma 4.16 (Estimate of η_b). *Under assumptions **(A-NC)**, the term η_b satisfies*

$$\eta_b \lesssim h^k \Big(\sum_{E \in \Omega_h} \|\boldsymbol{\beta}\|^2_{[W^k_\infty(E)]^2} |u|^2_{k+1,E} \Big)^{1/2} . \tag{4.77}$$

Proof. We have

$$\eta_b = \sup_{v_h \in V_h^{k,nc}(\Omega_h)} \frac{\sum_{E \in \Omega_h} (\boldsymbol{\beta} \cdot \nabla u, v_h)_{0,E} - (\boldsymbol{\beta} \cdot \nabla \Pi_k^{0,E} u_{\mathcal{I}}, \Pi_k^{0,E} v_h)_{0,E}}{\|v_h\|_{\mathrm{cip}}} .$$

We proceed locally element by element. For any fixed element $E \in \Omega_h$, we have

$$\begin{aligned}(\boldsymbol{\beta}\cdot\nabla u,&v_h)_{0,E} - (\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}u_{\mathcal{I}},\Pi_k^{0,E}v_h)_{0,E}\\ &= (\boldsymbol{\beta}\cdot\nabla(u-u_{\mathcal{I}}),v_h)_{0,E} + ((I-\Pi_k^{0,E})(\boldsymbol{\beta}\cdot\nabla u_{\mathcal{I}}),(I-\Pi_k^{0,E})v_h)_{0,E}\\ &\quad + (\boldsymbol{\beta}\cdot\nabla(I-\Pi_k^{0,E})u_{\mathcal{I}},\Pi_k^{0,E}v_h)_{0,E}\\ &=: T_{b,1}^E + T_{b,2}^E + T_{b,3}^E\,.\end{aligned}$$

We consider each of the three terms. On the first one, we use the Cauchy-Schwarz inequality, the interpolation estimate, and the definition of $\|\cdot\|_{\mathrm{cip},E}$, and obtain

$$T_{b,1}^E = (\boldsymbol{\beta}\cdot\nabla(u-u_{\mathcal{I}}),v_h) \lesssim \|\boldsymbol{\beta}\|_{[L^\infty(E)]^d}|u-u_{\mathcal{I}}|_{1,E}\|v_h\|_{0,E} \lesssim h^k\|\boldsymbol{\beta}\|_{[L^\infty(E)]^d}|u|_{k+1,E}\|v_h\|_{\mathrm{cip},E}\,.$$

For the second one, we have

$$\begin{aligned}T_{b,2}^E &= ((I-\Pi_k^{0,E})(\boldsymbol{\beta}\cdot\nabla u_{\mathcal{I}}),(I-\Pi_k^{0,E})v_h)_{0,E}\\ &= ((I-\Pi_k^{0,E})(\boldsymbol{\beta}\cdot\nabla u),(I-\Pi_k^{0,E})v_h)_{0,E} + ((I-\Pi_k^{0,E})(\boldsymbol{\beta}\cdot\nabla(u_{\mathcal{I}}-u)),(I-\Pi_k^{0,E})v_h)_{0,E}\\ &\lesssim \big(\|(I-\Pi_k^{0,E})(\boldsymbol{\beta}\cdot\nabla u)\|_{0,E} + \|(I-\Pi_k^{0,E})(\boldsymbol{\beta}\cdot\nabla(u_{\mathcal{I}}-u))\|_{0,E}\big)\|(I-\Pi_k^{0,E})v_h\|_{0,E}\\ &\lesssim \big(h^k|\boldsymbol{\beta}\cdot\nabla u|_{k,E} + \|\boldsymbol{\beta}\|_{[L^\infty(E)]^d}|u-u_{\mathcal{I}}|_{1,E}\big)\|v_h\|_{\mathrm{cip},E}\\ &\lesssim h^k\|\boldsymbol{\beta}\|_{[W^k_\infty(E)]^d}\|u\|_{k+1,E}\|v_h\|_{\mathrm{cip},E}\,.\end{aligned}$$

The third term can be estimated as

$$\begin{aligned}T_{b,3}^E &= (\boldsymbol{\beta}\cdot\nabla(I-\Pi_k^{0,E})u_{\mathcal{I}},\Pi_k^{0,E}v_h)_{0,E}\\ &\lesssim \|\boldsymbol{\beta}\|_{[L^\infty(E)]^d}|(I-\Pi_k^{0,E})u_{\mathcal{I}}|_{1,E}\|\Pi_k^{0,E}v_h\|_{0,E}\\ &\lesssim \|\boldsymbol{\beta}\|_{[L^\infty(E)]^d}\big(|(I-\Pi_k^{0,E})u|_{1,E} + |(I-\Pi_k^{0,E})(u-u_{\mathcal{I}})|_{1,E}\big)\|\Pi_k^{0,E}v_h\|_{0,E}\\ &\lesssim h^k\|\boldsymbol{\beta}\|_{[L^\infty(E)]^d}|u|_{k+1,E}\|v_h\|_{\mathrm{cip},E}\,.\end{aligned}$$

Estimate (4.77) follows from the above three bounds, and from summing over all the elements $E \in \Omega_h$. □

Lemma 4.17 (Estimate of η_d)**.** *Under assumptions **(A1)** and **(A2)**, the term η_d can be estimated as follows:*

$$\eta_d \lesssim h^k\Big(\sum_{E\in\Omega_h}|u|^2_{k+1,\mathcal{D}(E)}\Big)^{1/2}\,. \tag{4.78}$$

Proof. We have

$$\eta_d = \sup_{v_h\in V_h^{k,nc}(\Omega_h)}\frac{-\frac12\sum_{E\in\Omega_h}\sum_{e\subset\partial E/\Gamma}\int_e\boldsymbol{\beta}\cdot[\![\Pi_k^0u_{\mathcal{I}}]\!]\{\Pi_k^0v_h\}\mathrm{d}s}{\|v_h\|_{\mathrm{cip}}}\,.$$

Again, we proceed element by element. Using also a trace inequality for polynomials, we estimate the numerator of the quotient above as follows:

$$\begin{aligned}-\sum_{e\subset\partial E\setminus\Gamma}&\frac12\int_e\boldsymbol{\beta}\cdot[\![\Pi_k^0u_{\mathcal{I}}]\!]\{\Pi_k^0v_h\}\mathrm{d}s = \sum_{e\subset\partial E\setminus\Gamma}\frac12\int_e\boldsymbol{\beta}\cdot[\![u-\Pi_k^0u_{\mathcal{I}}]\!]\{\Pi_k^0v_h\}\mathrm{d}s\\ &= \sum_{e\subset\partial E\setminus\Gamma}\frac12\int_e\boldsymbol{\beta}\cdot[\![u-\Pi_k^0u]\!]\{\Pi_k^0v_h\}\mathrm{d}s + \sum_{e\subset\partial E\setminus\Gamma}\frac12\int_e\boldsymbol{\beta}\cdot[\![\Pi_k^0(u-u_{\mathcal{I}})]\!]\{\Pi_k^0v_h\}\mathrm{d}s\\ &\lesssim \|\boldsymbol{\beta}\|_{[L^\infty(\mathcal{D}(E))]^d}\sum_{K\subset\mathcal{D}(E)}\big(h^{\frac12}|u-\Pi_k^0u|_{1,K} + h^{-\frac12}\|u-\Pi_k^0u\|_{0,K}\\ &\qquad + h^{-\frac12}\|\Pi_k^0(u-u_{\mathcal{I}})\|_{0,K}\big)h^{-\frac12}\|v_h\|_{0,\mathcal{D}(E)}\\ &\lesssim h^k|u|_{k+1,\mathcal{D}(E)}\|v_h\|_{\mathrm{cip},\mathcal{D}(E)}\,.\end{aligned}$$

Estimate (4.78) follows from the bound above, by summing over the elements and taking into account, as already noticed for (4.65), that each element is counted only a uniformly bounded number of times. □

Lemma 4.18 (Estimate of $\eta_{\mathcal{N}}$)**.** *Under assumptions* ***(A-NC)****, the term $\eta_{\mathcal{N}}$ can be estimated as follows:*

$$\eta_{\mathcal{N}} \lesssim (\epsilon^{\frac{1}{2}} h^k + h^{k+\frac{1}{2}}) \Big(\sum_{E\in\Omega_h} |u|^2_{k+1,E} \Big)^{1/2} . \tag{4.79}$$

Proof. We have

$$\begin{aligned}\eta_{\mathcal{N}} = \sup_{v_h \in V_h^{k,nc}(\Omega_h)} & \Big[\frac{1}{\|v_h\|_{\mathrm{cip}}} \sum_{E\in\Omega_h} \Big(\sum_{e\subset\Gamma_E} \epsilon \langle \mathbf{\Pi}^{0,E}_{k-1} \nabla u_{\mathcal{I}} \cdot \boldsymbol{n}^E - \nabla u \cdot \boldsymbol{n}^E, v_h \rangle_e \\ & + \sum_{e\subset\Gamma_E} \epsilon \langle u_{\mathcal{I}} - u, \mathbf{\Pi}^{0,E}_{k-1} \nabla v_h \cdot \boldsymbol{n}^E \rangle_e \\ & + \frac{\epsilon}{\delta h_E} \sum_{e\subset\Gamma_E} \langle u - u_{\mathcal{I}}, \Pi^{0,e}_{k-1} v_h \rangle_e + \sum_{e\subset\Gamma_{E,\mathrm{in}}} \langle |\boldsymbol{\beta}\cdot\boldsymbol{n}^E| (u - \Pi^{0,E}_k u_{\mathcal{I}}), \Pi^{0,E}_k v_h \rangle_e \Big) \Big] .\end{aligned}$$

We consider an element $E \in \Omega_h$. Thus, we need to estimate four different terms:

$$\begin{aligned}& \epsilon \langle \mathbf{\Pi}^{0,E}_{k-1} \nabla u_{\mathcal{I}} \cdot \boldsymbol{n}^E - \nabla u \cdot \boldsymbol{n}^E, v_h \rangle_{\Gamma_E} + \epsilon \langle u_{\mathcal{I}} - u, \mathbf{\Pi}^{0,E}_{k-1} \nabla v_h \cdot \boldsymbol{n}^E \rangle_{\Gamma_E} \\ & \quad + \frac{\epsilon}{\delta h_E} \sum_{e\subset\Gamma_E} \langle u - u_{\mathcal{I}}, \Pi^{0,e}_{k-1} v_h \rangle_e + \langle |\boldsymbol{\beta}\cdot\boldsymbol{n}^E| (u - \Pi^{0,E}_k u_{\mathcal{I}}), \Pi^{0,E}_k v_h \rangle_{\Gamma_{E,\mathrm{in}}} \\ & \quad = T^E_{N,1} + T^E_{N,2} + T^E_{N,3} + T^E_{N,4} .\end{aligned}$$

For $T^E_{N,1}$, we have

$$\begin{aligned} T^E_{N,1} &= \epsilon \langle \mathbf{\Pi}^{0,E}_{k-1} \nabla u_{\mathcal{I}} \cdot \boldsymbol{n}^E - \nabla u \cdot \boldsymbol{n}^E, v_h \rangle_{\Gamma_E} \\ &\leq \epsilon \|\nabla u - \mathbf{\Pi}^{0,E}_{k-1} \nabla u_{\mathcal{I}}\|_{0,e} \|v_h\|_{0,\Gamma_E} \\ &\lesssim \epsilon h^{-\frac{1}{2}} \left(\|\nabla u - \mathbf{\Pi}^{0,E}_{k-1} \nabla u_{\mathcal{I}}\|_{0,E} + h\, |\nabla u - \mathbf{\Pi}^{0,E}_{k-1} \nabla u_{\mathcal{I}}|_{1,E} \right) \|v_h\|_{0,\Gamma_E} .\end{aligned}$$

Since it holds (cf. Lemma 1.4 and Lemma 4.1):

$$\begin{aligned} \|\nabla u - \mathbf{\Pi}^{0,E}_{k-1} \nabla u_{\mathcal{I}}\|_{0,E} &\leq \|\nabla u - \mathbf{\Pi}^{0,E}_{k-1} \nabla u\|_{0,E} + \|\mathbf{\Pi}^{0,E}_{k-1} (\nabla u - \nabla u_{\mathcal{I}})\|_{0,E} \\ &\lesssim h^k |u|_{k+1,E} ,\end{aligned}$$

and

$$\begin{aligned} |\nabla u - \mathbf{\Pi}^{0,E}_{k-1} \nabla u_{\mathcal{I}}|_{1,E} &\leq |\nabla u - \mathbf{\Pi}^{0,E}_{k-1} \nabla u|_{1,E} + |\mathbf{\Pi}^{0,E}_{k-1} (\nabla u - \nabla u_{\mathcal{I}})|_{1,E} \\ &\lesssim |\nabla u - \mathbf{\Pi}^{0,E}_{k-1} \nabla u|_{1,E} + h^{-1} \|\mathbf{\Pi}^{0,E}_{k-1} (\nabla u - \nabla u_{\mathcal{I}})\|_{0,E} \\ &\lesssim h^{k-1} |u|_{k+1,E} ,\end{aligned}$$

we get

$$\begin{aligned} T^E_{N,1} &\lesssim \epsilon h^k |u|_{k+1,E}\, h^{-\frac{1}{2}} \sum_{e\subset\Gamma_E} \|v_h\|_{0,e} \\ &\lesssim \epsilon h^k |u|_{k+1,E}\, h^{-\frac{1}{2}} \sum_{e\subset\Gamma_E} \left(\|v_h - \Pi^{0,e}_0 v_h\|_{0,e} + \|\Pi^{0,e}_0 v_h\|_{0,e} \right) \\ &\lesssim \epsilon h^k |u|_{k+1,E} \left(|v_h|_{1,E} + \sum_{e\subset\Gamma_E} h^{-\frac{1}{2}} \|\Pi^{0,e}_{k-1} v_h\|_{0,e} \right) \\ &\lesssim \epsilon^{\frac{1}{2}} h^k |u|_{k+1,E} \|v_h\|_{\mathrm{cip},E} ,\end{aligned} \tag{4.80}$$

where, in the penultimate step, we have used Lemma 1.5.

For $T^E_{N,2}$, we get

$$\begin{aligned}
T^E_{N,2} &= \epsilon\langle u_{\mathcal{I}} - u, \mathbf{\Pi}^{0,E}_{k-1}\nabla v_h \cdot \boldsymbol{n}^E\rangle_{\Gamma_E} \\
&\le \epsilon\|u - u_{\mathcal{I}}\|_{0,\Gamma_E}\|\mathbf{\Pi}^{0,E}_{k-1}\nabla v_h\|_{0,\Gamma_E} \\
&\lesssim \epsilon\|u - u_{\mathcal{I}}\|_{0,\Gamma_E}\, h^{-\frac{1}{2}}\|\mathbf{\Pi}^{0,E}_{k-1}\nabla v_h\|_{0,E} \\
&\lesssim \epsilon h^{-\frac{1}{2}}\|u - u_{\mathcal{I}}\|_{0,\Gamma_E}|v_h|_{1,E} \\
&\lesssim \epsilon^{\frac{1}{2}}\left(|u - u_{\mathcal{I}}|_{1,E} + h^{-1}\|u - u_{\mathcal{I}}\|_{0,E}\right)\|v_h\|_{\mathrm{cip},E} \\
&\lesssim \epsilon^{\frac{1}{2}}h^k|u|_{k+1,E}\|v_h\|_{\mathrm{cip},E}\,.
\end{aligned} \tag{4.81}$$

For $T^E_{N,3}$, we have

$$\begin{aligned}
T^E_{N,3} &= \frac{\epsilon}{\delta h_E}\sum_{e\subset\Gamma_E}\langle u - u_{\mathcal{I}}, \Pi^{0,e}_{k-1}v_h\rangle_e \\
&\le \left(\frac{\epsilon}{\delta h_E}\sum_{e\subset\Gamma_E}\|u - u_{\mathcal{I}}\|^2_{0,e}\right)^{\frac{1}{2}}\left(\frac{\epsilon}{\delta h_E}\sum_{e\subset\Gamma_E}\|\Pi^{0,e}_{k-1}v_h\|^2_{0,e}\right)^{\frac{1}{2}} \\
&\lesssim \left(\frac{\epsilon}{\delta h_E}\sum_{e\subset\Gamma_E}\|u - u_{\mathcal{I}}\|^2_{0,e}\right)^{\frac{1}{2}}\|v_h\|_{\mathrm{cip},E} \\
&\lesssim \frac{\epsilon^{\frac{1}{2}}}{\delta^{\frac{1}{2}}}\left((|u - u_{\mathcal{I}}|^2_{1,E} + h^{-2}\|u - u_{\mathcal{I}}\|^2_{0,E})\right)^{\frac{1}{2}}\|v_h\|_{\mathrm{cip},E} \\
&\lesssim \epsilon^{\frac{1}{2}}h^k|u|_{k+1,E}\|v_h\|_{\mathrm{cip},E}.
\end{aligned} \tag{4.82}$$

Finally, for $T^E_{N,4}$, we get

$$\begin{aligned}
T^E_{N,4} &= \langle|\boldsymbol{\beta}\cdot\boldsymbol{n}^E|(u - \Pi^{0,E}_k u_{\mathcal{I}}), \Pi^{0,E}_k v_h\rangle_{\Gamma_{E,\mathrm{in}}} \\
&\lesssim \|u - \Pi^{0,E}_k u_{\mathcal{I}}\|_{0,\Gamma_{E,\mathrm{in}}}\,\||\boldsymbol{\beta}\cdot\boldsymbol{n}^E|^{\frac{1}{2}}\Pi^{0,E}_k v_h\|_{0,\Gamma_{E,\mathrm{in}}} \\
&\lesssim \left(\|u - \Pi^{0,E}_k u\|_{0,\Gamma_{E,\mathrm{in}}} + \|\Pi^{0,E}_k(u - u_{\mathcal{I}})\|_{0,\Gamma_{E,\mathrm{in}}}\right)\|v_h\|_{\mathrm{cip},E} \\
&\lesssim \left(h^{-\frac{1}{2}}\|u - \Pi^{0,E}_k u\|_{0,E} + h^{\frac{1}{2}}|u - \Pi^{0,E}_k u|_{1,E} + h^{-\frac{1}{2}}\|\Pi^{0,E}_k(u - u_{\mathcal{I}})\|_{0,E}\right)\|v_h\|_{\mathrm{cip},E} \\
&\lesssim \left(h^{-\frac{1}{2}}\|u - \Pi^{0,E}_k u\|_{0,E} + h^{\frac{1}{2}}|u - \Pi^{0,E}_k u|_{1,E} + h^{-\frac{1}{2}}\|u - u_{\mathcal{I}}\|_{0,E}\right)\|v_h\|_{\mathrm{cip},E} \\
&\lesssim h^{k+\frac{1}{2}}|u|_{k+1,E}\,\|v_h\|_{\mathrm{cip},E}\,.
\end{aligned} \tag{4.83}$$

Estimate (4.79) now follows by considering estimates (4.80)-(4.83) and summing all the local contributions. □

Combining Lemmas 4.13–4.18 with Proposition 4.12, we obtain the following result.

Theorem 4.19. *Let u be the solution of problem* (4.2) *and $u_h \in V^{k,nc}_h(\Omega_h)$ be the solution of the discrete problem* (4.18)*. Under assumptions **(A-NC)**, it holds true that*

$$\|u - u_h\|_{\mathrm{cip}} \lesssim \left(\epsilon^{\frac{1}{2}}\,h^k + h^{k+\frac{1}{2}}\right)\Big(\sum_{E\in\Omega_h}\Theta^2_E\Big)^{1/2}, \tag{4.84}$$

with constants Θ_E depending on $|u|_{k+1,E}$, $|f|_{k+\frac{1}{2},E}$, and $\|\beta\|_{[W^{k+1,\infty}(E)]^2}$, but independent of h and ε.

4.3 Numerical results

In this section, we numerically test our method in two space dimensions, by considering two model problems in the domain $\Omega = (0,1)\times(0,1)$. We use two different families of meshes:

- `octag:` meshes obtained by perturbing structured triangular meshes: each hypotenuse is split into two edges, then all nodes are perturbed, finally one extra node (the midpoint) is introduced on each edge; the elements of the obtained meshes are octagons;
- `voro:` Voronoi meshes.

The meshes are the same used in the previous chapter and are depicted in Figure 3.1 The analytic expression of the VEM solution u_h is unknown and we cannot compute the difference $u - u_h$ in closed form. Therefore, we consider the following quantities:

- H^1-**seminorm error**
$$e_{H^1} := \sqrt{\sum_{E\in\Omega_h} \left\|\nabla(u - \Pi_k^{\nabla,E} u_h)\right\|_{0,E}^2}\,;$$
- L^2-**norm error**
$$e_{L^2} := \sqrt{\sum_{E\in\Omega_h} \left\|(u - \Pi_k^{0,E} u_h)\right\|_{0,E}^2}\,.$$

We also consider the error in the CIP-norm defined in (4.43):

$$\begin{aligned}\|u-u_h\|_{\text{cip}}^2 \approx &\sum_{E\in\Omega_h} \epsilon \left\|\nabla(u - \Pi_k^{\nabla,E} u_h)\right\|_{0,E}^2 + h \sum_{E\in\Omega_h} \left\|\boldsymbol{\beta}\cdot\nabla\Pi_k^{0,E}(u-u_h)\right\|_{0,E}^2 \\ &+ \sum_{E\in\Omega_h} \sigma \left\|(u - \Pi_k^{0,E} u_h)\right\|_{0,E}^2 + \frac{\epsilon}{\delta h} \sum_{e\in\mathcal{E}_h^\partial} \left\|\Pi_{k-1}^{0,e}(u-u_h)\right\|_{0,e}^2 \\ &+ \sum_{e\in\mathcal{E}_h^\partial,\; e\subset\Gamma_{\text{in}}} \left\||\boldsymbol{\beta}\cdot\boldsymbol{n}^E|^{\frac{1}{2}}\Pi_k^{0,E}(u-u_h)\right\|_{0,e}^2 + J(u-u_h, u-u_h).\end{aligned}$$

We assume that the analytic solutions of problem (4.1) are the functions

$$u_1(x,y) = \frac{1}{2}\left(1 - \tanh\left(\frac{x-0.5}{0.05}\right)\right) \qquad \text{(first test)}\,,$$

$$u_2(x,y) = (y-y^2)\left(x - \frac{e^{\frac{x-1}{0.05}} - e^{\frac{-1}{0.05}}}{1 - e^{\frac{-1}{0.05}}}\right) \qquad \text{(second test)}\,,$$

$$u_3(x,y) = \sin(\pi x)\sin(\pi y) \qquad \text{(third test)}\,.$$

The first solution exhibit an internal layer in the middle of the domain. The second solution vanishes along the whole boundary and has a boundary layer at $x = 1$. The last one is a smooth solution. For all the tests, the parameter σ is set to 1, while the convective coefficient is

$$\boldsymbol{\beta}(x,\,y) := \begin{bmatrix} -2\,\pi\,\sin(\pi\,(x+2\,y)) \\ \pi\,\sin(\pi\,(x+2\,y)) \end{bmatrix}.$$

The Nitsche parameter is selected as $\delta = 0.1$, while for the parameters γ_e and γ_E, see (4.13), we set $\kappa_e = \kappa_E = 0.025$.

Convergence rates of the schemes. We investigate the convergence rates of the approximation error, choosing $\epsilon = 10^{-5}$ (hence the problem is in the advection-dominated regime). We consider both the mesh families described above. In Figure 4.1, we observe the results for u_1 and u_2, when the method orders are $k = 1, 2, 3$. Optimal rates of convergence in the L^2-norm and H^1-seminorm can be appreciated. For the CIP-norm and $k = 1$, we observe a super-linear convergence rate. This behavior can be explained by considering the error estimate of Theorem 4.19: most likely, for such errors and small ϵ, the dominating part is the second one in the right-hand side of (4.84), which correspond to a convergence rate of order $\frac{3}{2}$.

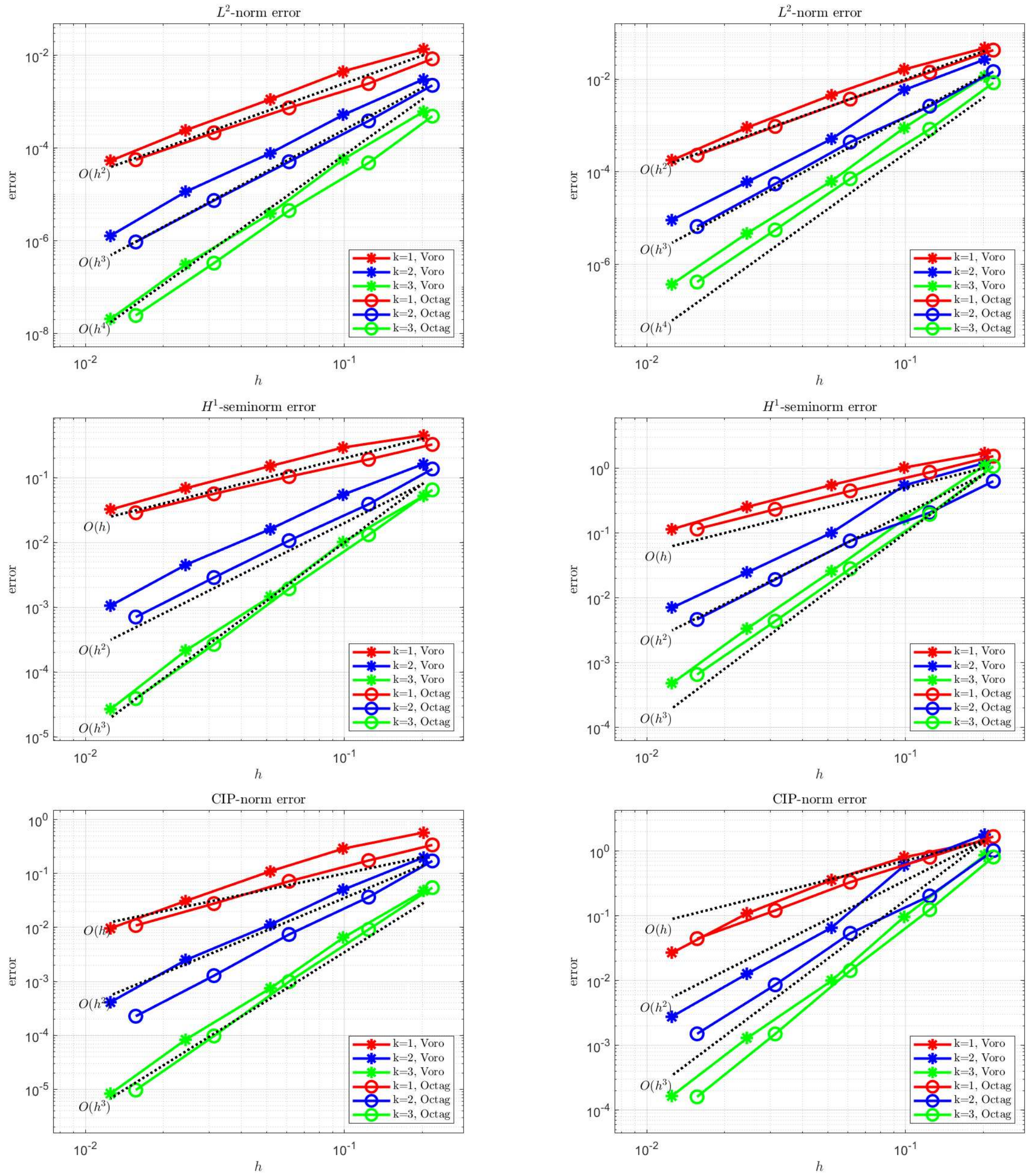


Figure 4.1: Convergences for u_1 (left column) and u_2 (right column). The red lines correspond to the case $k = 1$, the blue lines to the case $k = 2$, and the green lines to the case $k = 3$. The stars refer to the Voronoi mesh family and the circle correspond to the octagonal mesh family. We show the results for the error measured in the L^2-norm, the H^1−seminorm, and the CIP-norm.

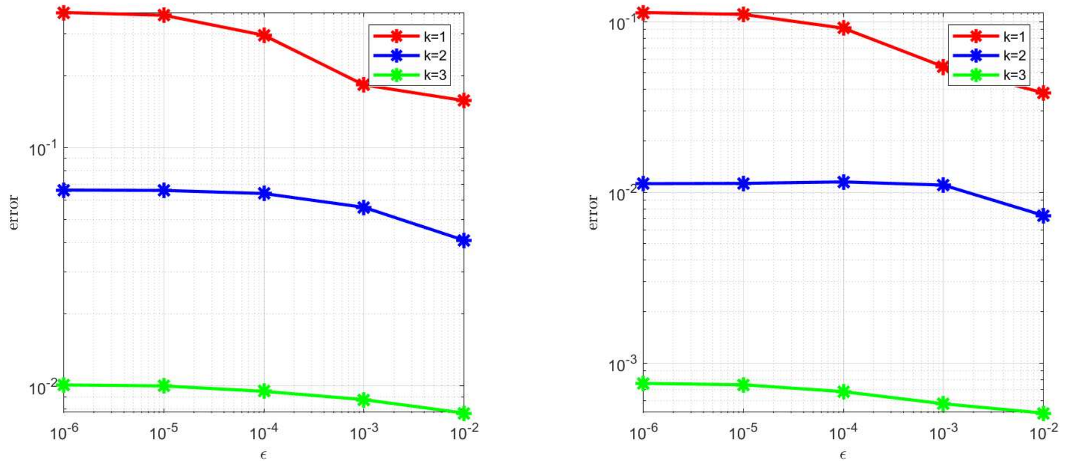


Figure 4.2: Numerical results for various choice of ϵ. The red lines correspond to the case $k = 1$, the blue lines to the case $k = 2$, and the green lines to the case $k = 3$. The error is measured in the H^1-seminorm.

Robustness with respect to the parameter ϵ. We now numerically assess the robustness of the method with respect to the diffusion parameter ϵ. For this purpose, we test the method on a Voronoi tessellation with 1024 polygons. As expected, in Figure 4.2, we observe that the CIP norm of the error is almost constant. Left column is related to u_1, right column is related to u_2. Similar result not reported here have been obtained using other meshes and other test cases. It is important to note that, by varying the parameter ϵ, we are modifying the load term f_ϵ. This variation is negligible when ϵ is very small, and we can assume it to be almost constant. The variations observed in the graph as ϵ approaches 1 can be explained by the fact that f is changing.

Conforming vs Nonconforming 1. We aim to compare the errors obtained using both the conforming and nonconforming methods, assessing their performance on different types of solutions. We employ Voronoi and we set the order of the methods to $k = 1, 2, 3$. Initially, we select the function $u_3(x, y)$ as the solution of equation (4.1). This function represents a smooth, continuous solution, allowing us to focus on the accuracy and stability of each method under ideal conditions. Figure 4.3 presents a visual comparison, with results from the conforming method shown on the left and those from the nonconforming method on the right. As observed, the graphs for the conforming and nonconforming methods are quite similar, showing no substantial differences in their error distributions. This similarity suggests that for highly regular solutions, both methods perform comparably well, providing consistent and reliable results. To further investigate, we then select a solution with an internal layer, represented by the function $u_2(x, y)$. The internal layer is intended to simulate conditions where the solution changes rapidly within a small area, a scenario commonly encountered in practical applications. The results for this test are presented in Figure 4.4. Here again, the conforming and nonconforming methods show comparable performance, with no major discrepancies in the error distribution observed between them.

Conforming vs Nonconforming 2. In this paragraph, we examine the sensitivity of the conforming and nonconforming methods with respect to the parameters γ_e and γ_E. We aim to understand how variations in these parameters affect the accuracy and robustness of the methods. For our analysis, we set the function $u(x, y) = u_3(x, y)$ as the target solution. To investigate a broad range of parameter values, we define $\kappa_e = \kappa_E = 10^p$, where p spans a set of 100 evenly distributed values in the interval $(0, 6)$. We solve the equation on a Voronoi mesh composed of 256 polygons. The errors are measured in the H^1-seminorm, which emphasizes the gradient of the error. The results are presented in Figure 4.5, where we plot the error

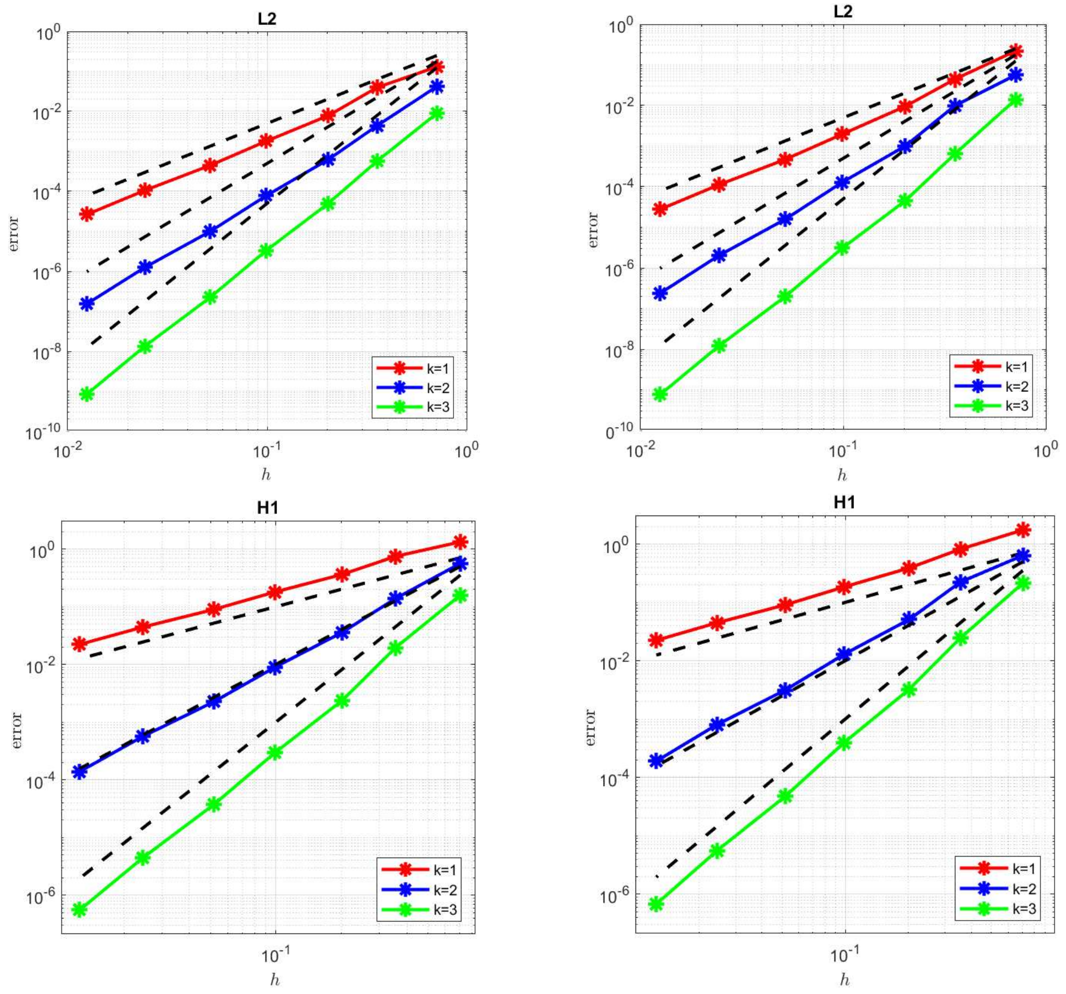


Figure 4.3: Convergences for the conforming method (left column) and the non conforming method (right column). The red lines correspond to the case $k = 1$, the blue lines to the case $k = 2$, and the green lines to the case $k = 3$. The dashed lines represent h^{k+1} for the L^2-error and h^k for the H^1-seminorm.

values as functions of p for both methods. As shown in the figure, both the conforming and nonconforming methods yield similar error patterns across the entire range of κ values. This similarity suggests that both methods are comparably stable and accurate with respect to variations in κ_e and κ_E, at least within the parameter range tested.

Simulation of a fluid inside a channel with two pipes. We consider the same situation presented in the previous chapter: a water-filled channel containing a pollutant, with two pipes blocking the flow in the middle of the channel. The parameters are identical to those used in Chapter 3; see Section 3.3 for further details. In Figure 4.6, we observe that the results are very similar to those obtained with the conforming methods, confirming both the accuracy and comparability of our approach.

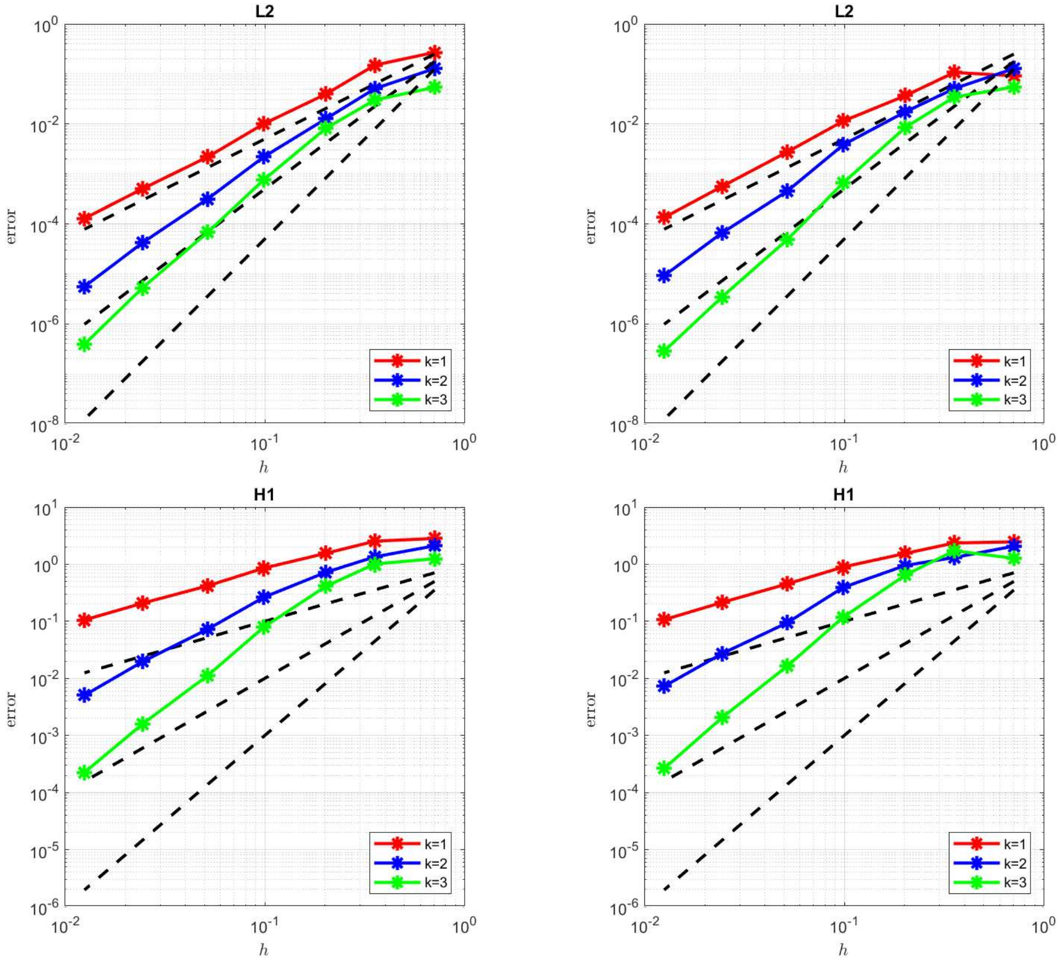


Figure 4.4: Convergences for the conforming method (left column) and the non conforming method (right column). The red lines correspond to the case $k = 1$, the blue lines to the case $k = 2$, and the green lines to the case $k = 3$. The dashed lines represent h^{k+1} for the L^2-error and h^k for the H^1-seminorm.

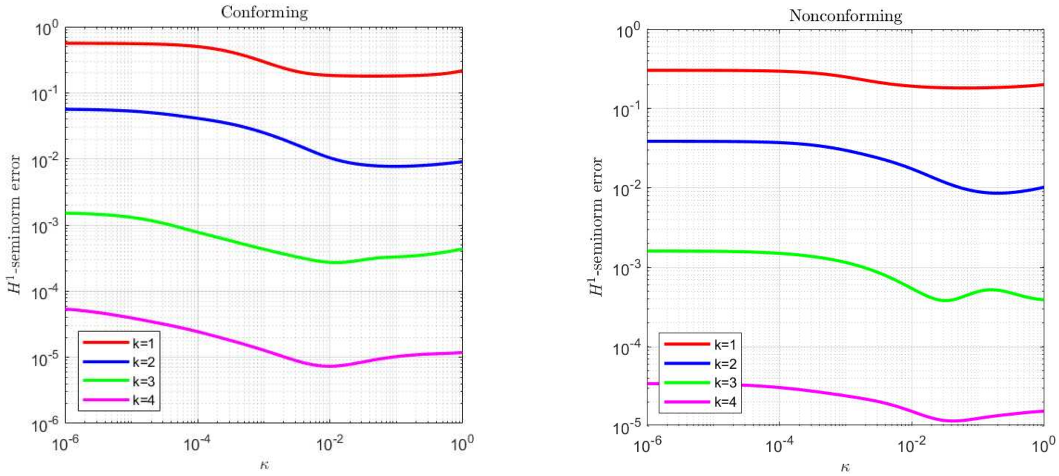


Figure 4.5: Results for the conforming method (left column) and the non conforming method (right column) for various values of κ_e and κ_E. The red lines correspond to the case $k = 1$, the blue lines to the case $k = 2$, the green lines to the case $k = 3$, and the magenta line to the case $k = 4$.

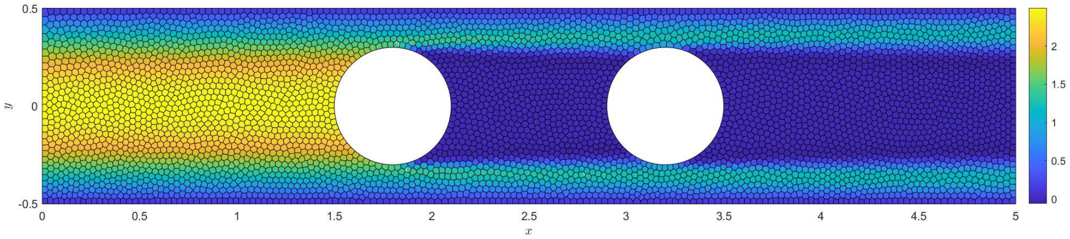


Figure 4.6: Numerical representation of a pollutant moves inside a channel with two pipes. The color represent the concentration of the pollutant.

Chapter 5

Three level CIP for Oseen equation

In recent years, there has been significant interest in developing pressure-robust numerical schemes [45, 37, 18]. These methods enable accurate approximations of velocity, particularly when dealing with non-smooth pressure, by eliminating the dependence on pressure in the error analysis of the method. This advancement is crucial to ensure the reliability and robustness of simulations in various fluid dynamics applications. This chapter aims to introduce a pressure-robust Virtual Element Method for the Oseen problem (the linearized version of the Navier-Stokes equations), which remains stable even in advection-dominated regimes. In [18, 13], the authors propose a VEM for Stokes that achieves divergence-free conditions by ensuring that the divergence of a virtual velocity is included in the space of the pressures in the definition of the local spaces. As mentioned in the Introduction, we recall that this requirement is not sufficient to eliminate the dependence on pressure in the error analysis of the velocity, since a slight dependence on the pressure still exists due to the approximation of load term.

This chapter concentrates on advection-dominated case, as the application of VEM to the Navier-Stokes equation in a diffusion-dominated regime has been covered in [14]. There exists a VEM for the Oseen problem that remains stable in the hyperbolic limit, as introduced in [15]. To achieve stabilization, this method incorporates local SUPG-like terms for the vorticity equation and a jump term similar to those discussed in previous chapters. Following the FEM method [7], we try to develop a VEM that achieves stability solely through jump operators applied to the skeleton of the mesh. The method controls the polynomial parts of the jumps of $(\boldsymbol{\nabla}\mathbf{u})\boldsymbol{\beta}$ through a three-level CIP-form. Specifically, we control the jumps of $(\boldsymbol{\nabla}\mathbf{u})\boldsymbol{\beta}$, the jump of the curl, and the jump of the gradient of the curl. Here, we are considering the scalar curl applied to vector-valued functions defined as

$$\operatorname{curl}(\mathbf{v}) := \frac{v_2}{\partial x} - \frac{v_1}{\partial y} .$$

where v_1 and v_2 denotes the two components of the vector-valued function $\mathbf{v}$. Currently, we can demonstrate the well-posedness of the discrete method by proving coercivity and presenting the numerical results. Although an error analysis has not been formulated, we have some ideas on how to approach it.

This chapter is organized as follows: the first section introduces the Oseen problem and the spaces used in the analysis. The second section describes the method. The third section presents the theoretical analysis. Finally, we conclude with some numerical results.

5.1 Model problem

Given a polygonal and simple connected domain $\Omega \subset \mathbb{R}^2$ with boundary Γ, we consider the steady Oseen equation with homogeneous Dirichlet boundary conditions.

$$\begin{cases} \text{find } (\mathbf{u}, p) \text{ such that:} & \\ -\nu\, \mathbf{div}(\boldsymbol{\varepsilon}(\mathbf{u})) + (\boldsymbol{\nabla}\mathbf{u})\boldsymbol{\beta} + \sigma\, \mathbf{u} - \nabla p = \mathbf{f} & \text{in } \Omega\,, \\ \operatorname{div}(\mathbf{u}) = 0 & \text{in } \Omega\,, \\ \mathbf{u} = 0 & \text{on } \Gamma\,. \end{cases} \tag{5.1}$$

As usual in this type of problems, $\mathbf{u}$ denotes the velocity of the fluid while p is the pressure. Furthermore, $\mathbf{div}$ (div), $\boldsymbol{\nabla}$, ∇, denote the vector (scalar) divergence operator, the gradient operator for vector fields and the gradient operator for scalar function while $\boldsymbol{\varepsilon}(\mathbf{u})$ is the symmetric gradient operator defined as

$$\boldsymbol{\varepsilon}(\mathbf{u}) := \frac{\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}^T\mathbf{u}}{2}\,.$$

The parameters $\nu, \sigma \in \mathbb{R}^+$ represent the diffusive and reaction coefficients, respectively. The transport advective field $\boldsymbol{\beta}$ is a sufficiently smooth vector-valued function that satisfies $\operatorname{div}\boldsymbol{\beta} = 0$. Finally, the load term $\mathbf{f}$ is assumed to be $\mathbf{f} \in [L^2(\Omega)]^2$. We introduce the spaces

$$\mathbf{V}(\Omega) := [H_0^1(\Omega)]^2 \quad \text{and} \quad Q(\Omega) := L_0^2(\Omega)\,.$$

A possible variational formulation for the problem (5.1) reads as

$$\begin{cases} \text{find } (\mathbf{u}, p) \in \mathbf{V}(\Omega) \times Q(\Omega) \text{ such that:} & \\ A(\mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = \mathcal{F}(\mathbf{v}) & \forall \mathbf{v} \in \mathbf{V}(\Omega)\,, \\ b(\mathbf{u}, q) = 0 & \forall q \in Q(\Omega)\,, \end{cases} \tag{5.2}$$

where the bilinear forms are defined as

$$A(\cdot,\cdot) : \mathbf{V}(\Omega) \times \mathbf{V}(\Omega) \to \mathbb{R}\,, \quad A(\mathbf{u}, \mathbf{v}) := \nu \int_\Omega \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v})\, \mathrm{d}\Omega + \sigma \int_\Omega \mathbf{u} \cdot \mathbf{v}\, \mathrm{d}\Omega\,,$$

$$b(\cdot,\cdot) : \mathbf{V}(\Omega) \times Q(\Omega) \to \mathbb{R}\,, \quad b(\mathbf{v}, q) := \int_\Omega q\, \operatorname{div}(\mathbf{v})\, \mathrm{d}\Omega\,,$$

$$c(\cdot,\cdot) : \mathbf{V}(\Omega) \times \mathbf{V}(\Omega) \to \mathbb{R}\,, \quad c(\mathbf{u}, \mathbf{v}) := \int_\Omega [(\boldsymbol{\nabla} u)\boldsymbol{\beta}] \cdot \mathbf{v}\, \mathrm{d}\Omega\,,$$

and as usual $\mathcal{F} : \mathbf{V}(\Omega) \to \mathbb{R}$ is the $[L^2(\Omega)]^2$ inner product against the function $\mathbf{f}$

$$\mathcal{F}(\mathbf{v}) := \int_\Omega \mathbf{f} \cdot \mathbf{v}\, \mathrm{d}\Omega\,.$$

Introducing the space

$$\mathbf{Z}(\Omega) := \Big\{ \mathbf{v} \in \mathbf{V}(\Omega) \text{ such that } \operatorname{div}\mathbf{v} = 0 \Big\}\,,$$

we can reformulate problem (5.1) in a pressure independent form

$$\begin{cases} \text{find } \mathbf{u} \in \mathbf{Z}(\Omega) \text{ such that:} \\ A(\mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{Z}(\Omega)\,. \end{cases} \tag{5.3}$$

This new formulation is useful for the theoretical analysis but is impractical for implementation due to the difficulty in identifying divergence-free functions. Since Ω is a smooth domain with Lipschitz boundary, we can associate a potential φ with a divergence-free function $\mathbf{v} \in \mathbf{Z}(\Omega)$ such that

$$\mathbf{curl}(\varphi) := \left(\frac{\partial \varphi}{\partial y}, -\frac{\partial \varphi}{\partial x} \right)^T = \mathbf{v}\,.$$

Therefore, we introduce a useful space for the analysis of the method. We define the space of the stream functions as

$$\Phi(\Omega) := \{\varphi \in H^2(\Omega) \text{ such that } \varphi|_\Gamma = \nabla\varphi|_\Gamma \cdot \boldsymbol{n} = 0\}\,.$$

Thanks to this space, the following sequence is exact on simple connected domains

$$0 \xrightarrow{\quad i \quad} \Phi(\Omega) \xrightarrow{\quad \mathbf{curl} \quad} \mathbf{V}(\Omega) \xrightarrow{\quad \mathrm{div} \quad} Q(\Omega) \xrightarrow{\quad 0 \quad} 0\,, \tag{5.4}$$

where i is the operator that associates to each real number the corresponding constant function. The term *exact* means that the range of each operator is equal to the kernel of the following operator. In particular the following equivalence holds

$$\mathbf{curl}(\Phi(\Omega)) = \mathbf{Z}(\Omega)\,.$$

Problem (5.2) is another example of problem in which the discretization needs a stabilization when the advective field becomes dominant with respect to the diffusive coefficient. Similarly to the previous chapters, we work under the assumption that the parameters are scaled such that

$$\|\boldsymbol{\beta}\|_{[L^\infty(\Omega)]^2} = 1\,.$$

In the following sections, we describe how devise a VEM that is stable in the hyperbolic limit of Oseen equation.

5.2 The method

5.2.1 Mesh assumptions

We keep the same notations used for the previous chapters. We consider a sequence of meshes $\{\Omega_h\}_h$ composed by non overlapping polygons E that fulfills the following assumptions:
(AO) Mesh assumption. There exists a positive constant ρ such that for any $E \in \{\,\Omega_h\,\}_h$:

- E is star-shaped with respect to a ball B_E of radius grater or equal than $\rho\, h_E$,
- any edge e of E has length grater or equal than $\rho\, h_E$.

Unlike in the previous two chapters, we do not impose the quasi-uniformity of the mesh. This is because the analysis of the method is still incomplete, and it is unclear whether this assumption is required. We expect that, at some stage of the analysis, we may need to use an Oswald interpolant similar to those discussed in the previous chapters, and quasi-uniformity might be necessary.

In this chapter we use this definition of the jump operator: given an interior edge $e \subset \partial E \cap \partial K$, we define for a sufficiently smooth function w

$$[\![w]\!] = \lim_{s\to 0} w(\mathbf{x} - s\boldsymbol{n}^E) + w(\mathbf{x} - s\boldsymbol{n}^K)\,.$$

If e is a boundary edge, we set $[\![w]\!] = 0$. Another important distinction from the previous two chapters is that we were dealing with scalar functions. This involved using scalar polynomial projectors on scalar functions or vector projectors on their gradients. To differentiate between scalar and vector projectors, we used the symbols Π and $\mathbf{\Pi}$ respectively. Now, since we are working with vector-valued functions, the symbol Π denotes a projection applied to a vector-valued function, while the bold symbol $\mathbf{\Pi}$ represents a projection operator applied to a 2×2 tensor. For completeness, we remind the reader that Lemma 1.4 holds for vector-valued functions without any additional assumptions.

5.2.2 Virtual element spaces

Given an integer $k \geq 2$ and an element $E \in \Omega_h$, we introduce the enhanced virtual element space

$$
\begin{aligned}
\mathbf{V}_h^k(E) := \Big\{ \mathbf{v}_h \in [C^0(\bar{E})]^2 \text{ s.t. } & (i)\, \boldsymbol{\Delta} \mathbf{v}_h + \nabla s \in \mathbf{x}^\perp \mathbb{P}_{k-1}(E), \quad \text{for some } s \in L_0^2(E)\,, \\
& (ii)\, \mathrm{div}(\mathbf{v}_h) \in \mathbb{P}_{k-1}(E)\,, \\
& (iii)\, \mathbf{v}_h|_e \in [\mathbb{P}_k(e)]^2\,, \quad \forall e \in \partial E\,, \\
& (iv)\, (\mathbf{v}_h - \Pi_k^{\nabla,E} \mathbf{v}_h, \mathbf{x}^\perp \hat{p}_{k-1})_{0,E} = 0\,, \quad \forall \hat{p}_{k-1} \in \hat{\mathbb{P}}_{k-1/k-3}(E) \Big\}\,,
\end{aligned}
\tag{5.5}
$$

where the vector $\mathbf{x}^\perp$ is defined as $\mathbf{x}^\perp := [x_2; -x_1]^T$. we consider the following set of linear operator as set of DoFs for the space $\mathbf{V}_h^k(E)$:

- the pointwise values of $\mathbf{v}_h$ at the vertices of the polygon E,
- the values of $\mathbf{v}_h$ at $k-1$ internal points of a Gauss-Lobatto quadrature for every edge $e \subset \partial E$,
- the moments of $\mathbf{v}_h$
$$\frac{1}{|E|} \int_E \mathbf{v}_h \cdot \mathbf{m}^\perp \left(\frac{\boldsymbol{x} - \boldsymbol{x}_E}{h_E} \right)^{\boldsymbol{\alpha}} \mathrm{d}E \quad |\boldsymbol{\alpha}| \leq k-3\,,$$
where $\mathbf{m}^\perp := \frac{1}{h_E}(x_2 - x_{2,E}, -x_1 + x_{1,E})$,
- the moments of $\mathrm{div}\, \mathbf{v}_h$
$$\frac{h_E}{|E|} \int_E \mathrm{div}\, \mathbf{v}_h \left(\frac{\boldsymbol{x} - \boldsymbol{x}_E}{h_E} \right)^{\boldsymbol{\alpha}} \mathrm{d}E \quad 0 < |\boldsymbol{\alpha}| \leq k-1\,.$$

The global space for the velocities is obtained as in the scalar case by gluing together the local spaces

$$\mathbf{V}_h^k(\Omega_h) := \Big\{ \mathbf{v}_h \in \mathbf{V} \text{ such that } \mathbf{v}_h|_E \in \mathbf{V}_h^k(E) \quad \forall E \in \Omega_h \Big\}\,.$$

The space of pressures, is discretized by the standard piecewise polynomials space

$$Q_h^k(\Omega_h) := \Big\{ q_h \in L^2(\Omega) \text{ such that } q_h|_E \in \mathbb{P}_{k-1}(E) \quad \forall E \in \Omega_h \Big\}\,.$$

In [18], it was proved that the couple $[\mathbf{V}_h^k(\Omega_h),\, Q_h^k(\Omega_h)]$ is inf-sup stable. It holds that

$$\sup_{\mathbf{v}_h \in \mathbf{V}_h^k(\Omega_h)} \frac{b(\mathbf{v}_h, q_h)}{\|\boldsymbol{\nabla} \mathbf{v}_h\|_{0,\Omega_h}} \geq \hat{\kappa} \|q_h\|_{0,\Omega_h} \quad \forall q_h \in Q_h^k(\Omega_h)\,,$$

where $\hat{\kappa}$ denotes the inf-sup constant that does not depend on the mesh size h. We now introduce the discrete kernel as

$$\mathbf{Z}_h(\Omega_h) := \{ \mathbf{v}_h \in \mathbf{V}_h^k(\Omega_h) \text{ such that } b(\mathbf{v}_h, q_h) = 0 \quad \forall q \in Q_h^k(\Omega_h) \}.$$

Thanks to property (ii) in (5.5), the following inclusion holds

$$\mathbf{Z}_h(\Omega_h) \subseteq \mathbf{Z}(\Omega)\,.$$

This means that the virtual functions in the discrete kernel are divergence-free. Now we construct the space of the discrete stream-functions. We define

$$
\begin{aligned}
\Phi_h^k(E) := \Big\{ \varphi \in [C^1(\bar{E})]^2 \text{ such that } & (i)\, \Delta^2 \varphi \in \mathbb{P}_{k-1}(E)\,, \\
& (ii)\, \varphi|_e \in \mathbb{P}_{k+1}(e)\,, \quad \forall e \in \partial E\,, \\
& (iii)\, \nabla \varphi|_e \in [\mathbb{P}_k(e)]^2\,, \quad \forall e \in \partial E\,, \\
& (iv)\, (\mathbf{curl}\varphi - \Pi_k^{\nabla,E} \mathbf{curl}\varphi, \mathbf{x}^\perp \hat{p}_{k-1})_{0,E} = 0 \quad \forall \hat{p}_{k-1} \in \hat{\mathbb{P}}_{k-1/k-3}(E) \Big\}\,,
\end{aligned}
$$

and the corresponding global space

$$\Phi_h^k(\Omega_h) := \{\varphi \in \Phi(\Omega) \text{ such that } \varphi|_E \in \Phi_h^k(E) \quad \forall E \in \Omega_h\}\,.$$

We have constructed an exact subcomplex of (5.4)

$$0 \xrightarrow{\ i\ } \Phi_h^k(\Omega_h) \xrightarrow{\ \mathbf{curl}\ } \mathbf{V}_h^k(\Omega_h) \xrightarrow{\ \mathrm{div}\ } Q_h^k(\Omega_h) \xrightarrow{\ 0\ } 0\,.$$

Finally, we recall from [15] two approximation results for the space $\mathbf{V}_h^k(\Omega_h)$ and the space $\Phi_h^k(\Omega_h)$ respectively.

Lemma 5.1 (Approximation with divergence-free virtual element functions)**.** *Under assumption **(AO)**, for any $\mathbf{v} \in \mathbf{V}(\Omega) \cap [H^{s+1}(\Omega_h)]^2$, there exists $\mathbf{v}_{\mathcal{I}} \in \mathbf{V}_h^{k,nc}(\Omega_h)$, such that for all $E \in \Omega_h$,*

$$\|\mathbf{v} - \mathbf{v}_{\mathcal{I}}\|_{0,E} + h_E\|\boldsymbol{\nabla}(\mathbf{v} - \mathbf{v}_{\mathcal{I}})\|_{0,E} \lesssim h_E^{s+1}|v|_{s+1,E}\,,$$

where $0 < s \leq k$.

Lemma 5.2 (Approximation property of $\Phi_h^k(\Omega_h)$)**.** *Under assumption **(AO)**, for any $\phi \in \Phi(\Omega) \cap H^{s+2}(\Omega_h)$ there exists $\varphi_{\mathcal{I}} \in \Phi_h^k(\Omega_h)$ such that for all $E \in \Omega_h$ it holds*

$$\|\varphi - \varphi_{\mathcal{I}}\|_{0,E} + h_E|\varphi - \varphi_{\mathcal{I}}|_{1,E} + h_E^2|\varphi - \varphi_{\mathcal{I}}|_{0,E} \lesssim h_E^{s+2}|\varphi|_{s+2,E}\,,$$

where $0 < s \leq k$.

5.2.3 Virtual element forms and the discrete problem

This section aims to construct a computable counterpart of the forms introduced in Section 5.1. The approach is similar to the methods used in the previous chapters, but in this case, we need to take into account projections applied to 2×2 tensors or vector-valued functions. As always, we recall that the forms introduced in Section 5.1 can be decomposed into local contributions by restricting the integral to an element $E \in \Omega_h$

$$A(\mathbf{u}, \mathbf{v}) =: \sum_{E \in \Omega_h} A^E(\mathbf{u}, \mathbf{v})\,, \qquad c^{\text{skew}}(\mathbf{u}, \mathbf{v}) =: \sum_{E \in \Omega_h} c^{\text{skew},E}(\mathbf{u}, \mathbf{v})\,,$$
$$b(\mathbf{v}, q) =: \sum_{E \in \Omega_h} b^E(\mathbf{v}, q)\,, \qquad \mathcal{F}(\mathbf{v}) =: \sum_{E \in \Omega_h} \mathcal{F}^E(\mathbf{v})\,.$$

The bilinear form $A^E(\cdot,\cdot)$ is discretized by $A_h^E(\cdot,\cdot) : \mathbf{V}_h^k(E) \times \mathbf{V}_h^k(E) \to \mathbb{R}$ defined as

$$\begin{aligned} A_h^E(\mathbf{u}_h, \mathbf{v}_h) := \nu \int_E \boldsymbol{\Pi}_{k-1}^{0,E}\boldsymbol{\varepsilon}(\mathbf{u}_h) : \boldsymbol{\Pi}_{k-1}^{0,E}\boldsymbol{\varepsilon}(\mathbf{v}_h)\,\mathrm{d}E + \sigma \int_E \Pi_k^{0,E}\mathbf{u}_h \cdot \Pi_k^{0,E}\mathbf{v}_h\,\mathrm{d}E \\ + \left(\nu + \sigma\, h_E^2\right) \mathcal{S}^E\left((I - \Pi_k^{0,E})\mathbf{u}_h, (I - \Pi_k^{0,E})\mathbf{v}_h)\right)\,, \end{aligned}$$

where $\mathcal{S}^E(\cdot,\cdot) : \mathbf{V}_h^k(E) \times \mathbf{V}_h^k(E) \to \mathbb{R}$ is a VEM stabilization term that satisfies

$$\alpha_*\|\boldsymbol{\nabla}\mathbf{v}_h\|_{0,E} \leq S_h^E(\mathbf{v}_h, \mathbf{v}_h) \leq \alpha^*\|\boldsymbol{\nabla}\mathbf{v}_h\|_{0,E} \quad \forall \mathbf{v}_h \in \mathbf{V}_h^k(E) \cap \ker(\Pi_k^{0,E})\,,$$

where α_* and α_* are two uniform constants. The convective term is replaced by $c_h^E(\cdot,\cdot)$: $\mathbf{V}_h^k(E) \times \mathbf{V}_h^k(E) \to \mathbb{R}$ defined as

$$c_h^E(\mathbf{u}_h, \mathbf{v}_h) := \int_E \left[\left(\boldsymbol{\Pi}_k^{0,E}\boldsymbol{\nabla}\mathbf{u}_h\right)\boldsymbol{\beta}\right] \cdot \Pi_k^{0,E}\mathbf{v}_h\,\mathrm{d}E\,.$$

We consider the skew-part of this bilinear form

$$c_h^{\text{skew},E}(\mathbf{u}_h, \mathbf{v}_h) := \frac{1}{2}\left(c_h^E(\mathbf{u}_h, \mathbf{v}_h) - c_h^E(\mathbf{v}_h, \mathbf{u}_h)\right)\,.$$

In order to stabilize the method, we introduce in the formulation of the method a three level CIP term. First, we introduce the operator

$$(\mathcal{B}\mathbf{v})|_E := \big(\mathrm{curl}\,(\boldsymbol{\nabla}\mathbf{v})\,\boldsymbol{\beta}\big) \quad \forall E \in \Omega_h\,,$$

defined as

$$J_h^E(\mathbf{u}_h, \mathbf{v}_h) := \sum_{i=1}^{3} \delta_i\, J_h^{E,i}(\mathbf{u}_h, \mathbf{v}_h) + (\delta_1 + \delta_2 + \delta_3)\, h_E\, \mathcal{S}^E\big((I - \Pi_k^{0,E})\mathbf{u}_h, (I - \Pi_k^{0,E})\mathbf{v}_h)\big)$$

where

$$\begin{aligned}
J_h^{E,1}(\mathbf{u}_h, \mathbf{v}_h) &:= \frac{1}{2}\int_{\partial E} h_E^2 \big[\!\big[(\boldsymbol{\Pi}_k^{0,E}\boldsymbol{\nabla}\mathbf{u}_h)\boldsymbol{\beta} \times \boldsymbol{n}\big]\!\big]\,\big[\!\big[(\boldsymbol{\Pi}_k^{0,E}\boldsymbol{\nabla}\mathbf{v}_h)\boldsymbol{\beta} \times \boldsymbol{n}\big]\!\big]\,\mathrm{d}s\,,\\
J_h^{E,2}(\mathbf{u}_h, \mathbf{v}_h) &:= \frac{1}{2}\int_{\partial E} h_E^4 \big[\!\big[\mathcal{B}\boldsymbol{\Pi}_k^{0,E}\mathbf{u}_h\big]\!\big]\,\big[\!\big[\mathcal{B}\boldsymbol{\Pi}_k^{0,E}\mathbf{v}_h\big]\!\big]\,\mathrm{d}s\,,\\
J_h^{E,3}(\mathbf{u}_h, \mathbf{v}_h) &:= \frac{1}{2}\int_{\partial E} h_E^6 \big[\!\big[\nabla\mathcal{B}\boldsymbol{\Pi}_k^{0,E}\mathbf{u}_h\big]\!\big]\,\big[\!\big[\nabla\mathcal{B}\boldsymbol{\Pi}_k^{0,E}\mathbf{v}_h\big]\!\big]\,\mathrm{d}s\,.
\end{aligned}$$

Thanks to the DoFs of $\mathbf{V}_h^k(\Omega_h)$, it is not necessary to introduce a discretization of $b(\cdot,\cdot)$. Now we introduce the local bilinear form

$$\mathcal{K}_h^E(\mathbf{u}_h, \mathbf{v}_h) := A_h^E(\mathbf{u}_h, \mathbf{v}_h) + c_h^{\mathrm{skew},E}(\mathbf{u}_h, \mathbf{v}_h) + J_h^E(\mathbf{u}_h, \mathbf{v}_h)\,.$$

The right-hand side is discretized by

$$\mathcal{F}_h^E(\mathbf{v}_h) := \int_E \mathbf{f} \cdot \Pi_k^{0,E}\mathbf{v}_h\,\mathrm{d}E\,.$$

Similarly to the previous chpaters, we introduce the global versions of the bilenar forms by summing over all the polygons $E \in \Omega_h$

$$\begin{aligned}
A_h(\mathbf{u}_h, \mathbf{v}_h) &:= \sum_{E\in\Omega_h} A_h^E(\mathbf{u}_h, \mathbf{v}_h)\,, & c_h^{\mathrm{skew}}(\mathbf{u}_h, \mathbf{v}_h) &:= \sum_{E\in\Omega_h} c_h^{\mathrm{skew},E}(\mathbf{u}_h, \mathbf{v}_h)\,,\\
J_h(\mathbf{u}_h, \mathbf{v}_h) &:= \sum_{E\in\Omega_h} J_h^E(\mathbf{u}_h, \mathbf{v}_h)\,, & \mathcal{F}_h(\mathbf{v}_h) &:= \sum_{E\in\Omega_h} \mathcal{F}_h^E(\mathbf{v}_h)\,,
\end{aligned}$$

and

$$\mathcal{K}_h(\mathbf{u}_h, \mathbf{v}_h) := \sum_{E\in\Omega_h} \mathcal{K}_h^E(\mathbf{u}_h, \mathbf{v}_h)\,.$$

The discrete problem reads as

$$\begin{cases}
\text{find } (\mathbf{u}_h, p_h) \in [\mathbf{V}_h^k(\Omega_h), Q_h^k(\Omega_h)] \text{ such that:}\\
\mathcal{K}_h(\mathbf{u}_h,\, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = \mathcal{F}_h(\mathbf{v}_h) & \forall \mathbf{v}_h \in \mathbf{V}_h^k(\Omega_h),\\
b(\mathbf{u}_h, q_h) = 0 & \forall q_h \in Q_h^k(\Omega_h),
\end{cases} \tag{5.6}$$

or, in a divergence-free formula

$$\begin{cases}
\text{find } \mathbf{u}_h \in \mathbf{Z}_h(\Omega_h) \text{ such that:}\\
\mathcal{K}_h(\mathbf{u}_h,\, \mathbf{v}_h) = \mathcal{F}_h(\mathbf{v}_h) \quad \forall \mathbf{v}_h \in \mathbf{Z}_h(\Omega_h).
\end{cases}$$

5.3 Theoretical analysis

We introduce the norm

$$\|\mathbf{v}_h\|^2_{\mathcal{K},E} := \nu\|\boldsymbol{\nabla}\mathbf{v}_h\|^2_{0,E} + \sigma\|\mathbf{v}_h\|^2_{0,E} + J_h^E(\mathbf{v}_h, \mathbf{v}_h)\,,$$

with global counterpart

$$\|\mathbf{v}_h\|^2_{\mathcal{K}} := \left(\sum_{E\in\Omega_h} \|\mathbf{v}_h\|^2_{\mathcal{K},E}\right)^{\frac{1}{2}}\,.$$

The well-posedness of (5.6) is guaranteed by this coercivity result.

Proposition 5.3. *Under assumption (**AO**), it holds that*

$$\|\mathbf{v}_h\|_{\mathcal{K}}^2 \lesssim \mathcal{K}_h(\mathbf{v}_h, \mathbf{v}_h) \quad \forall \mathbf{v}_h \in \mathbf{V}_h^k(\Omega_h)\,.$$

Proof. We restrict our attention to an element $E \in \Omega_h$. First, we note that since the bilinear form $c_h^{\text{skew},E}$ is skew-symmetric, it holds that

$$c_h^{\text{skew},E}(\mathbf{v}_h, \mathbf{v}_h) = 0\,.$$

Thanks to to property of the VEM-stabilization term, we have

$$A_h^E(\mathbf{v}_h, \mathbf{v}_h) \gtrsim \nu\, \|\boldsymbol{\nabla} \mathbf{v}_h\|_{0,E}^2 + \sigma\, \|\mathbf{v}_h\|_{0,E}^2\,.$$

By definition of the bilinear form $\mathcal{K}_h^E(\cdot,\cdot)$, it is clear that

$$\mathcal{K}_h^E(\mathbf{v}_h, \mathbf{v}_h) \gtrsim \|\mathbf{v}_h\|_{\mathcal{K},E}^2\,.$$

The proof is completed by summing over all the elements $E \in \Omega_h$. □

The error analysis for this method has not yet been developed and will be the subject of future research. We expect that the error analysis for the velocity can be conducted by leveraging the coercivity of the bilinear problem and following a procedure similar to the ones used in the previous two methods. Specifically, the norm of the error $\mathbf{e} = \mathbf{u} - \mathbf{u}_h$ can be decomposed as follows

$$\|\mathbf{u} - \mathbf{u}_h\|_{\mathcal{K}} \leq \|\mathbf{u} - \mathbf{u}_{\mathcal{I}}\|_{\mathcal{K}} + \|\mathbf{u}_{\mathcal{I}} - \mathbf{u}_h\|_{\mathcal{K}}\,.$$

We expect that the interpolation error will be controlled similarly to Lemma 3.20 and Lemma 4.13, considering each component in the definition of the norm $\|\cdot\|_{\mathcal{K}}$. To control the second term, we will exploit the coercivity of the bilinear form $\mathcal{K}_(\cdot,\cdot)$ and estimate each term in its definition. The approximation error between $A(\cdot,\cdot)$ and $A_h(\cdot,\cdot)$ can be handled by extending the procedures of Lemma 3.22 and Lemma 3.24 to vector valued functions. Under the assumption $(\boldsymbol{\nabla}\mathbf{u}_h)\boldsymbol{\beta} \in H^{\frac{5}{2}+\epsilon}$ for some $\epsilon > 0$, the error estimate for $J_h(\cdot,\cdot)$ follows from trace inequalities and approximation results. We foresee that the term $c^{\text{skew}}(\cdot,\cdot)$ will present the most challenges. In the analysis, it may be necessary to introduce an Oswald interpolant of the function

$$(\boldsymbol{\nabla}\Pi_k^0 \mathbf{u}_h)\boldsymbol{\beta}_h$$

into the space $\Phi_h^k(\Omega_h)$ where $\boldsymbol{\beta}_h$ is an appropriate approximation of $\boldsymbol{\beta}$. Then, we control the difference between $(\boldsymbol{\nabla}\Pi_k^0\mathbf{u}_h)\boldsymbol{\beta}_h$ and its interpolation by considering the jumps contained in the definition of $J_h(\cdot,\cdot)$. With respect to pressure, we expect that the error analysis can be performed in a way similar to other problems involving this type of equation.

5.4 Numerical results

Although we do not have an accurate error analysis for this method, we present some numerical results because the method has already been implemented in MATLAB. We consider a family of problems in the unit square $\Omega = (0,1) \times (0,1)$ and the following error quantities for the velocity will be considered:

- H^1**−seminorm error**

$$e_{H^1} := \sqrt{\sum_{E\in\Omega_h} \left\|\boldsymbol{\nabla}(\mathbf{u} - \Pi_k^{\nabla}\mathbf{u}_h)\right\|_{0,E}^2}\,,$$

- L^2**−norm error**

$$e_{L^2} := \sqrt{\sum_{E\in\Omega_h} \left\|\mathbf{u} - \Pi_k^0\mathbf{u}_h\right\|_{0,E}^2}\,.$$

For the pressures, we consider the quantity

$$e_p := \sqrt{\sum_{E\in\Omega_h} \left\|p - p_h\right\|_{0,E}^2}\,.$$

We consider a family of Voronoi meshes such as the one depicted in Figure 3.1.

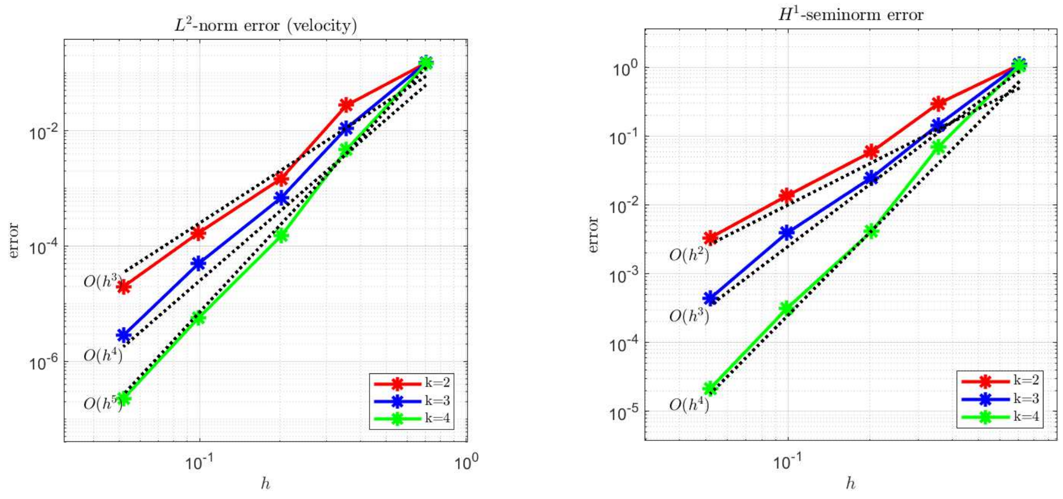


Figure 5.1: Convergences results for the velocity $\mathbf{u}$ in the L^2 norm (left column) and in the H^1-seminorm (right column). The red lines correspond to the case $k = 1$, the blue lines to the case $k = 2$, and the green lines to the case $k = 3$.

Convergence analysis. We consider as solution of (5.1) the couple $(\mathbf{u}, p)$ defined as

$$\mathbf{u}(x, y) := \begin{bmatrix} -\frac{1}{2}\sin(\pi x)^2\cos(\pi y)\sin(\pi y) \\ \frac{1}{2}\sin(\pi y)^2\cos(\pi x)\sin(\pi x) \end{bmatrix},$$

and

$$p(x, y) := 3\sin(x) - 3\sin(y)\,.$$

We choose as parameters of the method the values $\nu = 10^{-5}$, $\sigma = 1$ and the advective field defined as

$$\boldsymbol{\beta}(x, y) := \begin{bmatrix} \sin(2\pi x)\sin(2\pi y) \\ \cos(2\pi x)\cos(2\pi y) \end{bmatrix}.$$

The CIP-parameters are set equal to $\delta_1 = 0.1$ and $\delta_2 = \delta_3 = 0.01$. We solve the equation using virtual element of order $k = 2, 3, 4$. The results are depicted in Figure 5.1 and Figure 5.2. We can note that we have reached the optimal rate of convergence for the H^1-seminorm of the velocity and the L^2-norm of the pressure. Actually, the method converges with a rate of h^{k+1} in the L^2-norm of the velocity, which is a better result than the theoretical estimate for these types of problems that is $h^{k+\frac{1}{2}}$. This might be due to the solution being very smooth.

Solution with a boundary layer We consider a solution with a boundary layer. We select as solution of the problem (5.1) the velocity $\mathbf{u} = (u_1, u_2)^T$ defined as

$$u_1(x, y) = 0\,, \qquad u_2(x, y) = x - \frac{\exp(\frac{x-1}{\nu}) - \exp(\frac{-1}{\nu})}{1 - \exp(\frac{-1}{\nu})}\,,$$

and the pressure

$$p(x, y) = \frac{1}{2} - y\,.$$

We choose $\nu = 10^{-9}$, $\sigma = 1$ and the advective field

$$\boldsymbol{\beta}(x, y) := \begin{bmatrix} y^2 \\ x^2 \end{bmatrix}.$$

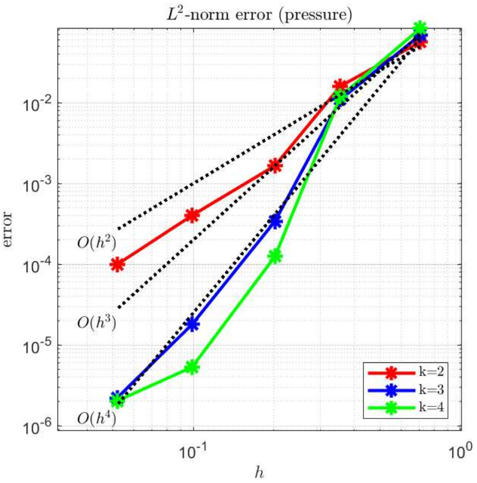


Figure 5.2: Convergences for the pressure p in the L^2-norm. The red lines correspond to the case $k = 1$, the blue lines to the case $k = 2$, and the green lines to the case $k = 3$.

To demonstrate the advantages of the CIP term, we consider different values for δ_1, δ_2 and δ_3. The problem is solved with $k = 2$ on a mesh consisting of 1024 unit squares. A square mesh was chosen to better visualize the numerical solutions. The results are shown in Figure 5.3. We observe that without any CIP term, the numerical solution does not match the analytical solution. By adding the first level of CIP, we obtain a numerical solution that includes a boundary layer and some spurious oscillations near the layer. These oscillations are reduced with the addition of the second level of CIP. No further improvement is observed when the final level of CIP is added.

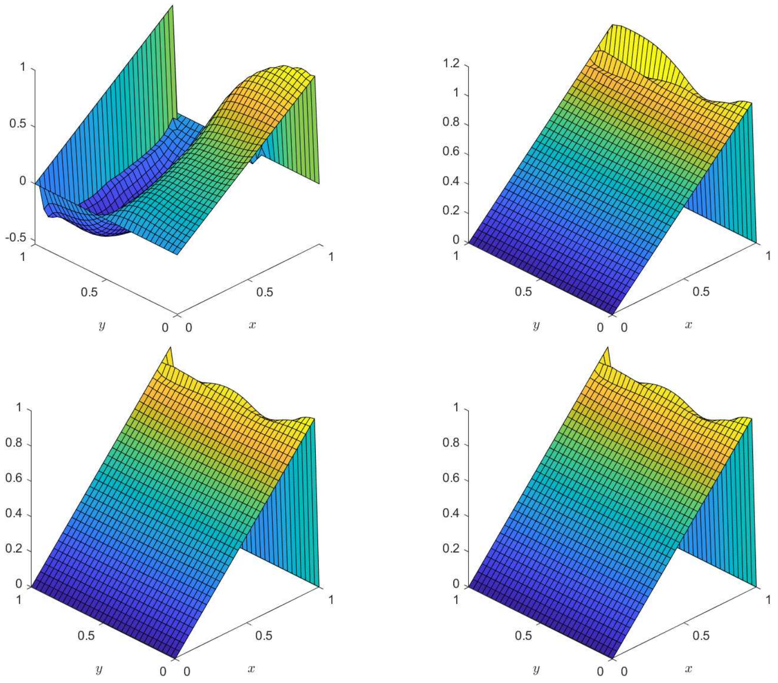


Figure 5.3: Numerical solutions obtained with different choices of the triple $(\delta_1, \delta_2, \delta_3)$.Top-left $(\delta_1, \delta_2, \delta_3) = (0, 0, 0)$, top-right $(\delta_1, \delta_2, \delta_3) = (0.1, 0, 0)$, bottom-left $(\delta_1, \delta_2, \delta_3) = (0.1, 0.01, 0)$, bottom-right $(\delta_1, \delta_2, \delta_3) = (0.1, 0.01, 0.001)$.

Pressure robustness. We want to verify if the method is pressure-robust in the VEM sense. We consider the solution $(\mathbf{u}, p)$ of (5.1) defined as

$$\mathbf{u}(x, y) = \underline{0}\,, \qquad p(x, y) = 3\cos(x) - 3\cos(y)\,.$$

The parameters are set as in the first test case and select the order of the method $k = 2, 3, 4$. Since velocity $\mathbf{u}$ belongs to the discrete space, if the method is pressure robust, we expect to obtain an error of the order of the precision of the machine. The VEM introduces in the error analysis of the velocity a little dependence on the pressure due to the approximation of the right-hand side. This is a typical situation in VEM for this type of problems. In Figure 5.4, we note that the errors are not of the order of the machine precision but they are quite low if compared to the previous cases.

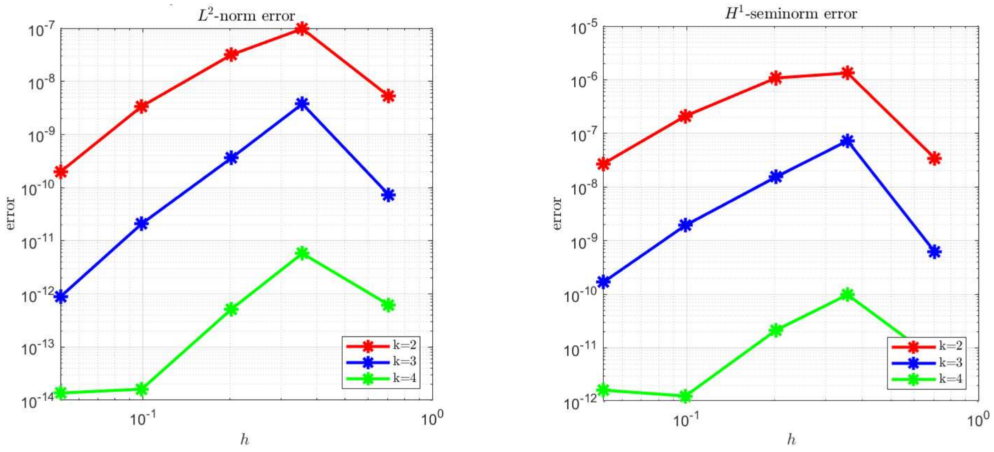


Figure 5.4: Result for the L^2−norm of the error (left column) and the H^1−seminorm of the error (right column). The red lines correspond to the case $k = 1$, the blue lines to the case $k = 2$, and the green lines to the case $k = 3$.

Simulation of a fluid inside a channel with two pipes. In this test, we assume that a fluid, like water, is moving from left to right inside a channel, driven by an imposed flow velocity. This channel contains two cylindrical obstacles, modeled as pipes positioned at different locations along the channel's length. The domain is the same used in Section 3.3. For simplicity, we assume that the flow velocity in the channel is uniform and directed along the horizontal axis, given by $\boldsymbol{\beta} = (1, 0)^T$. We consider a scenario with low diffusion, setting the diffusion coefficient ν to 10^{-5}, while neglecting the reaction term. To ensure stability in the numerical solution, we apply the CIP stabilization method with parameters set to $(\delta_1, \delta_2, \delta_3) = (0.1, 0.01, 0)$. No-slip boundary conditions are imposed along the top and bottom boundaries of the channel, as well as on the surfaces of the pipes. On the left boundary, we prescribe an inflow condition with a parabolic velocity profile of the form

$$-10\,(y - 0.5)\,(y + 0.5)\,.$$

Finally, on the right boundary, we impose homogeneous Neumann boundary conditions to allow the fluid to exit the domain without further interaction. The numerical solution in Figure 5.5 shows that the fluid retains its shape as it moves towards the first pipe. Upon reaching the first obstacle, the fluid divides and flows both above and below the pipe, maintaining a symmetric shape. This behavior is repeated at the second pipe, with the fluid continuing its trajectory in a similar pattern until it reaches the right boundary. We also tested the setup with square-based pipes to examine whether the corners would produce singularities, but we did not observe any significant differences. We note that the following test cases do not fit within our theoretical analysis, as the domain contains two holes. Despite this, we attempt to assess the accuracy of our method. This test case is inspired by the one that appears in [53].

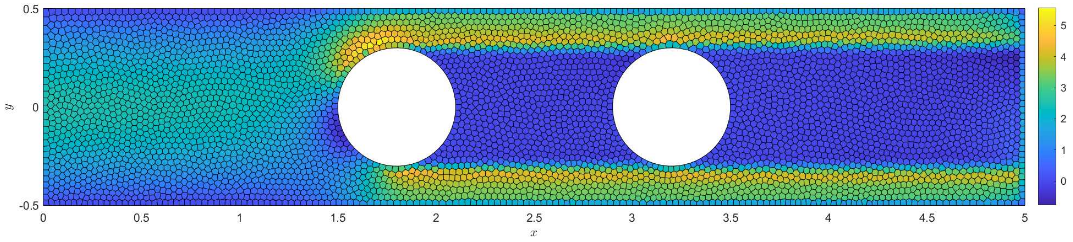


Figure 5.5: Numerical representation of a fluid that moves inside a channel with two pipes. The color represent the norm of the velocity.

Chapter 6

Implementation of the code

This chapter focuses on describing the implementation of the codes. We will primarily concentrate on the method presented in Chapter 3, as the implementations for the other methods are quite similar. The codes were fully implemented in MATLAB by the authors, except for the function `read_mesh()`, which was sourced from the VEMLAB library[1]. We chose to use their meshes and its associated reader instead of developing our own, allowing us to focus on other aspects of the implementation.

This chapter discusses the implementation of the main file and several auxiliary functions. It is organized as follows: the first section describes the preprocessing steps, the second explains the assembly of the global matrix, while the third focuses on the construction of the local matrices. The final two sections cover the solution of the linear system and the computation of the error, respectively.

6.1 Preprocessing

We begin by clearing the memory, closing any open figures, and clearing the command window. Next, we capture the start time using the `tic` function. We then use the `fixpath()` function to add the paths of all folders containing the necessary functions. The `fixpath()` function does not require any input parameters and internally calls `addpath()` for each of the folders needed in the implementation.

```
%% INITIALIZATION
clear; close; clc; tic;

fixpath();
```

Following the initialization, we construct a structure `options`. The fields within this structure contain information about the PDE and its discretization. In details:

- `epsilon` and `sigma`: contain the values of the diffusive coefficient and the reaction coefficient, respectively,
- `beta{1}` and `beta{2}`: contain the value of the two components of the advective field,
- `gamma` and `gamma_p`: contain the values of $\kappa_e = \kappa_E$, see (3.10). The field `gamma_p` is needed if we implement the jump term as in (3.78),
- `delta`: contains the value of the Nitsche parameter,
- `stiffness`: describes in which way we want to discretize the bilinear form $a^E(\cdot, \cdot)$. If the value of this field is `"PiNabla"`, we are using (1.9); if it is `"PiBar"`, we are using (3.7),
- `stabilization`: describes which VEM stabilization term we are considering,

[1] `https://camlab.cl/vemlab/`

- `advection`: describes how we discretize the bilinear form $b^E(\cdot,\cdot)$. If its value is equal to `"Boundary"`, we are using the discretization that appears in Chapter 3. If it is equal to `"Zero"`, we are using

$$b_h^E(u_h, v_h) = \int_E \boldsymbol{\beta} \cdot \boldsymbol{\Pi}_k^{0,E} u_h \, \Pi_k^{0,E} v_h \, \mathrm{dE} \, ,$$

- `jump`: describe which jump term term we are considering. The choice is between $J_h^E(\cdot,\cdot)$, $K_h^E(\cdot,\cdot)$ and $L_h^E(\cdot,\cdot)$,

- `boundary`: indicate how we impose the boundary condition (strongly or weakly).

```
%% OPTIONS & PARAMETERS
options.problem = "ADR";

%Parameters of the PDE
options.epsilon = 10^(-5);
options.sigma   = 1;
options.beta{1} = @(x,y) 1 + 0.*x + 0.*y;
options.beta{2} = @(x,y) 2 + 0.*x + 0.*y;
options.gamma   = 0.025; options.gamma_p = 0.01;
options.delta   = 0.1;

%Options for the discretization
options.stiffness      = "PiBar";
options.stabilization  = "DofiDofi";
options.advection      = "Boundary";
options.jump           = "Normal";
options.boundary       = "Nitsche";
```

After that, we call the function `problem_test(n, options)`. This function has two input parameters:

- `n`: this parameter indicates the number of the problem that we are solving. Each number corresponds to a specific solution u with its corresponding f,

- `options`: this parameter is a structure containing the values of the parameters in the PDE.

It returns as output parameters the solution u, the gradient of the solution, the load term f and the boundary data g.

```
%% CONSTRUCTION OF THE PDE
[f, g, u, grad_u] = test_problems(1, options);
```

The initialization of the method continues by specifying the order k of the VEM and creating a structure `polynomial`. This structure contains information about the polynomials that we will frequently use in the implementation. It is designed to speed up the code by avoiding repeated computations of frequently requested quantities. This task is handled by the function `get_polynomial_info(k)`, which takes the order of the method k as its input parameter. Some of the fields of the structure `polynomial` are

- `dim`: an array containing the dimensions of $\mathbb{P}_n(E)$ for $n = k-2, k-1, k$,

- `lobatto`: the positions of k points in the interval $[0,1]$ for a Gauss-Lobatto quadrature rule,

- `gauss`: the positions of k points in the interval $[0,1]$ for a Gauss quadrature rule,

- `lweights`: the weights for the Gauss-Lobatto quadrature rule,

- `gweights`: the weights for the Gauss quadrature rule,
- `val`: the values of the basis functions at the Gauss quadrature points in the interval $[0, 1]$. These basis functions are polynomial of order k equal to 1 at a Gauss-Lobatto quadrature point and zero at the others.

```
%% CONSTRUCT THE INFORMATION ON THE POLYNOMIALS
k           = 1;
polynomial = get_polynomial_info(k);
```

We conclude this section of the code by defining the mesh Ω_h. The name of the mesh is stored in the variable `mesh`. We then call the function `read_mesh(mesh)`, which creates a structure `Mesh` containing information about the mesh we are using. This is the only function sourced from the VEMLAB library. We did not develop this function ourselves because we are still using their mesher and chose not to focus on implementing our own mesher and its associated reader.

The function `add_points(Mesh, n)` allows for the addition of n equidistributed points along each edge of the mesh. This function is used to construct the `octag` meshes described in Section 3.3 from a quadrilateral mesh. If $n = 0$, the function does not perform any action.

The function `add_edges(Mesh, k)` inserts fields into the structure `Mesh` related to the edges of the mesh Ω_h and their connectivity. It also includes the coordinates of the points associated with the degrees of freedom in $\mathcal{E}_c^k$. This function provides information on the endpoints of each edge, which pairs of vertices are connected by edges, and the elements that contain each edge.

Next, we obtain the indices of the edges located on the boundary using the function `get_boundary_edges(Mesh)`. The indices of the internal edges are then determined by subtracting the array of boundary edge indices from the array of all edge indices.

Finally, we conclude this section by computing the indices of the DoFs located on the boundary. This step is essential if we choose to impose the boundary conditions strongly.

```
%% READ THE MESH
mesh = 'polygon_256.txt';
Mesh = read_mesh(mesh);

%% OBTAIN INFORMATION ON THE MESH
fprintf('[%.2f] Obtaining information on the mesh... \n',toc);

Mesh = add_points(Mesh,0);
Mesh = add_edges(Mesh, k);

Mesh.boundary_edges  = get_boundary_edges(Mesh);
Mesh.internal_edges  = setdiff(1:Mesh.nedges,Mesh.boundary_edges);

boundary_vertex  = Mesh.boundary_nodes.all;
[boundary_dofs, boundary_intdofs] = get_boundary_dofs(Mesh,
    boundary_vertex, k);
```

6.2 Assembly of the linear system

After the preprocessing part, we construct the linear system associated to the problem (3.14). We need to assemble the matrix `A` associated with the bilinear form

$$\mathcal{A}_{\text{cip}}(u_h, v_h) = \epsilon\, a_h(u_h, v_h) + b_h^{\text{skew}}(u_h, v_h) + \sigma\, c_h(u_h, v_h) + \mathcal{N}_h(u_h, v_h) + J_h(u_h, v_h), ,$$

as well as the right-hand side vector $\underline{\mathtt{F}}$ associated with the form $\mathcal{F}_h(\cdot)$. We then solve the linear system

$$\mathtt{A}\underline{\mathtt{U}} = \underline{\mathtt{F}},$$

where $\underline{\mathtt{U}}$ is the vector containing the DoFs of the numerical solution u_h. The volume terms

$$\epsilon\, a_h(u_h, v_h) + b_h^{\mathrm{skew}}(u_h, v_h) + \sigma\, c_h(u_h, v_h) + \mathcal{N}_h(u_h, v_h)$$

are assembled in the function `ADR_assembly`, while the jump term is assembled in the function `CIP_assembly`. We will provide a detailed description of these two functions.

```
%% ASSEMBLYING GLOBAL MATRIX
fprintf('[%.2f] Assemblying element matrices...\n',toc);
[A_global, f_global, Mesh] = ADR_assembly(Mesh, options,
   polynomial, f, g);

%% ASSEMBLYING CIP MATRIX
fprintf('[%.2f] Assemblying CIP matrix...\n',toc);
[J_global] = cip_assembly(Mesh, polynomial, options);
```

6.2.1 Assembly of a sparse matrix

Before describing the function `ADR_assembly`, it is necessary to describe an efficient way to assembly a sparse matrix in MATLAB. It is well known that the matrices obtained with the Galerkin method are often sparse matrices. Hence, to save memory, it is appropriate to initialize the matrix `A` as a sparse matrix and not with the command `zeros(dim)`. Recall that in MATLAB, a sparse matrix is stored using a three-column vector format. The columns of this vector represent:

1. the row indices of the nonzero elements,
2. the column indices of the nonzero elements,
3. the values of the nonzero elements.

This format is efficient for storing and manipulating sparse matrices, as it only records the positions and values of the nonzero entries. It is important to note that the conventional method for assembling a global matrix

$$\mathtt{A(index,index)} = \mathtt{A(index,index)} + \mathtt{A\_loc},$$

where `A_loc` is a local matrix from an element $E \in \Omega_h$ and `index` denotes the positions in the global matrix where `A_loc` is inserted, can be very slow if `A` is sparse. This inefficiency arises because at each iteration, MATLAB needs to sort the indices of the nonzero elements in `A`, which can become slow for large matrices. A more efficient procedure is described in [39]. The idea is to construct three arrays: `I`, `J`, and `X`. The array `I` stores the row indices, the array `J` stores the column indices, and the array `X` stores the values of nonzero entries. When updating the global matrix by adding the local matrix `A_loc`, we simply append the corresponding `I_loc`, `J_loc`, and `X_loc` (which represent the row indices, column indices, and values of `A_loc` in the global matrix, respectively) to the arrays `I`, `J`, and `X`. This approach efficiently manages matrix updates without needing to repeatedly sort indices. Then, the matrix `A` is constructed with the command

$$\mathtt{A} = \mathtt{sparse(I,J,X)}.$$

6.2.2 The function `ADR_assembly`

The function `ADR_assembly` takes as input parameters the structures `Mesh`, `options` and `polynomial` and the functions f and g. It returns three output parameters:

- the global matrix `A_global`. This is a sparse matrix and is stored as a structure with three fields: `I`, `J`, and `X`, as described in the previous section,
- the vector `f_global` corresponding to the right-hand side of the linear system that we want to solve,
- the structure Mesh that will be updated by this function.

First, we need to allocate the memory for all the elements that we need in the discretization of the problem. We compute the dimension of the linear system

$$N_{\Omega_h} = n_v + (k-1)|\mathcal{E}_h| + n_p \frac{(k-2)(k-1)}{2}\,,$$

and store the result in the variable `size`. After that, we allocate the memory for the vector $\underline{\mathtt{F}}$ with the command

`f_global = zeros(size, 1).`

In the structure `Mesh`, we initialize the field `diameter` which represents the diameter h of the mesh Ω_h. Initially, this field is set to 0. We also initialize the field `polygon`. This field is a cell array with a dimension equal to the number of polygons, where each cell contains a structure of type `polygon`. The details of the `polygon` structure will be described later.

A variable `init` gives an upper bound of the elements contained in `I`, `J` and `X`. These three vectors are initialized with the command `zeros(init,1)`. A variable `count` counts how many elements we have inserted in each of the three vectors.

```
%% MEMORY ALLOCATION
size = Mesh.nvertex + (polynomial.k - 1) * Mesh.nedges +
    polynomial.dim(1) * Mesh.npolygon;

f_global = zeros(size, 1);

Mesh.polygon  = {1:Mesh.npolygon};
Mesh.diameter = 0;

init = 2*Mesh.npolygon * (8*polynomial.k + polynomial.dim(1));

A_global.I = zeros(init,1);
A_global.J = zeros(init,1);
A_global.X = zeros(init,1);

count = 0;
```

After allocating memory for the matrix and the right-hand side, we begin a loop over the elements $E \in \Omega_h$. The variable `i` is the variable of the for loop and represents the element under consideration. At each iteration, we obtain the DoF indices in the element E.

This task is handled by the function `get_local_dofs`. This function takes the following input parameters: the structures `Mesh` and `polynomial`, and the index `i`. It returns the indices of the DoFs, the coordinates of the vertices of the polygon, and a boolean array `boundary_edges`. If the element in position j of `boundary_edges` is `true`, it indicates that the j-th edge is a boundary edge.

The next step is constructing the local matrix associated to the bilinear form

$$\epsilon\, a_h^E(u_h, v_h) + b_h^{\mathrm{skew},E}(u_h, v_h) + \sigma\, c_h^E(u_h, v_h) + \mathcal{N}_h^E(u_h, v_h)\,,$$

and the associated right-hand side. This task is handled by the function `ADR_element`. For now, it is sufficient to mention that it returns the local matrix `A_loc`, the right-hand side `f_loc`, and the structure `polygon`, which contains the information on the i-th polygon. We add the field `local_dofs` to this structure and save it in the i-th cell of `Mesh.polygon`. We store the information of the polygons because many quantities will be required in the computation of the CIP bilinear form and the computation of the error.

We check if the diameter of the i-th polygon is larger than the previous diameters. If so, we update the diameter of the mesh accordingly.

The next task is to update the fields of the global matrix `A_global`. We start by creating a square matrix `ld` by repeating the column vector of the local DoFs

$$\texttt{local\_dofs} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \quad \Rightarrow \quad \texttt{ld} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix}.$$

The field `A_global.I` is updated with the command `reshape(ld’,[],1)`. The function `reshape(B,[],1)` takes a matrix `B` and returns a vector in the following manner:

$$\texttt{B} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \Rightarrow \quad \texttt{ans} = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 4 \end{bmatrix}.$$

In this example, the matrix `B` is reshaped into a column vector by stacking its columns on top of each other. Similarly, we update `A_global.J` and `A_global.X`. This implementation for updating `A` avoids the use of a for loop or similar and completes the task using MATLAB functions. Once we have updated the matrix `A`, we increment the counter `count` by the size of `A_loc`. Finally, we update the right-hand side by adding the local vector `f_loc` into the global vector `f_global` in the positions indicated by the local DoFs.

After completion of the loop over all the elements, the last task of the function is to trim the fields in `A_global` by taking the number of elements indicated by `count`.

```
%% CONSTRUCT THE LINEAR SYTEM
for i=1:Mesh.npolygon

    [local_dofs, vertex, boundary_edges] = get_local_dofs(Mesh, polynomial, i);
    [A_loc, f_loc, polygon] = ADR_element(vertex, polynomial, f, g, boundary_edges, options);

    polygon.local_dofs     = local_dofs;
    Mesh.polygon{i}        = polygon;

    if (Mesh.polygon{i}.diameter > Mesh.diameter)

        Mesh.diameter = Mesh.polygon{i}.diameter;

    end

    ld = repmat(local_dofs,1,numel(local_dofs));

    A_global.I(count + (1:numel(A_loc))) = reshape(ld'   ,[],1);
    A_global.J(count + (1:numel(A_loc))) = reshape(ld    ,[],1);
    A_global.X(count + (1:numel(A_loc))) = reshape(A_loc',[],1);

    count = count + numel(K_loc);

    f_global(local_dofs) = f_global(local_dofs) + f_loc;
```

```
end

A_global.I = A_global.I(1:count);
A_global.J = A_global.J(1:count);
A_global.X = A_global.X(1:count);
```

6.2.3 The function CIP_assembly

This function is very similar to the previous ones so we will omit some details. It takes as input parameters the structures Mesh, polynomial, and options. It returns the sparse matrix J_global corresponding to the bilinear form $J_h(u_h, v_h)$. We remark that the VEM stabilization term of this bilinear form is computed in another part of the code by the function ADR_element. The first task of this function is to allocate the memory as in ADR_assembly.

```
%% MEMORY ALLOCATION
init = 2*Mesh.npolygon * (8*polynomial.k + polynomial.dim(1));

J_global.I = zeros(init,1);
J_global.J = zeros(init,1);
J_global.X = zeros(init,1);

count = 0;
```

We start a loop over the internal edges of Ω_h. We begin by obtaining the indices of the two elements that share the edge index. We store in the variable coords, which is a 2×2 matrix, the coordinates of the two endpoints of the edge. We then compute the edge length and one of the two outward normal vectors. Finally, we compute the Gauss quadrature points and store them in the variable points.

Now, we call the function cip_dofs(dofs1, dofs2). It takes as input the arrays of indices of the local DoFs of the elements that share the edge index and merges them. It returns three vectors:

- totalindex: a merged vector of the two arrays of local DoFs without repetitions,
- loc1: indicates the positions of the local DoFs of the first element in the array totalindex,
- loc2: indicates the positions of the local DoFs of the second element in the array totalindex.

There is a switch on the variable options.jump. Depending on its value, we choose between the possible discretizations: $J_h(\cdot,\cdot)$ corresponding to the value "Normal", $K_h(\cdot,\cdot)$ corresponding to the value "Gradient", and $L_h(\cdot,\cdot)$ corresponding to the value "CrossWind". We will focus only on the case "Normal" since it is the one that appears in the theoretical analysis. With this choice, we call the function cip_element_normal, which computes the local matrix associated with a single interior edge. Its implementation will be discussed later.

The function cip_assembly terminates similarly to cip_assembly and we omit the details.

```
%% J_GLOBAL ASSEMBLY
for index = Mesh.internal_edges

    elem1 = Mesh.adj(index,1);
    elem2 = Mesh.adj(index,2);

    coords = [Mesh.coords(Mesh.map(index,1),:);
              Mesh.coords(Mesh.map(index,2),:)];

    edge = Mesh.length(index);

    vnormal = [coords(2,2) - coords(1,2), -coords(2,1) + coords(1,1)];
    vnormal = vnormal ./ norm(vnormal,2);

    points = coords(1,:)' + [0, polynomial.gauss, 1].* (coords(2,:)' - coords(1,:)');

    [loc1, loc2, totalindex] = cip_dofs(Mesh.polygon{elem1}.local_dofs, Mesh.polygon{elem2}.local_dofs);

    % Local Switch
    switch(options.jump)

        case "Gradient"

            [J_loc] = cip_element_gradient(polynomial, points, edge, options, loc1, loc2, Mesh.polygon{elem1}, Mesh.polygon{elem2});

        case "Normal"

            [J_loc] = cip_element_normal(polynomial, points, edge, vnormal, options, loc1, loc2, Mesh.polygon{elem1}, Mesh.polygon{elem2});

        case "CrossWind"

            [J_loc] = cip_element_crosswind(polynomial, points, edge, options, loc1, loc2, Mesh.polygon{elem1}, Mesh.polygon{elem2});

        otherwise

            error("Invalid field jump in options!");

    end
%
    ld = repmat(totalindex,1,numel(totalindex));

    J_global.I(count + (1:numel(J_loc))) = reshape(ld'   ,[],1);
    J_global.J(count + (1:numel(J_loc))) = reshape(ld    ,[],1);
    J_global.X(count + (1:numel(J_loc))) = reshape(J_loc',[],1);

    count = count + numel(J_loc);

end

J_global.I = J_global.I(1:count);
J_global.J = J_global.J(1:count);
J_global.X = J_global.X(1:count);

end
```

We conclude this section by noting that, in the nonconforming method described in Chapter 4, this function also constructs the matrix associated with the bilinear form $d_h(\cdot,\cdot)$ by calling the function dh_element on each interior edge.

6.3 Assembly of the local elements

This section aims to describe how we assemble the local matrices. Specifically, the focus is on two functions: ADR_element and cip_element_normal. The first function, ADR_element, assembles the matrix related to the bilinear forms $a_h^E(\cdot,\cdot)$, $b_h^{\text{skew},E}(\cdot,\cdot)$, $c_h^E(\cdot,\cdot)$, $\mathcal{N}_h^E(\cdot,\cdot)$, and their VEM stabilization terms. It also includes the VEM stabilization of the bilinear form $J_h^E(\cdot,\cdot)$. The second function, cip_element_normal, constructs the CIP term restricted to an interior edge.

6.3.1 The function ADR_element

This function takes as input parameters: the structures polynomial and options, the coordinates of the vertices, the functions g and f, and the array boundary_edges. It return the local matrix A_loc, the local righ-hand side f_loc and the structure polygon. The first step is to compute the information related to the polygon with the function get_polygon_info. It takes the vertices of the polygon as input and returns a structure of type polygon. This structure is designed to contain all the quantities related to the polygon that we need for the computations. Specifically, we compute:

- the area of the polygon $|E|$,
- the diameter of the polygon h_E,
- the centroid of the polygon $\mathbf{x}_E$,
- the length of each edge $|e|$,
- the number of edges n_E,
- the perimeter of the polygon $|\partial E|$,
- the outward unit normal of each edge $\boldsymbol{n}^e$.

Next, we compute the dimension N_E of the space $V_h^k(E)$ and store it in the polygon structure. We conclude this part by defining the polynomial bases that will be used for the computations.

```matlab
%% GET THE INFORMATION OF THE POLYGON
polygon = get_polygon_info(vertex);

polygon.size   = polygon.nedges + polygon.nedges*(polynomial.k - 1) + polynomial.dim(1);

%% DEFINITION OF THE BASE FUNCTIONS AND THEIR GRADIENTS
base = @(x,y,k) ((x - polygon.centroid(1)) ./ polygon.diameter).^k(:,1) .* ((y - polygon.centroid(2)) ./ polygon.diameter).^k(:,2);
```

We construct a quadrature formula on the polygon E. To avoid triangulating the polygon, we define a quadrature formula over the entire domain. We follow the procedure described in [55]. It is possible to reduce the number of quadrature points to the optimal number by following [51], though we have not implemented this technique yet. The quadrature in the domain is computed in the function polygon_quadrature(polygon, m) which takes as input parameter the polygon and the order m, ensuring that the quadrature is exact up to order $2m-1$. The output is stored in the structure p_quad that contains the quadrature points and the associated weights. The fields in p_quad are multidimensional arrays with three dimension to match the construction of [55]. We also compute a quadrature rule on the boundary with the function boundary_quadrature and save the result in the variable b_quad.

We evaluate the polynomial basis functions and their gradients at the quadrature points. This approach allows us to compute the basis functions only once, thereby saving time. It is important to note that `base_val_int` and similarly `grad_val_int` are multidimensional arrays with four dimensions. In these arrays, the first three dimensions correspond to the quadrature points, while the last dimension indicates the basis function.

```
%% GET QUADRATURE NODES AND WEIGHTS
p_quad = polygon_quadrature (polygon, polynomial.k+1);
b_quad = boundary_quadrature(polygon, polynomial.k+1);

%% POINTWISE VALUES OF THE BASIS FUNCTION
base_val_int   = base_evaluation_interior(p_quad, polynomial,
    polygon);
grad_val_int   = grad_evaluation_interior(p_quad, polynomial,
    polygon);

base_val_bound = base_evaluation_boundary(b_quad, polynomial,
    polygon);
grad_val_bound = grad_evaluation_boundary(b_quad, polynomial,
    polygon);
```

Now, we explain how to compute the matrix representation of the polynomial projection $\Pi_k^{\nabla,E} : V_h^k(E) \to \mathbb{P}_k(E)$. A more detailed description can be found in [10]. We denote by $\{m_\alpha\}_{\alpha=1}^{N_p}$ a basis of the space $\mathbb{P}_k(E)$, where N_p is the dimension of $\mathbb{P}_k(E)$. Given a virtual function v_h, we seek coefficients $\{s^\alpha\}_{\alpha=1}^{N_p}$ such that

$$\Pi_k^{\nabla,E} v_h = \sum_{\alpha=1}^{N_p} s^\alpha m_\alpha \,.$$

The coefficients $\{s^\alpha\}_{\alpha=1}^{N_p}$ can be determined by solving the linear system

$$\begin{bmatrix} P_0 m_1 & P_0 m_2 & \dots & P_0 m_{n_p} \\ 0 & (\nabla m_2, \nabla m_2)_{0,E} & \dots & (\nabla m_2, \nabla m_{N_p})_{0,E} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & (\nabla m_{N_p}, \nabla m_2)_{0,E} & \dots & (\nabla m_{N_p}, \nabla m_{N_p})_{0,E} \end{bmatrix} \begin{bmatrix} s^1 \\ s^2 \\ \vdots \\ s^{N_p} \end{bmatrix} = \begin{bmatrix} P_0 v_h \\ (\nabla v_h, \nabla m_2)_{0,E} \\ \vdots \\ (\nabla v_h, \nabla m_{N_p})_{0,E} \end{bmatrix} ,$$

where P_0 is the projection onto constants and it is chosen as in (1.6). We define the square matrix $\mathtt{G} \in \mathbb{R}^{N_p \times N_p}$ as

$$\mathtt{G} := \begin{bmatrix} P_0 m_1 & P_0 m_2 & \dots & P_0 m_{N_p} \\ 0 & (\nabla m_2, \nabla m_2)_{0,E} & \dots & (\nabla m_2, \nabla m_{N_p})_{0,E} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & (\nabla m_{N_p}, \nabla m_2)_{0,E} & \dots & (\nabla m_{N_p}, \nabla m_{N_p})_{0,E} \end{bmatrix} ,$$

and the matrix $\mathtt{B} \in \mathbb{R}^{N_p \times N_E}$ as

$$\mathtt{B} := \begin{bmatrix} P_0 \varphi_1 & \dots & P_0 \varphi_{N_E} \\ (\nabla \varphi_1, \nabla m_2)_{0,E} & \dots & (\nabla \varphi_{N_E}, \nabla m_2)_{0,E} \\ \vdots & \ddots & \vdots \\ (\nabla \varphi_1, \nabla m_{N_p})_{0,E} & \dots & (\nabla \varphi_{N_E}, \nabla m_{N_p})_{0,E} \end{bmatrix} ,$$

where $\{\varphi_i\}_{i=1}^{N_E}$ is the nodal basis function of $V_h^k(E)$. The matrix of the coefficients $\Pi_*^\nabla \in \mathbb{R}^{N_p \times N_E}$ can be determined by solving the linear system

$$\mathtt{G}\,\Pi_*^\nabla = \mathtt{B} \,.$$

Each column of Π_*^∇ contains the coefficient of $\Pi_k^{\nabla,E} \varphi_i$ with respect to the basis $\{m_\alpha\}_{\alpha=1}^{N_p}$.

To construct the matrix corresponding to the stabilization term `dofi-dofi`, we have to express each $\{\Pi_k^{\nabla,E}\varphi_i\}_{i=1}^{N_E}$ with respect to the VEM basis function $\{\varphi_j\}_{i=1}^{N_E}$

$$\Pi_k^{\nabla,E}\varphi_i = \sum_{j=1}^{N_E} \pi_i^j \varphi_j \,.$$

We want to determine the coefficients $\{\pi_i^j\}_{j=1}^{N_E}$ for each $i = 1, \dots, N_E$. We observe that

$$\Pi_k^{\nabla,E}\varphi_i = \sum_{\alpha=1}^{N_p} s_i^\alpha m_\alpha = \sum_{\alpha=1}^{N_p} s_i^\alpha \sum_{j=1}^{N_E} \mathrm{dof}_j(m_\alpha)\varphi_j \,,$$

which implies

$$\pi_i^j = \sum_{\alpha=1}^{N_p} s_i^\alpha \mathrm{dof}_j(m_\alpha) \,.$$

We define the matrix $\mathtt{D} \in \mathbb{R}^{N_E \times N_p}$ as

$$\mathtt{D} := \begin{bmatrix} \mathrm{dof}_1(m_1) & \mathrm{dof}_1(m_2) & \dots & \mathrm{dof}_1(m_{N_p}) \\ \mathrm{dof}_2(m_1) & \mathrm{dof}_2(m_2) & \dots & \mathrm{dof}_2(m_{N_p}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{dof}_{N_E}(m_1) & \mathrm{dof}_{N_E}(m_2) & \dots & \mathrm{dof}_{N_E}(m_{N_p}) \end{bmatrix} .$$

Hence, the previous equation can be written in a matrix form as

$$\pi_i^j = \sum_{\alpha=1}^{N_p} (\mathtt{G}^{-1}\mathtt{B})_{\alpha i}\, \mathtt{D}_{j\alpha} = (\mathtt{D}\mathtt{G}^{-1}\mathtt{B})_{ji} \,.$$

The matrix representation of the operator $\Pi_k^{\nabla,E} : V_h^k(E) \to V_h^k(E)$ in the canonical basis $\{\varphi_j\}_{j=1}^{N_E}$ is

$$\Pi^\nabla = \mathtt{D}\Pi_*^\nabla = \mathtt{D}\mathtt{G}^{-1}\mathtt{B} \,.$$

As demonstrated in [10], the following equivalence holds

$$\mathtt{G} = \mathtt{B}\mathtt{D} \,.$$

Consequently, computing matrix G is redundant, leading to potential computational savings. The matrix representation of the `dofi-dofi` stabilization term

$$\sum_{i=1}^{N_E} \mathrm{dof}_i(I - \Pi_k^{\nabla,E})u_h \, \mathrm{dof}_i(I - \Pi_k^{\nabla,E})v_h$$

is given by

$$(\mathtt{I} - \Pi^\nabla)^T \, (\mathtt{I} - \Pi^\nabla) \,.$$

The matrix representations of the projections $\Pi_k^{0,E}$ and $\boldsymbol{\Pi}_k^{0,E}$ (Π^0 and $\bar{\Pi}_k^0$, respectively) are constructed in a very similar manner, so we will omit the details. In these constructions, the role of `G` is taken by the matrices `H` and `S`, respectively, while the role of `B` is taken by `C` and `E`. In details, the matrices $\mathtt{H} \in \mathbb{R}^{N_P \times N_p}$ and $\mathtt{C} \in \mathbb{R}^{N_P \times N_E}$ are defined as

$$(\mathtt{H})_{ij} := (m_j, m_i)_{0,E} \,, \qquad \text{and} \qquad (\mathtt{C})_{ij} := (m_j, \varphi_i)_{0,E} \,,$$

while $\mathtt{S} \in \mathbb{R}^{2N_P \times 2N_p}$ and $\mathtt{E} \in \mathbb{R}^{2N_P \times N_E}$ are defined as

$$(\mathtt{S})_{ij} := (\mathbf{m}_j, \mathbf{m}_i)_{0,E} \,, \qquad \text{and} \qquad (\mathtt{E})_{ij} := (\mathbf{m}_j, \nabla\varphi_i)_{0,E}$$

where $\{\mathbf{m}_j\}_{j=1}^{2N_P}$ is a polynomial basis function of $[\mathbb{P}_k(E)]^2$.

These projections are computed by the following lines of code. We remark that in order to compute the matrices `C` and `E`, we have to calculate the quantity

$$\int_E v_h\, \hat{p}_k \,\mathrm{d}E = \int_E \Pi_k^{\nabla,E} v_h\, \hat{p}_k \,\mathrm{d}E \quad \forall \hat{p}_k \in \mathbb{P}_{k/k-2}(E)\,.$$

The function `Pi_evaluate` computes the values of each $\Pi_k^{\nabla,E}\varphi_i$ at each quadrature point. It does so by taking as input parameters the coefficients of the projection and the evaluations of the basis functions at the quadrature points.

```
%% CONSTRUCT MATRIX D
D = create_matrix_D(base, polynomial, polygon, p_quad,
    base_val_int);

%% CONSTRUCT THE PINABLA(K) PROJECTION
%CONSTRUCT THE MATRIX B
B = create_matrix_B(grad_val_bound, polynomial, polygon);

%CONSTRUCT THE MATRIX G AND GTILDE
G = B*D;

%PINABLA(K) PROJECTION
PiNabla_k  = G \ B;

%% CONSTRUCT THE PI0(K) PROJECTION
% EVALUATE THE PINABLA_K PROJECTION
PiNabla_val_int = Pi_evaluate(PiNabla_k, base_val_int);

% CONSTRUCT THE MATRIX C
C = create_matrix_C(base_val_int, PiNabla_val_int, p_quad,
    polynomial, polygon);

% CONSTRUCT THE MATRIX H
H = C*D;

% CONSTRUCT THE MATRIX PI0(K)
Pi0_k = H \ C;

% Evaluate the Pi0 projection
Pi0_val_int = Pi_evaluate(Pi0_k, base_val_int);

%% CONSTRUCT THE PIBAR(K) PROJECTION
% CONSTRUCT THE MATRIX S
S = [H zeros(polynomial.dim(3)); zeros(polynomial.dim(3)) H];

% CONSTRUCT THE MATRIX P
E = create_matrix_E(base_val_bound,polynomial,polygon,C);

% CONSTRUCT THE MATRIX PIBAR
PiBar_k = S \ E;

%% CONSTRUCT THE PIBAR(K-1) PROJECTION
n_rows = polynomial.dim(2);
index  = 1:n_rows;

% CONSTRUCT THE MATRIX S
S_minus = [H(index,index) zeros(n_rows); zeros(n_rows) H(index,
    index)];
```

```

% CONSTRUCT THE MATRIX P
E_minus = E([1:n_rows, polynomial.dim(2)+1:polynomial.dim(2)+
    n_rows],:);

% CONSTRUCT THE MATRIX PIBAR
PiBar_kminus = S_minus \ E_minus;
```

Now, we add each form to the local matrix A_loc. We check the values of the fields option.stiffness, option.stabilization, and option.advection, and compute the corresponding discretizations. Depending on option.stiffness, the matrix representation of the bilinear form $a_h^E(\cdot,\cdot)$ without the VEM stabilization is either

$$(\Pi_*^\nabla)^T\,\tilde{\mathtt{G}}\,\Pi_*^\nabla \qquad \text{or} \qquad (\bar{\Pi}_{k-1}^0)^T\,\mathtt{S}_{k-1}\,\bar{\Pi}_{k-1}^0\,.$$

where $\mathtt{S}_{k-1}$ is the restriction of $\mathtt{S}$ to a basis of $[\mathbb{P}_{k-1}(E)]^2$, and $\tilde{\mathtt{G}}$ is equal to $\mathtt{G}$ except that its first row is zero. The reaction bilinear form is represented by:

$$(\Pi^0)^T\,\mathtt{H}\,\Pi^0\,.$$

Finally, depending on the field options.advection, the bilinear form $b_h(\cdot,\cdot)$ is either

$$(\Pi^0)^T\,\mathtt{BetaH}\,\bar{\Pi}_k^0\,, \qquad \text{with} \qquad (\mathtt{BetaH})_{ij} := (\boldsymbol{\beta}\cdot\mathbf{m}_j, m_i)_{0,E}\,,$$

or

$$(\Pi^0)^T\,\mathtt{BetaG}\,\Pi^0 + (\Pi^0)^T\,\mathtt{BetaGb}\,(\mathtt{I} - \mathtt{D} * \Pi^0)\,,$$

with

$$(\mathtt{BetaG})_{ij} := (\boldsymbol{\beta}\cdot\nabla m_j, m_i)_{0,E}\,, \qquad \text{and} \qquad (\mathtt{BetaGb})_{ij} := \langle\boldsymbol{\beta}\cdot\boldsymbol{n}^E m_j, m_i\rangle_{0,E}\,.\,.$$

```
%% CONSTRUCT THE DIFFUSION MATRIX
if (options.stiffness == "PiNabla")

    Gtilde      = G;
    Gtilde(1,:) = zeros(1, polynomial.dim(3));

    A_loc = options.epsilon .* PiNabla_k' * Gtilde * PiNabla_k;

elseif (options.stiffness == "PiBar")

    A_loc = options.epsilon .* PiBar_kminus' * S_minus *
        PiBar_kminus;

else

    error("Invalid field stiffness in opt!")

end

%% CONSTRUCT THE CONVECTION TERM
b_val1_int = f_evaluation_interior(options.beta{1}, p_quad);
b_val2_int = f_evaluation_interior(options.beta{2}, p_quad);

if (options.advection == "Zero")

    BetaH = create_matrix_betaH_1D(b_val1_int, b_val2_int,
        base_val_int, p_quad, polynomial);

    T = Pi0_k'*BetaH*PiBar_k;

elseif (options.convection == "Boundary")

    b_val1_bound = f_evaluation_boundary(options.beta{1},b_quad);
    b_val2_bound = f_evaluation_boundary(options.beta{2},b_quad);

    BetaG = create_matrix_betaG(b_val1_int, b_val2_int, base_val_int, grad_val_int, ...
                               p_quad, polynomial);

    BetaGb = create_matrix_betaGb(b_val1_bound, b_val2_bound, base_val_bound, polynomial, polygon);

    T = Pi0_k'*BetaG*Pi0_k + Pi0_k'*BetaGb*(I-D*Pi0_k);

else

    error("Invalid field stabilization in opt!")

end

% ADD CONVECTION MATRIX
A_loc = K_local + 0.5 * (T - T');

%% CONSTRUCT THE REACTION MATRIX
A_loc = A_loc + options.sigma .* Pi0_k' * H * Pi0_k;

%% ADD THE STABILIZATION TERM
I       = eye(polygon.size);
Iminus  = I - Pi;
A_loc   = A_loc + (options.epsilon + options.gamma*polygon.diameter + options.sigma*polygon.area)*(Iminus' * Iminus);
```

Finally, we construct the right-hand side. We compute the values of f and each $\Pi_k^{0,E}\varphi_i$ at the quadrature point and evaluate each integral.

```
%% CONSTRUCT f_loc
f_loc     = zeros(polygon.size,1);
f_val_int = f_evaluation_interior(f, p_quad);

for i=1:polygon.size

    integrand  = Pi0_val_int(:,:,:,i) .* f_val_int;
    f_loc(i) = quadrature_2D(p_quad, integrand,"Evaluated");

end
```

If we have chosen to use the Nitsche method, it remains to impose the boundary conditions. The task is performed by the function `create_matrix_N` that computes the Nitsche matrix and the corresponding right-hand side.

```
if (options.boundary == "Nitsche")

    [N_loc, fN_loc] = create_matrix_N(boundary_edges, grad_val_bound, options, polygon, polynomial);

    A_loc = A_loc + N_loc;
    f_loc = f_loc + fN_loc;

end
```

Finally, we need to store some quantities that will be required later for the computation of the errors and for the CIP term.

```
polygon.PiN = PiNabla_k;
polygon.Pi0 = Pi0_k;
```

6.3.2 The function `cip_element_normal`

The first task of this function is to evaluate the function

$$[\![\Pi_k^0 \varphi_i]\!]$$

at each quadrature point and for each virtual basis function. The results are stored in the matrix `N_Diff_val`, where each column corresponds to the evaluation of a basis function. The number of columns of this matrix equals the number of unique DoFs in the elements E and K that contain the internal edge e. After that, it is possible to compute the CIP matrix corresponding to the edge e with a for loop.

```
for i = 1:total

    J_loc(i,i) = polynomial.gweights * (N_diff_val(i,:)').^2;

    for j = 1:i-1

        J_loc(i,j) = polynomial.gweights * (N_diff_val(i,:)' .*
           N_diff_val(j,:)');
        J_loc(j,i) = J_loc(i,j);


    end

end

J_loc = gamma * edge^3 .* J_loc;
```

6.4 Solving the linear system

Returning to the main file, we need to solve the linear system. We assemble the global matrix `A_global` using the MATLAB function `sparse(I,J,X)`. If we choose to solve the PDE with the weakly imposed boundary condition, we only need to solve the linear system using the backslash operator `\`. If we impose the boundary conditions in the classical manner, we split the matrix `A` and the load vector as

$$\texttt{A} = \begin{bmatrix} \texttt{AII} & \texttt{AIB} \\ \texttt{ABI} & \texttt{ABB} \end{bmatrix}, \quad \text{and} \quad \texttt{F} = \begin{bmatrix} \texttt{FI} \\ \texttt{FB} \end{bmatrix},$$

where `I` represents the indices of the interior DoFs, and `B` represents the indices of the boundary DoFs. The linear system reads as

$$\begin{bmatrix} \texttt{AII} & \texttt{AIB} \\ \texttt{ABI} & \texttt{ABB} \end{bmatrix} \begin{bmatrix} \texttt{UI} \\ \texttt{UB} \end{bmatrix} = \begin{bmatrix} \texttt{FI} \\ \texttt{FB} \end{bmatrix}.$$

The component UB is known thanks to the boundary conditions. To compute the values of u_h in the interior DoFs we solve the linear

$$\texttt{AI} * \texttt{UI} = \texttt{fI} - \texttt{AIB} * \texttt{UB}\,.$$

```
%% CONVERTING SYSTEM MATRIX TO A SPARSE MATRIX
siz = size(f_global,1);

A = sparse([A_global.I; J_global.I], [A_global.J; J_global.J], [A_global.X; J_global.X], siz, siz);

%% CALCULATE BOUNDARY DATAS

if (options.boundary == "Classic")

    fprintf('[%.2f] Enforcing Dirichlet boundary conditions...\n',toc);
    f_global(boundary_vertex)  = g(Mesh.coords(boundary_vertex,1), Mesh.coords(boundary_vertex,2));
    f_global(boundary_intdofs) = g(Mesh.intcoords(boundary_intdofs-Mesh.nvertex,1), Mesh.intcoords(boundary_intdofs-Mesh.nvertex,2));

end

%% SOLVE THE SYSTEM
fprintf('[%.2f] Solving system of linear equations...\n',toc);

if (options.boundary == "Classic")

    internal_dofs = setdiff(1:size(f_global,1), boundary_dofs);

    AII = A(internal_dofs,internal_dofs);
    AIB = A(internal_dofs, boundary_dofs);
    fI  = f_global(internal_dofs);
    UB  = f_global(boundary_dofs);
    UI  = AII \ (fI - AIB * UB);

    U                = zeros(size(A,1),1);
    U(internal_dofs) = UI;
    U(boundary_dofs) = UB;

end


if (options.boundary == "Nitsche")

    U = A \ f_global;

end
```

It is possible to visualize the numerical solution using the following lines of code. The function fill3 allows you to draw a function on a polygon by specifying the values of the function at its vertices. The first two input parameters are the x-coordinates and y-coordinates of the vertices, respectively. The third parameter is a value that determines the color of the polygon.

```
figure(1)

for i = 1:Mesh.npolygon
    hold on
    fill3( Mesh.coords(Mesh.connect{i},1), Mesh.coords(Mesh.connect{i},2), U(connect{i}), mean(U(Mesh.connect{i})))
end
```

6.5 Computation of the errors

The final step is to compute the errors in the L^2-norm and H^1-seminorm. This task is handled by the function `compute_errors`, which takes as input the structures `Mesh` and `polynomial`, the vector of DoFs of the numerical solution `U`, the analytic solution u, and its gradient.

```
fprintf('[%.2f] Computing errors...\n',toc);
[errL2, errH1] = compute_errors(Mesh,polynomial,U,u, grad_u);

fprintf("[%.2f] Errors computed: ",toc)
fprintf("\nL2 norm     : %f", errL2)
fprintf("\nH1 seminorm: %f", errH1)
```

6.5.1 The function compute_errors

We begin by initializing the errors to 0 and then consider each element of the decomposition Ω_h. For each polygon, we construct a quadrature rule that is more accurate than the one used for constructing the linear system. Next, we evaluate the polynomial basis functions, their gradients, the analytic solution u, and its gradient at the quadrature points. We then compute the coefficients of $\Pi_k^{\nabla,E} u_h$ and $\Pi_k^{0,E} u_h$ with respect to the polynomial basis functions and calculate their values at the quadrature points using the function `Pi_evaluate`, as described in previous sections. square root to obtain the final error values. With the values of the analytic solution and the numerical solution at the quadrature points, we compute their difference. The error is then calculated using the quadrature formula. After summing all the contributions from each polygon, we take the square root to obtain the final error values.

```
errL2 = 0;
errH1 = 0;

for i=1:Mesh.npolygon

    %% COMPUTE THE QUADRATURE
    polygon = Mesh.polygon{i};

    [p_quad] = polygon_quadrature(polygon{i},polynomial.k+2);

    base_val = base_evaluation_interior(p_quad, polynomial, polygon{i});

    grad_val = grad_evaluation_interior(p_quad, polynomial, polygon{i});

    %% U EVALUATION
    u_val      = f_evaluation_interior(u,        p_quad);
    deru_x_val = f_evaluation_interior(grad_u.x, p_quad);
    deru_y_val = f_evaluation_interior(grad_u.y, p_quad);

    %% COMPUTE PI_NABLA U_h & EVALUATION
    Pistar = polygon.G \ (polygon.B * U(polygon.local_dofs));

    derPistar_val_x = Pi_evaluate(Pistar, grad_val(:,:,:,1:polynomial.dim(3)));
    derPistar_val_y = Pi_evaluate(Pistar, grad_val(:,:,:,polynomial.dim(3)+1:end));

    %% COMPUTE PI_0 U_h & EVALUATION
    P0 = polygon.H \ (polygon.C * U(polygon.local_dofs));

    P0_val = Pi_evaluate(P0, base_val);

    %% COMPUTE THE DIFFERENCES
    diff_sq      = (u_val - P0_val).^2;                         %Integrands
    diff_sq_grad = (deru_x_val - derPistar_val_x).^2 + (deru_y_val - derPistar_val_y).^2;

    errL2 = errL2 + quadrature_2D(p_quad,diff_sq, "Evaluated");

    errH1 = errH1 + quadrature_2D(p_quad,diff_sq_grad, "Evaluated");

    end

    errL2 = sqrt(errL2);
    errH1 = sqrt(errH1);

end
```

Bibliography

[1] B. Ahmad, A. Alsaedi, F. Brezzi, L. D. Marini, and A. Russo. Equivalent projectors for virtual element methods. *Comput. Math. Appl.*, 66(3):376–391, 2013.

[2] O. Andersen, H. M. Nilsen, and X. Raynaud. Virtual element method for geomechanical simulations of reservoir models. *Computational Geosciences*, 21:877–893, 2017.

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

[4] P.F. Antonietti, L. Beirão da Veiga, and G. Manzini. *The Virtual Element Method and its applications*, volume 31. SEMA-SIMAI springer series, 2021.

[5] B. Ayuso De Dios, K. Lipnikov, and G. Manzini. The nonconforming virtual element method. *ESAIM Math. Model. Numer. Anal.*, 50(3):879–904, 2016.

[6] H. J. C. Barbosa and T. J. R. Hughes. The finite element method with lagrange multipliers on the boundary: circumventing the babuška-brezzi condition. *Computer Methods in Applied Mechanics and Engineering*, 85(1):109–128, 1991.

[7] G. Barrenechea, E. Burman, E. Cáceres, and J. Guzmán. Continuous interior penalty stabilization for divergence-free finite element methods. *IMA Journal of Numerical Analysis*, 44(2):980–1002, 06 2023.

[8] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, and A. Russo. Basic principles of Virtual Element Methods. *Math. Models Methods Appl. Sci.*, 23(1):199–214, 2013.

[9] L. Beirão da Veiga, F. Brezzi, and L. D. Marini. Virtual elements for linear elasticity problems. *SIAM Journal on Numerical Analysis*, 51(2):794–812, 2013.

[10] L. Beirão da Veiga, F. Brezzi, L. D. Marini, and A. Russo. The Hitchhiker's Guide to the Virtual Element Method. *Math. Models Methods Appl. Sci.*, 24(8):1541–1573, 2014.

[11] L. Beirão da Veiga, F. Brezzi, L. D. Marini, and A. Russo. Virtual Element Method for general second-order elliptic problems on polygonal meshes. *Math. Models Methods Appl. Sci.*, 26(4):729–750, 2016.

[12] L. Beirão da Veiga, F. Brezzi, L.D. Marini, and A. Russo. The virtual element method. *ACTA Numerica*, 32:123–202, 2023.

[13] L. Beirão da Veiga, F. Dassi, D. Di Pietro, and J. Droniou. Arbitrary-order pressure-robust ddr and vem methods for the stokes problem on polyhedral meshes. *Computer Methods in Applied Mechanics and Engineering*, 397:115061, 2022.

[14] L. Beirão da Veiga, F. Dassi, C. Lovadina, and G Vacca. SUPG-stabilized virtual elements for diffusion-convection problems: A robustness analysis. *ESAIM Math. Model. Numer. Anal.*, 55(5):2233–2258, 2021.

[15] 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, 11 2021.

[16] L. Beirão da Veiga, C. Lovadina, and A. Russo. Stability analysis for the virtual element method. *Math. Mod.and Meth. in Appl. Sci.*, 27(13):2557–2594, 2017.

[17] L. Beirão da Veiga, C. Lovadina, and M. Trezzi. CIP-stabilized virtual elements for diffusion-convection-reaction problems. *IMA Journal of Numerical Analysis*, page drae020, 05 2024.

[18] L. Beirão da Veiga, C. Lovadina, and G. Vacca. Divergence free virtual elements for thestokes problem on polygonal meshes. *ESAIM Math. Model. Numer. Anal.*, 51(2):509–535, 2017.

[19] L. Beirão da Veiga and G. Vacca. Sharper error estimates for Virtual Elements and a bubble-enriched version. *ArXiv preprint*, 2020.

[20] M. F. Benedetto, S. Berrone, A. Borio, S. Pieraccini, and S. Scialò. A hybrid mortar virtual element method for discrete fracture network simulations. *J. Comput. Phys.*, 306:148–166, 2016.

[21] M. F. Benedetto, S. Berrone, A. Borio, S. Pieraccini, and S. Scialò. Order preserving SUPG stabilization for the virtual element formulation of advection-diffusion problems. *Comput. Methods Appl. Mech. Engrg.*, 293:18–40, 2016.

[22] M. F. Benedetto, S. Berrone, S. Pieraccini, and S. Scialò. The virtual element method for discrete fracture network simulations. *Comput. Methods Appl. Mech. Engrg.*, 280:135–156, 2014.

[23] S. Berrone, A. Borio, and G. Manzini. SUPG stabilization for the nonconforming virtual element method for advection–diffusion–reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 340:500–529, 2018.

[24] S. Bertoluzza, G. Manzini, M. Pennacchio, and D. Prada. Stabilization of the nonconforming virtual element method. *Computers & Mathematics with Applications*, 116:25–47, 2022. New trends in Computational Methods for PDEs.

[25] S. Bertoluzza, M. Pennacchio, and D. Prada. Weakly imposed dirichlet boundary conditions for 2d and 3d virtual elements. *Computer Methods in Applied Mechanics and Engineering*, 400:115454, 2022.

[26] S. C. Brenner, Q. Guan, and L.Y. Sung. Some estimates for virtual element methods. *Comput. Methods Appl. Math.*, 17(4):553–574, 2017.

[27] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer, New York, third edition, 2008.

[28] S. C. Brenner and L.Y. Sung. Virtual element methods on meshes with small edges or faces. *Math. Models Methods Appl. Sci.*, 28(7):1291–1336, 2018.

[29] F. Brezzi, K. Lipnikov, and M. Shashkov. Convergence of the mimetic finite difference method for diffusion problems on polyhedral meshes. *SIAM J. Numerical Analysis*, 43:1872–1896, 01 2005.

[30] F. Brezzi and E. Süli. Discontinuous galerkin methods for first-order hyperbolic problems. *Mathematical Models and Methods in Applied Sciences*, 14, 02 2004.

[31] A. Brooks and T. 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(1):199–259, 1982.

[32] E. Burman. A unified analysis for conforming and nonconforming stabilized finite element methods using interior penalty. *SIAM Journal on Numerical Analysis*, 43(5):2012–2033, 2005.

[33] E. Burman and A. Ern. Continuous interior penalty hp-finite element methods for advection and advection-diffusion equations. *Mathematics of Computation*, 76(259):1119–1140, 2007.

[34] E. Burman, M. Fernández, and P. Hansbo. Continuous interior penalty finite element method for oseen's equations. *SIAM J. Numerical Analysis*, 44:1248–1274, 07 2006.

[35] E. Burman and P. Hansbo. Edge stabilization for galerkin approximations of convection–diffusion–reaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193(15):1437–1453, 2004.

[36] A. Cangiani, G. Manzini, and O. Sutton. Conforming and nonconforming virtual element methods for elliptic problems. *IMA J. Numer. Anal.*, 37(3):1317–1354, 2017.

[37] D. Castanon Quiroz and D. Di Pietro. A pressure-robust HHO method for the solution of the incompressible Navier–Stokes equations on general meshes. *IMA Journal of Numerical Analysis*, 44(1):397–434, 04 2023.

[38] M. Cicuttin, A. Ern, and N. Pignet. *Hybrid high-order methods: a primer with applications to solid mechanics.* Springer, 2021.

[39] T. Davis. *Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms 2).* Society for Industrial and Applied Mathematics, USA, 2006.

[40] D. Di Pietro and J. Droniou. The hybrid high-order method for polytopal meshes. *Number 19 in Modeling, Simulation and Application*, 2020.

[41] D. Di Pietro and A. Ern. *Mathematical aspects of discontinuous Galerkin methods*, volume 69. Springer Science & Business Media, 2011.

[42] Z. Dong, E. H. Georgoulis, and T. Pryer. Recovered finite element methods on polygonal and polyhedral meshes. *ESAIM: M2AN*, 54(4):1309–1337, 2020.

[43] J. Douglas and T. Dupont. Interior penalty procedures for elliptic and parabolic galerkin methods. In R. Glowinski and J. L. Lions, editors, *Computing Methods in Applied Sciences*, pages 207–216, Berlin, Heidelberg, 1976. Springer Berlin Heidelberg.

[44] T.P. Fries and T. Belytschko. The extended/generalized finite element method: An overview of the method and its applications. *International Journal for Numerical Methods in Engineering*, 84:253 – 304, 10 2010.

[45] V. John, A. Linke, C. Merdon, M. Neilan, and L. Rebholz. On the divergence constraint in mixed finite element methods for incompressible flows. *SIAM Review*, 59(3):492–544, 2017.

[46] M. Juntunen and R. Stenberg. Nitsche's method for general boundary conditions. *Mathematics of Computation*, 78:1353–1374, 2009.

[47] P. Knobloch and G. Lube. Local projection stabilization for advection–diffusion–reaction problems: One-level vs. two-level approach. *Applied numerical mathematics*, 59(12):2891–2907, 2009.

[48] Y. Li and M. Feng. A local projection stabilization virtual element method for convection-diffusion-reaction equation. *Applied Mathematics and Computation*, 411:126536, 2021.

[49] C. Lovadina, I. Perugia, and M. Trezzi. A nonconforming virtual element method for advection-diffusion-reaction problems with cip stabilization. *arxiv*, 2024.

[50] G. Matthies, P. Skrzypacz, and L. Tobiska. A unified convergence analysis for local projection stabilisations applied to the oseen problem. *ESAIM: Mathematical Modelling and Numerical Analysis*, 41(4):713–742, 2007.

[51] S. E. Mousavi, H. Xiao, and N. Sukumar. Generalized gaussian quadrature rules on arbitrary polygons. *International Journal for Numerical Methods in Engineering*, 82(1):99–113, 2010.

[52] J. Nitsche. Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36:9–15, 1971.

[53] Maxim A. Olshanskii and Leo G. Rebholz. Longer time accuracy for incompressible navier–stokes simulations with the emac formulation. *Computer Methods in Applied Mechanics and Engineering*, 372:113369, 2020.

[54] B. Rivière. *Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation*. SIAM, 2008.

[55] A. Sommariva and M. Vianello. Product gauss cubature over polygons based on green's integration formula . *Bit Numer Math*, 47:441–453, 2007.

[56] N. Sukumar and A. Tabarraei. Conforming polygonal finite elements. *International Journal for Numerical Methods in Engineering*, 61(12):2045–2066, 2004.

[57] G. Vacca. An h1-conforming virtual element for darcy and brinkman equations. *Mathematical Models and Methods in Applied Sciences*, 28(01):159–194, 2018.

[58] J. Wang and X. Ye. A weak galerkin finite element method for second-order elliptic problems. *Journal of Computational and Applied Mathematics*, 241:103–115, 2013.